Multicode Multirate Compact Assignment of OVSF Codes for QoS Differentiated Terminals

Yang Yang, *Member, IEEE,* and Tak-Shing Peter Yum, *Senior Member, IEEE*

***Abstract*—Orthogonal variable spreading factor (OVSF) codes are used in both universal terrestrial radio access–frequency division duplex (UTRA-FDD) and time division duplex (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 different quality of service (QoS) requirements. Priority differentiation between multirate realtime traffic and best-effort data traffic is also supported in MMCA. Analytical and simulation results show that MMCA is efficient and fair.**

***Index Terms*—Dynamic channel assignment (DCA), orthogonal variable spreading factor (OVSF) code, quality of service (QoS), universal terrestrial radio access (UTRA).**

I. Introduction

ORTHOGONAL variable spreading factor (OVSF) codes [1] are adopted in universal terrestrial radio access–frequency division duplex (UTRA-FDD) and time division duplex (UTRA-TDD) of the third-generation (3G) mobile communication systems. According to UTRA-FDD and UTRA-TDD specifications [2], [3], multiple parallel code (channel) transmissions are possible for a single user to support multirate applications. Multicode transmission has the advantages of finer granularity in bandwidth assignment and therefore higher bandwidth efficiency.

Traffic can typically be classified as fixed data-rate realtime calls and best-effort data packets. Realtime calls have priority over data packets in code assignment. From the system perspective, users are heterogeneous in that they have different quality of service (QoS) requirements, and their mobile terminals have different capabilities in supporting multicode transmission.

Code assignment schemes can be of the *nonrearrangeable* and *rearrangeable* type. Specifically, rearrangeable code assignment schemes allow 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]–[14]. Among them, the priority issue between realtime traffic and best-effort traffic was considered in [4], [7] and [9]. Several single-code nonrearrangeable code assignment schemes were proposed. Specifically, the scheme in [8] is based on the "first-fit" policy for the bin-packing problem. In [10], the number of OVSF codes for each service class is found for maximizing the average throughput. In [11], the performance of random, leftmost, and crowded-first schemes are compared. 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 for code assignment and the compact assignment (CA) scheme is proposed. Multicode rearrangeable code assignment schemes were proposed in [15] and [16] for uniform mobile terminals having exactly the same capability in supporting multicode transmission and in [17] and [18] for different multicode capable terminals. All these multicode schemes are designed for only multirate realtime traffic.

In this paper, based on the concept of "compact index," we design and analyze a multicode nonrearrangeable code assignment scheme, namely "multicode multirate compact assignment" (MMCA), for accommodating both multirate realtime traffic and best-effort data traffic. MMCA allows the coexistence of mobile terminals with different multicode transmission capabilities and different QoS requirements. It consists of four parts: 1) the "S_K to S Transformation Algorithm" for identifying the remaining candidate OVSF codes for assignment; 2) the "Multicode Solution Generator" for identifying all possible multicode solutions under a mobile terminal's bandwidth requirement and multicode transmission capability; 3) the criteria for selecting the most feasible multicode solution to serve the mobile terminal; and 4) the "compact index" based code assignment strategy for realizing the selected solution.

In the following, the tree structure and some basic concepts of OVSF codes are reviewed, and the code assignment problem for accommodating mobile terminals with different multicode transmission capabilities and different QoS requirements is formulated in Section II. The MMCA scheme is presented in Section III and the performance of MMCA is evaluated in Section IV.

II. Code Assignment Problem

The major notations used in this paper are summarized in Table I.

Manuscript received August 31, 2004; revised January 25, 2005. This work was supported in part by the Hong Kong Research Grants Council under Grant CUHK 4325/02E and by the Short-Term Research Fellowship program of British Telecommunications (BT). The review of this paper was coordinated by Prof. X. (Sherman) Shen.

Y. Yang is with the Department of Electronic and Electrical Engineering, University College London, London WC1E 6BT, U.K. (e-mail: y.yang@ee.ucl.ac.uk).

T.-S. P. Yum is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (e-mail: yum@ie.cuhk.edu.hk).

Digital Object Identifier 10.1109/TVT.2005.858166

TABLE I
GLOSSARY OF NOTATION

Notation	Definition
$(d_0, d_1, \cdots, d_K)$	A multicode solution. Each element d_k $(0 \le k \le K)$ denotes the number of candidate codes needed in a particular layer k.
F_D	Fairness index of sojourn time for data packets from different multicode capable mobile terminals.
F_R	Fairness index of success probability for realtime calls with different bandwidth requirements.
$g^{(k,m)}$	Compact index of candidate code (k, m).
G_D	Offered traffic of data packets.
G_j	Offered traffic of class-j realtime calls.
G_R	Total offered traffic of realtime calls.
$I^{(k,m)}$	Status index of code (k, m).
J	Number of classes of realtime calls. The class-j $(1 \le j \le J)$ calls have arrival rate λ_j, bandwidth requirement $j \cdot R$, and average call holding time μ_j^{-1}.
(k, m)	The m^{th} code (from left) in layer k.
K	Size of the code tree. There are 2^k codes in layer k $(0 \le k \le K)$.
L	Total offered load of realtime calls.
L_j	Offered load of class-j realtime calls.
λ_D	Arrival rate of data packets.
λ_j	Arrival rate of class-j realtime calls.
μ_D^{-1}	Average packet length of data packets.
μ_j^{-1}	Average holding time of class-j realtime calls.
N	Number of types of mobile terminals. The type-n $(1 \le n \le N)$ terminals can support the simultaneous transmission of n codes.
$\vec{\nu} = (\nu_1, \cdots, \nu_J)$	State vector of the Markov chain for realtime traffic. Each element ν_j $(1 \le j \le J)$ denotes the number of ongoing class-j realtime calls in the system.
p_n	Penetration rate of type-n mobile terminals.
P_j	Blocking probability of class-j realtime calls.
P_R	Overall blocking probability of realtime calls.
Φ	State space of the Markov chain for realtime traffic.
π_0	Limiting probability of the empty state, i.e. no realtime call in the system.
$\pi_{\vec{\nu}}$	Limiting probability of state $\vec{\nu}$.
r	Assignable capacity (in unit of R) of the code tree.
R	Basic/unit data rate supported by a leaf code in layer K.
S	Set of all candidate codes in the code tree.
S_K	Set of leaf candidate codes in layer K.
S_S	Set of all multicode solutions.
$S_A^{(k,m)}$	Ancestor code set of code (k, m).
$S_D^{(k,m)}$	Descendant code set of code (k, m).
$S_i^{(k,m)}$	Set of the i^{th}-layer neighbors of code (k, m).
T	Total throughput of realtime calls.
T_j	Throughput of class-j realtime calls.
$\lvert x \rvert$	Size of set x.
$\lceil x \rceil$	Ceiling function of real number x.
$\lfloor x \rfloor$	Floor function of real number x.

