Dynamic Bandwidth Allocation for Multimedia Traffic with Rate Guarantee and Fair Access in WCDMA Systems

Chih-Min Chao, Yu-Chee Tseng, *Senior Member*, *IEEE Computer Society*, and
Li-Chun Wang, *Member*, *IEEE*

Abstract—Packet scheduling in a WCDMA system poses a new challenge due to its nature of variable bit rates and location-dependent, time-varying channel conditions. In this work, three new downlink scheduling algorithms for a WCDMA base station are proposed to support multimedia transmissions. Using a credit management and a compensation mechanism, our algorithms provide rate guarantee and fair access to mobile terminals. In particular, we propose to allow a user to simultaneously use multiple OVSF codes in a time-sharing manner, which we call a *multicode*, *shared* model. Using multiple codes allows us to compensate those users suffering from bad communication quality or even errors. The proposed schemes can tolerate a multistate link condition (compared to the typically assumed two-state, or good-or-bad, link condition) by adjusting the number of OVSF codes and the spreading factor of each code. Simulation results show that the proposed schemes do achieve higher bandwidth utilization while keeping transmission delay low.

Index Terms—Mobile computing, OVSF, personal communication services, WCDMA, wireless communication, 3G.

1 Introduction

Supporting multimedia applications with quality of service (QoS) requirements is one of the ultimate goals in next-generation wireless systems. The current second-generation (2G) systems, such as IS-95 (CDMA) and GSM (TDMA), are circuit-switched, fixed-rate, and voice-traffic-oriented and, thus, not appropriate to support multimedia services. The third-generation (3G) wireless standards [1], [2], [12] will be based on the WCDMA technologies and can flexibly support mixed and variable-rate services. Two transmission schemes are proposed in the 3G wireless standard: multicode-CDMA (MC-CDMA) and Orthogonal Variable Spreading Factor CDMA (OVSF-CDMA). In MC-CDMA, multiple Orthogonal Constant Spreading Factor (OCSF) codes can be assigned to a user [10], [13]. The maximum data rate a user can receive depends on the number of transceivers in the device. In OVSF-CDMA, a single OVSF code can provide a data rate that is several times than that of an OCSF code, depending on its spreading factor [2].

In CDMA systems, multiple connections are allowed to receive packets simultaneously. This is opposite to TDMA systems where only one connection can be active at any moment. Thus, the scheduling problem for WCDMA systems is harder than that for TDMA systems. It can neither be solved directly by wireline scheduling disciplines, such as WFQ [19], virtual clock [27], and EDD [15], since they do not consider the variability of wireless connections, nor be solved by wireless scheduling strategies, such as IWFQ [16], CIF-Q [18], and SBFA [21], since they only consider a single-server, two-state link model.

This paper considers the bandwidth allocation problem in an OVSF WCDMA system. In the literature, solutions to this problem can be classified depending on two factors:

- *single-code/multicode*: Whether a user can simultaneously utilize multiple OVSF codes or not.
- *dedicated/shared*: Whether a code can be time-shared by multiple users or not.

In Table 1, we categorize existing solutions and the solutions proposed in this paper based on such classification.

The OVSF code assignment strategy, though playing an important role in system performance, is not explicitly addressed in the current 3G WCDMA standard. Most works in the literature [3], [6], [9], [17], [23], [26] fall into the dedicated, single-code class. One OVSF code is exclusively assigned to one client until the call is terminated or reallocated. Intelligently assigning codes to calls can reduce *code blocking* [23], [26]. Solutions [5], [7], [22], [25] allow a user to occupy multiple but dedicated OVSF codes. Such solutions are more flexible and, thus, can reduce code blocking.

Since wireless bandwidths are limited resources, it is usually desirable that a code can be time-shared by multiple users. The advantage is higher flexibility in utilizing bandwidth, especially in handling bursty traffic. Allowing sharing means that we need to decide "which code can be used by which user at what time." So, a solution should comprise a code assignment strategy and a scheduling

- C.-M. Chao is with the Department of Computer Science, National Taiwan Ocean University, 20224, Taiwan. E-mail: cmchao@axp1.csie.ncu.edu.tw.
- Y.-C. Tseng is with the Department of Computer Science and Information Engineering, National Chiao-Tung University, 30050, Taiwan. E-mail: yctseng@csie.nctu.edu.tw.
- L.-C. Wang is with the Department of Communication Engineering, National Chiao-Tung University, 30050, Taiwan. E-mail: lichun@mail.nctu..edu.tw.

Manuscript received 20 Aug. 2003; revised 26 Jan. 2004; accepted 9 Mar. 2004; published online 27 July 2005.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0131-0803.

1536-1233/05/$20.00 © 2005 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

TABLE 1
Classification of Bandwidth Allocation Solutions for WCDMA Systems

	dedicated	shared
single-code	[3, 6, 9, 17, 23, 26]	[14, 24]
multi-code	[5, 7, 22, 25]	ours

scheme. Recently, two solutions in the single-code, shared category are proposed [14], [24]. In [24], a packet scheduling scheme is provided but the code allocation issue is not addressed. A *credit-based* scheduling scheme is proposed in [14]. However, this scheme does not consider channel quality, which may impact the selected spreading factor. A more comprehensive review is in Section 1.1.

In this paper, we propose three new credit-based bandwidth allocation schemes that allow a user to utilize multiple time-shared codes. This is more flexible than what was proposed in existing works. With our credit management mechanism and the compensation mechanism provided by additional codes, the schemes can provide fair access and data rate guarantee. Using multiple OVSF codes has two purposes. First, it is for compensation purpose when a terminal encounters errors. Second, it can adapt to an environment with multiple link states, in the sense that a higher spreading factor can be used when the link quality is bad; however, the user can still enjoy the guaranteed bandwidth supported by multiple codes. Simulation results show that the proposed schemes can achieve higher bandwidth utilization and keep average delay low.

The rest of this paper is organized as follows: In Section 2, we introduce the system model. Section 3 presents the proposed dynamic bandwidth allocation algorithms. Performance evaluations are in Section 4. Concluding remarks are given in Section 5.

1.1 Related Works

In the dedicated, single-code class, [23] shows that significant improvement can be obtained by using a *leftmost-first* or a *crowded-first* strategy, compared to a random assignment. When a code tree is used for long time, it is possible that the tree may become too fragmented, due to calls arriving and leaving the system. Code replacement can be conducted to improve bandwidth utilization. The *dynamic code assignment* (DCA) scheme [17] is designed for this purpose. In this case, code assignment is *semidedicated*, in the sense that a call can be moved to a different code when relocation is conducted.

