

# OVSF Code Tree Management for UMTS with Dynamic Resource Allocation and Class-Based QoS Provision

Huei-Wen Ferng, Jin-Hui Lin, Yuan-Cheng Lai<sup>†</sup>, and Yung-Ching Chen

Department of Computer Science and Information Engineering  
National Taiwan University of Science and Technology, Taipei 106, Taiwan

<sup>†</sup>Department of Information Management  
National Taiwan University of Science and Technology, Taipei 106, Taiwan  
E-mail: hwferng@mail.ntust.edu.tw

**Abstract**—Rather than simply keeping the code tree as compact as possible merely depending on the request rate, we design a scheme on the orthogonal variable spreading factor (OVSF) code tree management for the universal mobile telecommunication system (UMTS) by further considering characteristics of traffic and dynamic resource allocation (DRA) to make the management efficient and satisfy different quality-of-service (QoS) requirements requested by diverse multimedia applications. Through simulations, we demonstrate that the proposed scheme can not only adequately take care of QoS requirements but also significantly reduce the number of code reassignment, making it charming and more suitable for the real and complicated system.

## I. INTRODUCTION

Compared to the second generation (2G) mobile communication systems, which mainly provide voice and low-rate data transmission, the third generation (3G) ones are targeted to provide high-rate transmission for multimedia applications. To fulfill the goal, UMTS employs wide-band code division multiple access (WCDMA) to support multi-rate applications utilizing OVSF codes. Then, how to efficiently manage the code tree becomes a research issue [2]–[10] because of *code fragments*, resulting in *code blocking* [2] although the capacity is still enough. The situation gets worse when requested rates are diverse. In [2], a dynamic assignment scheme was designed to solve code blocking and minimize the number to reassign OVSF codes. The authors showed that the proposed scheme can not only reduce the blocking probability but also increase the system utilization. [3] proposed two strategies, called *leftmost* and *crowded-first*, to reduce the call blocking probability as well as the code reassignment cost for multi-rate services. The OVSF code tree management schemes proposed by [2] and [3] focus on single type of services rather than multiple services. To consider multiple services and differentiated QoS requirements, [4] proposed the scheme described as follows: the code tree is divided into different regions for voice and video services so as to guarantee certain degree of QoS (but this will reduce the system utilization). Similar to [4], [5] proposed a new scheme to take care

of users with different rate requirements by evaluating call blocking probability and number of reassignments. Taking four service classes specified in the 3GPP standard [1], i.e., conversational, streaming, interactive, and background classes into consideration, traffic classes in [6] are grouped into high-rate and low-rate ones. The high-rate traffic takes the OVSF code from the left-hand side and the low-rate traffic takes the OVSF code from the right-hand side. By reassigning codes once a code is released and endowing with queues for the delay-tolerable traffic, the proposed scheme can improve the performance for the streaming traffic, but it increases the reassignment cost. [7] allocates codes based on a compact index to keep the code tree as compact as possible. This compact index represents the distribution of neighboring codes but it needs to be recalculated before code assignments. To improve system utilization, [8] presented a dynamic OVSF code assignment algorithm that efficiently shares resources among traffic with diverse requirements. As for the scheme proposed in [9], it only supports delay-tolerant traffic and constructs the code tree according to the number of users in the system. It would be useful to maximize the transmission rate in the homogeneous communication situation as suggested by simulation results. Finally, [10] provided a framework of traffic conditioning with QoS provisioning in the 3G radio access network, which employs traffic shaping at each user equipment (UE) and traffic policing at the radio network controller (RNC).

In the literature, both *single-code* and *multi-code* OVSF code assignments are proposed to support multi-rate transmission. However, we focus on the single-code one in this paper because of its simplicity in the hardware design. We take multimedia applications with differentiated QoS into account. According to the 3GPP specification [1], the four service classes are also considered. Moreover, DRA in UMTS terrestrial radio access network (UTRAN) is included to guarantee QoS for multimedia applications. The ideas for the proposed scheme are briefly described as follows. Based on the requested rate by the traffic, this scheme can keep the

OVSF code tree as compact as possible in a new way with prioritized classes to reduce code blocking and the number of reassignments. Using the DRA scheme, queues and service priorities for different classes are incorporated. By doing so, we can demonstrate that our scheme performs well and is more suitable for real and complicated systems as compared to [6] which is the closest paper to ours in the literature.

