# Scheduling for time-division based shared channel allocation for UMTS\*

Chai-Hien Gan  $\cdot$  Phone Lin  $\cdot$  Nei-Chiung Perng  $\cdot$  Tei-Wei Kuo  $\cdot$  Ching-Chi Hsu

Published online: 8 May 2006 © Springer Science + Business Media, LLC 2007

Abstract The Universal Mobile Telecommunications System (UMTS) adopts the WCDMA technology as the radio access interface to provide variable transmission rate services. There are four classes of connections identified in UMTS, which are the conversational, streaming, interactive, and background connections. To efficiently utilize radio bandwidth, the shared channel approach is proposed to deliver the packets for the interactive and background connections. This paper proposes a "Shared-Channel As-

\*A preliminary version [11] of this work has been accepted by IEEE Wireless Communications and Networking Conference 2004. This paper is an extension of the proposed algorithm, and simulation and analysis are conducted to investigate the performance of the proposed algorithm.

C.-H. Gan Department of Computer Science, National Chiao Tung University, R.O.C. e-mail: chgan@csie.nctu.edu.tw

P. Lin (🖂)

Department of Computer Science & Information Engineering, and Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, R.O.C e-mail: plin@csie.ntu.edu.tw

#### N.-C. Perng

Department of Computer Science & Information Engineering, National Taiwan University, R.O.C e-mail: d90011@csie.ntu.edu.tw

T.-W. Kuo Department of Computer Science & Information Engineering, National Taiwan University, R.O.C. e-mail: ktw@csie.ntu.edu.tw

#### C.-C. Hsu

Executive Vice President, Institute for Information Industry (III), Taiwan, R.O.C. e-mail: cchsu@iii.org.tw signment and Scheduling" (SCAS) algorithm to periodically allocate shared channels to serve interactive and background connections. We conduct formal mathematical proofs and simulation experiments to investigate the performance of the SCAS algorithm. We formally prove that with SCAS, a shared channel can be fully utilized (i.e., the utilization of a shared channel can be up to 100%) to serve the interactive connections. Our analysis indicates that compared with the previously proposed shared channel allocation and scheduling algorithms, there are less computation and communication overheads introduced in the SCAS algorithm. The results of the simulation experiments indicate that it is preferred to set up the Transmission Time Interval (TTI; that is, the unit of time interval for shared channel allocation) smaller to optimize the performance of the SCAS algorithm, including the shared channel utilization and the average waiting time of a connection before getting transmission service.

**Keywords** Downlink shared channel · Orthogonal variable spreading factor · Universal mobile telecommunications system · Wide-band code division multiple access

## 1. Introduction

The Universal Mobile Telecommunications System (UMTS) [2, 7] provides high and variable transmission rate services for mobile users. With UMTS, wireless multimedia communication can be enhanced with high quality. Packet access to information and services in the Internet will be enhanced by higher data rates and new flexible communication capabilities. In UMTS, four classes of traffics are identified, which are *conversational, streaming, interactive,* and *background* traffics. The conversational and streaming



RNC: Radio Network Controller UE: User Equipment UTRAN: UMTS Terrestrial Radio Access Network







traffics have the constant bandwidth characteristics. The typical applications with the conversational and streaming traffics are speech and multimedia applications. The interactive traffic is highly dependent on request-response patterns of end users (where a response is timely critical), which has the burstness characteristics (i.e., lots of packets arrive within a short time period). The typical applications with the interactive traffic are common Internet applications (e.g., web browsing and network gaming). The applications with the background traffic are without stringent responsetime requirement, such as the background e-mail application. They are usually served by the best-effort transmission service.

As shown in Fig. 1, UMTS consists of the *Terrestrial Radio Access Network* (UTRAN) and the core network. A *User Equipment* (UE) communicates with UTRAN through the air interface Uu [1] adopting the WCDMA radio access technology. WCDMA is an interference-constrained radio access technology. The total bandwidth (i.e., transmission rate) that can be supported within the coverage area of a Node B is dependent on the physical factors including power control, frame error rate, signal to interference ratio, and multi-path propagation [7]. In this paper, we focus on the *Media Access Control* (MAC) layer [1] for the WCDMA system, where the total transmission rate (that can be supported within the coverage area of a Node B) is treated as the radio resource.

In WCDMA, the Orthogonal Variable Spreading Factor (OVSF) [7] technique is used to preserve the orthogonality among the radio channels for different users. To provide transmission services with variable rates, physical channels are assigned the OVSF codes with different Spreading Factors (SFs). The transmission rate supported by a physical channel increases with the decreasing of SF of the OVSF code assigned to the channel. The rule for the generation of OVSF codes is based on a complete binary tree with h + 1layers (from Layer 0 to Layer h) as shown in Fig. 2. Each OVSF code corresponds to a node in the tree, which can be denoted as  $C_{k,n}$  where k is the layer number  $(0 \le k \le h)$ , and *n* is the position number in Layer  $k(1 \le n \le 2^k)$ . The SF of an OVSF code in Layer k is  $2^k$ . Assume that the OVSF code  $C_{h,n}$  supports transmission rate r bits/s. Typically, the r value is  $8 \times 1024$ , and it may vary according to the network status (e.g., signal-to-interference ratio). The maximum SF is 512 for downlink transmission in WCDMA [7]. The transmission rate  $R_k$  supported by  $C_{k,n}$  is

$$R_k = 2^{h-k} r \text{ bits/s} \tag{1}$$

As mentioned previously, the conversational and streaming traffics have the fairly constant characteristics. Thus the dedicated physical channels are preferred channels to serve these two kinds of traffics. For the interactive traffic, due to the burstness characteristics, the shared channel technology is proposed to serve it, where multiple users may share the same physical channel, Physical Downlink Shared Channel (PDSCH). The efficient scheduling of a PDSCH to serve interactive traffics may increase the system utilization and the QoS of UMTS. For this issue, a packet scheduling function is accommodated in the UTRAN [7]. Two approaches, code division or time division, may be adopted for packet scheduling as shown in Fig. 3. In the code division manner (see Fig. 3(a)), a PDSCH channel is divided into several OVSF sub-code channels to simultaneously serve different users with a lower transmission rate. When the number of users increases, the transmission rate (that can be allocated for a single user) decreases. In the time division manner (see Fig. 3(b)), the PDSCH is assigned to one user at each moment of time. Thus a user can be served with a high transmission rate in a short time period.

