Multicode Multirate Compact Assignment of OVSF Codes for QoS Differentiated Terminals*

Yang Yang[1] and Tak-Shing Peter Yum[2]

[1] Department of Electronic and Computer Engineering,
Brunel University, Uxbridge, London, UB8 3PH, United Kingdom
[2] Department of Information Engineering,
The Chinese University of Hong Kong, Shatin, Hong Kong

Abstract. Orthogonal Variable Spreading Factor (OVSF) codes are used in both UTRA-FDD and UTRA-TDD of the third-generation (3G) mobile communication systems. They can support multirate transmissions for mobile terminals with multicode transmission capabilities. In this paper, a new OVSF code assignment scheme, namely "Multicode Multirate Compact Assignment" (MMCA), is proposed and analyzed. The design of MMCA is based on the concept of "compact index" and takes into consideration mobile terminals with different multicode transmission capabilities and Quality of Service (QoS) requirements. Specifically, priority differentiation between multirate realtime traffic and best-effort data traffic is supported in MMCA. Analytical and simulation results show that MMCA is efficient and fair.

1 Introduction

Orthogonal Variable Spreading Factor (OVSF) codes [1] are adopted in UTRA-FDD and TDD (Universal Terrestrial Radio Access – Frequency Division Duplex and Time Division Duplex) of the third-generation (3G) mobile communication systems to identify traffic channels for different users. According to the technical specifications [2,3], multiple parallel code (channel) transmissions are possible for a single user to support multirate multimedia applications. Although single-code transmission is simpler, multicode transmission has the advantages of finer granularity in bandwidth assignment, more flexible code assignment solutions, and therefore higher bandwidth efficiency.

From a user's perspective, traffic can be classified as either realtime calls or best-effort data packets. Realtime calls require realtime transmission with a fixed bandwidth or, in other words, at a fixed data rate. This traffic class includes audio and video telephonies, on-line TV/movie watching and so on. Best-effort data packets are those generated from the Internet and audio and video file transfers. Realtime calls have priority over data packets in code assignment. On

* This work was supported in part by the Hong Kong Research Grants Council under Grant CUHK 4325/02E and the Short Term Research Fellowship programme of British Telecommunications (BT).

the other hand, from the system's perspective, users are heterogeneous. First, they have different Quality of Service (QoS) requirements, e.g. realtime or best-effort transmission, fixed or variable bandwidth assignment, and fixed or variable packet size. Second, mobile terminals have different capabilities in supporting multicode transmission.

Code assignment schemes can be of the *non-rearrangeable* and *rearrangeable* type. Specifically, rearrangeable code assignment schemes allow the OVSF codes to be rearranged so that they have better performance at the expense of higher computational complexity. Many single-code rearrangeable code assignment schemes were proposed in literature [4,5,6,7,8,9,10,11,12,13,14]. Among them, the priority issue between realtime and best-effort traffic was considered in [4,7,9]. Several single-code non-rearrangeable code assignment schemes were proposed in [8,10,11,12,13]. Specifically, the algorithm in [8] is based on the first-fit scheme for the bin-packing problem. In [10], a fixed code configuration, which specifies the number of OVSF codes for each service class, is used for maximizing the average throughput. Tseng and Chao compare the performance of *random*, *leftmost* and *crowded-first* schemes in [11]. The concept of *crowded-first* is extended in [12] and a new code selection scheme based on the "weights" of candidate codes is proposed. In [13], a new measure called "compact index" is defined as the criterion for code assignment. By using this criterion, the proposed *Compact Assignment* (CA) scheme can offer comparable performance to rearrangeable schemes. Multicode rearrangeable code assignment schemes were proposed in [15,16] for uniform mobile terminals having exactly the same capability in supporting multicode transmission, and in [17,18] for different multicode capable terminals. All these multicode schemes consider only multirate realtime traffic class.

In this paper, based on the concept of "compact index", we design and analyze a non-rearrangeable multicode code assignment scheme, namely "Multicode Multirate Compact Assignment" (MMCA), for accommodating both multirate realtime and best-effort traffic. The design considers the coexistence of mobile terminals with different multicode transmission capabilities and QoS requirements. When multicode transmission is introduced, many slack capacities in the code tree can be taken up by the second and third codes and renders code re-arrangement not essential. Also, when data packets are introduced, they can absorb these "wasted" capacity (usable but not used by realtime traffic).

The rest of this paper is organized as follows. In Section 2, the tree structure and some basic concepts of OVSF codes are reviewed. Based on that, the code assignment problem for accommodating mobile terminals with different QoS requirements and different multicode transmission capabilities is formulated. The algorithm of *Multicode Multirate Compact Assignment* (MMCA) is proposed and discussed in Section 3. In Section 4, the performance of MMCA is studied. Both the analytical and simulation results are given and compared.