A. Code Tree

OVSF codes can be represented by the nodes in a binary tree [1]. Fig. 1 shows a K-layer code tree.[1] 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. The capacity of the codes in layer k is then $2^{K-k}R$, as shown in Fig. 1.

Layer k has 2^k codes and they are sequentially labeled from left to right, starting from one. The mth code in layer k is referred to as code (k, m). The total capacity of all the codes in each layer is $2^K R$, which is also referred to as the capacity of a K-layer code tree.

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

[1]In some other notational convention, this is referred to as a $(K+1)$-layer tree.

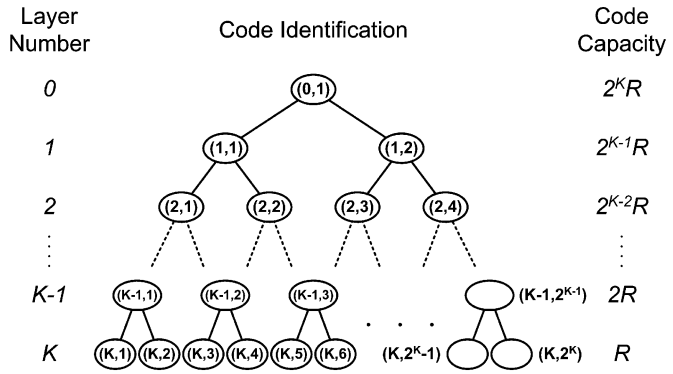

Fig. 1. K-layer code tree.

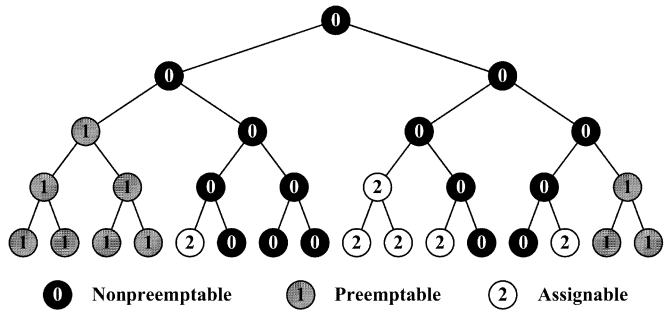

Fig. 2. Occupancy status of a four-layer code tree. Codes $\{(3,4),(4,6),(4,12),(4,13)\}$ are carrying realtime calls, and codes $\{(2,1),(3,8)\}$ are carrying data packets.

root code (0,1). Its descendant code set, denoted by $S_D^{(k,m)}$, contains all the codes in the branch under (k,m). As an example, the ancestor code set of (2,3) is $S_A^{(2,3)} = \{(0,1),(1,2)\}$, and the descendant code set of $(K-1,2)$ is $S_D^{(K-1,2)} = \{(K,3),(K,4)\}$. According to the generation process of OVSF codes [2], [3], code (k,m) is orthogonal to all other codes in the same layer, but not orthogonal to its ancestor or descendant codes.

The positional relationship between the 2^k orthogonal codes in layer k can be identified by tracing their common ancestor codes. Referring to Fig. 1, codes (2,3) and (2,4) are called first-layer neighbors because they share a common parent code (1,2) or, in other words, they are connected by a one-layer sub-tree consisting of these three codes. Codes (2,2) and (2,3) are not first-layer neighbors, but second-layer neighbors because they share a common grandparent code (0,1), or they are connected by a two-layer sub-tree. Besides $\{(2,2),(2,3)\}$, codes (2,1) and (2,4) belong to the same set of these second-layer neighbors. In general, let $S_i^{(k,m)}$ denote the set of the ith-layer neighbors of code (k,m). It contains 2^i layer-k codes (including code (k,m)) that are connected by an i-layer sub-tree, i.e.,

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

The positional relationship between code (k,m) and other layer-k codes can then be represented by k different sets $S_i^{(k,m)} (1 \le i \le k)$. For example, since $S_1^{(2,3)} = \{(2,3),(2,4)\}$ and $S_2^{(2,3)} = \{(2,1),(2,2),(2,3),(2,4)\}$, we know that code (2,3) is "closer" to code (2,4), while equally "far away" from code (2,1) and code (2,2).

B. Occupancy Status

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 *nonpreemptable*. 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*. Similarly, an ancestor code is nonpreemptable if it has some nonpreemptable descendant codes. Preemptable codes can be reassigned to realtime calls by suspending some ongoing packet transmissions. Besides nonpreemptable and preemptable codes, all remaining codes in the tree are *assignable*. They can be freely assigned to carry either realtime calls or data packets. These assignable, preemptable and *nonpreemptable* codes form a partition of the code tree. They can be characterized by status index $I^{(k,m)}$, defined as

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

As an example, consider a four-layer code tree. When codes $\{(3,4),(4,6),(4,12),(4,13)\}$ and codes $\{(2,1),(3,8)\}$ are assigned respectively to realtime calls and data packets, the corresponding status index values of all codes are show in Fig. 2.