The rest of the paper is organized as follows. Section II describes the proposed scheme. Then, numerical examples and discussions are given in Section III. Finally, Section IV concludes the paper.

## II. DESCRIPTION OF THE PROPOSED SCHEME

The proposed scheme consists of three parts. The first part is OVSF code tree segmentation. The second part concerns the OVSF code assignment (with or without reassignment), including *class-based unit assignment* (CUA) and *class-based rearrangeable unit assignment* (CRUA). Finally, the dynamic resource allocation forms the third part.

### A. OVSF Code Tree Segmentation

After a period of operation, available codes will be scattered around the code tree. This definitely affects the performance of the system. Hence, keeping the code tree as compact as possible is a general goal to manage the code tree well. In our scheme, we divide the code tree into sub-trees (assignment units) according to the highest requested rate by the service class, i.e.,  $16R$ , the requested rate of the streaming class, where  $R$  is the requested rate of the conversational class. Therefore, the whole tree with capacity of  $256R$  is divided into 16 sub-trees. Fig. 1 illustrates the  $16R$ -based code tree segmentation. Unlike regions used by [4], sub-trees are not reserved for any particular class but are used to facilitate the code tree management. For example, ongoing calls in an sub-tree can be moved to other sub-trees if *code blocking* happens. Also, the computation complexity can be kept at a low level and radio resources can be efficiently utilized with the code tree segmentation.

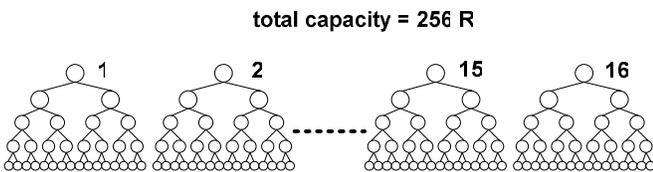


Fig. 1. OVSF code tree segmentation.

### B. Class-Based Unit Assignment (CUA) and Class-Based Rearrangeable Unit Assignment (CRUA)

Let us first describe the procedure of CUA as follows:

- 1) Since the streaming class requests the highest rate, they will take an assignment unit at a time. Giving an arriving call belonging to the streaming class, assign a randomly chosen unit to the new call *if* there are still available assignment units; *otherwise*, block the call.

- 2) For an arriving call not belonging to the streaming class, choose an available assignment unit occupied by the most calls of the same requested rate or block the call. After choosing an assignment unit, code selection will be performed. In this paper, we adopt the *compact index* denoting the relationship between codes proposed by [7] to select a code so that a compact subtree/tree is formed. In this way, the calls belonging to the same class will be put together. This can be regarded as a *partial crowded-first* strategy and can reduce the possibility of code blocking.

As mentioned earlier, the dynamic nature of code usage eventually affects the code status. Hence, even though we do our best to keep the code tree as compact as possible, multi-rate traffic can easily lead to *code fragments*, resulting in code blocking. Therefore, rearranging ongoing calls is inevitable to improve the system performance and to make the utilization high. Considering a state change of the code tree, CRUA perform the rearrangement procedure at an arrival epoch. Also, with differentiated QoS constraints for different traffic classes, CRUA rearranges delay-sensitive (real-time) ongoing calls only, i.e., calls of conversational and streaming classes. The algorithm of CRUA is depicted as follows (also shown diagrammatically in Fig. 2):

- 1) *If* a new real-time call arrives and code blocking happens, choose a sub-tree with the least occupied calls and swap the ongoing calls to other sub-trees. The procedure continues iteratively from codes in the leaves.
- 2) *If* the chosen sub-tree can accommodate the new real-time call after the above procedure, the arriving real-time call uses a code within this sub-tree; *else*, block the call.
- 3) *If* code blocking happens but the new call is not delay-sensitive, i.e., non-real-time, then no reassignment is performed.