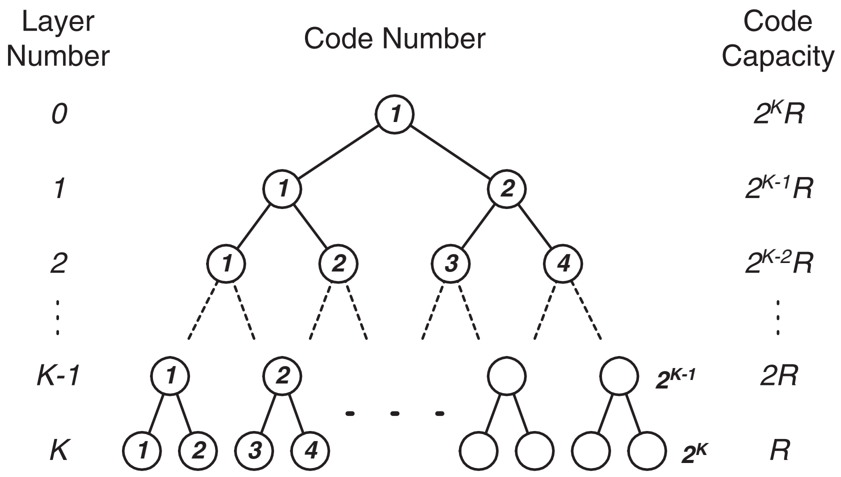

Fig. 1. A K-layer code tree.

2 The Code Assignment Problem

All OVSF codes in the system can be represented by the nodes in a binary tree [1]. Fig. 1 shows a K-layer code tree. Each layer corresponds to a particular spreading factor, so all codes in the same layer can support the same data rate. The data rate a code can support is called its capacity. Let the capacity of the leaf codes (in layer K) be R. Then, the capacity of the codes in layer k is $2^{K-k}R$, as shown in Fig. 1.

Layer k has 2^k codes and they are sequentially labeled from left to right, starting from one. The m^{th} code in layer k is referred to as code (k,m). The total capacity of all the codes in each layer is $2^K R$, irrespective of the layer number. For a typical code (k,m), its ancestor code set, denoted by $S_A^{(k,m)}$, contains all the codes on the path from (k,m) to the root code $(0,1)$. Its descendant code set, denoted by $S_D^{(k,m)}$, contains all the codes in the branch under (k,m).

Codes in the same layer that are connected by an i-layer sub-tree are defined as the i^{th}-layer neighbours. Let $S_i^{(k,m)}$ denote the set of i^{th}-layer neighbours of code (k,m). Then,

$$S_i^{(k,m)} = \left\{(k, m-p+q)\,|\, p = (m-1)\ mod\ 2^i,\, 0 \le q \le 2^i - 1\right\} \ . \tag{1}$$

Take code $(K,3)$ in Fig. 1 as an example, the sets of 1^{st}- and 2^{nd}-layer neighbours are $S_1^{(K,3)} = \{(K,3),(K,4)\}$ and $S_2^{(K,3)} = \{(K,1),(K,2),(K,3),(K,4)\}$, respectively. The positional relationship between (k,m) and other layer-k codes are represented by the k different sets $S_i^{(k,m)}$ $(1 \le i \le k)$.

2.1 Assignable Codes

Consider code (k, m). When it is assigned to carry a realtime call, we stipulate that code (k, m) and all its ancestor and descendant codes are *non-preemptable.* When code (k, m) is assigned to carry a best-effort data packet, we stipulate that code (k, m) and all its descendant codes are *preemptable.* In this case, an ancestor code of (k, m) is also *preemptable* if it is not *non-preemptable.* Therefore, an ancestor code is *non-preemptable* if it has both *non-preemptable* and *preemptable* descendant codes. Preemptable codes can be only assigned and reassigned to realtime calls by suspending some ongoing packet transmissions. Besides *non-preemptable* and *preemptable* codes, all remaining codes in the tree are *assignable.* They can be freely assigned to carry either realtime calls or data packets. More importantly, assignable codes have the following properties.

Property 1: If code (k, m) (where $k \leq K - 1$) is assignable, so are all its descendant codes [13].

Property 2: If all the leaf descendant codes of code (k, m) (where $k \leq K - 1$) are assignable, so is code (k, m).

These assignable codes, preemptable codes and non-preemptable codes form a partition of the code tree. They can be characterized by the status index $I^{(k,m)}$, defined as

$$I^{(k,m)} = \begin{cases} 0, & \text{code } (k, m) \text{ is non-preemptable } ; \\ 1, & \text{code } (k, m) \text{ is preemptable } ; \\ 2, & \text{code } (k, m) \text{ is assignable } . \end{cases} \tag{2}$$

As an example, Fig. 2 shows the status index values of all codes in a 4-layer code tree.

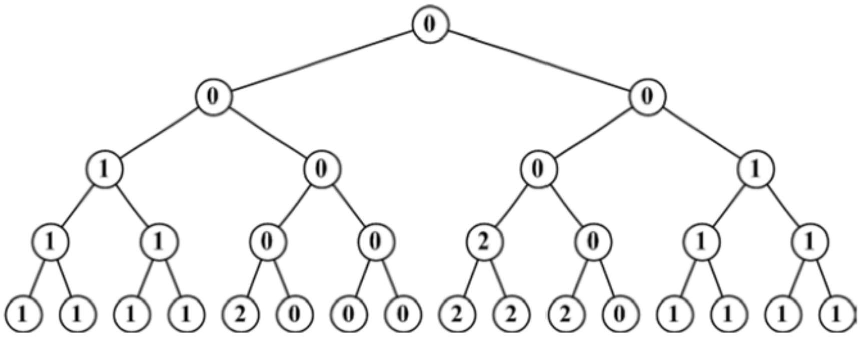

Fig. 2. Status index in a 4-layer code tree. Codes $\{(3, 4), (4, 6), (4, 12)\}$ are carrying realtime calls, and codes $\{(2, 1), (2, 4)\}$ are carrying data packets.