C. Problem Formulation

Given the occupancy status of the code tree, the code assignment problem for a realtime call from a terminal is to satisfy its bandwidth requirement under its multicode transmission capability. For a data packet, the problem is to make full use of the terminal's multicode capability for transmitting the data packet as quickly as possible, i.e., at the highest possible data rate. In addition, priority differentiation between realtime calls and best-effort data packets should be supported by the code assignment scheme.

Compared with single-code transmission, multicode transmission is more flexible and, therefore, limits the advantage of code rearrangement on system performance. Many slack capacities in the code tree can be now taken up by the second and third codes, which renders code rearrangement not essential. Also, data packets can absorb the remaining usable capacity left by realtime traffic, so system utilization is improved. Another important issue is that codes in different layers may be assigned to the same mobile user for a single transmission/application. These codes have different spreading factors and hence offer different transmission qualities. This difference should be balanced in code selection.

III. Multicode Multirate Compact Assignment

We propose in this section a multicode nonrearrangeable code assignment scheme called MMCA. The objective 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 newly arrived calls and data 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.
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 qualities 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.

A. Candidate Code Set

Upon receiving a new transmission request (realtime call or data packet), the base station needs to 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. However 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 (3), shown at the bottom of the page. The corresponding candidate code set S can then be derived from S_K by using the "S_K to S Transformation Algorithm" in Appendix A. For example, consider the four-layer code tree shown in Fig. 2. For data packets, $S_K = \{(4,5),(4,9),(4,10),(4,11),(4,14)\}$ and $S = S_K \cup \{(3,5)\}$. While for realtime calls, we have (4) and (5), shown at the bottom of the page.

Before the process of code selection, the base station calculates the assignable capacity of the system according to traffic class. In unit of R, assignable capacity r is defined as the size of leaf candidate code set S_K, or

$$r = |S_K| = \begin{cases} \sum_{m=1}^{2^K} \left\lfloor \frac{I^{(K,m)}}{2} \right\rfloor, & \text{for data packets} \\ \sum_{m=1}^{2^K} \left\lceil \frac{I^{(K,m)}}{2} \right\rceil, & \text{for realtime calls} \end{cases} \tag{6}$$

where $|x|$ denotes the size of set x, and $\lfloor x \rfloor$ and $\lceil x \rceil$ denote, respectively, the floor function and the ceiling function of real number x. For the code tree shown in Fig. 2, assignable capacity is $r = 5$ for data packets and $r = 11$ for realtime calls. Note that a realtime call with bandwidth requirement larger than the assignable capacity is immediately blocked (Condition 1 in Section III-C).

B. Multicode Solution

For a mobile terminal requiring bandwidth $j \cdot R$ and that can transmit n codes, several code combinations, or solutions, may be used. A solution, denoted by $(d_0, d_1, \ldots, d_K)$, consists of $(K+1)$ integers with d_k representing the number of candidate codes needed in layer k. The set S_S of all multicode solutions can be obtained by enumerating all integer combinations under the constraints of bandwidth requirement and multicode transmission capability, i.e.,

$$\sum_{k=0}^{K} d_k \cdot 2^{K-k} = j \tag{7}$$

and

$$\sum_{k=0}^{K} d_k \le n. \tag{8}$$

We propose to use a more efficient algorithm called "Multicode Solution Generator." It starts from the solution $(0, 0, \ldots, 0, j)$, which requires j leaf candidate codes.[2] The next solution $(0, 0, \ldots, 1, j-2)$ is obtained by replacing two leaf codes by one $(K-1)$-layer code in the first solution. Continuing this way, all possible multicode solutions satisfying (7) can be obtained. Next, we use (8) to screen out solutions requiring more than n codes. The detailed algorithm is given in the Appendix.

As an example, consider a four-layer code tree with each solution represented by five integers. Table II lists all multicode solutions for different combinations of bandwidth requirement (from $j = 1$ to $j = 16$) and multicode transmission capability (from $n = 1$ to $n = 6$) obtained from the "Multicode Solution Generator." 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 Table II.

For some combinations of j and n, e.g., $j = 14$ and $n = 2$, no solution exists and symbol "—" is used to indicate that. To accommodate such realtime calls, we gradually increase the value of j until the first solution is found in the table. For the case of $j = 14$ and $n = 2$, multicode solutions

[2]The constraint on multicode transmission capability, or (8), is not taken into consideration at this moment.

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

$$S_K = \{(4,1),(4,2),(4,3),(4,4),(4,5),(4,9),(4,10),(4,11),(4,14),(4,15),(4,16)\} \tag{4}$$

$$\begin{aligned} S &= S_K \cup \{(2,1),(3,1),(3,2),(3,5),(3,8)\} \\ &= \left\{ \begin{array}{c} (2,1),(3,1),(3,2),(3,5),(3,8),(4,1),(4,2),(4,3), \\ (4,4),(4,5),(4,9),(4,10),(4,11),(4,14),(4,15),(4,16) \end{array} \right\}. \end{aligned} \tag{5}$$

TABLE II
MULTICODE SOLUTIONS

Data Rate	Multicode Transmission Capability					
	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

$\{(1,0,0,0,0),(0,2,0,0,0)\}$ are identified when j is increased to 16. The difference between the assigned bandwidth 16 R and the required bandwidth 14 R is called "wasted bandwidth." This waste can be reduced by increasing a terminal's multicode transmission capability.

Multiple solutions may exist for a particular realtime call. To identify the most feasible solution, we apply the following two criteria:

Criterion 1: Choose the solution requiring a larger number of codes. These codes have smaller code capacity and is more "system-friendly" because small-capacity codes are often easier to find and have better transmission qualities at larger spreading factors.

Criterion 2: To break ties, we choose the solution with the minimum variance in code capacity (or spreading factor) so as to balance transmission qualities among these codes.

As an example, for a realtime call with $j = 6$ and $n = 3$, there are three multicode solutions: $\{(0,0,1,1,0),(0,0,0,3,0),(0,0,1,0,2)\}$. According to Criterion 1, $\{(0,0,0,3,0)\}$ and $\{(0,0,1,0,2)\}$, both requiring three candidate codes, are identified as more system-friendly. The solution (0,0,0,3,0), which requires three candidate codes with capacity 2 R, is the final choice since it has a smaller capacity variance.