Since the interactive traffic has the burstness characteristics, lots of packets may arrive in a short period. Thus the number of packets queued in the buffer increases rapidly in a short period. In the previously proposed scheduling algorithms [6, 8, 15], the network reassigns OVSF sub-code channels to the connections according to the number of packets to be transmitted. The algorithms in [6, 8, 15] relied on the OVSF sub-code channel approach. In this approach, each interactive connection is allocated with a dedicated OVSF sub-code, which behavior is similar to that of the dedicated channel approach. The difference is that the allocated OVSF sub-code may change when the traffic of the connection becomes bursty, and the complexity of the algorithms in [6, 8,15] is considered high. Instead, the time division scheduling approach allocates total transmission rate of a PDSCH to a connection for a short period, which satisfies the burstness characteristics of the interactive traffic. Few previous works conducted complete study on applying the time division approach for packet scheduling.

Consider the time division approach as shown in Fig. 3(b). To assign the PDSCH to an interactive connection, an extra control signaling is required to inform the UE the assignment. Thus, if the time interval reserved for a connection to use the PDSCH is too short, lots of control signaling will be introduced. However, if the time interval is too long, other connections have to wait for a long time to get the service from the PDSCH. Thus, it is a very important issue to carefully schedule the PDSCH to serve different connections in the time division manner, which significantly affects the radio resource utilization as well as the QoS of the network. In this paper, we propose a highly efficient scheduling algorithm named Shared-Channel Assignment and Scheduling (SCAS) for PDSCH allocation. In SCAS, the PDSCH is scheduled to serve different interactive connections periodically. We can properly set up the time interval within which an interactive connection can be served so that a connection need not wait for a long time to get service. Furthermore, with "periodicity", the control signaling is required only before the periodicity begins, and thus the number of signaling can be reduced.

In this paper, we propose a highly efficient scheduling algorithm SCAS with a simple implementation. Three important properties of the SCAS algorithm are shown as follows: (i) SCAS guarantees that the packets of the interactive connections can be delivered with a transmission rate no less than the requested transmission rate; (ii) any two interactive connections served by the same PDSCH would not be scheduled for services in the same time intervals; (iii) each PDSCH can be fully utilized to serve interactive connections, and SCAS is considered an optimal periodic scheduling algorithm. Furthermore, simulation experiments are conducted to investigate the performance of the SCAS algorithm in terms of the packet waiting time and channel utilization. In our study, we compare the channel utilization and average waiting time performance between our SCAS algorithm and the dedicated



Fig. 3 The code and time division manners

channel approach. The simulation results show that the average waiting time performance of our proposed algorithm is close to that of the dedicated channel approach, and the channel utilization of the interactive connection is close to 100%. To clearly compare the complexity of the proposed SCAS algorithm and that of the algorithms in [6, 8, 15], we analyze the computation complexity, signaling exchange overhead, and traffic adaptability. Our analysis shows that the SCAS algorithm has the lowest computation complexity and slight signaling exchange overhead with traffic adaptability.

The rest of this paper is organized as follows. Section 2 details the SCAS algorithm. Section 3 explores the properties of the SCAS algorithm. In Section 4, we conduct simulation experiments to investigate the performance of the SCAS algorithm. Section 5 gives a concluding remark.

#### 2. The SCAS algorithm

Figure 4 illustrates the packet scheduler architecture for the PDSCH with  $C_{k,n}$ . In this figure, there are *L* interactive connections (labeled with  $Q_1, Q_2, \ldots, Q_L$ ) and several background connections served by this PDSCH. Each  $Q_i$  connection has its own buffer to queue its packets to be transmitted, and a single buffer is shared by all of the background connections. Let *S* be the set of the *L* interactive connections delivered by  $C_{k,n}$ , which is denoted as

$$S = \{Q_1, Q_2, \ldots, Q_L\}$$

Our algorithm periodically reserves the *Transmission Time Intervals* (TTIs) [7] to serve the interactive connections, where a TTI is the basic time interval unit within which the WCDMA system can serve a connection. The packet transmission for a connection starts at the beginning of a TTI. The length of a TTI can be 10, 20, 40, or 80 ms [7]. Let *D* denote the length of a TTI, and  $T_i$  be the time period between two consecutive TTIs reserved for  $Q_i$ . The SCAS algorithm exercises as follows:

The SCAS algorithm consists of two phases: the assignment phase and the scheduling phase. The flowcharts for the SCAS algorithm are shown in Figs. 5 and 6, which consists of



the assignment phase and the scheduling phase. The assignment phase assigns a PDSCH to the interactive connection request. When the request of a new interactive connection is granted, or the transmission for an interactive connection is completed by a PDSCH, the scheduling phase exercises to reschedule the transmissions for the connections. The details of the two phases are given below.

**The Assignment Phase:** Suppose that the new interactive connection  $Q_j$  requests the transmission rate  $R_{q,j}$  bits/s. As shown in Block 1 of Fig. 5, the network attempts to allocate  $Q_j$  the transmission rate

$$\gamma_i = 2^{a_j} r \text{ bits/s}$$
 where  $a_i \in \{0, 1, 2, 3, \dots, h\}$  (2)

such that  $\gamma_i$  is larger than or equal to  $R_{q,i}$ .

In Block 2 of Fig. 5, we select a PDSCH (served with  $C_{k,n}$ ) to support the  $Q_j$  connection based on the utilization U of the PDSCH, which is defined as

$$U = \frac{\sum_{Q_i \in S} \gamma_i}{R_k} \tag{3}$$

The PDSCH satisfying the following two conditions will be used to serve the  $Q_i$  connection.

**Condition 1.** The remaining transmission rate of the PDSCH is larger or equal to  $\gamma_i$ , i.e.,

$$R_k - \sum_{Q_i \in S} \gamma_i \ge \gamma_j \tag{4}$$

**Condition 2.** The utilization of the PDSCH (i.e.,  $U = \frac{\gamma_j + \sum_{Q_j \in S} \gamma_j}{R_k}$ ) is the largest.

In Block 3 of Fig. 5, if such a PDSCH is available, then this PDSCH is used to serve the request, and its corresponding *S* is updated as

$$S \leftarrow S \cup \{Q_j\}$$

Otherwise (i.e., no such PDSCH exists), the request  $Q_j$  is rejected.



Background Connection Requests



Fig. 5 The flowchart of the assignment phase

Note that if the r value changes due to the change of the network status, the assignment phase is re-executed to continue the packet transmission for the  $Q_i$  connection.

**The Scheduling Phase:** This phase exercises after a new interactive connection request is granted. The flowchart for this phase is shown in Fig. 6. Suppose that the PDSCH is allocated the OVSF code  $C_{k,n}$  and can support a transmission rate  $R_k$ . This phase re-reserves the TTIs for the interactive connections served by this PDSCH. As shown in Block 1 of Fig. 6, first, the network determines the length (known as  $T_i$ ) of time period between two consecutive

$$T_i = D\left(\frac{R_k}{\gamma_i}\right) \tag{5}$$

The network will then periodically reserve the TTIs for the connection  $Q_i$  according to its  $T_i$ . The "reserve" operations for different connections in *S* are performed in the increasing order of  $T_i$  to guarantee that the SCAS algorithm would not reserve the same TTIs for any two interactive connections. This property will be proven in Theorem 2. We reserve the TTIs for the connections with smaller  $T_i$  first. If tie-breaking occurs (i.e., there exists two connections with the same the  $T_i$  value), the TTIs for the two connections can be reserved in any order. If the TTI beginning at *t* is reserved for the connection  $Q_i$ , the next TTI reserved for  $Q_i$  must be apart from *t* for  $T_i$  (i.e., at  $t + T_i$ ).

