Overflow Control for UMTS High-Speed Downlink Packet Access

Phone Lin, *Member, IEEE*, Yi-Bing Lin, *Fellow, IEEE*, and Imrich Chlamtac, *Fellow, IEEE*

Abstract—**This paper proposes overflow control schemes to support high-speed downlink packet access (HSDPA) mechanism in the universal mobile telecommunication system (UMTS). To access the UMTS services, a user equipment (UE) communicates with all cells (base stations) in an active set. However, multiple links between the UE and the cells in the active set may reduce the transmission speed due to interference. Third-Generation Partnership Project specification TR 25.950 proposes HSDPA. In this mechanism, the UE only selects one cell (referred to as the serving cell) in the active set for high-speed downlink transmission. In HSDPA, the radio network controller sends the packet frames to the cells in the active set. For the serving cell, the packet frames are forwarded to the UE. On the other hand, every nonserving cell in the active set queues the packet frames in a buffer. If the link quality between the serving cell and the UE degrades below some threshold, the UE selects the best cell in the active set as the new serving cell. Since the nonserving cells do not send packet frames to the UE, their buffers may overflow. In this paper, we propose schemes to address the buffer overflow issue. Our schemes guarantee that when the buffer of a nonserving cell is full, the previously received packet frames in the buffer can be safely dropped, and after the UE has switched wireless link to the new serving cell, no packet frames are lost.**

Index Terms—**Frame synchronization, high-speed downlink packet access (HSDPA), overflow control, universal mobile telecommunication system (UMTS).**

I. INTRODUCTION

UNIVERSAL mobile telecommunications system (UMTS) [9], [6] is a third-generation system proposed by the Third-Generation Partnership Project (3GPP), which is designed to support higher data transmission rate for mobile users, and to provide streaming, interactive, and background services with better quality of service [14]. UMTS terrestrial radio access network (UTRAN) consists of Node Bs and radio network controllers (RNCs). To access the UMTS services, a user equipment (UE) communicates with cells (Node Bs) in an active set through the air interface Uu which is based on wideband-CDMA radio access technology [1]. If the quality of the wireless link between the UE and a cell is above some threshold, then this cell is included in the active set. When the quality of the wireless link of a cell in the active set is below the threshold, then the cell is removed from the active set. Typically, there are three cells in the active set. In standard UTRAN [5], multiple paths exist between the UE and all Node Bs in the active set. This mechanism does not support high-speed downlink transmission because multiple links for a UE may increase the overall interference within an UTRAN, and thus the data transmission rate decreases.

3GPP TR 25.950 [4] proposes a mechanism to support high-speed downlink packet access (HSDPA) [2]–[4], where a UE only communicates with one cell (called the serving cell) in the active set. This "serving cell" is selected by the fast cell selection mechanism [3] based on the common pilot-channel received signal code power measurements of the cells in the active set. Two physical channels, high-speed physical downlink shared channel (HS-PDSCH) and dedicated physical control channel (DPCCH) are used for downlink packet frame transmission and uplink/downlink signaling, respectively. While multiple cells may be members of the active set, only one of them transmits at any time in the HSDPA mode. Therefore, the interference within a cell is potentially decreased, and the system capacity is increased. Several feasibility studies have been contributed to HSDPA [7], [8], [10]–[12], [15], [17]. The topics addressed in these studies include adaptive modulation and coding, hybrid automatic repeat request (ARQ), packet scheduler, and fast cell selection. In this paper, we will focus on the buffer overflow control issue for HSDPA.

Fig. 1 illustrates the network architecture of HSDPA with the active set $\{\mathrm{Cell}_1, \mathrm{Cell}_2, \mathrm{Cell}_3\}$ and the serving cell Cell_1. In HSDPA, the RNC sends the packet frames to all cells in the active set. For the serving cell, the packet frames are forwarded to the UE. For each nonserving cell, the packet frames are queued in a buffer. The stop-and-wait hybrid ARQ (SAW-Hybrid ARQ) [16] algorithm is exercised between the UE and the serving cell for flow control of the wireless link. If the link quality for high-speed downlink transmission degrades below some threshold, the UE selects the best cell in the active set as the serving cell. Then the next packet frames are transmitted from the new serving cell to the UE. In HSDPA, the buffer in a