Upon receiving a new transmission request (realtime call or data packet), the base station needs to first identify all candidate codes suitable for assignment.

Let S denote the set of all candidate codes in the tree and let S_K denote the set of leaf candidate codes in layer K. For data packets, S and S_K consist of assignable codes only. But for realtime calls, preemptable codes are also included in S and S_K since realtime calls have priority over data packets in code assignment. In other words, S_K is given by

$$S_K = \begin{cases} \{(K,m) \,|\, I^{(K,m)} = 2,\ 1 \leq m \leq 2^K\}, & \text{for data packets ;} \\ \{(K,m) \,|\, I^{(K,m)} \geq 1,\ 1 \leq m \leq 2^K\}, & \text{for realtime calls .} \end{cases} \tag{3}$$

According to *Property 2*, candidate code set S can be derived from S_K [1].

2.2 Compact Index $g^{(k,m)}$

To expand the capability of the code tree in supporting different data rates, new code assignments should be packed as tightly as possible into the existing busy codes, i.e. the candidate codes in the most congested positions are used to carry the new calls/packets. In this way, the code tree is kept as compact (and hence flexible for accommodating multiple data rates) as possible after each code assignment.

The candidate codes in the most congested positions can be identified by their *compact index* $g^{(k,m)}$, which is defined as the total number of candidate codes in the k different neighbourhoods of code (k, m) [13].

$$g^{(k,m)} = \sum_{i=1}^{k} |S_i^{(k,m)} \cap S| \ , \tag{4}$$

where $|x|$ denotes the size of set x. Given layer k, a smaller value of $g^{(k,m)}$ implies that candidate code (k, m) is surrounded by less number of other candidate codes in the same layer and is, therefore, located in a more congested position. For a newly arrived data packet seeing the code tree shown in Fig. 2, we have $g^{(4,5)} = 7$, $g^{(4,9)} = g^{(4,10)} = 12$ and $g^{(4,11)} = 11$, which implies code $(4, 5)$ is in the most congested position.

3 Multicode Multirate Compact Assignment

We propose in this section a multicode assignment scheme, namely *Multicode Multirate Compact Assignment* (MMCA). The objective of MMCA is to keep the remaining candidate codes in the most compact state after each code assignment without rearranging codes. This can be achieved by finding the candidate codes in the most congested positions for the new calls/packets. In summary, MMCA is a natural extension of *Compact Assignment* (CA) [13] with the following features.

1. MMCA does not perform code rearrangement and is therefore simple.

[1] The mapping from S_K to S is a bijection.

2. MMCA provides priority differentiation between realtime calls and data packets.

3. MMCA supports mobile terminals with different multicode transmission capabilities.

4. MMCA balances transmission quality among the multiple codes assigned to the same user.

5. MMCA supports multirate realtime calls and keeps the code tree as flexible as possible in accepting new multirate calls.

3.1 Multicode Solutions

For a mobile terminal requiring bandwidth (or data rate) $j \cdot R$ and can transmit n codes, several code assignment solutions may be available for use. A solution, denoted by $(d_0, d_1, \cdots, d_K)$, consists of $K+1$ integers with d_k representing the number of candidate codes needed in layer k. The set of all solutions can be obtained by enumerating all integer-combinations under the constraints of bandwidth requirement and multicode transmission capability, i.e. $\sum_{k=0}^{K} d_k \cdot 2^{K-k} = j$ and $\sum_{k=0}^{K} d_k \leq n$.

We propose for use a more efficient algorithm called *Multicode Solution Generator*. It starts from the solution $(0, 0, \cdots, 0, j)$, which requires j leaf candidate codes. The next solution can be obtained by replacing two leaf codes by one $(K-1)$-layer code in the first solution, i.e. $(0, 0, \cdots, 1, j-2)$. Continuing this way, all possible multicode solutions satisfying the bandwidth requirement can be obtained. Next, we use multicode transmission capability to screen out solutions requiring more than n codes [2].

As an example, consider a 4-layer code tree with each solution represented by five integers. Table 1 lists all multicode solutions for different combinations of bandwidth requirement (from $j = 1$ to $j = 16$) and multicode transmission capability (from $n = 1$ to $n = 6$) [3].