C. Compact Code Assignment

After selecting the most feasible solution $(d_0, d_1, \ldots, d_K)$, the base station needs to identify and assign d_k candidate codes in layer k (from 0 to K) for accommodating the new mobile user. For single-code transmission $(n = 1)$, compact assignment can offer better performance (in terms of blocking, throughput, and fairness) than random and first-fit assignments [13]. Specifically, compact assignment uses the candidate code in the most congested position so as to keep the resulting code tree as flexible as possible in supporting different bandwidth requirements after each code assignment.

For multicode transmission $(n > 1)$, the assignment of d_k candidate codes in layer k can be seen as multiple single-code assignments for a batch of d_k simultaneous new arrivals with the same bandwidth requirement $(2^{K-k} \cdot R)$ and single-code transmission capability. So the advantage of compact assignment over other nonrearrangeable schemes carries to the multicode case with the exception that 1) only leaf candidate codes (in layer K) can be used for multicode assignments and 2) all mobile terminals are capable of transmitting any number of codes, i.e., no constraint on multicode transmission capability. For this exception case, all code assignment schemes (nonrearrangeable and rearrangeable) will offer the same performance.

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

$$g^{(k,m)} = \sum_{i=1}^{k} |S_i^{(k,m)} \cap S|. \tag{9}$$

Given (k, m) a candidate code in layer k, a smaller value of $g^{(k,m)}$ implies that 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.

The process of code selection and assignment starts from the highest layer with a nonzero integer d_k. After each code assignment, the candidate code set S is updated accordingly. For transmitting a data packet, the base station makes full use of the terminal's multicode capability and select up to n largest capacity codes from S.[3] The objective is to maximize system utilization and transmit the data packet as quickly

[3]When S is empty (i.e., $r = 0$), data packet transmission requests are put in a queue at the base station. These packets will be transmitted as soon as codes are available.

as possible. As an example, for a newly arrived data packet seeing the code tree of Fig. 2, the candidate code set is found to be $S = \{(3,5),(4,5),(4,9),(4,10),(4,11),(4,14)\}$. If the mobile terminal is two-code capable, the base station will assign two candidate codes to carry the transmission of this data packet. As code (3,5) is the only candidate code in layer three, code (3,5) and its descendant codes $\{(4,9),(4,10)\}$ become preemptable. The set S is updated to be $S = \{(4,5),(4,11),\ (4,14)\}$. In layer four, we have $g^{(4,5)} = 6$ and $g^{(4,11)} = g^{(4,14)} = 7$, which implies candidate code (4,5) is in the most congested position and should be selected for assignment.

Now consider a realtime call from a mobile terminal with bandwidth requirement $j = 5$ and multicode transmission capability $n = 3$, the corresponding candidate code set S for Fig. 2 is given by (5). According to the criteria given in Section III-B, multicode solution (0,0,0,2,1) is identified as the feasible choice for supporting this realtime call. Giving high priority to realtime calls, the base station performs code assignments assuming the absence of data packet traffic. In layer three, we get four candidate codes with their compact indices: $g^{(3,1)} = g^{(3,2)} = 8$ and $g^{(3,5)} = g^{(3,8)} = 7$. So codes $\{(3,5),(3,8)\}$ are equivalent in this case and one of them, say (3,5), is randomly selected for assignment. The set S is then updated. The compact indices for the remaining three layer-3 candidate codes are now $g^{(3,1)} = g^{(3,2)} = 7$ and $g^{(3,8)} = 5$. Obviously, code $(3,8)$ is selected for assignment this time and the set S is updated again. Following the same procedure in layer four, codes (4,11) and (4,14) are identified and one of them is selected at random. Note that by using compact assignment, candidate code (2,1) and its descendant codes are kept available for new realtime calls so that the flexibility of the code tree in supporting different bandwidth requirements is maintained.

A realtime call 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: $j \le r$, but the multicode solutions all have bandwidth larger than r, i.e., $\sum_{k=0}^{K} d_k \cdot 2^{K-k} > r$.
- Condition 3: $j \le r$ and $\sum_{k=0}^{K} d_k \cdot 2^{K-k} \le r$, but the number of candidate codes is not sufficient in some layers, 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 is $r = 11$. However, a new call 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 16 R). Another call with $j = 10$ and $n = 3$ will be blocked due to Condition 3 (all multicode solutions, namely (0,1,0,1,0), (0,0,2,1,0) and (0,1,0,0,2), 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 enhancing a 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) [or (4,14)], codes $\{(2,3),(3,6),(4,12)\}$ become assignable. The realtime call with $j = 10$ and $n = 3$ can then be carried in the code tree by using the solution (0,0,2,1,0) and suspending all ongoing packet transmissions. As seen in Table II, when mobile terminals are multicode capable, a number of multicode solutions are usually available. Condition 3 is therefore much less likely to occur, compared to the single-code transmission scenario.

D. Data Packet Transmission

As packet transmissions can be preempted by new realtime calls, some mobile terminals may need to reduce their transmission data rates, or even totally suspend their packet transmissions, to make bandwidth available for accommodating new realtime calls.

To illustrate, let us assume code (2,1) and code (3,8) in Fig. 2 are currently used for packet transmissions by two single-code capable terminals named Terminal-A and Terminal-B. When a new call request from Terminal-C with j=8 and n=6 is received by the base station, multicode solution (0,0,0,2,4) will be identified for code assignment. This means that code (3,8) for Terminal-B is reassigned to Terminal-C for carrying the realtime call. Terminal-B thus has to suspend its packet transmission and record the transmission break point. At the same time, the identity of Terminal-B is put in a queue at the base station. Later, when some codes are released, all assignable capacity in the code tree will be shared by these suspended mobile terminals as fairly as possible.

