Time Division Based Shared Channel Allocation Algorithm for UMTS

Phone Lin[1], Chai-Hien Gan[2], Nei-Chiung Perng[3], Tei-Wei Kuo[4], and Ching-Chi Hsu[5]

Abstract—**The** *Universal Mobile Telecommunications System* **(UMTS) adopts WCDMA as the air interface to provide variable transmission rate services. Four traffic classes are identified in UMTS, which are conversational, streaming, interactive, and background classes. To efficiently utilize the radio bandwidth, the shared channel technology is proposed to deliver interactive and background traffics. This paper proposes a "Shared-Channel Assignment and Scheduling" (SCAS) algorithm to allocate shared channels to interactive and background connections. We formally prove that the SCAS algorithm can satisfy the transmission rate requirement of each interactive connection and can efficiently utilize the radio bandwidth.**

Keywords: Downlink Shared Channel, QoS, OVSF, UMTS, WCDMA

I. INTRODUCTION

The *Universal Mobile Telecommunications System* (UMTS) [2], [5] provides high and variable transmission rate services to mobile users. With UMTS, wireless multimedia communication can be enhanced with high quality, and packet access to information and services on Internet will be enhanced by higher data rates and new flexible communication capabilities. In UMTS, four traffic classes for connections are identified, which are *conversational*, *streaming*, *interactive*, and *background*. During a connection, the traffic for the conversational and streaming classes has the fairly constant characteristics. The typical applications for the conversational and streaming classes are speech and multimedia applications. The traffic of the interactive class is highly dependent on request-response patterns of end users (where a response is timely critical), which has the bursty characteristics. The typical applications for the interactive class include common Internet applications (e.g., web browsing and network gaming). The background class is for applications without stringent response-time requirement, such as the background e-mail application. It is usually served by the best-effort transmission service.

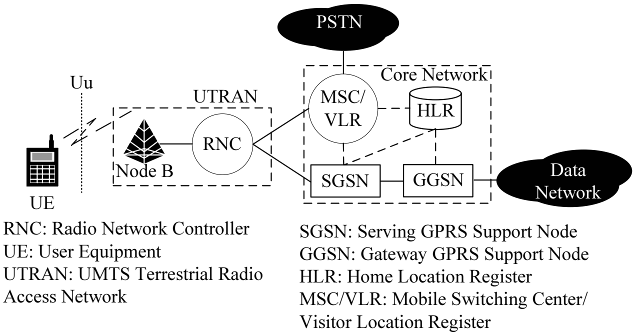
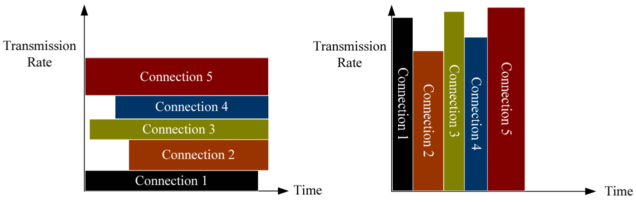


Fig. 1. The UMTS Network Architecture



(a) Code Division (b) Time Division

Fig. 2. The code and time division manners

As shown in Figure 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. In WCDMA, the *Orthogonal Variable Spreading Factor* (OVSF) [5] technique is used to preserve the orthogonality among the radio channels for different users. To provide variable transmission rate services, 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 $K + 1$ layers (from Layer 0 to Layer K). Each OVSF code is corresponding to a node in the tree, which can be denoted as $C_{k,n}$ where k is the layer number ($0 \leq k \leq K$), and n is the position number in Layer k ($1 \leq n \leq 2^k$). An OVSF code in Layer k has $SF = 2^k$. Let the OVSF code $C_{K,n}$ support transmission rate r bits/s. Typically, the r value approximates to 8×1024 and the maximum SF is 512 for downlink transmission in WCDMA [5]. The transmission rate R_k supported by $C_{k,n}$ is