### C. Dynamic Resource Allocation

DRA is a good approach to fulfill differentiated QoS. With DRA, we can set different priorities to different services. Services with the highest priority can access the resource first. Also, queues can be employed for non-real-time services to reduce the blocking probability of non-real-time services. However, with limited resources, there are always some tradeoffs. For example, when the blocking probability for high-rate services decreases, the blocking probability for low-rate services may (slightly) increase. Due to the delay-sensitive property, real-time services should be quickly served, while non-real-time services can endure a bit longer delay. In general, DRA can improve the whole system performance for multimedia applications.

Now, how to apply priorities and queues is depicted as follows:

- **Priorities:** To ensure QoS requirements, real-time traffic of conversational and streaming classes should always have higher priorities over non-real-time traffic of interactive and background classes. As aforementioned, non-real-time services can endure longer delay time and this

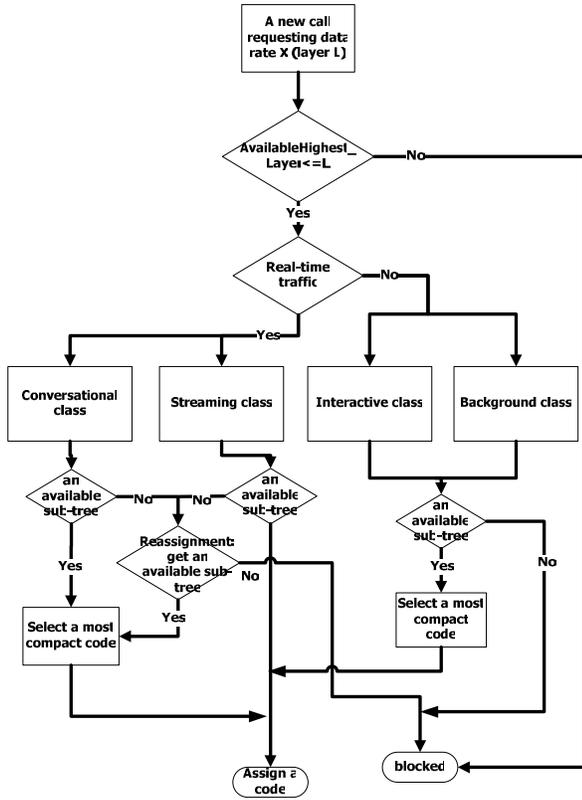


Fig. 2. CRUA Algorithm.

gives real-time services a chance to get an available code. Moreover, if new calls and handoff calls are considered separately, handoff calls of conversational and streaming classes should have higher priorities over new calls of these two classes, whose priorities are subsequently higher than other calls because of the real-time nature and the fact that less satisfaction is raised by handoff failure than new call blocking.

- **Queues:** This strategy can reduce blocking probabilities for non-real-time services at the expense of delay time of non-real-time services. But it is acceptable since longer delay can be tolerated by non-real-time services. To work with the priority strategy, we employ the following useful rules involved queues:

- 1) If the system load exceeds a preset threshold ( $T_{nr}$ , where  $nr$  denotes non-real-time), non-real-time traffic is forced to enter the queue first. During staying in the queue, the non-real-time traffic can re-gain the resource if the system load falls below the threshold.
- 2) For higher priority traffic, such as handoff calls and real-time traffic, the traffic is forced to enter the queue when there is not any available code after the reassignment. During staying in the queue, this call can immediately be served once the system load falls below a preset threshold for the corresponding traffic ( $T_{con}$  and  $T_{str}$ , where  $con$  and  $str$  denote conversational and streaming, respectively) and there is

an available code after the reassignment.

Applying DRA strategies, we propose a new scheme called *classed-based QoS provisioning rearrangeable unit assignment* (CQPRUA), which includes three variants, i.e., CQPRUA<sub>1</sub>, CQPRUA<sub>2</sub> and CQPRUA<sub>3</sub>. CQPRUA<sub>1</sub> is designed based on CRUA but queues for non-real-time traffic are added. CQPRUA<sub>2</sub> further employs queues for conversational and streaming classes as compared to CQPRUA<sub>1</sub>. At last, CQPRUA<sub>3</sub> differentiates handoff calls and new calls.