Allowing a user to occupy multiple dedicated codes provides more adaptability. The works in [7], [22] address the relationship between a requested data rate and the number of codes to be assigned to the request. However, the placement of codes in the OVSF code tree is left unspecified. A multicode scheduling scheme to be run on the base station is proposed in [25], where fair scheduling for users to share the resources is addressed. Again, the important code placement issue is left unaddressed. In [5], both the code placement and replacement issues are addressed for such an environment.

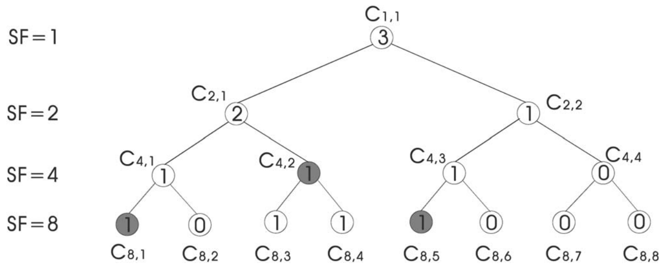


Fig. 1. A code blocking scenario (busy codes are marked by gray).

The scheme proposed in [14] is a member in the single-code, shared class. Initially, each user is assigned a leaf code (of the lowest rate). However, the scheduling algorithm would allow a user to utilize any ancestor code of his/her assigned code. In each time frame, the algorithm will repeatedly pick the user with the highest credit, say A, and try to move A's code one level up from its current stand if A still has backlog packets to be sent. This increase also means sacrificing other users' assigned, but unused, codes. That is, the latter will be inhibited from transmission in this round. The process is repeated until all capacities in the code tree are consumed or no packet is pending. Inhibited users' credits will be increased in the next frame. This scheme fails to consider the possible higher bit error rate (BER) that is caused by using a lower spreading factor. Depending on the channel condition, the spreading factor should not be unlimitedly reduced. Otherwise, a lot of retransmissions may be incurred, resulting in even worse performance. The scheme also includes an exception handling mechanism to solve the starvation problem that may occur during scheduling.

2 System Model

In the WCDMA system [20], data transmission involves two steps. The first step is *channelization*, which transforms every data bit into a code sequence. The length of the code sequence per data bit is called the *spreading factor (SF)*, which is typically a power of two. The second step is *scrambling*, which applies a scrambling code to the spread signal. Scrambling codes are used to separate signals from different sources, while channelization codes multiplex data to different users from the same source.

The OVSF codes are used as the channelization codes in the WCDMA system. The possible OVSF codes can be represented by a code tree as shown in Fig. 1. Each OVSF code is denoted as $C_{SF,k}$, where SF is the spreading factor and k is the branch number, $1 \leq k \leq SF$. The number of codes at each level is equal to the value of SF. All codes in the same layer are orthogonal, while codes in different layers are orthogonal only if they do not have ancestor-descendant relationship. Leaf codes have the minimum data rate, which is denoted by $1R_b$. The data rate is doubled whenever we go one level up the tree. For example, in Fig. 1, $C_{4,1}$ has rate $2R_b$, and $C_{2,1}$ has rate $4R_b$. The resource units in a code tree are codes. Fig. 1 shows an example where codes $C_{4,2}$, $C_{8,1}$, and $C_{8,5}$ are occupied by users. The remaining capacity is $4R_b$. If the dedicated, single-code model is assumed, a new call requesting a rate of $4R_b$ will be rejected because there is no such code available. Such a situation is called *code blocking*. However, the problem can be solved if up to three codes can be used by a user under the multicode model.

A main characteristic of wireless communications is link variability, which could be *time-dependent* and *location-dependent*. It is time-dependent because interference can come to hit at any time. One example frequently seen is bursty errors. The capacity of a wireless link is location-dependent because a longer distance typically has to suffer lower transmission rate. These properties necessitate the design of dynamic bandwidth allocation mechanisms.

Since the quality of a wireless link is both time and location-dependent, we assume that its capacity follows a multistate model. Specifically, a wireless link's symbol error rate is inversely proportional to its received signal strength, which is given by $\frac{E_b}{N_0} = SNR \times SF$, where E_b is the energy per bit, N_0 is two times the power spectral density of additive white Gaussian noise, SNR is the signal-to-noise ratio, and SF is the spreading factor. To achieve a target E_b/N_0, one can either increase SNR through power control or increase the spreading factor SF. When the transmission power is up to a limit, increasing SF is the only way to reduce bit error rate. In this work, we assume that there exists a target BER for all users. As a result, at any moment, there is always a maximum data rate (and, thus, minimum spreading factor) for each user. In such an environment, the scheduling problem is still an open question [4]. Most existing scheduling strategies [8], [11], [16], [18], [21] consider a simple two-state channel model where a channel can be either *good* or *bad*, receiving full or none capacity, respectively.

This paper considers the downlink bandwidth allocation problem for a base station. The resource at the base station is an OVSF code tree. Each user i, when entering the system, needs to specify its peak data rate p_iR_b and guaranteed data rate a_iR_b. Both p_i and a_i are powers of two and $p_i \geq a_i$. The base station maintains a queue Q_i for user i's packets to be delivered. So, the inputs to our scheduling algorithms are p_i, a_i, and Q_i, for all users $i = 1..n$. The base station is responsible for utilizing its code tree to serve users efficiently and fairly. According to the current 3G standard, each frame is 10 ms, which consists of 15 slots. So, in every 10 ms, we can reschedule codes for users. The purpose of bandwidth allocation is to achieve high utilization while providing guarantee rates for individual users.

We assume the shared, multicode model, where a user can simultaneously use up to N_{max} codes. This model has several advantages: 1) when a user suffers severe interference, we can increase the spreading factor to improve reliability but maintain the promised data rate by using multiple codes, and 2) when a user suffers temporary degradation, compensation can be made up to it by using multiple codes for fairness.

3 Dynamic Bandwidth Allocation Strategies

This section presents our dynamic bandwidth allocation strategies to provide QoS-guaranteed services. Our solutions consist of three parts: code assignment, credit management, and packet scheduling.

3.1 Code Assignment Part

This part decides which codes should be assigned to each user to satisfy his/her demand. We assume that there is an admission control mechanism such that a user is accepted only if the sum of guaranteed data rates over all users does not exceed the capacity of the code tree. For each code k in the code tree, the base station maintains a variable (SC_k) (read as shared count) to keep track of the number of users who are currently sharing k, either partially or completely. Specifically, a user is said to share code k if the user is using code k, a descendant code of k, or an ancestor code of k. For example, in Fig. 1, the number in each node is the corresponding code's shared count. It is easy to maintain the shared counts. Whenever a code k is allocated to a user, the shared counts of k and k's ancestor and descendant codes are all increased by one. These counts are decreased by one when this user leaves.

Note that, in contrast to the code assignment in [14], which assigns a leaf code to each user initially, we allocate codes to users according to their maximal data rates. During transmission, a user can use any of the descendent codes of the initially assigned one, which is compatible to the 3gpp standard [1].