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

Within the coverage area of a Node B, the total transmission rate that can be assigned to users is fixed. As mentioned above, the traffic for the conversational and stream classes has the fairly constant characteristics. Thus the dedicated physical channels are preferred to serve these two classes. For the interactive class, due to the bursty characteristics, a shared-channel-based technology is proposed to serve the interactive class, where different users share the same physical channel, *Physical Downlink Shared Channel* (PDSCH). Efficiently scheduling a PDSCH to serve different interactive connections may increase the system utilization and the QoS of UMTS, which becomes a very important issue. For this issue, a packet scheduling function is accommodated in the UTRAN [5], which can be done in two approaches, in a code or time division manner as shown in Figure 2. In the code division manner (see Figure 2 (a)), a PDSCH

---

[1] Contact Author: P. Lin, Department of Computer Science & Information Engineering, National Taiwan University, Taipei 106, R.O.C.; TEL: +886-2-23625336 ext. 329; FAX: +886-2-23628167; Email: plin@csie.ntu.edu.tw.

[2] C.-H. Gan, Dept. of Computer Science & Information Engineering, National Taiwan University, R.O.C.; Email: chyen@pcs.csie.ntu.edu.tw.

[3] N.-C. Perng, Dept. of Computer Science & Information Engineering, National Taiwan University, R.O.C.; Email: d90011@csie.ntu.edu.tw.

[4] T.-W. Kuo, Dept. of Computer Science & Information Engineering, National Taiwan University, R.O.C.; Email: ktw@csie.ntu.edu.tw.

[5] C.-C. Hsu, Dept. of Computer Science & Engineering, Yuan Ze University, Taiwan, R.O.C.; Email: cchsu@csie.ntu.edu.tw.

channel is divided into several OVSF sub-code channels (supporting lower transmission rates). A large number of users can be served with these low transmission rate channels simultaneously. When the number of users waiting for transmission services increases, the transmission rate (that can be allocated for a single user) decreases. In the time division manner (see Figure 2 (b)), the transmission rate of a PDSCH is given to one user at each moment of time. Thus a user can have a very high transmission rate but can use it only very briefly.

The studies [7], [6], [4] have been contributed for efficiently scheduling a PDSCH to serve different connections. The study [7] proposed the Code-Division Generalized Processor Sharing algorithm to fairly schedule a PDSCH to users in the code division manner. In this algorithm, the connections are assigned different weights. This algorithm divides a PDSCH into several sub-code channels (with different transmission rates) to serve different connections. The code division operation is performed at the beginning of each time slot based on the weights of the served connections and the ratio of the volume of the data of the connections currently queued in the buffer. In [6], the connection is initially assigned a leaf sub-code channel (i.e., the sub-code channel supporting the lowest transmission rates) of the PDSCH when there is data to be transmitted. When bursty data traffic arrives for this connection, the system reassigns a ancestor code of the leaf code, which supports higher transmission rate. The longer no data is transmitted for a connection, the larger credit is given to the connection. The system assigned the OVSF sub-code channels to the connections based on its credit. For the connection with larger credit, it is assigned the sub OVSF code that supports higher transmission rate. In [4], traffics are classified as the *high QoS* traffic and *best-effort* traffic. High QoS traffic requires a guaranteed bandwidth to support its required peak transmission rate. The bandwidth reserved for the high QoS traffic is dynamically changed according to its burstness. Best-effort traffic is allowed to utilize an OVSF sub-code channel when high QoS traffic is not served at peak transmission rate.

In the above previous studies, heavy computation may be introduced for scheduling in every transmission, and the need for changing the OVSF code is frequent during the transmission for a connection. To efficiently schedule a PDSCH to serve the connections, we proposes a time division approach named "Shared-Channel Assignment and Scheduling" (SCAS) algorithm (with low computation complexity and with QoS guarantee) to allocate PDSCHs to the interactive and background connections. We formally prove that

- our algorithm can guarantee that the interactive connections can be delivered with the requested transmission rates,
- any two interactive connections served by the same PDSCH would not be scheduled to be served at the same time (i.e., no collision occurs), and
- each PDSCH can be fully utilized to serve the interactive connections or background connections.

The reset of this paper is organized as follows. Section II details the SCAS algorithm. Section III proves the correctness of the SCAS algorithm. Section IV gives a concluding remark.