### III. NUMERICAL EXAMPLES AND DISCUSSIONS

In the following, we use a simulation approach to study the performance of schemes we proposed and the scheme proposed in [6] (denoted by CWH01), including call blocking probability, system utilization, and the number of reassignments. In addition, we also show how the ratio of handoff calls and new calls affects the performance of various service classes.

#### A. Simulation Parameters

We assume the total capacity for the system is  $256R$ , where  $R$  is the basic rate and is approximately equal to 8 Kbits/sec. For each traffic classes, we use Poisson processes to model the arrivals (the mean arrival rate is varied from 2 to 5 (requests/minute)). In addition, all the four service classes are mutually independent and have their own queue, if any. In Table I, we summarize detailed parameters about traffic classes.

TABLE I  
SIMULATION PARAMETERS

service type	data rate	service time	timeout period	buffer size
conversational	$1R$	180 sec.	10 sec.	4
streaming	$16R$	180 sec.	100 sec.	4
interactive	$4R$	10 sec.	360 sec.	$\infty$
background	$2R$	5 sec.	3600 sec.	$\infty$

#### B. Simulation Results

In Figs. 3–6, we compare the proposed schemes with the scheme proposed in [6] and show blocking probabilities for the four service classes, respectively. Fig. 3 shows the new call blocking probability for the conversational class. From this figure, we observe that CQPRUA<sub>1</sub> performs almost equivalently to CWH01 and is inferior to CRUA, CQPRUA<sub>2</sub>, and CUA. Since a rearrangement policy, which rearranges codes for real-time traffic, is employed in CRUA, CQPRUA<sub>1</sub>, and CQPRUA<sub>2</sub>, it will assign available codes to the high-rate real-time traffic, i.e., streaming class, thus increasing new call blocking probability for the low-rate traffic. As for the comparison between CQPRUA<sub>1</sub> and CQPRUA<sub>2</sub>, CQPRUA<sub>2</sub> performs better because of the setting of queues. Similarly, new call blocking probability for the streaming class using CQPRUA<sub>2</sub> is still better than that using CQPRUA<sub>1</sub>. In fact, CQPRUA<sub>2</sub> outperforms other schemes because of the implementation of the rearrangement policy, priority and queue as shown in In Fig. 4. In Figs. 5 and 6, the new call blocking probabilities for

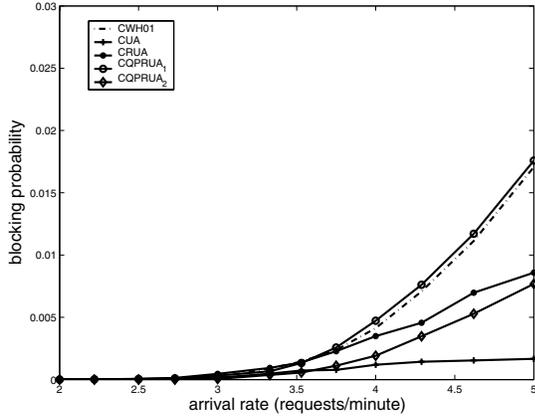


Fig. 3. New call blocking probabilities among different schemes for the conversational service type.

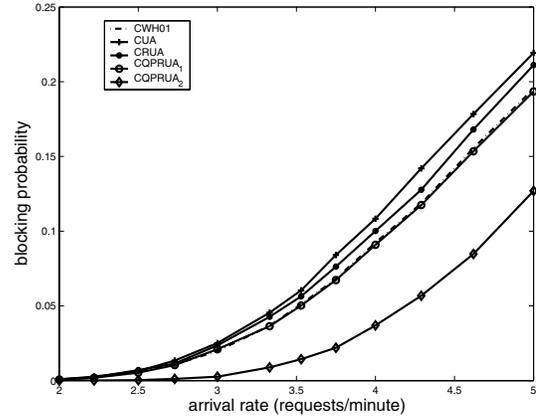


Fig. 4. New call blocking probabilities among different schemes for streaming service type.