3.1.1 Scheme 1: Multiple Fixed Codes

In this scheme, we allocate N_{max} fixed codes, each of rate $p_i \cdot R_b$, to user i. Under normal situations, only one code is sufficient. The additional codes can be used for interference reduction or bandwidth compensation. This will be further elaborated on in Section 3.3.

Given user i's request, the scheme sequentially picks N_{max} codes. To allocate each code, the following rules are applied:

1. Scan all codes in the code tree with rate p_iR_b. Pick the one(s) with the least shared count. If there is only one candidate, assign this code to user i. Otherwise, go to step 2.
2. If the shared counts of these candidate codes are 0s, we follow the *crowded-first* rule to do the allocation. Specifically, if x and y are two codes such that $SC_x = SC_y = 0$, we compare x's and y's parent codes. The one with a larger SC is selected. If they tie again, we further compare their parents. This is repeated until the tie breaks. One special case is that x's and y's parents could be the same code. If so, we simply select the one on the left-hand side.
3. If the shared counts of the candidate codes are nonzero, we follow the *fairness* rule. Specifically, if x and y are two codes such that $SC_x = SC_y \neq 0$, we compare x's and y's parent codes similar to the recursive procedure in step 2. However, we select the code with a smaller shared count, instead of a larger one.

Intuitively, when there are still free codes, we try to place users' requests in the code tree as compact as possible (this is what suggested in [23]). Otherwise, some degree of sharing among users is necessary, and we try to assign codes as fairly as possible.

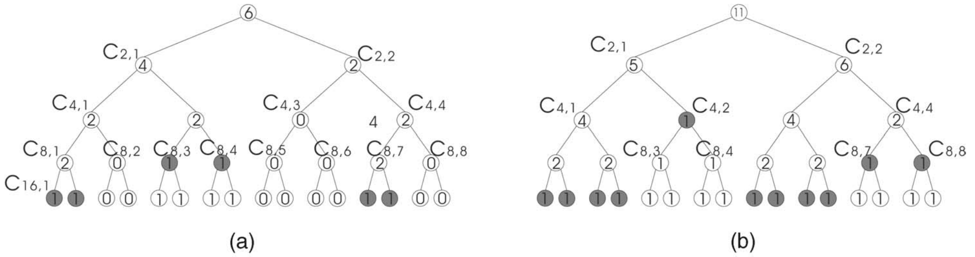


Fig. 2. A code assignment example when the code tree has (a) free $2R_b$ codes and (b) no free $2R_b$ code. The number in each node is its shared count.

Take the scenario in Fig. 2a as an example. Suppose $N_{max} = 2$ and a new user requests for a $2R_b$ rate. After searching for all $2R_b$ codes, we find that $C_{8,2}$, $C_{8,5}$, $C_{8,6}$, and $C_{8,8}$ all have zero shared counts. So, we further compare their ancestors, $C_{4,1}$, $C_{4,3}$, and $C_{4,4}$. Among them, $C_{4,1}$ and $C_{4,4}$, which have the same shared counts of two, tie again. After going one level up, we see that $C_{2,1}$ has a larger shared count than $C_{2,2}$. So, $C_{8,2}$ is allocated to the user. The same procedure is applied to allocate the second code. This time $C_{8,8}$ will be picked. Another example where the code tree is fully occupied is shown in Fig. 2b. Again, assuming the same situation, step 1 will pick $C_{8,3}$, $C_{8,4}$, $C_{8,7}$, and $C_{8,8}$, all of which have the same shared count $= 1$. After going one level up, we see that $C_{4,2}$ has a smaller shared count. Since $C_{8,3}$ and $C_{8,4}$ tie, following the leftmost-first rule, we will select $C_{8,3}$. To assign the second code, we see that $C_{8,4}$, $C_{8,7}$, and $C_{8,8}$ all have the same least SC value. Their ancestors, $C_{4,2}$ and $C_{4,4}$, tie again with the same SC value of two. Another tie is found when coming to codes $C_{2,1}$ and $C_{2,2}$. Since $C_{2,1}$ and $C_{2,2}$ have the same ancestor, the leftmost candidate, $C_{8,4}$, will be assigned to the user.

3.1.2 Scheme 2: Single Fixed Code with Multiple Dynamic Codes

This scheme only assigns one fixed code to each user. The other $N_{max} - 1$ codes are all assigned in a dynamic manner. Specifically, the OVSF code tree is partitioned into two areas: *partially shared area* on the left and *fully shared area* on the right (how to select a good partition point will be evaluated in Section 4). Given a user requesting a rate $p_i R_b$, it is assigned a code of rate $p_i R_b$ in the partially shared area. The assignment follows the same rule in the previous scheme except that only the partially shared area is assignable. The remaining $N_{max} - 1$ codes will be assigned to the fully shared area, but the assignment will not be done in this stage and will be decided at the scheduling stage.

3.1.3 Scheme 3: No Fixed Codes

In contrast to the previous schemes, this scheme does not assign any fixed code to users. All N_{max} codes will be allocated dynamically at the scheduling stage. This scheme provides the maximal flexibility (alternatively, one can consider the whole OVSF code tree as a fully shared area in Scheme 2).

3.1.4 Remark on Code Notification

The codes used in our schemes can be directly mapped to the transport channels in 3gpp. To schedule users, the *Forward Access Channel (FACH)* can be used to notify them that their transmission data rates as well as codes. An additional ID field is needed in the header to distinguish users (we will analyze the signaling overhead in Section 3.4). In schemes 1 and 3, all codes allocated to a user can be mapped to the *Dedicated Transport Channel (DCH)*. In scheme 2, the single fixed code can be mapped to DCH, while the remaining dynamic codes to *Downlink Share Channel (DSCH)*. According to 3gpp, a DSCH is always associated with a DCH, which provides necessary signaling for the DSCH.

3.2 Credit Management Part

In the above part, we already assigned each user multiple codes which exceed the user's guaranteed data rate. These codes are not necessarily always used by the user. We employ a credit management mechanism to dynamically allocate bandwidths to users. The base station maintains a credit C_i for each user i. Initially, $C_i = 0$. After every 10 ms, a_i credits, i.e., user i's guaranteed rate, are granted to user i and added to C_i. However, for every code of rate $2^k R_b$ that is consumed by user i, 2^k credits are paid by decreasing 2^k from C_i. Hence, as a user's perceived data rate is below its guaranteed rate, its credits can be accumulated for later use. The value of C_i can also be negative, if the user is over-served (which happens, for example, if other users under-use their capacities due to poor channel condition). Long-term fairness and rate guarantee are provided if the base station honors all users' credits.