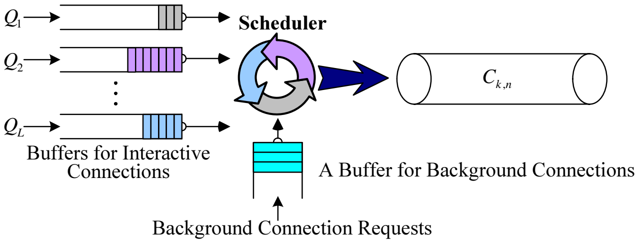


Fig. 3. The packet scheduler architecture

II. THE SCAS ALGORITHM

This section describes the SCAS algorithm. Figure 3 illustrates an example for the packet scheduler for the PDSCH with $C_{k,n}$, where the L interactive connections labelled with $Q_1, Q_2, \cdots, Q_L$, and background connections are served. The Q_i connection has its own buffer to queue the 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

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

In UMTS, the transmission rate γ_i requested by the Q_i connection is

$$\gamma_i = 2^{a_i} r \text{ bits/s, where } a_i \in \{0, 1, 2, 3, ..., K\} \tag{2}$$

Our algorithm periodically schedules the interactive connections to be served. Let D denote the length of a *Transmission Time Interval* (TTI). Typically, the D value is 10 ms in WCDMA [5]. Let T_i be the time period between two consecutive transmissions for Q_i, and D_i be the transmission duration for every transmission for Q_i.

The SCAS algorithm consists of two phases: the assignment phase and the scheduling phase. The assignment phase selects a PDSCH for an interactive connection request. When the service of a new interactive connection is granted, or an interactive connection is completed by a PDSCH, the scheduling phase exercises to reschedule connections which share that PDSCH. The details of the two phases are given below.

The Assignment Phase: For a new incoming interactive connection request Q_n requesting data rate γ_n, the network finds a PDSCH with the OVSF code $C_{k,n}$ which transmission rate R_k satisfies the following condition.

$$R_k - \sum_{Q_i \in S} \gamma_i \geq \gamma_n \tag{3}$$

If such a PDSCH is available, then this channel is allocated to serve the request, and its corresponding S is updated as

$$S \leftarrow S \cup \{Q_n\}$$

Otherwise (i.e., no such PDSCH exists), the request Q_n is blocked.

The Scheduling Phase: This phase exercises when the request of a new interactive connection is granted, or an interactive connection is completed by a PDSCH. Suppose

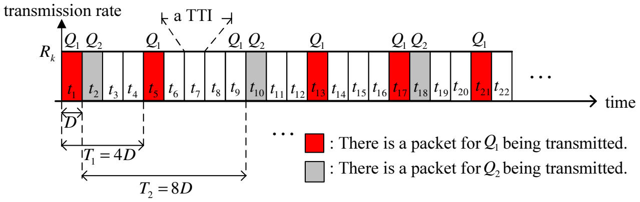


Fig. 4. An example for scheduled connections sharing the PDSCH with $C_{k,n}$

that the PDSCH is with the OVSF code $C_{k,n}$, and can provide transmission rate R_k. This phase reschedules all the interactive connections served by this PDSCH. First, the network determines the values of T_i and D_i for $Q_i \in S$. For all $Q_i \in S$, the transmission duration D_i is set to

$$D_i = mD \text{ where } m \in \text{positive integers} \tag{4}$$

We use the following equation to determine the time period T_i between two consecutive transmissions for Q_i.

$$T_i = D_i \left\lceil \frac{R_k}{\gamma_i} \right\rceil \tag{5}$$

The network will then periodically transmit the packets for a connection Q_i according to its D_i and T_i. The scheduling of the connections in S is in the increasing order of T_i. A connection with smaller T_i will be scheduled first. If tie-breaking occurs, the two connections with the same T_i can be transmitted in any order. When a connection Q_i is scheduled at time t, the next transmission duration for Q_i must be at least apart from t for T_i transmission durations (i.e., at $t + T_i$).