In Block 2 of Fig. 6, the packets for  $Q_i$  are transmitted in the reserved TTIs. Figure 7 shows an example for the transmission scheduling of the connections that share the PDSCH with  $C_{k,n}$ . The time period of the  $Q_i$  connection is 4D (i.e.,  $T_1 = 4D$ ), and  $Q_1$  is scheduled to transmit packets at the TTIs  $t_0, t_4, t_{12}, ...$  The time period of the  $Q_2$  connection is 8D (i.e.,  $T_2 = 8D$ ), and  $Q_2$  is scheduled to transmit packets at the TTIs  $t_1, t_9, t_{17}, ...$ 

When there are packets for the background connections to be transmitted, the SCAS algorithm schedules them to be transmitted in two kinds of TTIs: (i) the TTIs that are not reserved to serve any interactive connection, or (ii) the TTIs that have been reserved for an interactive connection  $Q_i$  but are idle (i.e., no packets for  $Q_i$  to be transmitted). As shown in Fig. 7, the TTIs, e.g.,  $t_2, t_3$ , and  $t_5$ , can be scheduled to serve background connections. Note that in this example, the TTI  $t_8$  can be scheduled to serve a background connection when  $Q_1$  has no packets to be delivered.

In Block 3 of Fig. 6, the network checks it any new interactive connection is granted to be served with this PDSCH. If so, the PDSCH is rescheduled. Otherwise, the network transmits the packets in the next TTI.

#### 3. The properties of the SCAS algorithm

In this section, we explore important properties of the SCAS algorithm. We formally prove that

• the SCAS algorithm guarantees that packets of the interactive connections can be delivered with a transmission rate no less than the requested transmission rate,



Fig. 6 The flowchart of the scheduling phase

- any two interactive connections served by the same PDSCH would not be scheduled to be transmitted in the same TTIs (i.e., no collision occurs), and
- each PDSCH can be fully utilized to serve the interactive connections, and SCAS is considered an optimal periodic scheduling algorithm.

bursty period constitutes a bursty sequence of packets. Hence, within a bursty period, there are packets in the buffer to be transmitted.

Let V be the total number of bits in a bursty period, and  $T_t$  be the time difference between the transmission of the first and last bits generated within the bursty period. Define the average transmission rate  $R_a$  in the bursty period as

As shown in Fig. 8, the packet service connection for an interactive application contains one or several bursty periods, which is known as the ETSI packet data model [5]. The

 $R_a = \frac{V}{T_t}$ 

**Fig. 7** An example for scheduled connections sharing the PDSCH with  $C_{k,n}$ 







 $X_{i,i}$ : the *j*th bursty period for the  $Q_i$  connection

In Theorem 1, we show that by using the SCAS algorithm, in the bursty periods, the packets of the served interactive connection  $Q_i$  can be transmitted with rate  $R_a$  no less than the requested transmission rate  $\gamma_i$ .

**Theorem 1.** Suppose that by using (4), the SCAS algorithm assigns the PDSCH with  $C_{k,n}$  (supporting transmission rate  $R_k$ ) to the interactive connection  $Q_i$  (that requests transmission rate  $\gamma_i$ ), and by using (5), SCAS periodically transmits the packets for  $Q_i$ . Then during any bursty period  $X_{i,i}$  of the connection  $Q_i$ ,  $R_a$  is larger than or equal to  $\gamma_i$ 

Proof: Assume that the total volume of data to be delivered in a bursty period is V bits. By using PDSCH with transmission rate  $R_k$ , there are  $DR_k$  bits that can be delivered in each TTI reserved for  $Q_i$ . The number N of TTIs required to deliver the packets in the bursty period is

$$N = \left\lceil \frac{V}{DR_k} \right\rceil$$

Based on (5),  $T_i$  is equal to  $\frac{DR_k}{\gamma_i}$  or  $\gamma_i$  is equal to  $\frac{DR_k}{T_i}$ . Therefore, the average transmission rate over any integral number of such periods (i.e.,  $T_i$ ) will remain equal to  $\gamma_i$ . Among the N TTIs, the last TTI may not be fully utilized to transmit the packet data for  $Q_i$ . Considering the incomplete part of the last TTI, the transmission rate is  $R_k$ , which is always larger than or equal to  $\gamma_i$ . Thus, we have that the packets of the served interactive connection  $Q_i$  can be transmitted with rate  $R_a$  no less than the requested transmission rate  $\gamma_i$  in the bursty periods.  $\square$ 

Because SCAS periodically reserves TTIs to serve different interactive connections, one may question whether a TTI is reserved for more than one connection, and thus collision occurs. In the following, we will prove that such event would not occur in SCAS.

Lemma 1. Suppose that the TTIs in a PDSCH are sequentially labelled by  $t_0, t_1, t_2, t_3, \ldots$  Let  $t_x$  be the first TTI reserved for the connection  $Q_i$ . Then the number x is less than  $\frac{T_i}{D}$ .

**Proof:** The proof is completed by contradiction. Suppose that  $t_x$  be the first TTI reserved for the  $Q_i$  connection. Let  $z = \frac{T_i}{D}$ . Assume that  $x \ge z$ . In SCAS, the TTIs are reserved in the increasing order of the calculated time period. Since  $t_x$  is the first TTI for  $Q_i$ , and x is larger than or equal to z (i.e.,  $x \ge 1$ z), we know that the TTIs whose indexes are less than z (i.e.,  $t_0, t_1, \ldots, \text{ and } t_{z-1}$ ) are already reserved for other interactive connections. Without loss of generality, we denote these interactive connections as  $Q_0, Q_1, Q_2, Q_3, \dots, Q_{z-1}$ , and let S' be the set of  $Q_0, Q_1, Q_2, Q_3, ...,$  and  $Q_{z-1}$ , i.e., S' =  $\{Q_0, Q_1, Q_2, Q_3, \dots, Q_{z-1}\}$ . Suppose that the time periods for  $Q_0, Q_1, Q_2, Q_3, \ldots$ , and  $Q_{z-1}$  are  $T_0, T_1, T_2, T_3, \ldots$ , and  $T_{z-1}$ , respectively. Then,  $T_0, T_1, T_2, T_3, \ldots$ , and  $T_{z-1}$ must be less than or equal to  $T_i$ . From (1), (2), and (5), the time period  $T_i$  for any interactive connection  $Q_i$  can be presented by  $2^{h-k-a_j}D$  where  $h-k-a_j$  is an integer. Then for any connection  $Q_i$  in S', the following relationship with  $Q_i$ holds:

$$T_i = 2^{bj} T_j$$
 where  $0 \le j \le z$  and  $b_j \ge 0$  (6)