Care should be taken if a user's perceived data rate is below his/her request rate for long time, due to reasons such as poor channel condition, sudden congestion, or simply low data arrival rate. In this case, a lot of credits may be accumulated for the user. Later on, if a large amount of traffic arrives for the user, the user may take up a lot of bandwidth, thus blocking the opportunity of other users. This is alright if only long-term rate guarantee is required, but it does cause a problem viewed for a short term. Short-term fairness is not guaranteed if we allow a user to accumulate credits unlimitedly. To resolve this problem, we propose to maintain a credit limit L_i for each user i such that $C_i \leq L_i$ is always true. The value of L_i can be negotiated with the base station in a per-user basis when the connection was first established according to the user's priorities, degree of burstiness, or even the system loading.

Note that we do not distinguish the situation that a user has lower traffic than expected from that it suffers from bad channel condition. In both situations, it can accumulate credits. An alternative is to apply the credit limit only in the

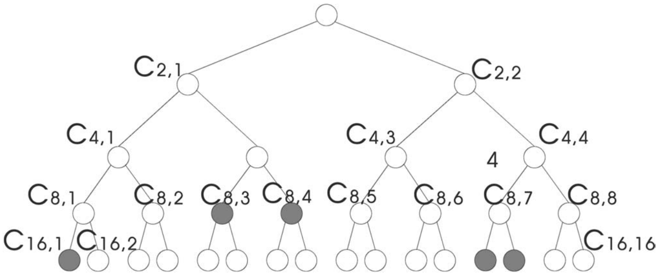


Fig. 3. A scheduling example. Codes marked by gray are occupied.

former case [14]. However, we believe that it is more reasonable to apply the limit to both cases because wireless channels may suffer from burst errors more frequently.

3.3 Packet Scheduling Part

The scheduling algorithm is executed once every 10 ms (frame length). It examines all users with backlog data at the base station and determines which users are to be served by which codes. The goal is to provide a fair and rate-guaranteed service to each user.

3.3.1 Scheduling for Scheme 1 (Multiple Fixed Codes)

To facilitate the scheduling scheme, we introduce the term *normalized credit* of user i, which is defined to be the ratio C_i/a_i. The normalized credit represents the amount of services a user i should receive relative to his/her guaranteed rate a_i. A user with the greatest normalized credit, instead of credit, should be scheduled first. The scheduling algorithm for scheme 1 has six steps:

- STEP 1. Sort all users according to their normalized credits.
- STEP 2. Pick the user, say i, with the greatest normalized credit such that user i has backlog packets and at least one of user i's codes is free. If there is no such user, the procedure terminates.
- STEP 3. Let M_i be the maximum bit rate for user i such that the BER is satisfied (this can be obtained by monitoring the channel condition).
- STEP 4. Let $T_i = min\{M_i, p_i\}$, which represents the feasible transmission rate considering user i's channel quality. We search all free codes of user i. If there exists at least one free code, we allocate the leftmost one for user i. Otherwise, we go one level down by searching all descendant codes (of rate $\frac{T_i}{2}R_b$) of user i's codes. If there exist at least one free code, we allocate the leftmost one for user i. Otherwise, we repeat the same procedure and search the descendant codes of rates $\frac{T_i}{4}, \frac{R_i}{8}, \cdots$, etc.
- STEP 5. For the code allocated in STEP 4, we decrease user i's credit C_i by the corresponding amount. Then, user i is put in to the sorted list.
- STEP 6. Go back to STEP 2.

Take Fig. 3 as an example. Suppose that user i has the greatest normalized credits and $C_{8,1}$, $C_{8,2}$, and $C_{8,3}$ are user i's codes. When user i is selected for the first time, only $C_{8,2}$ is free. So, $C_{8,2}$ is allocated to serve user i in the next frame and two credits are subtracted from C_i. If user i is selected for the second time, $C_{16,2}$ will be allocated to it.

3.3.2 Scheduling for Scheme 2 (Single Fixed Codes with Multiple Dynamic Codes)

The scheduling algorithm is similar to the previous case. The difference lies on how a user's codes are used. All steps are the same as the previous case, except Step 4, which is described below:

- STEP $4'$. If this is the first time user i is selected in this frame, go to (a). Otherwise, go to (b).
  - (a) Let $T_i = min\{M_i, p_i\}$. We search the single fixed code of user i. We allocate it to user i if it is free. Otherwise, we go one level down by searching all descendant codes (of rate $\frac{T_i}{2}R_b$) of user i's code. If there exists at least one free code, we allocate the leftmost one to user i. Otherwise, we repeat the same procedure and search the descendant codes of rates $\frac{T_i}{4}, \frac{T_i}{8}, \cdots$, etc. If we cannot find a free code, user i is skipped for further scheduling in the current time frame.
  - (b) Allocate a free code at the fully shared area (this is similar to case (a) but we have more freedom because any free code can be allocated).

Again, take Fig. 3 as an example. Let the quarter of the code tree on the right-hand side be the fully shared area. Suppose that user i has the highest normalized credit and $C_{8,1}$ is his/her initial code. The scheduling algorithm will allocate $C_{16,2}$ to user i in the first round. If user i is selected again for next two rounds, $C_{16,15}$ and $C_{16,16}$ in the fully shared area will be allocated to it. On the contrary, if $C_{8,3}$ is user i's initial code, it cannot be scheduled in the current frame because $C_{8,3}$ is already occupied by another user.

3.3.3 Scheduling for Scheme 3 (No Fixed Codes)

The algorithm is similar to the previous two schemes. It follows scheme 1, except the following modifications:

- STEP $2''$. The same as STEP 2 except that we only need to make sure that user i has not exhausted its N_{max} codes.
- STEP $4''$. The same as STEP 4 except that any free space in the code tree can be used by user i.

3.4 Time Complexity and Signaling Overhead Analysis

Next, we analyze the time complexity of our schemes. Consider the code assignment part. There are SF codes in the code tree with a spreading factor of SF. So, the cost is $O(SF)$ to search for such a code. For the fixed code assignment part of scheme 1, we may need to compare the shared counts of candidate codes' ancestors. Each time when we go one level up the code tree, the number of ancestors reduces by half. It is easy to see that the cost to allocate a code with a specific SF is

$$SF + \frac{SF}{2} + \cdots + 1 = O(SF).$$

Thus, to assign N_{max} codes, the searching cost is $O(N_{max} \times SF) = O(SF)$ because N_{max} is a constant. The time complexity for the fixed code assignment part of scheme 2 is similar to that of scheme 1, except that the searching domain is restricted to the partially shared area. So, the time complexity is the same.