interactive and background traffic are examined for different schemes. As shown in Fig. 4, we note that the rearrangement policy is indeed good for CQPRUA<sub>1</sub> and CQPRUA<sub>2</sub>. However, CQPRUA<sub>1</sub> and CQPRUA<sub>2</sub> downgrade performance of other traffic. This tradeoff can be easily examined from Figs. 3 and 5. Fig. 7 shows the system utilization for different schemes. We witness that CQPRUA<sub>1</sub> performs better than CRUA and CUA.

To decrease the code tree fragments resulting from allocating and releasing codes, reallocation of ongoing calls is an imperative procedure to enhance system performance. In [6], allocated codes are reallocated once a code is released and the procedure is performed recursively to find an adequate code for the tree-structure codes. However, we implement the reallocation at the arrival epoch for delay-sensitive traffic only in our proposed schemes to reduce the cost of reassignments. In Table II, we show that the proposed scheme can greatly reduce the number of reassignments (about 100% of improvement) although CQPRUA<sub>1</sub> and CWH01 [6] almost have the same performance as shown in Figs. 3–7.

To further apply the proposed scheme to real systems, the discrimination between handoff calls and new calls is necessary and is incorporated into CQPRUA<sub>3</sub>. Fig. 8 shows the performance under different ratios of handoff calls and new calls with the total arrival rate of 5 requests/minute. As the arrival rate of handoff calls increases, the blocking probability for the handoff streaming traffic increases naturally; so does that for the new streaming traffic because of lower priority and higher rate requested. In addition, handoff and new conversational traffic streams get served due to low requested rate and background class can have a low blocking probability since longer delay is acceptable when applying a queue. As for the blocking probability for the interactive class, it increases first when resources are taken by traffic with higher priorities, but it then decreases because of the left fragments.

#### IV. CONCLUSIONS

This paper proposes a new scheme CQPRUA (containing three variants) for the OVFS code tree management by taking

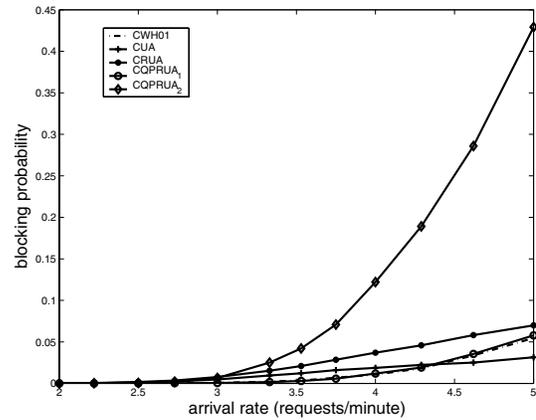


Fig. 5. New call blocking probabilities among different schemes for interactive service type.

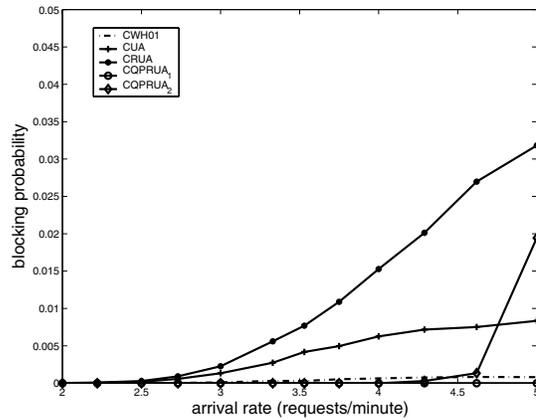


Fig. 6. New call blocking probabilities among different schemes for background service type.