In the SCAS algorithm, the TTIs for the interactive connection  $Q_i$  are reserved in every  $T_i$  period. That is, the TTIs,  $t_j, t_{\frac{T_j}{D}+j}, t_{\frac{2T_j}{D}+j}, t_{\frac{3T_j}{D}+j}, \dots$  are reserved for the  $Q_j$  connection. Thus, we have that for  $m \ge 0$ , the TTIs

$$t_{\frac{mT_0}{D}}, t_{\frac{mT_1}{D}+1}, t_{\frac{mT_2}{D}+2}, t_{\frac{mT_3}{D}+3}, \dots,$$
  
and  $t_{\frac{mT_{z-1}}{D}+z-1}$ 

are reserved for  $Q_0, Q_1, Q_2, Q_3, \ldots, Q_{z-1}$ , respectively. Also, we have that for  $m \ge 0$ , the TTIs

$$t_{(2_0^b m T_0)/D}, t_{(2_1^b m T_1)/D+1}, t_{(2^b 2 m T_2)/D+2}, t_{(2_3^b m T_3)/D+3}, \dots,$$
  
and  $t_{(2_{z-1}^b m T_{z-1})/D+z-1}$  (7)

are reserved for  $Q_0, Q_1, Q_2, Q_3, \ldots, Q_{z-1}$ , respectively. By applying (6) into (7), we have that for  $m \ge 0$ , the TTIs

$$t_{mz}, t_{mz+1}, t_{mz+2}, t_{mz+3}..., \text{ and } t_{mz+z-1}$$
 (8)

are reserved for  $Q_0, Q_1, Q_2, Q_3, \ldots, Q_{z-1}$ , respectively. Thus, by (8) all TTIs in this PDSCH are reserved to serve the connections in S' which does not include  $Q_i$ . This contradicts

the assignment rule (4), since the  $Q_i$  connection is also served by this PDSCH. Thus we have that x is less than  $\frac{T_i}{D}$ .

**Theorem 2.** Consider any two connections  $Q_i$  and  $Q_j$  (requesting the transmission rates  $\gamma_i$  and  $\gamma_j$ , respectively) served by the PDSCH with  $C_{k,n}$ . The SCAS algorithm would not reserve the same TTIs for  $Q_i$  and  $Q_j$ .

**Proof:** From (5), we have the lengths  $T_i$  and  $T_j$  between two consecutive TTIs reserved for  $Q_i$  and  $Q_j$  as follows.

$$T_i = D\left(\frac{R_k}{\gamma_i}\right) \quad \text{and} \quad T_j = D\left(\frac{R_k}{\gamma_j}\right)$$
(9)

Suppose that the TTIs in the PDSCH are sequentially labelled by  $t_0, t_1, t_2, t_3, ...$  Without loss of generality, we assume that the TTIs reserved for  $Q_i$  and  $Q_j$  are  $t_{x+O(T_j/D)}$  and  $t_{y+p(T_j/D)}$ .  $t_x$  and  $t_y$  are the first TTIs reserved to serve  $Q_i$  and  $Q_j$ , and o and p are any of positive integers. Since  $t_x$  and  $t_y$  are the first TTIs reserved for  $Q_i$  and  $Q_j$ , by Lemma 1, we have that x is less than  $T_i/D$  and y is less than  $T_j/D$ . We complete the proof by considering the relationship between the requested transmission rates  $\gamma_i$  and  $\gamma_j$  in the following two cases:

**Case I.**  $\gamma_i = \gamma_j$ , i.e., the two connections request the same transmission rate. Since  $\gamma_i = \gamma_j$ , from (9), we have that  $T_i$  is equal to  $T_j$ . The all indexes of the TTIs (reserved for  $Q_i$  and  $Q_j$ ) have the following relationship. For any o and  $p \in \text{positive integers}$ ,

$$[x + o(T_i/D)] - [y + p(T_j/D)]$$
  
= (x - y) + [o(T\_i/D) - p(T\_j/D)] (10)

As described in the Scheduling phase of SCAS, the TTIs for these two connections can be scheduled to be served in any order for the first transmission since  $\gamma_i = \gamma_j$  (i.e., tie-breaking occurs). Hence, *x* is no equal to *y*. Since

$$x > T_i/D$$
,  $y < T_j/D$ , and  $T_i/D = T_j/D$ 

we have

$$x - y < T_i/D$$
 and  $(x - y) + (o - p)(T_i/D) \neq 0$ 

Thus

$$[x + o(T_i/D)] - [y + p(T_i/D)] \neq 0$$

or

$$x + o(T_i/D) \neq y + p(T_j/D)$$

Deringer

Thus in this case, no TTIs are reserved for both  $Q_i$  and  $Q_j$  at the same time.

**Case II.**  $\gamma_i \neq \gamma_j$ , i.e., the two connections request different transmission rates. Without loss of generality, we suppose that  $\gamma_i$  is less than  $\gamma_j$ . Then from (9), the following inequality holds:

$$T_i > T_j \tag{11}$$

From (1) and (2), we know that  $R_k$ ,  $\gamma_i$ , and  $\gamma_j$  are powers of 2. Then, by using (4), we have that  $\gamma_i$  and  $\gamma_j$  are less than  $R_k$ . According to (9), we can rewrite  $T_i$  and  $T_j$  as follows.

$$T_i = 2^{a_i} D$$
 and  $T_j = 2^{a_j} D$  where  $a_i > a_j$ 

or

$$T_i = 2^b T_j \quad \text{where} \quad b = a_i - a_j \tag{12}$$

Consider all indexes of the TTIs (reserved to serve  $Q_i$  and  $Q_j$ ), which are

$$x + o(T_i/D)$$
 for  $Q_i$  and  $y + p(T_j/D)$  for  $Q_j$ 

where *o* and *p* are positive integers. As described in the scheduling phase in the SCAS algorithm, since  $\gamma_i < \gamma_j$ , the TTIs for  $Q_j$  would be reserved before the TTIs for  $Q_i$  are reserved. Hence we have that

$$x > y \tag{13}$$

and

$$x \neq y + p(T_j/D)$$
 or  $x - [y + p(t_j/D)] \neq 0$ 
  
(14)

From (13), for any negative integer p', the following inequality holds.

$$x > y > y + p'(T_i/D)$$

Thus,

$$x \neq y + p'(T_j/D)$$
 or  $x - [y + p'(t_j/D)] \neq 0$  (15)

The all indexes of the TTIs reserved for  $Q_i$  and that for  $Q_j$  have the following relationship. By using (12), i.e.,  $T_i = 2^b T_j$ , for all *o* and  $p \in$  positive integers, we have

$$[x + o(T_i/D)] - [y + p(T_j/D)]$$
  
= x - [y + (p - o2<sup>b</sup>)(T\_j/D)] (16)

By using (14) and (15), from (16), we obtain the following inequality.

$$[x + o(T_i/D)] - [y + p(T_i/D)] \neq 0$$
(17)

Hence, the statement "no TTIs are reserved for both  $Q_i$  and  $Q_i$ " holds in this case.