In addition, one of the following preemptable codes $\{(4,1),(4,2),(4,3),(4,4)\}$, say code (4,1), will be selected and assigned to the realtime call from Terminal-C. As a result, Terminal-A needs to reduce its packet transmission data rate. This reduction in data rate should take into account Terminal-A's multicode transmission capability. Specifically, Terminal-A cannot fully utilize the remaining bandwidth $(4R - R = 3R)$ due to its constraint $n = 1$ on multicode transmission capability. As a result, code (3,2) is used for packet transmission and code (4,2) is released as an assignable code. The bandwidth assigned to Terminal-A is therefore 2 R, although the remaining bandwidth is $3R$.

IV. Performance Analysis

A. Traffic Model

We generalize the traffic model in [13] for performance analysis. Specifically, let there be N types of mobile terminals in the system where the type-$n(1 \le n \le N)$ terminals can support the simultaneous transmission of n codes. Let p_n be the penetration rate of type-n terminals. Further, let there be J classes of realtime calls where the class-$j(1 \le j \le J)$ calls are characterized by

1) Poisson arrivals with rate λ_j;
2) bandwidth requirements equal to $j \cdot R$; and
3) exponentially distributed call holding time with mean μ_j^{-1}.

Let $G_j = \lambda_j/\mu_j (1 \le j \le J)$ 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. Packet length is chosen at equal probabilities from four exponential random variables with means R, 2 R, 4 R and 8 R. 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 Case 1 is the single-code transmission case. 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.

B. Blocking Probability of Realtime Calls

Blocking probability is the most important measure of QoS for realtime calls. Since 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 Condition 2 and Condition 3 (Section III-C) 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 for the restrictive multicode cases studied here. The following is a derivation of this lower bound.

Let ν_j denote the number of ongoing class-j realtime calls in the system. The occupancy status of realtime calls in the code tree can be characterized by a vector $\vec{\nu} \triangleq (\nu_1, \nu_2, \cdots, \nu_J)$. Using $\vec{\nu}$ as the state vector, the code assignment and release process for realtime calls can be modeled by a Markov chain. As an example, Fig. 3 shows the Markov chain model for a two-layer code tree with four classes of realtime calls.

Let Φ denote the state space of the above Markov chain. It contains all possible combinations of ν_j's under the capacity constraint, i.e.,

$$\Phi = \left\{ \vec{\nu} \left| \sum_{j=1}^{J} j \cdot \nu_j \le 2^K \right. \right\}. \tag{10}$$

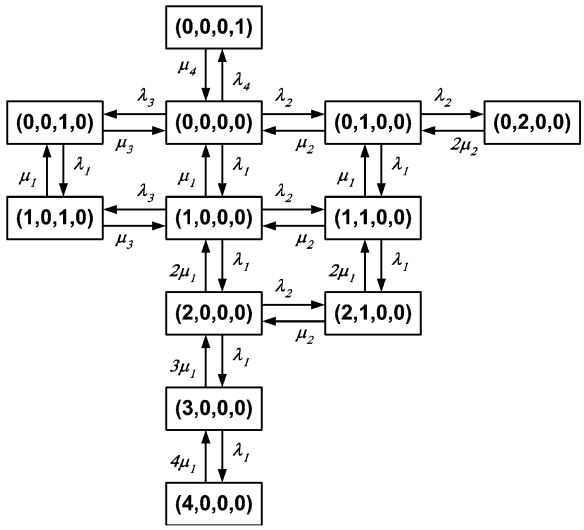

Fig. 3. Markov chain model for realtime calls, $K = 2$ and $J = 4$.

Let $\pi_{\vec{\nu}}$ denote the limiting probability of state $\vec{\nu}$. The solution of $\pi_{\vec{\nu}}$ has a product form [19], or

$$\pi_{\vec{\nu}} = \pi_0 \cdot \prod_{j=1}^{J} \frac{1}{\nu_j!} (G_j)^{\nu_j} \tag{11}$$

where π_0 is the limiting probability of the empty state $(0, 0, \ldots, 0)$ and is given by

$$\pi_0 = \left[\sum_{\vec{\nu} \in \Phi} \prod_{j=1}^{J} \frac{1}{\nu_j!} (G_j)^{\nu_j} \right]^{-1}. \tag{12}$$

At a particular state $\vec{\nu}$, a new class-j realtime call will be blocked if and only if the assignable capacity r is less than j (Condition 1 in Section III-C). Therefore, the blocking probability P_j of class-j realtime calls is given as

$$P_j = \sum_{\vec{\nu} \in \xi_j} \pi_{\vec{\nu}} \tag{13}$$

where $\xi_j \triangleq \{\vec{\nu} | 2^K - \sum_{i=1}^{J} i \cdot \nu_i < j\}$. The overall blocking probability P_R is simply the weighted sum of P_j's, or

$$P_R = \frac{\sum_{j=1}^{J} G_j \cdot P_j}{G_R}. \tag{14}$$

Fig. 4 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 IV-A 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 bound can be achieved by letting all mobile terminals capable of transmitting any number of codes. This result

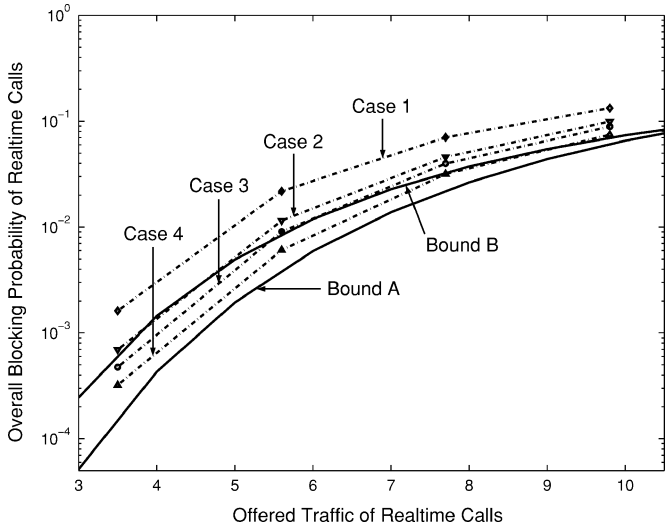

Fig. 4. Blocking probability of realtime calls.