Figure 4 shows an example for scheduling connections sharing the PDSCH with $C_{k,n}$, where $D_i = D$. The Q_1 connection has $T_1 = 4D$, and is scheduled to transmit packets at the transmission durations t_1, t_5, t_9, ... The Q_2 connection has $T_2 = 8D$, and is scheduled to transmit packets at the transmission durations t_2, t_{10}, t_{18}, ...

For the background connections, the SCAS algorithm schedules them to be transmitted in two kinds of transmission durations: (i) the transmission durations that are not scheduled to serve any interactive connection, or (ii) the transmission durations that have been scheduled to serve an interactive connection Q_i, but is idle (i.e., no packets for Q_i to be transmitted). As shown in Figure 4, the transmission durations, e.g., t_3, t_4, and t_6, can be scheduled to serve background connections. Note that in this example, the transmission duration t_9 can be scheduled to serve a background connection when Q_1 has no packets to be delivered.

III. PROOF FOR THE CORRECTNESS OF THE SCAS ALGORITHM

As shown in Figure 5, the packet service connection for an interactive application contains one or several bursty periods, which is known as the ETSI packet data model [3]. The bursty period constitutes a bursty sequence of packets. Hence, in a bursty period, the buffer for an interactive connection always contains packets to be transmitted (i.e., the buffer is not empty during the bursty period). In this section, we formally prove that

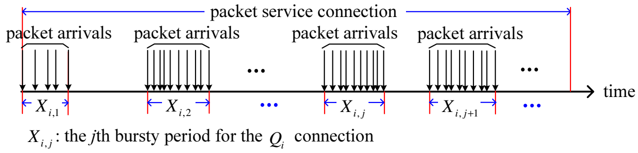


Fig. 5. Packet traffic characteristic for interactive connections

- the SCAS algorithm can guarantee that the interactive connections can be delivered with a transmission rate no less than the requested transmission rate, and
- each PDSCH can be fully utilized to serve the interactive connections or background connections.

Let the average transmission rate R_a be

$$R_a = \frac{\text{total number of bits}}{\text{total transmission time}} \tag{6}$$

We show, in Theorem 1, that by using the SCAS algorithm, in the bursty periods for a served interactive connection Q_i, the average transmission rate R_a is no less than the requested transmission rate γ_i.

Theorem 1: Suppose that by using (3), the SCAS algorithm assigns the PDSCH with the OVSF code $C_{k,n}$ (supporting the transmission rate R_k) to the interactive connection Q_i (that requests the transmission rate γ_i), and periodically transmits the packets for Q_i by using (4) and (5). Then, in any bursty period $X_{i,j}$ for the connection Q_i, $R_a \geq \gamma_i$.

Proof: Consider a bursty period $X_{i,j}$ for the Q_i connection. Assume that in this period, there are Y packet arrivals, and the kth packet arrival has size z_k bits. Then the total volume of data V bits that should be delivered in this period is

$$V = \sum_{k=1}^{Y} z_k$$

Since $D_i = mD$ (obtained from (4)) and this PDSCH provides the transmission rate R_k, the channel can deliver mDR_k bits in each transmission duration for Q_i. The number N of transmission durations required to deliver the packet arrivals in the bursty period can be obtained by

$$N = \left\lceil \frac{V}{mDR_k} \right\rceil \tag{7}$$

Note that the last transmission duration may not be fully utilized to transmit the packet data for Q_i. We calculate the number V^* of bits transmitted in the last transmission duration as follows:

$$V^* = V - (N-1)mDR_k \tag{8}$$

Then the time T^* spent for the data transmitted in the last transmission duration is

$$\begin{aligned} T^* &= V^*/R_k \\ &= \frac{V - (N-1)mDR_k}{R_k} \end{aligned} \tag{9}$$

We obtain the total transmission time T_t required to transmit packets in this bursty period as follows.

$$T_t = (N-1)T_i + T^* \tag{10}$$

Applying (5) and (9) into (10), (10) is rewritten as

$$T_t = \frac{V}{R_k} + (N-1)mDR_k\left(\frac{1}{\gamma_i} - \frac{1}{R_k}\right) \tag{11}$$

Since $R_a = V/T_t$ or $1/R_a = T_t/V$ (see (6)), from (11), we have

$$\frac{1}{R_a} = \frac{1}{R_k} + \left[\frac{(N-1)mDR_k}{V}\right]\left(\frac{1}{\gamma_i} - \frac{1}{R_k}\right) \tag{12}$$