The above two cases complete the proof.

In [14], Liu and Layland have shown that for any periodic scheduling algorithm, to enure that no collision occurs, the bandwidth utilization U is bounded by  $n(2^{1/n} - 1)$ , i.e.,

$$U \le n(2^{1/n} - 1)$$

where *n* is the number of connections. For a large *n*,  $n(2^{1/n} - 1)$  approximates to ln 2. In UMTS, the requested transmission rate  $\gamma_i$  of a connection is  $2^{a_i}r$  bits/s (i.e.,  $\gamma_i = 2^{a_i}r$  bits/s) where  $a_i$  is a non-negative integer, and  $a_i$  is no less than 0 and no larger than *h*. This releases the bandwidth utilization bound in [14]. With the release, our SCAS algorithm can fully utilize a PDSCH (i.e., *U* can be achieved 100%) to serve interactive connections, and thus the SCAS algorithm is considered an optimal periodic scheduling algorithm. In the following corollary, we show this property of the SCAS algorithm.

**Corollary 1.** Consider a PDSCH with the OVSF code  $C_{k,n}$ , which supports the transmission rate  $R_k$ . The SCAS algorithm can fully utilizes the PDSCH to serve interactive connections (i.e.,  $U \leq 100\%$ ), and no collision occurs.

**Proof:** Suppose that there are *L* interactive connections served by the PDSCH. From Theorem 2, if the total requested transmission rate for the *L* interactive connections is no larger than  $R_k$ , that is,

$$\sum_{i=1}^{L} \gamma_i \le R_k \tag{18}$$

then SCAS can serve these connections without collisions. From (18) and (3), we have that

$$U = \frac{\sum_{i=1}^{L} \gamma_i}{R_k} \le 100\% \tag{19}$$

Hence we have  $U \leq 100\%$ , and U = 100% occurs when

$$\sum_{i=1}^{L} \gamma_i = R_k$$

In this section, we conduct simulation experiments to investigate the performance of the SCAS algorithm. Then, we analyze the computation complexity, signaling exchange overhead, and traffic adaptability for the SCAS algorithm and the algorithms in the previous studies.

#### 4.1. The performance of SCAS

This subsection examines the following two issues for the SCAS algorithm.

- **Issue 1.** As pointed out in the proof for Theorem 1, in a bursty period, the last TTI may not be fully utilized to serve an interactive connection. Obviously, this phenomenon is closely related to the length of a TTI D. That is, as D increases, the utilization of TTIs used to transmit packets for  $Q_i$  may decrease.
- **Issue 2.** In the assignment phase (i.e.,  $R_k \sum_{Qi \in S} \gamma_i \ge \gamma_n$ ; see (4)), the selection of a PDSCH that will serve the  $Q_i$  connection may affect the time period  $T_i$  (i.e.,  $T_i = D(R_k/\gamma_i)$ ; see (5)) between two TTIs for  $Q_i$ . From (5), it is obvious that as the rate (i.e.,  $R_k$ ) supported by the selected PDSCH is larger, a larger value of  $T_i$  is required for the transmissions of  $Q_i$ . This may increase the waiting time of a buffered packet of  $Q_i$ .

If an interactive connection  $Q_i$  is served by a dedicated physical channel (supporting the transmission rate  $R_d$  no less than the requested transmission rate  $\gamma_i$ ), the packets for  $Q_i$  may have shorter waiting times than that served by a PDSCH. However, poor channel utilization may be incurred due to the bursty traffic characteristic of the interactive connection. In this paper, we conducts simulation experiments to compare the performances of the SCAS algorithm and the dedicated physical channel approach in terms of the waiting time and channel utilization. To simplify our description, we use "DCA" to denote the dedicated physical channel approach.

The simulation technology, the discrete event-driven approach, is used in our study, which has been used in many mobile network studies [9, 10, 13]. In the simulation model, we define two types of events, the Packet-Arrival and Schedule\_Trans events, to represent a packet arrival and data transmission for an interactive connection, respectively. The events are inserted into an event list, and are deleted/processed from the event list in non-decreasing timestamp order. A simulation clock is maintained to indicate the progress of the simulation. In each experiment, more than 100,000 incoming packet arrivals are simulated to ensure that the results are stable.

As mentioned in Section 2, the packet transmission for a connection starts at the beginning of a TTI in WCDMA. From the scheduling phase of SCAS, we know that when a TTI (reserved for an interactive connection) is not used to transmit the packets for the interactive connection, this TTI can be used to serve the background connections. Based on our simulation experiments, we calculate the utilization  $U_i$ (SCAS) for the TTIs scheduled to transmit the packets for an interactive connection by using

$$\frac{\sum(\text{actual time spent in the TTIs used to transmit packets for } Q_i)}{\sum(\text{the length of the TTIs scheduled to transmit packets for } Q_i)}$$

On the other hand, in DCA, all TTIs of a dedicated physical channel are reserved for an interactive connection  $Q_i$ . If a TTI is not used to transmit the packets for an interactive connection, this TTI cannot be utilized to serve the other connections (i.e., the other interactive and background connections). Therefore, we measure the utilization  $U_i$  (DCA) for the TTIs reserved to transmit the packets for an interactive connection in DCA by using the following equation

$$U_i(\text{DCA}) = \frac{\sum(\text{actual time spent in the TTIs used to transmit packets for } Q_i)}{\sum(\text{the length of all TTIs reserved for } Q_i)}$$
(21)

Let  $w_j$  denote the period between the time when the *j*th packet of  $Q_i$  arrives and the time when the *j*th packet begins being transmitted, and  $n_i$  be the total number of packets of  $Q_i$  to be transmitted. In either SCAS or DCA, we can have the average waiting time  $W_i$  for a packet of  $Q_i$  in the buffer as follows:

$$W_{i} = \frac{\sum_{j=1}^{n_{i}} w_{j}}{n_{i}}$$
(22)