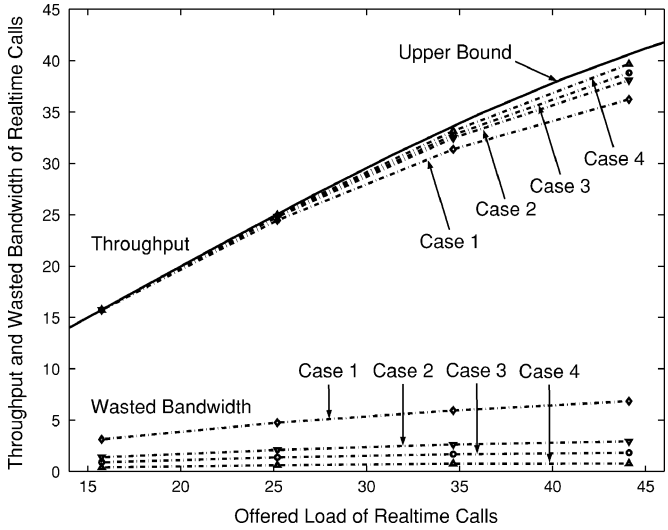

Fig. 5. Throughput and wasted bandwidth of realtime calls.

indicates that the use of multicode is an effective alternative to the rearrangeable single-code scheme in [13]. For comparison purpose, the lower bound for rearrangeable single-code assignment schemes is also shown (marked as Bound B).

C. Throughput and Wasted Bandwidth of Realtime Calls

The offered load L_j of class-j realtime calls is the offered traffic G_j times the bandwidth requirement, i.e., $L_j = j \cdot G_j$. The total offered load L of realtime calls is simply

$$L = \sum_{j=1}^{J} L_j = \sum_{j=1}^{J} j \cdot G_j. \tag{15}$$

The throughput of realtime calls, denoted by T, is

$$T = \sum_{j=1}^{J} (1 - P_j) \cdot L_j. \tag{16}$$

This is the time-averaged required bandwidth by the realtime terminals. The total assigned bandwidth for realtime calls may be larger. For example, to accommodate a type-1 realtime terminal with bandwidth requirement 6 R, the base station needs to assign a layer-$(K-3)$ code (with code capacity 8 R). The gap between these two values is the "wasted bandwidth."

Fig. 5 shows the throughput and wasted bandwidth of realtime calls as a function of offered load L. The solid line is the analytical upper bound, or (16), 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 bandwidth. In the limiting case where all terminals are capable of transmitting any number of codes, the wasted bandwidth is zero.

D. Sojourn Time of Data Packets

For data packets, average sojourn time is a typical QoS measure. It is defined as the time between a transmission request and the successful transmission of the whole packet. Fig. 6

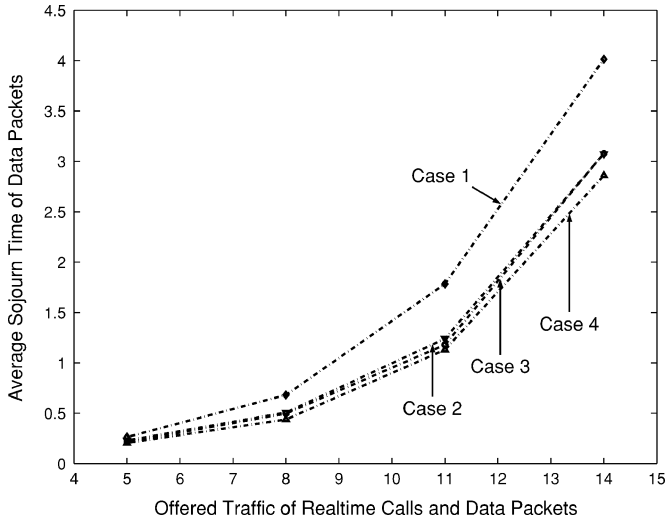

Fig. 6. Average sojourn time of data packets, $G_R : G_D = 7 : 3$.

shows the average sojourn time as a function of total offered traffic $G_R + G_D$. We assume the ratio between realtime traffic 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, i.e., Case 1. This indicates that 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.

E. Fairness Comparison

For realtime calls, 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 realtime terminals with different bandwidth

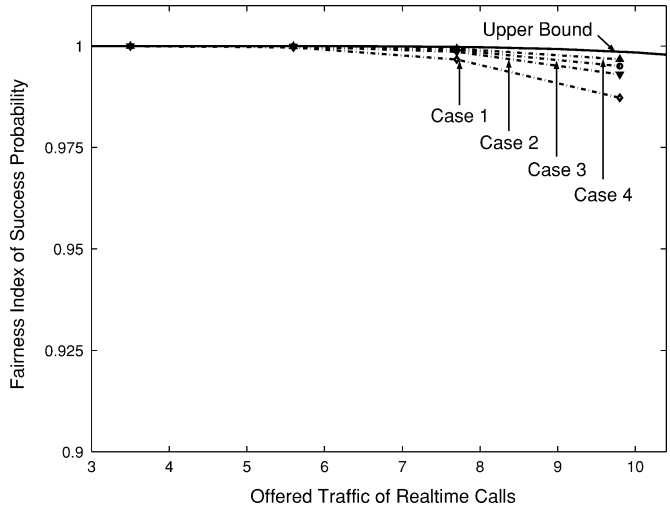

Fig. 7. Fairness index for realtime calls with different bandwidth requirements.