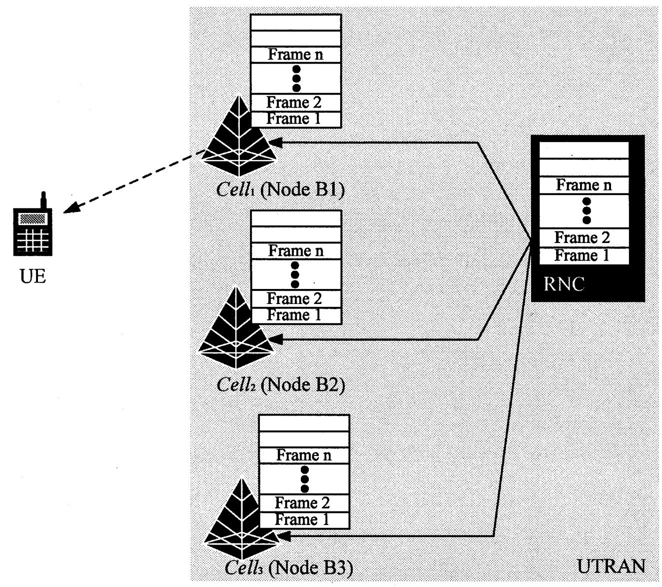
Manuscript received February 21, 2002; revised December 16, 2002; accepted January 15, 2003. The editor coordinating the review of this paper and approving it for publication is W. Lu. The work of P. Lin was supported in part by the National Science Council (NSC), R.O.C., under Contract NSC92–2213-E-002–094, in part by FarEastone, and in part by Computer and Communications Research Laboratories/Industrial Technology Research Institute (CCL/ITRI), and in part by the Institute for Information Industry. The work of Y.-B. Lin was supported in part by the NSC Program for Promoting Academic Excellence of Universities, in part by the Chair Professorship of Providence University, in part by the Institute of Information Science (IIS)/Academic Sinica, in part by FarEastone, in part by CCL/ITRI, and in part by the Lee and MTI Center for Networking Research/National Chiao Tung University (NCTU), R.O.C.

P. Lin is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: plin@csie.ntu.edu.tw).

Y.-B. Lin is with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan, R.O.C. (e-mail: liny@csie.nctu.edu.tw).

I. Chlamtac is with the Center for Advanced Telecommunications Systems and Services, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: chlamtac@utdallas.edu).

Digital Object Identifier 10.1109/TWC.2003.821152



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

Fig. 1. Network architecture of HSDPA.

nonserving cell may be full, and a scheme is required to avoid buffer overflow at that nonserving cell. Furthermore, when the UE switches to a new serving cell for downlink packet access, the new serving cell should be informed the status of the buffer (i.e., the number of packet frames received by the UE) in the old serving cell. This action is referred to as *frame synchronization* [4]. Since the nonserving cells do not send packet frames to the UE, their buffers may overflow. The buffer overflow issue is not addressed in 3GPP TR 25.950. In this paper, we propose four overflow control schemes: basic overflow control (BOFC); network-based overflow control (NOFC); combined BOFC and NOFC (COFC); and COFC with counter rest (COFCR); and prove the correctness of these schemes. In BOFC, the information needed for frame synchronization is carried by the uplink DPCCH. When the size of frame synchronization information exceeds the capacity of an uplink DPCCH, this information must be carried through multiple uplink DPCCH transmissions. To avoid multiple HSDPA transmissions, we propose NOFC to guarantee one uplink DPCCH transmission for frame synchronization through message exchange between the old and new serving cells. Then we propose COFC to take advantage of both BOFC and NOFC. In this scheme, the old serving cell decides whether to transmit frame synchronization information through the network or uplink DPCCH. COFCR is enhanced from COFC. In COFCR, the counters for the frame synchronization are reset after each frame synchronization, and thus, the size of frame synchronization information is reduced. We have constructed analytical and simulation models to investigate the delays of the frame synchronization for the four schemes. The study indicated that COFCR outperforms BOFC, NOFC, and COFC in terms of the cell switching delay for frame synchronization. For the details, readers are referred to [13].

This paper details the procedures required for overflow control and frame synchronization for HSDPA. We formally prove the correctness of these schemes. Our schemes guarantee that when the buffer of a nonserving cell is full, the previously received packets in the buffer can be safely dropped, and when the UE switches wireless link to the new serving cell, no packet frames are lost.

II. BOFC SCHEME

This section presents the BOFC scheme, and proves that the scheme is correct. We first describe the flow control algorithms exercised during downlink transmission. These algorithms are executed by the RNC (i.e., OFC1), the serving cell (i.e., OFC2 and OFC3), and the nonserving cell (i.e., OFC4). Then we describe the basic frame synchronization (BFS) algorithm that involves the UE, the old serving cell, and the new serving cell.