For the credit assignment part, it is easy to see that the time complexity is $O(n)$ for all three schemes, where N is the number of users currently in the system. For the packet scheduling part, all three schemes need to sort all users, giving a cost of $O(n \log n)$. The STEP 4 of scheme 1 may spend $1 + 2 + \cdots + \frac{SF_{max}}{SF} = O(\frac{SF_{max}}{SF})$ time to find a free code, where SF_{max} is the maximum allowable SF. So, the total searching cost is $O(n' \times \frac{SF_{max}}{SF})$, where n' is the actual number of transmissions that are scheduled in this time frame. STEP 5 takes $O(n)$ to put the scheduled user back to the sorted list. For scheme 2, the searching cost of STEP 4′(a) is $O(n_p \times \frac{SF_{max}}{SF})$, where n_p is the number of scheduled transmissions in the partially shared area. In STEP 4′(b), each code assignment takes $O(SF)$ time, so the total cost is $O(n_f \times SF)$, where n_f is the number of scheduled transmissions in the fully shared area. Similarly, for scheme 3, the searching cost of STEP 4″ is $O(n' \times SF)$.

Overall, the time complexity for scheme 1 is

$$O\left(SF + n \log n + n' \times \frac{SF_{max}}{SF}\right).$$

For scheme 2, the time complexity is

$$O\left(SF + n \log n + n_p \times \frac{SF_{max}}{SF} + n_f \times SF\right).$$

For scheme 3, the time complexity is

$$O(SF + n \log n + n' \times SF).$$

The complexities are typically low. For example, in our simulations, with $SF_{max} = 256$, the value of n is around 68, which should take less than tens of microseconds to complete these calculations in modern processors.

The signaling overhead for scheme 1 is $n' \times \lceil \lg S_1 \rceil$, where S_1 is the maximum number of users that use the same code. For scheme 2, the signaling overhead is $n_p \times \lceil \lg S_2 \rceil + n_f \times \lceil \lg n_2 \rceil$, where S_2 is the maximum number of users that use the same code in the partially shared area and n_2 is the number of codes in the fully shared area. The signaling overhead for scheme 3 is $n' \times \lceil \lg n_3 \rceil$, where n_3 is the number of codes in the code tree.

4 Performance Evaluation

We have implemented a simulator to evaluate the performance of the proposed strategies. The max SF is set to 256. We control the call generation such that the overall guaranteed traffic load falls between 10 percent and 90 percent. Three traffic models are tested [14]:

- Model I (constantly backlogged model): Each user has queued packets all the time. Each user's peak transmission rate is uniformly distributed between $4R_b$ and $32R_b$, and guaranteed rate one fourth of the peak rate (i.e., $a_i = \frac{1}{4}p_i$).
- Model II (highly bursty traffic model): Calls are generated into the system with a peak rate uniformly distributed between $4R_b$ and $32R_b$ and a guaranteed rate equal to one fourth of the peak rate. Packet arrival of each connection follows a 2-state Markov model. A connection can transit between an *idle* state or an *active* state. No packet is generated during the idle state, while P packets per 10 ms are generated during the active state, where P is uniformly distributed between $2a_i$ and $4a_i$. A state transition can be made every 10 ms, with probabilities of 1/3 and 2/3 from idle to active and from active to idle, respectively.
- Model III (lowly bursty traffic model): Calls are generated into the system with a peak rate uniformly distributed between $2R_b$ and $16R_b$ and a guaranteed rate equal to one half of the peak rate. Packet arrival also follows the above 2-state Markov model, but P is uniformly distributed between a_i and $2a_i$, while state transition probabilities are 2/3 and 1/3 from idle to active and from active to idle, respectively.

Each transport block is set to 150 bits, which means a $1R_b$ code can transmit 150 bits every 10 ms. A transport block will be retransmitted if an error is detected by the cyclic redundancy check (we assume no channel coding is applied). In schemes 1, 2, and 3, N_{max} ranges from 2 to 4. In scheme 2, the first three quarters of the code tree is the partially shared area, and the last quarter is the fully shared area.

Using the QPSK modulation, we model the symbol error probability by

$$P_E \approx 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \qquad (1)$$

where

$$Q(x) = \int_x^{\infty} \frac{e^{-y^2/2}}{\sqrt{2\pi}} dy$$

and

$$\frac{E_b}{N_0} = SNR \times SF.$$

Intuitively, the transmission rate is inversely proportional to the SF being applied. A lower SF has a higher transmission rate but the potential penalty, according to (1), is a higher P_E. To achieve a better throughput, we should choose the smallest SF that meets the required P_E. A scheme without considering channel condition, such as [14] (denoted by KMS below), is impractical because it may choose an improper SF and, thus, suffer from severe performance degradation. In this paper, the upper bound of P_E is set to 10^{-5}. We assume a Rayleigh fading channel, so signal power is modeled by an exponential random variable X with mean γ, i.e., $f(x) = \frac{1}{\gamma} e^{\frac{-x}{\gamma}}$ for $x \geq 0$ [20].

We first experiment on the KMS scheme on traffic model II by varying γ and evaluating the *unsatisfactory transmission*, which is defined to be the percentage of frames that experiences a symbol error rate P_E exceeding the

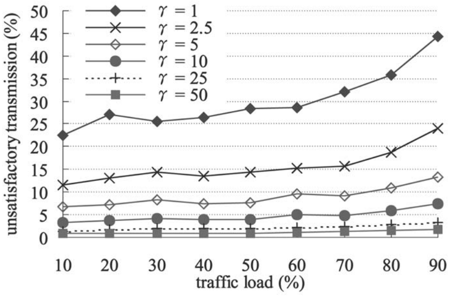


Fig. 4. Unsatisfactory transmission versus traffic load at different γ for the KMS scheme [14].

threshold 10^{-5}. As shown in Fig. 4, KMS performs highly depending on the channel condition. At low γ, a lot of packets may experience failures. So, the scheme is not channel-sensitive. To keep the failure rate low, say under 5 percent, γ should be maintained 25 or higher. Note that traffic load would increase the error rate because the KMS scheme favors users with more backlogged packets. This would aggravate error transmission.