As seen in Table 1, for some combinations of j and n, e.g. $j = 15$ and $n = 2$, no solution exists. These cases are marked by symbol "–" in the table. On the other hand, given j, mobile terminal with a larger n may have more choices in multicode assignment. As an example, for the case $j = 6$ and $n = 3$, there are three multicode solutions: $(0, 0, 1, 1, 0)$, $(0, 0, 0, 3, 0)$ and $(0, 0, 1, 0, 2)$. Intuitively, the solution requiring the least number of candidate codes (with large code capacity), i.e. $(0, 0, 1, 1, 0)$, is appealing. However, we found that the solution requiring a larger number of codes (with small code capacity) is more "system-friendly". Here are the reasons:

Reason 1: It is usually much easier to find small-capacity candidate codes for assignment, especially when the system is heavy loaded.

[2] Another approach using dynamic programming technique is given in [18].

[3] For simplicity, $(d_0, d_1, d_2, d_3, d_4)$ is represented by "$d_0 d_1 d_2 d_3 d_4$" in this table.

Table 1. Multicode solutions.

Data	Multicode Transmission Capability					
Rate	1	2	3	4	5	6
1R	00001					
2R	00010	00002				
3R	–	00011	00003			
4R	00100	00020	00012	00004		
5R	–	00101	00021	00013	00005	
6R	–	00110	00030 00102	00022	00014	00006
7R	–	–	00111	00031 00103	00023	00015
8R	01000	00200	00120	00040 00112	00032 00104	00024
9R	–	01001	00201	00121	00041 00113	00033 00105
10R	–	01010	00210 01002	00130 00202	00050 00122	00042 00114
11R	–	–	01011	00211 01003	00131 00203	00051 00123
12R	–	01100	00300 01020	00220 01012	00140 00212 01004	00060 00132 00204
13R	–	–	01101	00301 01021	00221 01013	00141 00213 01005
14R	–	–	01110	00310 01030 01102	00230 00302 01022	00150 00222 01014
15R	–	–	–	01111	00311 01031 01103	00231 00303 01023
16R	10000	02000	01200	00400 01120	00320 01040 01112	00240 00312 01032 01104

Reason 2: The use of small-capacity codes can offer better transmission quality because they have larger spreading factors.

Reason 3: There are more small-capacity candidate codes in the congested positions. They should be used first so as to keep the resulting code tree as flexible as possible in supporting different data rates.

When there are multiple choices for assigning the same number of candidate codes, we choose the solution with the minimum variance in code capacity (or spreading factor) so as to balance the transmission quality of these codes. Thus for the above example with $j = 6$ and $n = 3$, two suitable solutions both requiring three candidate codes, i.e. $(0, 0, 0, 3, 0)$ and $(0, 0, 1, 0, 2)$, are identified. Multicode solution $(0, 0, 0, 3, 0)$ (which requires three candidate codes with capacity $2R$) is chosen as it has smaller capacity variance.

3.2 Code Assignment

Upon receiving the transmission request from a particular mobile terminal, the base station first calculates the system assignable capacity according to traffic class. In unit of R, assignable capacity r is defined as

$$r = |S_K| = \begin{cases} \sum_{m=1}^{2^K} \left\lfloor \frac{I^{(K,m)}}{2} \right\rfloor , & \text{for data packets } ; \\ \\ \sum_{m=1}^{2^K} \left\lceil \frac{I^{(K,m)}}{2} \right\rceil , & \text{for realtime calls } . \end{cases} \tag{5}$$

where $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the floor and the ceiling functions, respectively. For the code tree shown in Fig. 2, assignable capacity r is equal to 4 for data packets and 12 for realtime calls.

For transmitting a data packet, the mobile terminal does not need to specify the bandwidth requirement. When assignable capacity r is non-zero, it is used as "Bandwidth Requirement" (i.e. let $j = r$) in the code assignment algorithm. In other words, we apply the "greedy" policy and try to use all assignable capacity for data packet transmission so as to achieve full system utilization. Under the constraint n of multicode transmission capability, all multicode solutions can be pre-computed by the *Multicode Solution Generator* and stored in a table such as Table 1. If no solution exists (indicated by symbol "–" in the table) for some combinations of j and n, the code assignment algorithm will *reduce* the value of j gradually until the first solution is identified. On the other hand, when assignable capacity is zero, the data packet transmission request is put into a queue at the base station if the queue size limit is not exceeded, or blocked otherwise.

Now consider a realtime call request from a mobile terminal with bandwidth requirement j and multicode transmission capability n. When $r \geq j$, the base station performs code assignment assuming the absence of data packet traffic. Some data packets may need to reduce their transmission data rates, or even totally suspend their transmissions, to make codes available for the newly arrived realtime call. For some combinations of j and n, no solution exists and symbol "–" is observed in the table of multicode solutions. The code assignment algorithm will then *increase* j gradually until the first solution is identified. After a particular solution is chosen according to the criteria given in Section 3.1, compact index $g^{(k,m)}$ is used for code selection and assignment in each layer.

A realtime call request will be blocked if the system cannot meet the bandwidth or multicode requirements. Specifically, there are three blocking conditions.

Condition 1: The required bandwidth is larger than the assignable capacity, i.e. $j > r$.

Condition 2: $j \leq r$, but the summed bandwidth of every multicode solution is larger than the assignable capacity, i.e. $\sum_{k=0}^{K} d_k \cdot 2^{K-k} > r$.