The traffic model used in our study is described as follows. We assume that the times between any two consecutive packet arrivals of  $Q_i$  form a Pareto distribution. The Pareto distribution is widely used to approximate the WWW packet traffic very well [3, 4]. In the study [6], the Pareto assumption has also been adopted to investigate the performance for the dynamic code assignment algorithm for WCDMA shared channel approach. The Pareto distribution has the density function

$$f_p(t) = \left(\frac{\beta}{l}\right) \left(\frac{l}{t}\right)^{\beta+1}$$
(23)

and the expected value

(20)

$$E[t] = \left(\frac{\beta}{\beta - l}\right)l\tag{24}$$

where  $\beta$  describes the "heaviness" of the tail of the distribution. If  $\beta$  is between 1 and 2, then the variance for the distribution becomes infinity. Once a suitable value for  $\beta$  is selected to describe the traffic characteristics, then l can be determined by using (24). In this study, we assume that the mean for the packet inter-arrival times of  $Q_i$  is  $E[t] = 1/\lambda_i$ , and we select  $\beta = 1.21$  as that used in [6]. By substituting  $\beta = 1.21$  and  $E[t] = 1/\lambda_i$  into (24), we have  $l = \frac{2l}{121\lambda_i}$ . Suppose that the sizes for the packets of the  $Q_i$  connection are Exponentially distributed with mean  $1/\mu_i$ , [12]. If the physical channel with transmission rate  $R_k$  is allocated to serve  $Q_i$ , then the times required to deliver each packet of  $Q_i$  also form the Exponential distribution with mean  $1/(R_k \mu_i)$ . In our study, we simulate that an interactive connection  $Q_i$  requests transmission rate  $\gamma_i$  equal to 8r bits/s (where r is equal to  $8 \times 1024$ ) for packet delivery. Following the traffic model in [4, 15], the mean  $1/\mu_i$  of the size of each packet is set to 500 bytes.

In the following, we study the effects of  $\lambda_i$ , D, and  $R_k$  on SCAS algorithm in terms of the  $U_i$  and  $W_i$  performances. We compare the performances of the SCAS and DCA (where the dedicated channel can support the transmission rate  $R_d$  equal to 8r bits/s.

- **Effects of**  $\lambda_i$ . Figure 9 plots the  $U_i$  and  $W_i$  values against  $\lambda_i$  for the SCAS algorithm for different *D* setups (i.e., D = 10 ms, D = 20 ms, D = 20 ms, D = 40 ms, and D = 80 ms; see the solid lines), where packet arrival rate  $\lambda_i$  ranges from 1/sec to 9/sec. Figure 9(a) shows that as  $\lambda_i$  increases, the better  $U_i$  performance is observed. This reflects that as the packet traffic becomes large, the TTIs reserved to serve the  $Q_i$  connection are more likely to be occupied, and thus better channel utilization is observed. As shown in Fig. 9(b),  $W_i$  is an increasing function of  $\lambda_i$  for the SCAS algorithm. This is due to that as  $\lambda_i$  increases, more packets arrive in a short time period, and these packets compete for the transmission services. Hence, the packets spend more time waiting in the buffer of  $Q_i$ .
- **Effects of** *D*. In Fig. 9, we also observe that as the length of a TTI (i.e., *D*) decreases, the better  $U_i$  and  $W_i$  performances can be achieved. As *D* increases, the number of bits that can be delivered in a TTI increases, and it is more likely that the last TTI (used to transmit the packets in a bursty period) cannot be fully utilized. The phenomenon shown in this figure supports the statement in Issue 1 properly. For the  $W_i$  performance, as *D* becomes large, the  $T_i$  between two consecutive TTIs reserved for  $Q_i$  gets larger (as pointed out in Issue 2). Thus, when a packet of  $Q_i$  arrives, it is



**Fig. 9** Effects of  $\lambda_i$  and  $D(R_k = 32r \text{ bits/s}, \gamma_i = 8r \text{ bits/s}, 1/\mu_i = 500 \text{ bytes})$ 

likely to spent more time waiting transmission service. To conclude, in our SCAS algorithm, the length of a TTI is preferred to be set small, and better system performance can be achieved.

**Effects of**  $R_k$ . In the assignment phase of SCAS, one of the PDSCHs (supporting different transmission rate  $R_k$ ) may be selected to serve  $Q_i$  (see (4)). In Fig. 10, we study the effects of the transmission rate  $R_k$  of the PDSCH selected to serve the  $Q_i$  connection on the  $U_i$  and  $W_i$  performance for SCAS (see solid lines), where  $R_k = 8r$  bits/s, 16r bits/s, 32r bits/s, or 64r bits/s, D = 10 ms,  $\gamma_i = 8r$ bits/s, and  $1/\mu_i = 500$  bytes. Figure 10(a) indicates that as  $R_k$  is smaller, the  $U_i$  performance improves. The number of bits that can be transmitted in a TTI is  $DR_k$ . Hence as  $R_k$  increases, there are more bits that can be delivered in a TTI, and it is more likely that the system does not fully utilize the last TTI to transmit the packets in a bursty period. As shown in Fig. 10 (b), the  $W_i$  performance decreases lightly as  $R_k$  increases. From (5), it is clear that  $T_i$ is proportioned to  $R_k$ . Hence a larger  $R_k$  implies a longer  $T_i$ , and the packet is likely to spend more time waiting in the buffer for service. To summarize, in the assignment phase of the SCAS algorithm, we prefer to select a PDSCH which transmission rate is closer to the requested transmission rate  $\gamma_i$  so that system performs more efficiently.





**Fig. 10** Effects of  $R_k$  (D = 10 ms,  $\gamma_i = 8r$  bits/s,  $1/\mu_i = 500$  bytes)

**Comparison between SCAS and DCA.** As mentioned previously, the system with DCA may have shorter waiting time but poor channel utilization than that with SCAS. In this study, we also run the simulation experiments for DCA, where the dedicated physical channel supporting transmission rate  $R_d$  equal to 8r bits/s is used to serve the  $Q_i$  connection with a requested transmission rate  $\gamma_i$  equal to 8r bits/s. Compare the solid lines (for SCAS) and the dashed lines (for DCA) shown in Figs. 9 and 10. We observe that when *D* is small (i.e. D = 10 ms), the SCAS significantly outperforms DCA in terms of the channel utilization  $U_i$  by slightly introducing waiting time  $W_i$ . This observation supports that the SCAS algorithm is preferred to be adopted to serve the interactive connections for better system utilization without decreasing Qos.

# 4.2. Comparison among the SCAS, CDGPS, CBA, and DOCA algorithms

In this subsection, to see the advantages of the proposed SCAS algorithm over the previously proposed algorithms, CDGPS [15], CBA [8], and DOCA [6], this subsection analyzes the computation complexity  $T_c$  introduced in each TTI of a shared channel, the number  $N_s$  of signaling exchanges for a connection, and traffic adaptability.