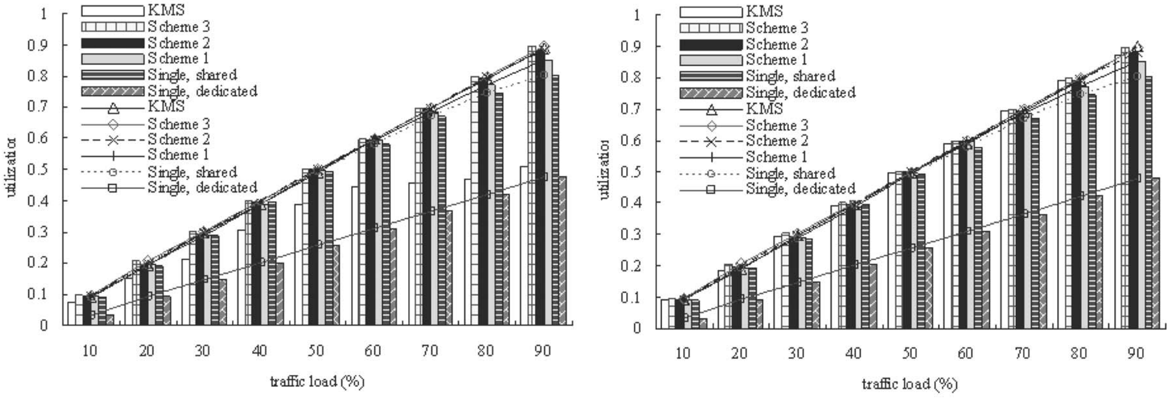
Below, we further experiment on four aspects. Each result below is from an average of 50 simulation runs, where each run contains at least 3,000 frames:

1. *Impact of Code Assignment Strategies:* This experiment tests different code assignment strategies. We evaluate three metrics: *code utilization*, *effective code utilization*, and *average delay*. Code utilization is the average capacity of the code tree that is actually used for transmission. However, considering those transmissions experiencing failure, effective code utilization counts only those successfully transmitted bits. In addition to the KMS scheme, we also simulate the "single-code, shared" scheme and the "single-code, dedicated" scheme.

   Fig. 5 shows the code utilization and effective code utilization at various traffic loads. For our schemes, N_{max} is set to 2. For code utilization, our scheme 3 performs the best, which is sequentially followed by the KMS scheme, our scheme 2, our scheme 1, and then the single-code schemes. After taking failure transmissions into account, effective code utilization of KMS scheme reduces significantly when $\gamma = 1$. For example, when the code tree is 90 percent fully loaded, KMS's effective utilization reduces to around 51 percent, while our schemes can still maintain an effective utilization over 85 percent. Only when the channel condition is extremely good (such as $\gamma = 25$), can KMS maintain high effective code utilization. Since our schemes have considered channel condition, the code utilization and effective code utilization are very close.

   Fig. 6 shows the average delay, which is defined as the average time, measured in time frames, a transport block is queued at the base station. With $\gamma = 1$, our scheme 3 experiences the least delay, which is subsequently followed by our scheme 2, scheme 1, single-code, shared scheme, and KMS. Our scheme 3 produces low delay since there is no code blocking. The delays for scheme 1 and single-code, shared scheme increase significantly when the code tree is above 80 percent loaded, which means the system is unstable then. The KMS scheme makes the system unstable after traffic load is over 30 percent. These results reveal that our scheme 3 and scheme 2 can accept more time-bounded services. The delays of KMS reduces significantly when $\gamma = 25$, which is reasonable since the channel condition is pretty good.
2. *Fairness Properties:* Next, we verify how our schemes support rate guarantee and fairness. We set $N_{max} = 4$ since it achieves better performance than $N_{max} = 2$ and 3 (performance comparisons of different N_{max} are not reported here due to space limit). To see rate guarantee, we randomly choose a user i with request $a_i = 1, 2, 4$ or $8R_b$ and observe the number of transmitted transport blocks. As shown in Fig. 7, both schemes 2 and 3 support rate guarantee well. For scheme 1, there is some lag for $a_i = 4$ and $8R_b$, which stems from the higher failure probabilities when allocating codes for such calls.

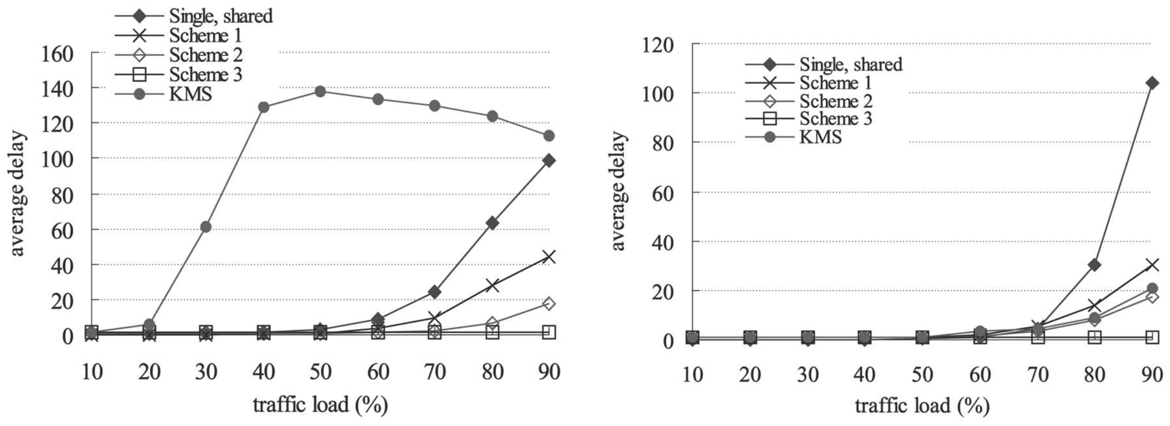
   To verify access fairness, we introduce a metric called *interservice time*, which is the interval that a



(a)

(b)

Fig. 5. Code utilization (in curves) and effective code utilization (in bars) versus traffic load under traffic model II with (a) $\gamma = 1$ and (b) $\gamma = 25$.



(a)

(b)

Fig. 6. Average delay versus traffic load under traffic model II with (a) $\gamma = 1$ and (b) $\gamma = 25$.

backlogged user experiences, measured in time frames, between two successive transmissions. We observe the maximum difference of interservice times of two backlogged users who are waiting for scheduling. The result is in Table 2. In all three traffic models, scheme 3 has the least difference, which is followed by scheme 2 and then scheme 1. For the constantly backlogged model, the differences are low for all schemes and are proportional to traffic load. For example, at traffic load = 80 percent, the most unfortunate user will experience at most 15.1, 8.1, and 3.9 time frames of delay for schemes 1, 2, and 3, respectively. For both bursty traffic models, the difference is relatively independent to traffic load. Scheme 3 performs the best, and scheme 1 the worst. In general, the differences are larger for the highly bursty traffic model, which is reasonable because traffics with higher variation are more difficult to handle.

We also investigate the impact of different traffic models on average delay. The results are in Fig. 8. Scheme 3 always has the least delay and is quite independent of traffic models. Scheme 2 also has very low delay (e.g., the average delays are only 8 and 5 time frames at load = 90 percent for traffic models II and III, respectively). Scheme 1 has a longer delay for lowly bursty traffics because bursty traffics arrive more frequently (thus causing more backlogged users).