Subtract $1/\gamma_i$ in both sides of (12), and then we have

$$\frac{1}{R_a} - \frac{1}{\gamma_i} = \left(\frac{1}{R_k} - \frac{1}{\gamma_i}\right)\left[1 - \frac{(N-1)mDR_k}{V}\right] \tag{13}$$

From (3), we know that $R_k \geq \gamma_i$ or $1/R_k - 1/\gamma_i \leq 0$, and from (7), it is obvious that $V \geq (N-1)mDR_k$ or $1-(N-1)mDR_k/V \geq 0$. Therefore, by using (13), the following inequality holds.

$$\frac{1}{R_a} - \frac{1}{\gamma_i} \leq 0 \text{ or } R_a \geq \gamma_i$$

which completes the proof. ∎

From Theorem 1, we know that in a bursty period, the packets for an interactive connection can be delivered with an average rate R_a no less than the requested transmission rate γ_i. In the following, we will further prove that the SCAS algorithm would not schedule a transmission duration to serve more than one interactive connection (i.e., no collision occurs), and the PDSCH can be fully utilized to serve interactive connections.

Theorem 2: With the SCAS algorithm, any two connections Q_i and Q_j (requesting the requested transmission rates γ_i and γ_j, respectively) in the set S served by the PDSCH with the OVSF code $C_{k,n}$ would not be scheduled the same transmission duration.

Proof: From (5), we have

$$T_i = D_i\left[\frac{R_k}{\gamma_i}\right] \text{ and } T_j = D_j\left[\frac{R_k}{\gamma_j}\right] \tag{14}$$

Without loss of generality, we assume that the transmission durations allocated to Q_i and Q_j are $t_{x+o(T_i/D)}$ and $t_{y+p(T_j/D)}$ where t_x and t_y are the first transmission durations scheduled 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 transmission duration for Q_i and Q_j, we have $x < T_i/D$ and $y < T_j/D$, respectively. We complete the proof by considering the relationship between the requested transmission rates γ_i and γ_j in the following two cases:

Case I. $\gamma_i = \gamma_j$, i.e., the two connections request the same transmission rate. Since $D_i = D_j$ (obtained from (4)) and $\gamma_i = \gamma_j$, from (14), we have

$$T_i = T_j$$

The all indexes of the transmission durations (scheduled to serve Q_i and Q_j) have the following relationship. For any o and $p \in$ positive integers,

$$\begin{aligned}&[x+o(T_i/D)] - [y+p(T_j/D)]\\&= (x-y) + [o(T_i/D) - p(T_j/D)]\end{aligned} \tag{15}$$

As described in the scheduling phase of SCAS, 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,

$$x \neq y \tag{16}$$

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

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

or

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

Therefore in this case, Q_i and Q_j are not scheduled to be served in the same transmission durations.

Case II. $\gamma_i \neq \gamma_j$, i.e., the two connections request different transmission rates. Without loss of generality, we suppose that $\gamma_i < \gamma_j$. Then from (14), the following inequality holds:

$$T_i > T_j \tag{17}$$

From (1) and (2), we know that R_k, γ_i, and γ_j are powers of 2. Then, by using (3), we have γ_i and $\gamma_j < R_k$. According to (14), we can rewrite T_i and T_j as follows.

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

or

$$T_i = 2^bT_j \text{ where } b = a_i - a_j \tag{18}$$

Consider all indexes of the transmission durations (scheduled 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, respectively. As described in the scheduling phase in the SCAS algorithm, since $\gamma_i < \gamma_j$, the Q_j connection is scheduled before the Q_i connection (i.e., the transmission durations scheduled to serve Q_j would be reserved before the transmission durations scheduled to serve Q_i are reserved). Hence we have for all $p \in$ integers,

$$x \neq y + p(T_j/D) \text{ or } x - [y + p(T_j/D)] \neq 0 \tag{19}$$

The all indexes of the transmission durations scheduled to serve Q_i and Q_j have the following relationship. By using (18), i.e., $T_i = 2^bT_j$, for all o and $p \in$ positive integers, we have

$$\begin{aligned}&[x+o(T_i/D)] - [y+p(T_j/D)]\\&= x - [y + (p - o2^b)(T_j/D)]\end{aligned} \tag{20}$$

By using (19) and from (20), we obtain the following inequality.

$$[x+o(T_i/D)] - [y+p(T_j/D)] \neq 0 \tag{21}$$

Hence, the statement "Q_i and Q_j are not scheduled the same time slots" holds in this case.