A. Overflow Control (OFC) Algorithms

To exercise HSDPA, every cell Cell_i in the active set maintains a buffer of size $N_{i,\max}$ for each downlink transmission. Let K_i be the number of packet frames currently stored in the buffer of Cell_i in the active set. The UE maintains a counter CR_{UE} to indicate the number of received packet frames. When the UE switches the wireless link from the old serving cell to the new serving cell, the CR_{UE} value is sent to the new serving cell for frame synchronization. At Cell_i, two counters are maintained. The CR_i counter counts the number of packet frames received from the RNC. The CS_i counter counts the number of packet frames that have been processed by Cell_i. If Cell_i is a serving cell, then CS_i is the number of packet frames that have been received by the UE. If Cell_i is a nonserving cell, then CS_i is the number of packet frames deleted from the buffer. A counter CS_{RNC} is maintained by the RNC to record the number of packet frames that have been received by the serving cell. To initiate HSDPA, the UE selects the serving cell based on the fast cell selection criteria as described in the previous section, and CR_{UE}, CS_{RNC}, CR_i, CS_i, and K_i values are initially set to zero. In UTRAN, the ATM AAL2 is adopted for links between the cells and the RNC [5]. These links are considered reliable, and we assume that no packet frame is lost during transmission. If the rare events of packet frame loss do occur, these lost packet frames can be recovered by higher level protocols, which is out of the scope of this paper. A window-based flow control algorithm OFC1 with window size w is used for downlink transmission from the RNC to the serving cell. After the RNC has sent all packet frames of the current window, it must wait for an acknowledgement (ACK) message from the serving cell before it can proceed to send the packet frames in the next window.

Algorithm OFC1 (exercised by the RNC)

The RNC sends a packet frame to every cell Cell_i in the active set, and increments CS_{RNC} by one. Two cases are considered for flow control.

1) *Case OFC1.1.* If $CS_{\text{RNC}} \bmod w \neq 0$, the packet frames of the current window are not transmitted completely, and the RNC continues to send the next packet frame to Cell_i.
2) *Case OFC1.2.* If $CS_{\text{RNC}} \bmod w = 0$, the RNC has transmitted all packet frames in the current window. The RNC suspends the packet frame transmission until an ACK message is received from the serving cell.

Upon receipt of a packet frame, the serving cell Cell_s executes Algorithm OFC2 for flow control between the RNC and Cell_s. Two parameters, F_s and WI_s, are used in this algorithm. The *overflow flag* F_s indicates if the buffer of Cell_s overflows (where $F_s = 1$ indicates buffer overflow). Parameter WI_s indicates the number of packet frames received by Cell_s for the current window. Without loss of generality, we assume that the buffer size of the serving cell Cell_s is $N_{s,\max} > w$.

Algorithm OFC2 (exercised by the serving cell)

This algorithm performs flow control between the serving cell and the RNC.

1) *Step OFC2.1.* When a packet frame arrives, Cell_s increments the number of packet frames received from the RNC, i.e.,

$$CR_s \leftarrow CR_s + 1. \tag{1}$$

The number of packet frames in the buffer of Cell_s is set to

$$K_s \leftarrow CR_s - CS_s \tag{2}$$

and the number of received packet frames within a window is set to

$$WI_s \leftarrow CR_s \bmod w.$$

2) *Step OFC2.2.* Cell_s checks WI_s and K_s values to determine if it will receive the next packet frames from the RNC. There are three cases.

Case OFC2.2.1. If Cell_s has not received all packet frames in the current window (i.e., $WI_s \neq 0$), it continues to receive the next packet frame.

Case OFC2.2.2. All packet frames in the current window have been received (i.e., $WI_s = 0$), and there is enough space to accommodate the packet frames of the next window (i.e., $K_s \leq N_{s,\max} - w$). In this case, it is safe for the RNC to transmit the packet frames in the next window. Cell_s replies the RNC an ACK message, and the RNC will be triggered to send the packet frames in the next window.

Case OFC2.2.3. All packet frames in the current window have been received (i.e., $WI_s = 0$), and Cell_s does not have enough space to accommodate the next w packet frames (i.e., $K_s > N_{s,\max} - w$). Cell_s sets the overflow flag $F_s \leftarrow 1$, and no ACK message is sent to the RNC.

Algorithm OFC3 is exercised between the UE and Cell_s. Since the wireless link is not reliable, SAW-Hybrid ARQ [16] is used in OFC3 for flow control. In SAW-Hybrid ARQ, Cell_s sends a packet frame to the UE through the HS-PDSCH channel in the air interface. The UE replies an ACK or a negative acknowledgement (NACK) to Cell_s through uplink DPCCH, which indicates whether the packet frame is correctly received. Details of Algorithm OFC3 are given below.