3. *Signaling Overhead:* Since different values of N_{max} have similar results, here we use $N_{max} = 4$ as a representative case. We plot the signaling overhead of different schemes according to the formulation in Section 3.4. As shown in Fig. 9, in all traffic models, scheme 2 incurs the least signaling overhead while scheme 3 the highest. Note that we should also take into account the utilization issue while evaluating these schemes. When this factor is considered, scheme 2 will be the best for traffic models II and III. For example, for model II under load = 90 percent, the signaling overheads are 152 and 720 bits for schemes 2 and 3, respectively. The effective utilization of scheme 3 is only 1 percent higher than scheme 2, which means around $1\% \times 256 \times 150 = 384$ bits of reward at the expense of $720 - 152 = 568$ bits of overhead when comparing scheme 3 and scheme 2. However, for model I, scheme 3 performs the best when the utilization factor is considered.
4. *Impact of Fully Shared Code Space for Scheme 2:* Last, we investigate how the ratio of fully shared code space in scheme 2 affects its performance. Four ratios, $\frac{1}{8}$, $\frac{2}{8}$, $\frac{3}{8}$, and $\frac{4}{8}$, are tested with different N_{max} values. Due to the space limit, we report the results without figures. When $N_{max} = 2$, the one with $\frac{2}{8}$ fully shared code space performs the best. As N_{max} increases, larger ratios would be more beneficial. It is because calls can more freely utilize the fully shared space, causing less blocking problem. When

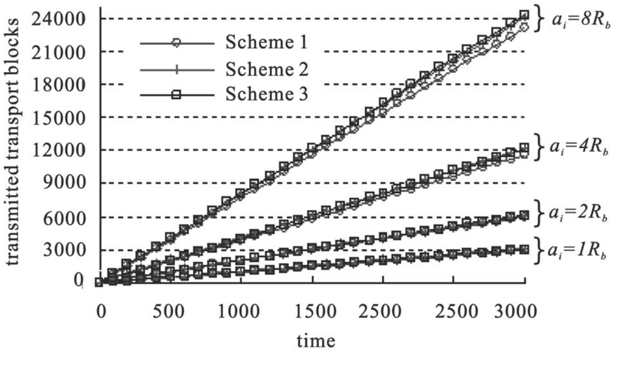


Fig. 7. Transmitted transport blocks versus time under traffic model II.

TABLE 2
The Maximum Difference of Interservice Times under Different Traffic Models

traffic model		I			II			III		
scheme		1	2	3	1	2	3	1	2	3
traffic load	20%	1.1	1.0	1.0	24.2	19.1	2.9	16.8	14.8	2.8
	40%	1.7	1.5	1.1	25.8	19.8	2.7	17.2	15.8	2.9
	60%	5.0	4.1	1.7	27.2	19.2	3.0	16.9	15.8	3.2
	80%	15.1	8.1	3.9	27.5	20.4	3.4	16.6	16.2	3.7

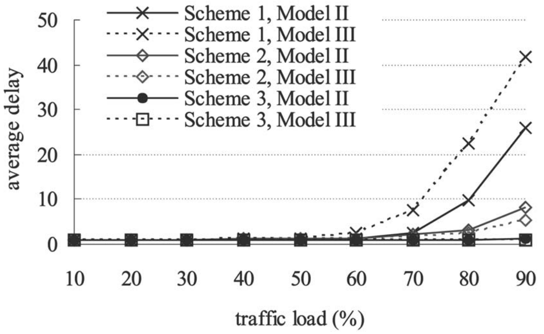


Fig. 8. The impact of traffic models on average delay.

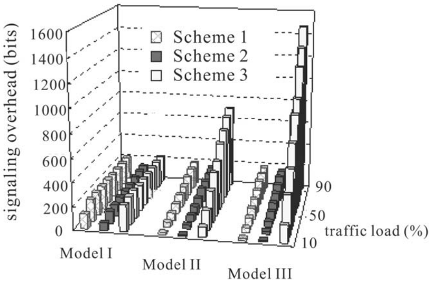


Fig. 9. Signaling overhead of different schemes.

N_{max} is large enough to better utilize the fully shared space, the utilization would be the best. (However, the advantage cannot be seen when N_{max} is too small.) When $N_{max} = 3$ and 4, the one with $\frac{3}{8}$ fully shared code space performs the best.

To conclude, we believe that a scheme that does not consider channel condition, such as the KMS scheme, may perform poorly. In our proposal, scheme 3 has the best behavior in terms of utilization, delay, rate guarantee, and fair access. However, it also imposes the highest signaling overhead. Scheme 2 falls behind scheme 3, but has the least signaling overhead. Taking all these issues into account, we recommend using scheme 3 under constantly backlogged traffic model (Model I) and scheme 2 under bursty traffic models (Model II and III) by properly tuning the fully shared code space depending on the given N_{max}.

5 Conclusions

Wireless bandwidth is a precious resource. Thus, resource management is an important issue for WCDMA. Existing mechanisms supporting multimedia traffic either do not consider channel condition or fail to address the exact code position in the code tree, which may result in inefficiency in resource utilization. In this paper, we have proposed three algorithms to provide fair and rate guaranteed service for multimedia traffic in a WCDMA system. The essence of our protocols is a credit-based scheduler which considers channel condition and explores the concept of compensation codes. With our channel-sensitive scheduling algorithms, a user with more credits will have more chance to transmit without compromising to the transmission quality. Simulation results do justify that schemes 2 and 3 work well.

Acknowledgments

Y.C. Tseng's research is cosponsored by the MOE Program for Promoting Academic Excellence of Universities under grant number 89-E-FA04-1-4, by the NSC of Taiwan under grant numbers NSC92-2213-E009-076 and NSC92-2219-E009-013, by the Institute for Information Industry and MOEA, R.O.C, under the Handheld Device Embedded System Software Technology Development Project, by the Lee and MTI Center of NCTU, and by Computer and Communications Research Labs., ITRI, Taiwan.

References

[1] Third Generation Partnership Project; Technical Specification Group Radio Access Network, Spreading and Modulation (FDD), http://www.3gpp.org, 1999.

[2] F. Adachi, M. Sawahashi, and H. Suda, "Wideband DS-CDMA for Next-Generation Mobile Communications Systems," *IEEE Comm. Magazine,* vol. 36, pp. 56-69, Sept. 1998.

[3] R. Assarut, K. Kawanishi, U. Yamamoto, Y. Onozato, and M. Masahiko, "Region Division Assignment of Orthogonal Variable-Spreading-Factor Codes in W-CDMA," *Proc. IEEE Vehicular Technology Conf.,* pp. 1184-1898, 2001.

[4] Y. Cao and V.O. Li, "Scheduling Algorithms in Broad-Band Wireless Networks," *Proc. IEEE INFOCOM,* vol. 89, no. 1, pp. 76-87, Jan. 2001.

[5] C.-M. Chao, Y.-C. Tseng, and L.-C. Wang, "Reducing Internal and External Fragmentations of OVSF Codes in WCDMA Systems with Multiple Codes," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC),* pp. 693-698, 2003.