 Table 1
 Comparison among SCAS, CDGPS, CBA, and DOCA

Algorithms	CDGPS	CBA	DOCA	SCAS
$T_c$ $N_s$ Traffic adapt- ability	$\Theta(n_c)$ $\Theta(n_{TTI})$ High	$\Theta(n_c)$ $\Theta(n_{TTI})$ High	$\Theta(n_{c,h})$ $\Theta(n_{bp})$ Low	1 or 2 $\Theta(n_{bp})$ Low

In Table 1, we list the comparison results. We first analyze the computation complexity  $T_c$  for the four algorithms. In CDGPS [15], the connections of different users are assigned different weights. The code division operation is performed at the beginning of each time slot (i.e., TTI) based on the weights of the served connections and the volume of data currently queued in the buffer. Let  $n_c$  denote the number of connections currently served by a shared channel. We have the computation complexity  $T_c$  for CDGPS as  $T_c$ (CDGPS)  $= \Theta(n_c)$ . In CBA [8], the connection is initially assigned a leaf OVSF sub-code channel (i.e., the channel supporting the lowest transmission rate) of a PDSCH. When the data traffic for this connection becomes bursty, the network reassigns the connection an ancestor code of the leaf code. which supports a higher transmission rate. In other words, the number of code reassignment operations is dependent on the number of currently served connections. The computation complexity for CBA can be represented as  $T_c(CBA) =$  $\Theta(n_c)$ . In DOCA [6], connections are classified as high QoS connections and best-effort connections. The transmission rate reserved for the high QoS connections is dynamically changed according to its burstness. When the traffic for the high QoS connection becomes bursty, the code reassignment operation is performed to allocate an OVSF code supporting its peak transmission rate. In most of time, an OVSF code with lower transmission rate is reserved to serve the high QoS connection. Let  $n_{c,h}$  denote the number of high QoS connections currently served by a PDSCH. Then we have the computation complexity for DOCA as  $T_c(\text{DOCA}) = \Theta(n_{c,h})$ . In the proposed SCAS algorithm, each TTI of a PDSCH is reserved for a specific interactive connection. When there is no packets (for the connection) to be delivered, the TTI is used to deliver the packets of the background connections. Thus, in SCAS, only one or two "check" operations (that is, one is for the buffer of the interactive connection and the other is for the buffer of the background connection) are required in each TTI. Thus, the complexity  $T_c$  for SCAS is  $T_c(SCAS) = 2$ , which is a constant number. Among the four algorithms, the SCAS algorithm has the lowest computation complexity.

For the signaling exchange overhead, in CDGPS [15] and CBA [8], the OVSF sub-codes may be reallocated to the connections at the beginning of each TTI. If the OVSF sub-codes

are changed, the network has to exchange signaling with the UE to notify this change. Let  $n_{TTI}$  denote the number of TTIs in a PDSCH used to serve a connection. With CDGPS and CBA, the number of signaling exchanges required for a connection can be represented as  $N_s$  (CDGPS) =  $N_s$ (CBA) =  $\Theta(n_{TTI})$ . In DOCA [6], the code reassignment operation may be performed at the beginning of a bursty period of a connection. At this moment, the signaling exchanges are required to inform the UE. Let  $n_{bp}$  denote the number of bursty periods of a connection. Thus, with DOCA, the number of signaling exchanges required for a connection is  $N_s(\text{DOCA}) =$  $\Theta(n_{bv})$ . In SCAS, the TTIs are reserved to serve the interactive connection in the bursty periods of the connection. In the non-bursty periods, the TTIs are used to serve the background connections. The signaling exchanges between the network and the UEs are required in the end of every bursty period of a connection. We have the number  $N_s$  for SCAS as  $N_s(SCAS) = \Theta(n_{bp})$ . The SCAS has the same signaling exchange overhead as that of DOCA. Since  $n_{TTI} > n_{bp}$ , SCAS and DOCA outperform CDGPS and CBA in terms of signaling exchange overhead.

Finally, we compare the traffic adaptability. Define traffic adaptability as that the allocated radio resource for a connection can serve other connections while the served connection has no packets to be transmitted. The traffic adaptability is high if the allocated radio resource for a connection can serve all types of connections (including interactive and background connections in this paper). On the other hand, the traffic adaptability is low if the allocated radio resource can only serve different type of connections. In both CDGPS and CBA, the OVSF sub-code can be rescheduled to serve other connections (including interactive and background connections) when a connection has no packets to be delivered. The traffic adaptability of both CDGPS and CBA is high. In DOCA, different OVSF codes may be used to serve the high QoS connections when the traffic for the connections is changed. When the traffic load is below a threshold, the OVSF sub-codes are scheduled to serve best-effort connections. The traffic adaptability of DOCA is considered low. In SCAS, when there are no packets (for an interactive connection) to be delivered, the reserved TTIs for the connection can be scheduled to serve the background connections. The traffic adaptability of SCAS is low. In terms of traffic adaptability, the CDGPS and CBA outperform our SCAS algorithm.

To summarize, among the four algorithms, our proposed SCAS algorithm has the lowest computation complexity and slight signaling exchange overhead with traffic adaptability. Except the traffic adaptability, SCAS has better performance than that of CDGPS, CBA, and DOCA in other aspects.

### 5. Concluding remarks

The UMTS provides wideband and high-speed wireless transmission services with high and variable rates for mobile users. In UMTS, four classes of traffics are identified, which are conversational, streaming, interactive, and background traffics. To efficiently utilize radio resource, the shared radio channel, PDSCH, is proposed to serve the interactive and background connections. This paper proposed a highly efficient time-division-based algorithm named "Shared Channel Assignment and Scheduling" (SCAS) with an extremely simple run-time implementation to schedule a PDSCH to serve different interactive connections, which is an optimal periodic scheduling algorithm. We formally showed the following three properties of the SCAS algorithm: Our algorithm guarantees that the packets of the interactive connections can be transmitted with a transmission rate no less than the requested transmission rate. Any two interactive connections served by the same PDSCH are not scheduled for service at the same time. The PDSCH can be fully utilized to serve the interactive connections.

Simulation experiments were conducted to investigate the performance of the SCAS algorithm. Our study indicated that to optimize the performance of the SCAS algorithm, we should set the length of a TTI small, and select a PDSCH (that supports the transmission rate closer to the requested transmission rate of an interactive connection  $Q_i$ ) to serve  $Q_i$ . Furthermore, the SCAS algorithm significantly outperforms the dedicated physical channel approach in terms of the channel utilization with slightly increasing average waiting time.

Acknowledgments The authors would like to thank the two anonymous reviewers. Their comments have significantly improved the quality of this paper. P. Lin's work was sponsored in part by the National Science Council (NSC), R.O.C., under the contract number NSC 94-2213-E-002-083-, NSC 94-2213-E-002-090-, and NSC 94-2627-E-002-001-, Ministry of Economic Affairs (MOEA), R.O.C., under contract number 93-EC-17-A-05-S1-0017, the Computer and Communications Researches Labs/Industrial Technology Research Institute (CCL/ITRL), Chunghwa Telecom Labs, Microsoft, Telcordia Applied Research Center, Taiwan Network Information Center (TWNIC), and Microsoft Corporation, Taiwan.