Condition 3: $j \leq r$ and $\sum_{k=0}^{K} d_k \cdot 2^{K-k} \leq r$, but the candidate codes found in some layers are not sufficient, i.e. the number is less than d_k.

To illustrate, consider the code tree shown in Fig. 2. For realtime calls, the assignable capacity $r = 12$. However, a new call request with $j = 10$ and $n = 1$ will be blocked due to *Condition 2* (the identified solution $(1,0,0,0,0)$ has summed bandwidth of $16R$). Another request with $j = 12$ and $n = 3$ will be blocked due to *Condition 3* (all multicode solutions, namely $(0,1,1,0,0)$, $(0,0,3,0,0)$ and $(0,1,0,2,0)$, cannot be supported by the code tree).

The blockings due to *Condition 1* are unavoidable. The blockings due to *Condition 2* can be avoided only by improving the mobile terminal's multicode transmission capability. The blockings due to *Condition 3* can be avoided by either rearranging codes or improving multicode capability. For example, in Fig. 2, if the realtime call on code $(4,12)$ is reassigned to code $(4,5)$, a realtime call request with $j = 12$ and $n = 3$ can then be carried in the code tree by suspending all ongoing data packet transmissions. As seen in Table 1, when mobile terminals are multicode capable, a number of multicode solutions are usually available. *Condition 3* of blocking is therefore much less likely to occur, compared to the single-code transmission scenario.

3.3 Data Rate Reduction and Transmission Resumption

As data packet transmissions can be preempted by realtime calls, some mobile terminals have to reduce their transmission data rates to make codes available for realtime calls. For the mobile terminals that totally suspend the data transmissions, their identifications and the corresponding break points are recorded at the base station. When some occupied codes are released, they will be shared by these suspended terminals as fairly as possible.

4 Performance Analysis

Let there be N types of mobile terminals in the system where the type-n $(1 \leq n \leq N)$ terminals can support the simultaneous transmission of n codes. Let p_n

be the fraction of type-n terminals. Further, let there be J classes of realtime calls where the class-j $(1 \leq j \leq J)$ calls are characterized by *(i)* Poisson arrivals with rate λ_j; *(ii)* bandwidth requirements equal to $j \cdot R$; and *(iii)* exponentially distributed call holding time with mean μ_j^{-1}. Let $G_j = \lambda_j/\mu_j$ $(1 \leq j \leq J)$ denote the offered traffic of class-j realtime calls. The total offered traffic G_R of realtime calls is simply the sum of G_j. For simplicity, we assume terminal type and service class are independent. Let λ_D and μ_D^{-1} denote the arrival rate and average packet length of data packets, respectively. The offered traffic of data packets is therefore given by $G_D = \lambda_D/\mu_D$.

Without loss of generality, a six-layer code tree $(K = 6)$ and eight classes realtime calls $(J = 8)$ with equal offered traffic $(G_1 = G_2 = \cdots = G_8)$ are considered in the computer simulation. The arrival of data packets is assumed to be a Poisson process and the packet length is chosen from four exponential random variables with means R, $2R$, $4R$ and $8R$ with equal probabilities. Let there be four types of mobile terminals $(N = 4)$ and let their combinations take on the following four cases.

Case 1: $p_1 : p_2 : p_3 : p_4 = 100 : 0 : 0 : 0$.
Case 2: $p_1 : p_2 : p_3 : p_4 = 40 : 30 : 20 : 10$.
Case 3: $p_1 : p_2 : p_3 : p_4 = 25 : 25 : 25 : 25$.
Case 4: $p_1 : p_2 : p_3 : p_4 = 10 : 20 : 30 : 40$.

Note that multicode transmission is not supported in *Case 1*. In the following figures, all simulation results are shown in dashed lines with markers. For each simulation experiment, the simulation time is increased until the 95% confidence interval is comparable to the marker size shown.

4.1 Blocking Probability of Realtime Calls

Blocking probability is the most important measure of QoS for realtime calls. Since the realtime calls have preemptive priority over data packets, as far as blocking performance is concerned, data packets are completely transparent to realtime calls. Consider the ideal case where all mobile terminals can use as many codes as required, i.e. $n = J$. Then, call blockings due to *Conditions 2* and *3* (section 3.2) can be completely avoided. The blocking probability in this case is the same as that under the "complete sharing policy" in shared resource environment [19]. This blocking result is therefore a lower bound (see Bound A in Fig. 3) for the restrictive multicode cases studied here. Due to the length limit, the derivation details of this lower bound are not shown in this paper.