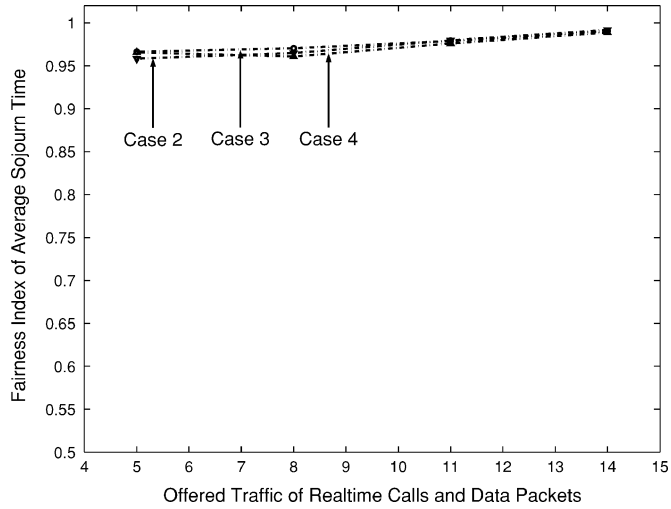

Fig. 8. Fairness index for data packets with different multicode transmission capabilities, $G_R : G_D = 7 : 3$.

requirements, denoted by F_R, is defined as

$$F_R = \frac{\left[\sum_{j=1}^{J}(1-P_j)\right]^2}{J\sum_{j=1}^{J}(1-P_j)^2}. \tag{17}$$

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

Fig. 7 shows F_R as a function of offered traffic of realtime calls. Even under heavy traffic, there is no significant 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 realtime terminals with different multicode transmission capabilities.

As to data packets, the major fairness concern is the average sojourn time for the data terminals with different multicode transmission capabilities. The fairness index F_D is given by

$$F_D = \frac{\left(\sum_{n=1}^{N} D_n\right)^2}{N\sum_{n=1}^{N} D_n^2}, \tag{18}$$

where D_n is the average sojourn time of data packets from type-n terminals.

As seen from Fig. 8, fairness index F_D is close to one for all multicode cases. This indicates that the presence of multicode terminals does not discriminate single-code terminals. So, no special procedure is needed to guard the fairness among data terminals.

V. Conclusion

Based on the concept of compact index, a new OVSF code assignment scheme, namely MMCA, has been proposed for 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 priorities 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 analytical and simulation results. In addition, MMCA is also shown to be a fair code assignment scheme for different service classes and different terminal types.

Compared with rearrangeable code assignment schemes, MMCA can offer comparable blocking and throughput performance with much less computational complexity, especially for the multicode transmission scenario. The complex process of code rearrangement is therefore not cost-effective for marginal performance improvement. In addition, by using the efficient algorithms for deriving candidate code set S and multicode solution set S_S, MMCA is able to achieve the minimum storage complexity; i.e., only the 2^K status index values of leaf codes need to be maintained by the base station for characterizing the occupancy status of the code tree.

Appendix

S_K to S Transformation Algorithm

For newly arrived realtime calls and data packets, the corresponding candidate code set S can be derived from S_K, or (3), by the following algorithm. Note that the mapping from S_K to S is a bijection.

S_K to S Transformation Algorithm

INPUT: the leaf candidate code set S_K.

OUTPUT: the candidate code set S.

1. Let $S = S_K$.
2. WHILE S_K is not empty, repeat the following:
   2.1 Arbitrarily select a code, say (K, m), from S_K.
   2.2 Generate $S_i^{(K,m)} (1 \le i \le K)$ by (1).

2.3 Compute $i_{\max} = \mathrm{Max}\{i | S_i^{(K,m)} \cap S_K = S_i^{(K,m)}\}$.

2.4 FOR $k = 1$ TO $i_{\max}$, repeat the following:

2.4.1 Generate $S_k = \{(K-k, \lceil n/2^k \rceil) | (K,n) \in S_{i_{\max}}^{(K,m)}\}$.

2.4.2 Update $S = S \cup S_k$.

2.5 Update $S_K = S_K - S_{i_{\max}}^{(K,m)}$.

3. Return S.

MULTICODE SOLUTION GENERATOR

For a mobile terminal with bandwidth requirement $j \cdot R$ and multicode transmission capability n, the corresponding set S_S of all possible multicode solutions can be derived by the following algorithm. Note that another approach using dynamic programming technique is given in [18].

Multicode Solution Generator

INPUT: the bandwidth requirement j and the multicode transmission capability n.

OUTPUT: the set of multicode solutions S_S.

1. Initialization: let $\mathrm{Index} = K$, $S_S = \phi$ (empty set), $\mathrm{NewSolution} = \phi$, and $\mathrm{SelectSolution} = \{(0, 0, \ldots, 0, j)\}$.

2. WHILE SelectSolution is not empty, repeat the following:

2.1 Arbitrarily SelectSolution, say

$(d_0^*, d_1^*, \ldots, d_{K-1}^*, d_K^*)$, from SelectSolution.

2.2 FOR $i = 1$ TO $\lfloor d_{\mathrm{Index}}^*/2 \rfloor$, repeat the following:

2.2.1 Update $\mathrm{NewSolution} = \mathrm{NewSolution} \cup$

$\{d_0^*, \ldots, d_{\mathrm{Index}-2}^*, d_{\mathrm{Index}-1}^* + i, d_{\mathrm{Index}}^* - 2i, d_{\mathrm{Index}+1}^*, \ldots, d_K^*\}$.

2.3 Update $S_S = S_S \cup \{(d_0^*, d_1^*, \ldots, d_{K-1}^*, d_K^*)\}$.

2.4 Update $\mathrm{SelectSolution} = \mathrm{SelectSolution} - \{(d_0^*, d_1^*, \ldots, d_{K-1}^*, d_K^*)\}$.

3. IF NewSolution is not empty, THEN do the following:

3.1 Update $\mathrm{Index} = \mathrm{Index} - 1$ and $\mathrm{SelectSolution} = \mathrm{NewSolution}$.

3.2 Let $\mathrm{NewSolution} = \phi$.

3.3 Repeat step 2 and step 3.

4. Update S_S by (8).

5. Return S_S.

ACKNOWLEDGMENT

The many useful comments given by the reviewers have significantly improved the quality of our presentation.

REFERENCES

[1] F. Adachi, M. Sawahashi, and K. Okawa, "Tree-structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile radio," *Electron. Lett.*, vol. 33, pp. 27–28, Jan. 1997.

[2] Spreading and Modulation (FDD). 3GPP TS 25.213 (V6.0.0), Technical Specification (Release 6), Technical Specification Group Radio Access Network, 3GPP, Dec. 2003.