[6] W.-T. Chen, Y.-P. Wu, and H.-C. Hsiao, "A Novel Code Assignment Scheme for W-CDMA Systems," *Proc. IEEE Vehicular Technology Conf.,* pp. 1182-1186, 2001.

[7] R.-G. Cheng and P. Lin, "OVSF Code Channel Assignment for IMT-2000," *Proc. IEEE Vehicular Technology Conf.,* pp. 2188-2192, 2000.

[8] D. Eckhardt and P. Steenkiste, "Effort-Limited Fair (ELF) Scheduling for Wireless Networks," *Proc. IEEE INFOCOM,* pp. 1097-1106, 2000.

[9] 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. Selected Areas in Comm.,* vol. 18, no. 8, pp. 1441-1454, Aug. 2000.

[10] V.K. Garg, *IS-95 CDMA and cdma2000.* Prentice Hall, 2000.

[11] J. Gomez, A.T. Campbell, and H. Morikawa, "The Havana Framework for Supporting Application and Channel Dependent QoS in Wireless Networks," *Proc. Int'l Conf. Network Protocols,* pp. 235-244, 1999.

[12] H. Holma and A. Toskala, *WCDMA for UMTS.* John Wiley & Sons, 2000.

[13] C.-L. I et al., "IS-95 Enhancements for Multimedia Services," *Bell Labs. Tech. J.,* pp. 60-87, Autumn 1996.

[14] A.Z. Kam, T. Minn, and K.-Y. Siu, "Supporting Rate Guarantee and Fair Access for Bursty Data in W-CDMA," *IEEE J. Selected Areas in Comm.,* vol. 19, no. 12, pp. 2121-2130, Nov. 2001.

[15] D. Kandlur, K. Shin, and D. Ferrari, "Real-Time Communication in Multihop Networks," *Proc. ACM SIGCOMM,* pp. 300-307, 1991.

[16] S. Lu and V. Bharghavan, "Fair Scheduling in Wireless Packet Networks," *IEEE/ACM Trans. Networking,* vol. 7, no. 4, pp. 473-489, 1999.

[17] T. Minn and K.-Y. Siu, "Dynamic Assignment of Orthogonal Variable-Spreading-Factor Codes in W-CDMA," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 8, pp. 1429-1440, Aug. 2000.

[18] T.S.E. Ng, I. Stoica, and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," *Proc. INFOCOM,* pp. 1103-1111, 1998.

[19] A. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. Networking,* vol. 1, pp. 334-357, June 1993.

[20] J. G. Proakis, *Digital Communications,* fouth ed. McGraw-Hill, 2001.

[21] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms to Wireless Networks," *Proc. ACM/IEEE MOBICOM,* pp. 1-9, 1998.

[22] F. Shueh, Z.-E.P. Liu, and W.-S.E. Chen, "A Fair, Efficient, and Exchangeable Channelization Code Assignment Scheme for IMT-2000," *Proc. IEEE Int'l Conf. Personal Wireless Comm. (ICPWC),* pp. 429-436, 2000.

[23] Y.-C. Tseng and C.-M. Chao, "Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree Management," *IEEE Trans. Mobile Computing,* vol. 1, no. 4, pp. 293-302, Oct.-Dec. 2002.

[24] J. Wigard, N.A.H. Madsen, P.A. Gutierrez, I.L. Sepulveda, and P. Mogensen, "Packet Scheduling with QoS Differentiation," *Wireless Personal Comm.,* vol. 23, pp. 147-160, 2002.

[25] L. Xu, X. Shen, and J.W. Mark, "Dynamic Bandwidth Allocation with Fair Scheduling for WCDMA Systems," *IEEE Wireless Comm.,* vol. 9, no. 2, pp. 26-32, Apr. 2002.

[26] Y. Yang and T.-S.P. Yum, "Nonrearrangeable Compact Assignment of Orthogonal Variable-Spreading-Factor Codes for Multi-Rate Traffic," *Proc. IEEE Vehicular Technology Conf. (VTC),* pp. 938-942, 2001.

[27] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks," *Proc. ACM SIGCOMM,* pp. 19-29, 1990.

Chih-Min Chao received the BS and MS degrees in computer science from Fu-Jen Catholic University and National Tsing-Hua University in 1992 and 1996, respectively. He was with SENAO International in 1996. He received the PhD degree in computer science and information engineering from National Central University in January of 2004. He now works in the Department of Computer Science at National Taiwan Ocean University. His research interests include mobile computing and wireless communication, with a current focus on resource management in WCDMA systems.

Yu-Chee Tseng received the BS and MS degrees in computer science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for the D-LINK Inc. as an engineer in 1990. He received the PhD degree in computer and information science from the Ohio State University in January of 1994. He was an associate professor at the Chung-Hua University (1994 to 1996) and at the National Central University (1996 to 1999), and a full professor at the National Central University (1999 to 2000). Since 2000, he has been a full professor in the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan. Dr. Tseng served as a program chair in the Wireless Networks and Mobile Computing Workshop, 2000 and 2001, as a vice program chair of the International Conference on Distributed Computing Systems (ICDCS), 2004, as a vice program chair of the IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), 2004, as an associate editor for *The Computer Journal*, and as guest editor of special issues for several journals including *ACM Wireless Networks*, *IEEE Transactions on Computers*, *Journal of Internet Technology*, etc. He is a two-time recipient of the Outstanding Research Award, National Science Council, ROC, in 2001-2002 and 2003-2005, and a recipient of the Best Paper Award at the International Conference on Parallel Processing, 2003. Several of his papers have been chosen as selected/distinguished papers in conferences and been included for publications in journals. He has guided students to participate in several national programming contests and received several awards. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng is a member of the ACM and a senior member of the IEEE Computer Society.

Li-Chun Wang received the BS degree from National Chiao Tung University, Taiwan, in 1986, the MS degree from National Taiwan University in 1988, and the MSci and PhD degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, in 1995, and 1996, respectively. From 1990 to 1992, he was with the Telecommunications Laboratories at the Ministry of Transportations and Communications in Taiwan (currently the Telecom Labs of Chunghwa Telecom Co.). In 1995, he was affiliated with Bell Northern Research of Northern Telecom, Inc., Richardson, Texas. From 1996 to 2000, he was with AT&T Laboratories, where he was a senior technical staff member in the Wireless Communications Research Department. Since August 2000, he has been an associate professor in the Department of Communication Engineering at National Chiao Tung University in Taiwan. His current research interests are in the areas of cellular architectures, radio network resource management, and crosslayer optimization for high-speed wireless networks. Dr. Wang was a corecipient of the Jack Neubauer Memorial Award in 1997 recognizing the best systems paper published in the *IEEE Transactions on Vehicular Technology*. He holds three US patents and one more pending. Currently, he is the editor of the *IEEE Transactions on Wireless Communications*. He is a member of the IEEE.