Fig. 3 shows the overall blocking probability as a function of realtime offered traffic G_R. The solid lines are the analytical lower bounds. The blocking probabilities of the four cases discussed in Section 4 are obtained by computer simulation. As seen, the overall blocking probability can be significantly reduced with the use of multicode. As an example, at $G_R = 5.6$ (Erlang), the blocking probabilities for the four simulation cases and the analytical lower bound (marked as Bound A) are 2.21%, 1.14%, 0.92%, 0.58% and 0.45%, respectively. This lower

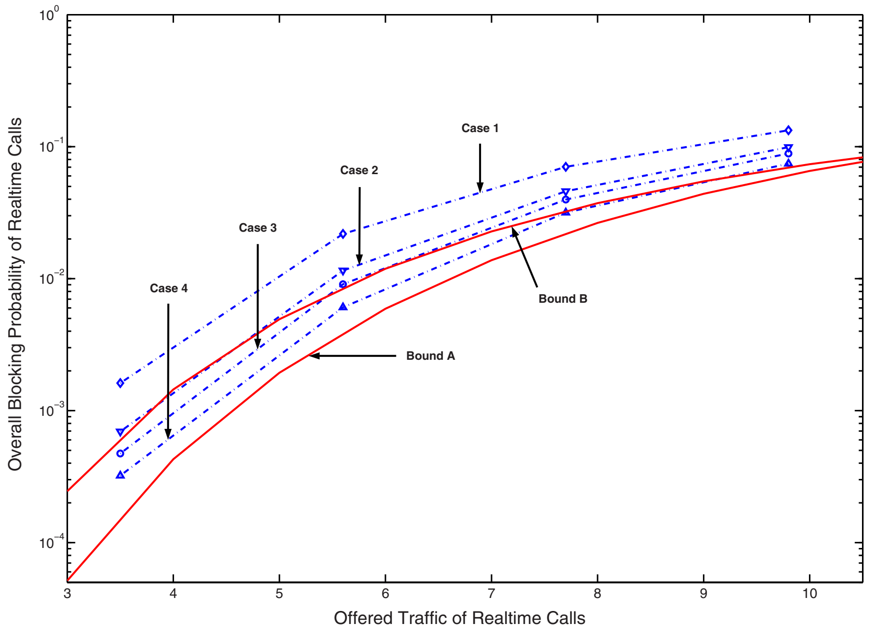

Fig. 3. Blocking probability of realtime calls.

bound can be achieved by letting all mobile terminals capable of transmitting any number of codes. This result indicates that the use of multicode can be an effective alternate to the rearrangeable single-code method in [13]. For comparison purpose, the lower bound for rearrangeable single-code system is also shown (marked as Bound B).

4.2 Throughput and Wasted Capacity of Realtime Calls

The throughput of realtime calls, denoted by T, is given by

$$T = \sum_{j=1}^{J} (1 - P_j) \cdot L_j \ , \tag{6}$$

where P_j and $L_j \triangleq j \cdot G_j$ denote the blocking probability and the offered load of class-j realtime calls, respectively. The total offered load L of realtime calls is simply the summation of L_j. The throughput in (6) gives the time-averaged required bandwidth from successful realtime terminals. The total assigned capacity for realtime calls may be larger. For example, to accommodate a type-1 realtime terminal with bandwidth requirement $6R$, the base station needs to assign a layer-$(K-3)$ code (with code capacity $8R$) to the terminal. The gap between these two values is called “wasted capacity”.

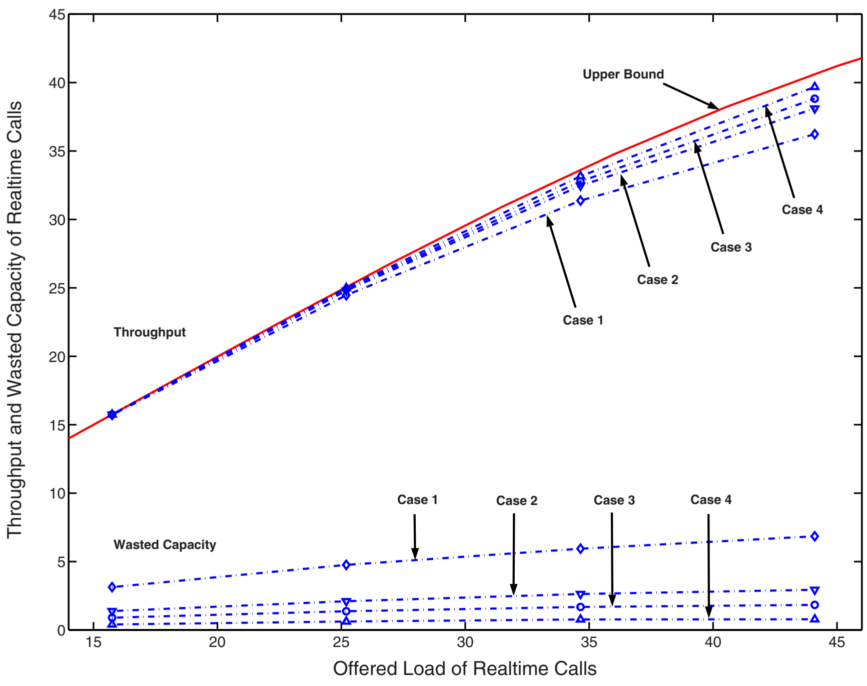

Fig. 4. Throughput and wasted capacity of realtime calls.

Fig. 4 shows the throughput and wasted capacity of realtime calls as a function of offered load L. The solid line is the analytical upper bound, or (6), on throughput. As seen, this upper bound can be approached by introducing more multicode-capable terminals. The same action can also reduce the amount of wasted capacity. In the limiting case where all terminals are capable of transmitting any number of codes, the wasted capacity is zero. Specifically, at offered load $L = 34.65$, the wasted capacity values in four simulation cases are 5.85, 2.65, 1.69 and 0.76, respectively.