[3] Spreading and Modulation (TDD). 3GPP TS 25.223 (V6.0.0), Technical Specification (Release 6), Technical Specification Group Radio Access Network, 3GPP, Dec. 2003.

[4] R. Fantacci and S. Nannicini, "Multiple access protocol for integration of variable bit rate multimedia traffic in UMTS/IMT-2000 based on wideband CDMA," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 8, pp. 1441–1454, Aug. 2000.

[5] T. Minn and K. Y. Siu, "Dynamic assignment of orthogonal variable-spreading-factor codes in W-CDMA," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 8, pp. 1429–1440, Aug. 2000.

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

[7] W. T. Chen, Y. P. Wu, and H. C. Hsiao, "A novel code assignment scheme for W-CDMA systems," in *Proc. IEEE Vehicular Technology Conf., 2001*, vol. 2, Atlantic City, NJ, Fall 2001, pp. 1182–1186.

[8] M. Dell'Amico, M. L. Merani, and F. Maffioli, "Efficient algorithms for the assignment of ovsf codes in wideband CDMA," in *Proc. IEEE Int. Conf. Communications (ICC'02)*, vol. 5, New York, NY, 2002, pp. 3055–3060.

[9] C. E. Fossa, Jr. and N. J. Davis, IV, "Dynamic code assignment improves channel utilization for bursty traffic in third-generation wireless networks," in *Proc. IEEE Int. Conf. Communications (ICC'02)*, vol. 5, New York, NY, 2002, pp. 3061–3065.

[10] J. S. Park and D. C. Lee, "On static and dynamic code assignment policies in the OVSF code tree for CDMA networks," in *Proc. IEEE Military Communications Conf. (MILCOM'02)*, vol. 2, Anaheim, CA, Oct. 2002, pp. 785–789.

[11] Y. C. Tseng and C. M. Chao, "Code placement and replacement strategies for wideband CDMA OVSF code tree management," *IEEE Trans. Mobile Comput.*, vol. 1, no. 4, pp. 293–302, Oct./Dec. 2002.

[12] A. N. Rouskas and D. N. Skoutas, "OVSF codes assignment and reassignment at the forward link of W-CDMA 3G systems," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC'02)*, vol. 5, Lisboa, Portugal, Sep. 2002, pp. 2404–2408.

[13] Y. Yang and T.-S. P. Yum, "Maximally flexible assignment of orthogonal variable spreading factor codes for multirate traffic," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 781–792, May 2004.

[14] Y. Sekine, K. Kawanishi, U. Yamamoto, and Y. Onozato, "Hybrid OVSF code assignment scheme in W-CDMA," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing (PACRIM'03)*, vol. 1, Victoria, BC, Canada, Aug. 2003, pp. 384–387.

[15] R. G. Cheng and P. Lin, "OVSF code channel assignment for IMT-2000," in *Proc. IEEE Vehicular Technology Conf., 2000*, vol. 3, Tokyo, Japan, Spring 2000, pp. 2188–2192.

[16] C. M. Chao, Y. C. Tseng, and L. C. Wang, "Reducing internal and external fragmentations of OVSF codes in WCDMA systems with multiple codes," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC'03)*, vol. 1, New Orleans, LA, Mar. 2003, pp. 693–698.

[17] F. Shueh and W. S. E. Chen, "Code assignment for IMT-2000 on forward radio link," in *Proc. IEEE Vehicular Technology Conf.*, vol. 2, Spring 2001, pp. 906–910.

[18] L. H. Yen and M. C. Tsou, "An OVSF code assignment scheme utilizing multiple rake combiners for W-CDMA," in *Proc. IEEE Int. Conf. Communications (ICC'03)*, vol. 5, Anchorage, AK, May 2003, pp. 3312–3316.

[19] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. COM-29, no. 10, pp. 1474–1481, Oct. 1981.

Yang Yang (S'99–M'02) received the B. Eng. and M. Eng. degrees in radio engineering from Southeast University, Nanjing, China, in 1996 and 1999, respectively, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2002.

He is currently a Lecturer with the Department of Electronic and Electrical Engineering at University College London (UCL), U.K. Prior to that, he served the Department of Information Engineering at The Chinese University of Hong Kong as an Assistant Professor from August 2002 to August 2003, and the Department of Electronic and Computer Engineering at Brunel University, U.K., as a Lecturer from September 2003 to February 2005. His general research interests include mobile *ad hoc* networks, wireless sensor networks, mobile IPv6, the third generation (3G) mobile communication systems and beyond, dynamic radio resource management (RRM) for integrated services, cross-layer performance evaluation and optimization, and medium access control (MAC) protocols.

Dr. Yang received the First Prize award at the IEEE Hong Kong Section Postgraduate Student Paper Contest in 2001, the Honorable Mention award at ACM Hong Kong Section Postgraduate Research Day in 2002, the Second Prize award at IEEE Region 10 Postgraduate Student Paper Contest in 2002, the Outstanding Ph.D. Thesis award from Faculty of Engineering, The Chinese University of Hong Kong, in 2002, the Young Scientist Award from Hong Kong Institution of Science in 2003, and the Short-Term Research Fellowship from British Telecommunications (BT) in 2004.

Tak-Shing Peter Yum (S'76–M'78–SM'86) was born in Shanghai, China. He received the B.S., M.S., and Ph.D. degrees from Columbia University, New York, in 1974, 1975, and 1978, respectively.

He joined Bell Telephone Laboratories, Holmdel, NJ, in April 1978, working on switching and signalling systems. In 1980, he accepted a teaching appointment at the National Chiao Tung University, Hsinchu, Taiwan, R.O.C. In 1982, he joined The Chinese University of Hong Kong, where he is now Professor of Information Engineering. He has published original research on packet switched networks with contributions in routing algorithms, buffer management, deadlock detection algorithms, message resequencing, and multiaccess protocols. He worked on the design and analysis of cellulars, lightwave, and video distribution networks. His recent works are on the technologies for 3G and IP networks. His website is at http://www.ie.cuhk.edu.hk/yum