#### References

- 3GPP. 3rd Generation Partnership Project; Radio Interface Protocol Architecture. Technical Report Technical Specification 3G TS 25.301 version 3.4.0 (2000–03), 2000.
- 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2. Technical Report Technical Specification 3G TS 23.060 version 4.1.0 (2001–06), 2001.
- M. Cheng and L.-F. Chang, Wireless dynamic channel assignment performance under packet data traffic, IEEE Journal on Selected Areas in Communications 17(7) (July 1999) 1257–1269.

- ETSI Universal Mobile Telecommunications System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS, Version 3.2.0. Technical Report TR 101 112, ETSI, 1998.
- ETSI/TC. UMTS Terrestrial Radio Access (UTRA); Concept Evaluation, Version 3.0.0. Technical Report Technical Report UMTS 30.06, ETSI, December 1997.
- C.E. Fossa, Jr. and N.J. Davis, IV. A dynamic code assignment algorithm for quality of service in 3G wireless networks, in: *Proceedings of IEEE WCNC2002* 1 (March 2002) 1–6.
- 7. H. Holma and A. Toskala, *WCDMA for UMTS*, John Wiley & Sons, 2000.
- A.-C. Kam, T. Minn and K.-Y. Siu, Supporting rate guarantee and fair access for bursty data traffic in W-CDMA, IEEE Journal on Selected Areas in Communications, 19(11) (November 2001) 2121– 2130.
- 9. P. Lin, Channel allocation for GPRS with buffering mechansims. *ACM/Baltzer Wireless Networks* 9(5) (September 2003) 431–441.
- P. Lin and Y.-B. Lin, Channel allocation for GPRS, IEEE Transactions on Vehicular Technology, 50(2) (March 2001) 375–387.
- P. Lin, C.-H. Gan, N.-C. Perng, T.-W. Kuo and C.-C. Hsu, Time division based shared channel allocation algorithm for UMTS, in: *Proceedings of IEEE WCNC2004*, (A-30–1), 2004.
- P. Lin, W.-R. Lai and C.-H. Gan, Modeling opportunity driven multiple access in UMTS. Accepted and to Appear in: IEEE Transactions on Wireless Communications, 2003.
- P. Lin, Y.-B. Lin and I. Chlamtac, Modeling frame syschronization for umts high-speed downlink packet access, IEEE Transactions on Vehicular Technology 52(1) (January 2003) 1321–141.
- C.L. Liu and J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, Journal of the Association for Computing Machinery (January 1973) 46–61.
- L. Xu, X. Shen and J.W. Mark, Dynamic bandwidth allocation with fair scheduling for WCDMA system, IEEE Wireless Communications 9(2) (April 2002) 26–32.



**Chai-Hien Gan** was born in Malaysia in 1971. He received his BS degree in computer science from Tamkang University in 1994, Taipei County, Taiwan, and both his MS. and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1996 and 2005, respectively. Since March 2005, he has been a Research Assistant Professor in Department of Computer Science, National Chiao Tung Univer-

sity, R.O.C. His current research interests include wireless mesh networks, mobile computing, personal communications services, and wireless Internet.



**Phone Lin** received his BSCSIE degree and Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C. in 1996 and 2001, respectively. From August 2001 to July 2004, he was an Assistant Professor in Department of CSIE and Graduate Institute of Graduate of Networking and Multimedia, National Taiwan University, R.O.C. Since August 2004, he has been an Associate Professor in Department of CSIE and Graduate Institute Graduate

of Networking and Multimedia, National Taiwan University, R.O.C. His current research interests include personal communications services, wireless Internet, and performance modeling. Dr. Lin is an Associate Editor for IEEE Transactions on Vehicular Technology, Editor for IEEE Wireless Communications special issue on Mobility and Resource Management and a Guest Editor for ACM/Springer MONET special issue on Wireless Broad Access. He is also an Associate Editorial Member for the WCMC Journal. P. Lin's email and website addresses are plin@csie.ntu.edu.tw and http://www.csie.ntu.edu.tw/~plin, respectively.



**Nei-Chiung Perng** is presently a Ph.D. student in the Department of Computer Science and Information Engineering, National Taiwan University. He received his Bachelor and Master degrees in the Department of Computer and Information Science, National Chiao Tung University in 1999 and 2001, respectively. His research interests include real-time systems and scheduling algorithms.



**Tei-Wei Kuo** received B.S.E. degree in computer science and information engineering from National Taiwan University in Taipei, Taiwan, in 1986. He received the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin in 1990 and 1994, respectively. He is currently a Professor and the Chairman of the Department of Computer Science and Information Engineering of the National Taiwan University, Taiwan, ROC.

He was an Associate Professor in the Department of Computer Science and Information Engineering of the National Chung Cheng University, Taiwan, ROC, from August 1994 to July 2000.

Dr. Kuo is a senior member of the IEEE computer society. His research interest includes embedded systems, real-time process scheduling, real-time operating systems, and real-time databases. He has over 100 technical papers published or been accepted in international journals and conferences and has a book "Real-Time Database Systems: Architecture and Techniques" published by Kluwer Academic Publishers (ISBN 0-7923-7218-2, USA). He is the Program Co-Chair of IEEE 7th Real-

Time Technology and Applications Symposium, 2001, and an associate editor of the Journal of Real-Time Systems since 1998.

He is an executive committee member of the IEEE Technical Committee on Real-Time Systems in 2005 and the steering committee chair of IEEE RTCSA'05. Dr. Kuo has consulted for government and industry on problems in various real-time and embedded systems designs. Dr. Kuo received several research awards in Taiwan, including the Distinguished Research Award from the ROC National Science Council in 2003 and the Young Scholar Research Award from Academia Sinica, Taiwan, ROC, in 2001.



Ching-Chi Hsu was born in Taipei, Taiwan in 1949. He received his BS degree in physics from National Tsing Hwa. University in 1971, Hsishu, Taiwan, and both his MS. and Ph.D. degrees in computer engineering from EE department of National Taiwan University, Taipei, Taiwan, in 1975 and 1982, respectively.

In 1977, he joined the faculty of the Department of Computer Science and Information

Engineering at National Taiwan University and became an associate professor in 1982. During the years between 1987 and 2002, he was first engaged as a professor and became the chairman of the department. During his tenure in National Taiwan University, Dr. Hsu was a visiting scholar of Computer Science Department, Stanford University from 1984 to 1985. After serving in National Taiwan University for over 25 years, Dr. Hsu had left and was promoted as the president of Kai Nan University in 2002. Starting from February 2004, Dr. Hsu has been the executive vice president of the Institute for Information Industry in which he is mainly in charge of accelerating the growth of information industry in the whole nation. His research interests include distributed processing of data and knowledge, mobile computing and wireless networks.