The above two cases complete the proof. ▮

Define the bandwidth utilization U of a PDSCH shared by a set of L interactive connections as

$$U = \sum_{i=1}^{L} \frac{D_i}{T_i}$$

In the following lemma, we will prove that the SCAS algorithm can fully utilize a PDSCH to serve a set of interactive connections.

Lemma 1: Consider a PDSCH with the OVSF code $C_{k,n}$, which supports the transmission rate R_k. The SCAS algorithm can schedule the PDSCH to serve interactive connections, such that the bandwidth utilization $U \leq 100\%$.

Proof: Suppose that there are L interactive connections served by the PDSCH. From (3), we know that the total requested data rate for the L interactive connections is no larger than R_k, that is,

$$\sum_{i=1}^{L} \gamma_i \leq R_k \tag{22}$$

In the scheduling phase of the SCAS algorithm, the time period T_i for the Q_i connection is obtained by $T_i = D_i \left[R_k/\gamma_i\right]$ that can be rewritten as

$$\gamma_i = \left[\frac{D_i}{T_i}\right] R_k \tag{23}$$

Apply (23) into (22), and we have

$$R_k \sum_{i=1}^{L} \left[\frac{D_i}{T_i}\right] \leq R_k \text{ or } \sum_{i=1}^{L} \left[\frac{D_i}{T_i}\right] \leq 100\% \tag{24}$$

Hence we have $U = \sum_{i=1}^{L} \left[D_i/T_i\right] \leq 100\%$. Note that the situation (where $U = 100\%$) occurs when $\sum_{i=1}^{L} \gamma_i = R_k$. ▮

IV. CONCLUDING REMARKS

The UMTS provides high and variable transmission rate services to support wireless transmission for mobile users. In UMTS, four traffic classes for connections are identified, which are conversational, streaming, interactive, and background. To efficiently utilize radio resource, the shared radio channel PDSCH is defined to serve the interactive and background connections. This paper proposed an efficient time-division-based algorithm named "Shared Channel Assignment and Scheduling" (SCAS) to schedule a PDSCH to serve different interactive connections. We formally prove the correctness for the SCAS algorithm. Our study contributed that

- the proposed SCAS algorithm is simple and with less complexity, which is practical to be implemented in the real UMTS system,
- our algorithm can guarantee that the interactive connections can be served with a transmission rate no less than the requested transmission rate, and
- with SCAS, the PDSCH can be fully utilized to serve the interactive connections.

REFERENCES

[1] 3GPP. 3rd Generation Partnership Project; Radio Interface Protocol Architecture. Technical Report Technical Specification 3G TS 25.301 version 3.4.0 (2000-03), 2000.

[2] 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.

[3] ETSI/TC. UMTS Terrestrial Radio Access (UTRA); Concept Evaluation, Version 3.0.0. Technical Report Technical Report UMTS 30.06, ETSI, December 1997.

[4] Fossa, C.E., Jr. and Davis, N.J., IV. A Dynamic Code Assignment Algorithm for Quality of Service in 3G Wireless Networks. *Proceedings of IEEE WCNC2002*, 1:1–6, March 2002.

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

[6] Kam, A.-C., Minn, T., and Siu, K.-Y. Supporting Rate Guarantee and Fair Access for Bursty Data Traffic in W-CDMA. *IEEE Journal on Selected Areas in Communications*, 19(11):2121–2130, November 2001.

[7] Xu, L., Shen, X., and Mark, J.W. Dynamic Bandwidth Allocation with Fair Scheduling for WCDMA System. *IEEE Wireless Communications*, 9(2):26–32, April 2002.