4.3 Sojourn Time of Data Packets

For data packets, the average sojourn time is the most important QoS measure. It is defined as the time between a transmission request and the successful transmission of the whole packet. Fig. 5 shows the average sojourn time as a function of total offered traffic $G_R + G_D$. We assume in this case the ratio between realtime and data traffic is fixed at $G_R : G_D = 7 : 3$. The performance of the three multicode cases are similar and are all about 30% better than the single-code case, *Case 1*. This indicates that the sojourn time cannot be effectively reduced by manipulating the multicode capability mixes. As an example, at offered traffic $G_R + G_D = 11$, the average sojourn time values for the four cases are 1.71, 1.24, 1.20 and 1.16, respectively. By Little's formula, the same conclusion can be drawn on queue length.

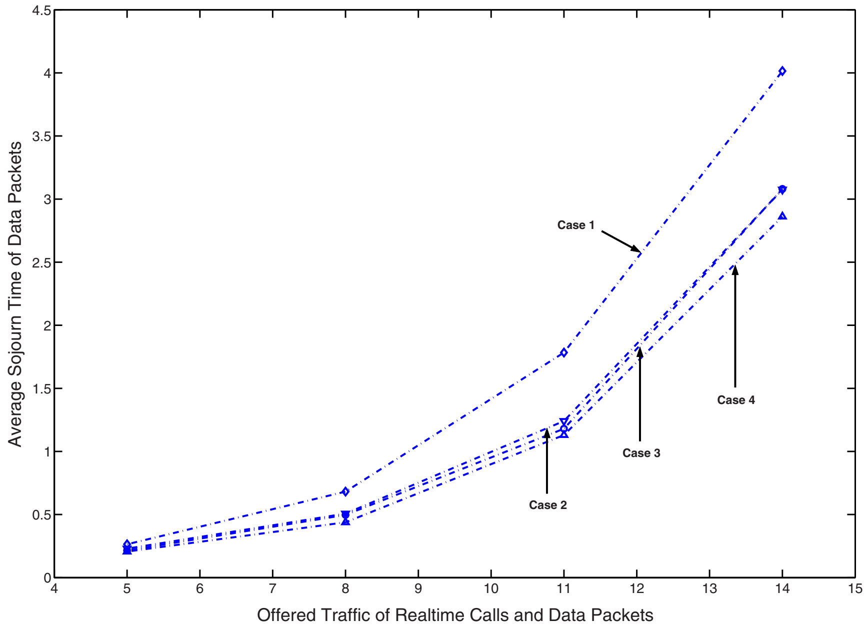

Fig. 5. Average sojourn time of data packets, $G_R : G_D = 7 : 3$.

4.4 Fairness Comparison

The major fairness concern is the chance of accessing system resource for different types of terminals with different bandwidth requirements. As an example, the fairness index for the terminals with different bandwidth requirements, denoted by F_R, is defined as [13]

$$F_R = \frac{\left[\sum_{j=1}^{J} (1 - P_j)\right]^2}{J \sum_{j=1}^{J} (1 - P_j)^2} \ . \tag{7}$$

In the ideal case where the terminals with different bandwidth requirements have the same opportunity of being served (i.e. the P_j values are equal), F_R achieves the maximum value of one.

Fig. 6 shows fairness index F_R as a function of offered traffic of realtime calls. As seen, even under heavy traffic, there is no substantial difference in the fair access among the realtime terminals with different bandwidth requirements. Although not shown, our results show that the same is true for the terminals with different multicode transmission capabilities.

5 Conclusions

Based on the concept of compact index, a new OVSF code assignment scheme, namely "Multicode Multirate Compact Assignment" (MMCA), is proposed for

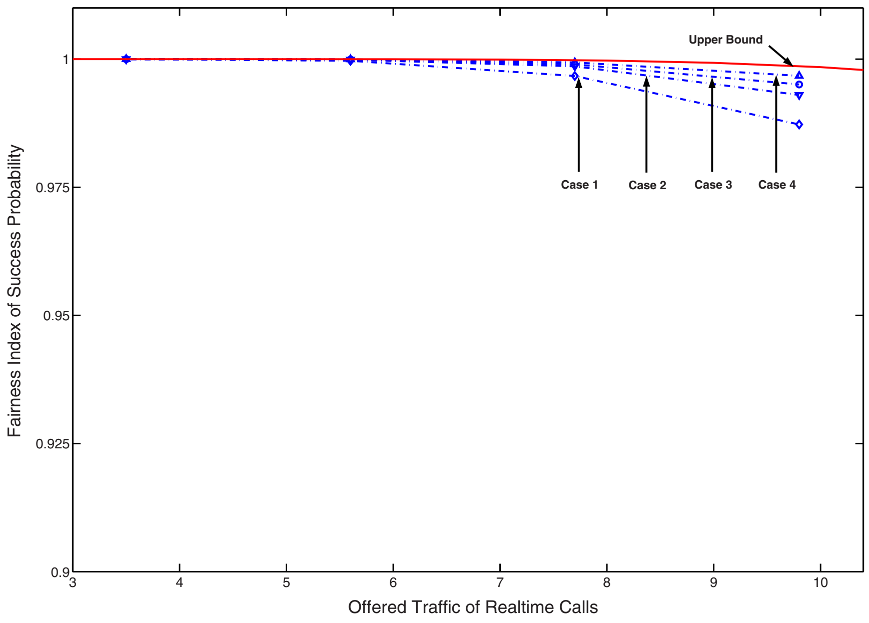

Fig. 6. Fairness index for realtime calls with different bandwidth requirements.