TABLE II  
NUMBER OF REASSIGNMENTS

arrival rate	2	3	4	5
CWH01	24158.1	41754.3	65186.9	90737.4
CQPRUA <sub>1</sub>	0.7	74.2	702.8	2046.1

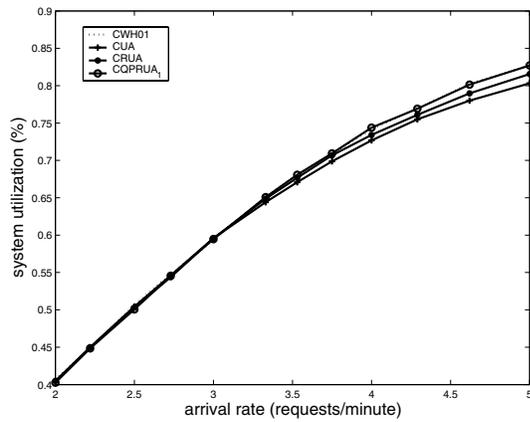


Fig. 7. System utilization.

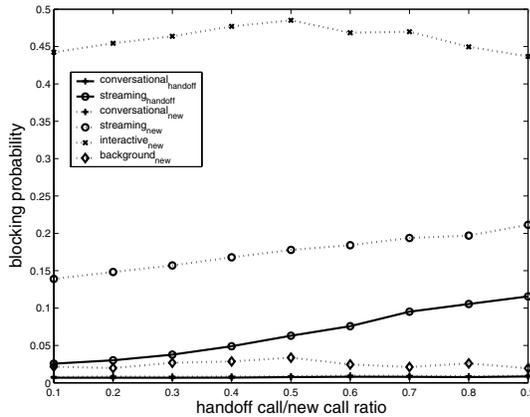


Fig. 8. Handoff call/new call ratio vs. blocking probability

the code tree segmentation, traffic classes with differentiated QoS and DRA into consideration. As far as the segmentation is concerned, the OVFSF code tree is divide into sub-trees (assignment units), which are not reserved for any particular traffic/service class, to facilitate the code tree management. Hence, the computation complexity can be kept at a low level and radio resources can be efficiently utilized. As for consideration of traffic characteristics, it allows the proposed scheme to significantly reduce the number of code reassignments as compared to the scheme in [6]. Of course, DRA also enhance the service discrimination. Through simulations, we have successfully demonstrate these merits for the proposed scheme as compared to that in [6].

#### ACKNOWLEDGMENT

The first author thanks the partial support in finance by the National Science Council, Taiwan under Contract NSC 93-2219-E-011-007.

#### REFERENCES

[1] ETSI, "Universal Mobile Telecommunication System (UMTS): Quality of Service (QoS) Concept and Architecture," ETSI, Tech. Rep. 3GPP TS 23.107 version 5.8.0, 2003.

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

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

[4] R. Assarut, K. Kawanishi, U. Yamamoto, Y. Onozato, and M. Masushita, "Region division assignment of orthogonal variable-spreading-factor codes in W-CDMA," in *Proc. IEEE VTC-Fall '01*, pp. 1884–1888, 2001.

[5] M. D. Amico, F. Maffioli, and M. L. Merani, "A tree partitioning dynamic policy for OVFSF codes assignment in wideband CDMA," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1013–1017, July 2004.

[6] W. T. Chen, Y. P. Wu, and H. C. Hsiao, "A novel code assignment scheme for W-CDMA systems," in *Proc. IEEE VTC-Fall '01*, pp. 1182–1186, 2001.

[7] Y. Yang and T. 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.

[8] C. E. Fossa and N. J. Davis, "Dynamic code assignment improves channel utilization for bursty traffic in third-generation wireless networks," in *Proc. IEEE WCNC '02*, vol. 5, pp. 139–143, Apr. 2002.

[9] R. Assarut, M. G. Husada, U. Yamamoto, and Y. Onozato, "Data rate improvement with dynamic reassignment of spreading codes for DS-CDMA," *Computer Communications*, vol. 25, no. 17, pp. 1575–1583, Nov. 2002.

[10] F. Y. Li and N. Stol, "QoS provisioning using traffic shaping and policing in 3rd-generation wireless networks," in *Proc. IEEE WCNC '02*, pp. 139–143, Mar. 2002.