accommodating QoS differentiated mobile terminals. These terminals have different multicode transmission capabilities. They can also support different traffic types (realtime calls and data packets) with different priority and bandwidth requirements. When more mobile terminals have multicode transmission capability, the bandwidth granularity in code assignment becomes smaller and the system is more flexible in supporting multirate multimedia traffic classes. As a result, higher bandwidth efficiency is observed in MMCA. This is demonstrated by both analysis and simulation. In addition, MMCA is also shown to be a fair code assignment scheme for different service classes.

References

1. Adachi, F., Sawahashi, M., Okawa, K.: Tree-structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile radio. Electronics Letters**33** (1997) 27–28
2. 3GPP TS 25.213 (V6.0.0): Spreading and modulation (FDD). Technical Specification (Release 6), Technical Specification Group Radio Access Network, 3GPP (2003)
3. 3GPP TS 25.223 (V6.0.0): Spreading and modulation (TDD). Technical Specification (Release 6), Technical Specification Group Radio Access Network, 3GPP (2003)

4. Fantacci, R., Nannicini, S.: Multiple access protocol for integration of variable bit rate multimedia traffic in UMTS/IMT-2000 based on wideband CDMA. IEEE Journal on Selected Areas in Communications**18** (2000) 1441–1454
5. Minn, T., Siu, K.Y.: Dynamic assignment of orthogonal variable-spreading-factor codes in W-CDMA. IEEE Journal on Selected Areas in Communications(2000) 1429–1440
6. Assarut, R., Kawanishi, K., Yamamoto, U., Onozato, Y., Matsushita, M.: Region division assignment of orthogonal variable-spreading-factor codes in W-CDMA. In: Proceedings of IEEE VTC 2001. Volume 3. (2001) 1884–1888
7. Chen, W.T., Wu, Y.P., Hsiao, H.C.: A novel code assignment scheme for W-CDMA systems. In: Proceedings of IEEE VTC2001. Volume 2. (2001) 1182–1186
8. Dell'Amico, M., Merani, M.L., Maffioli, F.: Efficient algorithms for the assignment of ovsf codes in wideband CDMA. In: Proceedings of IEEE ICC 2002. Volume 5. (2002) 3055–3060
9. Fossa Jr., C.E., Davis IV, N.J.: Dynamic code assignment improves channel utilization for bursty traffic in third-generation wireless networks. In: Proceedings of IEEE ICC 2002. Volume 5. (2002) 3061–3065
10. Park, J.S., Lee, D.C.: On static and dynamic code assignment policies in the OVSF code tree for CDMA networks. In: Proceedings of IEEE MILCOM 2002. Volume 2. (2002) 785–789
11. Tseng, Y.C., Chao, C.M.: Code placement and replacement strategies for wideband CDMA OVSF code tree management. IEEE Transactions on Mobile Computing**1** (2002) 293–302
12. Rouskas, A.N., Skoutas, D.N.: OVSF codes assignment and reassignment at the forward link of W-CDMA 3G systems. In: Proceedings of IEEE PIMRC 2002. Volume 5. (2002) 2404–2408
13. Yang, Y., Yum, T.S.P.: Maximally flexible assignment of orthogonal variable spreading factor codes for multirate traffic. IEEE Transactions on Wireless Communications**3** (2004) 781–792
14. Sekine, Y., Kawanishi, K., Yamamoto, U., Onozato, Y.: Hybrid OVSF code assignment scheme in W-CDMA. In: Proceedings of 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. Volume 1. (2003) 384–387
15. Cheng, R.G., Lin, P.: OVSF code channel assignment for IMT-2000. In: Proceedings of IEEE VTC2000. Volume 3. (2000) 2188–2192
16. Chao, C.M., Tseng, Y.C., Wang, L.C.: Reducing internal and external fragmentations of OVSF codes in WCDMA systems with multiple codes. In: Proceedings of IEEE WCNC 2003. Volume 1. (2003) 693–698
17. Shueh, F., Chen, W.S.E.: Code assignment for IMT-2000 on forward radio link. In: Proceedings of IEEE VTC 2001. Volume 2. (2001) 906–910
18. Yen, L.H., Tsou, M.C.: An OVSF code assignment scheme utilizing multiple rake combiners for W-CDMA. In: Proceedings of IEEE ICC 2003. Volume 5. (2003) 3312–3316
19. Kaufman, J.S.: Blocking in a shared resource environment. IEEE Transactions on Communications**COM-29** (1981) 1474–1481