Algorithm OFC3 (between the serving cell Cell_s and the UE)

When the UE receives a packet frame, it replies the status of transmission to Cell_s. There are two cases.

1) *Case OFC3.1.* If the UE receives an incorrect packet frame from Cell_s, it replies a NACK message to Cell_s. Cell_s retransmits the packet frame to the UE.

2) *Case OFC3.2.* If the UE receives a correct packet frame from Cell_s, it sets

$$CR_{\mathrm{UE}} \leftarrow CR_{\mathrm{UE}} + 1 \tag{3}$$

and replies Cell_s an ACK message. Then Cell_s deletes the last transmitted packet frame from the buffer, and the following counters are updated:

$$CS_s \leftarrow CS_s + 1 \tag{4}$$
$$K_s \leftarrow CR_s - CS_s. \tag{5}$$

Cell_s checks F_s and K_s to determine if an ACK message should be sent to the RNC. One of the following three cases occurs.

Case OFC3.2.1. After all packet frames of the previous window have been received by Cell_s, no ACK message has been sent to the RNC (i.e., $F_s = 1$), and after the counters CS_s and K_s have been updated in **(4) and (5)**, Cell_s is allowed to receive the packet frames in the next window (i.e., $K_s \leq N_{s,\max} - w$). Cell_s sends an ACK message to the RNC, and sets $F_s \leftarrow 0$. When the RNC receives the ACK message, it resumes sending the packet frames of the next window (see Algorithm OFC1).

Case OFC3.2.2. Like Case OFC3.2.1, $F_s = 1$, but Cell_s is not allowed to receive the packet frames in the next window (i.e., $K_s > N_{s,\max} - w$). In this case, Cell_s need not take any action.

Case OFC3.2.3. An ACK message has been sent to the RNC when all packet frames of the previous window were received by Cell_s (i.e., $F_s = 0$). No action is taken by Cell_s.

Cell_s continues to transmit the next packet frame to the UE.

From (3) and (4) in Case OFC3.2, it is clear that the following relationship holds:

$$CR_{\mathrm{UE}} = CS_s. \tag{6}$$

From Cases OFC2.2.2 and OFC3.2.1, an ACK message is sent to the RNC when $K_s \leq N_{s,\max} - w$.

For a nonserving cell Cell_i (where $i \neq s$), when a packet frame is received from the RNC, Algorithm OFC4 is performed to avoid buffer overflow. The buffer size of Cell_i is $N_{i,\max}$. In Section II-C, we show that the relationship

$$N_{s,\max} \leq N_{i,\max} - w \tag{7}$$

must hold, or packet frames may be lost at frame synchronization. Without loss of generality, we set that

$$N_{s,\max} = N_{i,\max} - w. \tag{8}$$

Algorithm OFC4 (exercised by a nonserving cell Cell_i)

1) *Step OFC4.1.* When Cell_i receives a packet frame from the RNC, it sets $K_i \leftarrow CR_i - CS_i$.
2) *Step OFC4.2.* If $K_i = N_{i,\max}$ (i.e., the buffer is full), Cell_i deletes a packet frame at the head of the buffer, and increments CS_i by one. This step guarantees that the K_i value is no larger than $N_{i,\max}$.
3) *Step OFC4.3.* Cell_i adds the received packet frame at the tail of the buffer, and increments CR_i by one.

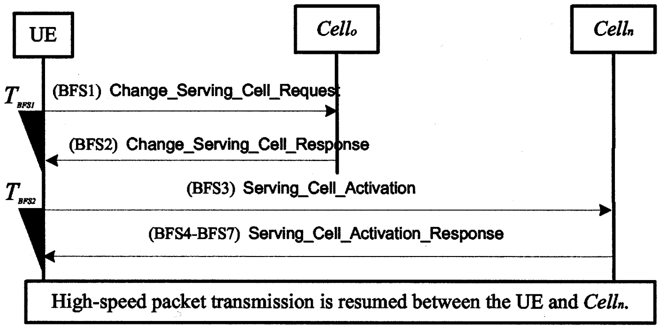


Fig. 2. Message flow for Algorithm BFS.

The above algorithm (specifically, Step OFC4.2) guarantees that

$$0 \leq K_i \leq N_{i,\max}. \tag{9}$$

B. BFS Algorithm

If the link quality between Cell_s and the UE degrades below some threshold, the UE selects another cell in the active set as the new serving cell based on the fast cell selection criteria described in the previous section. Let the old serving cell and the new serving cell have cell identities o and n, respectively. To switch the high-speed downlink packet frame transmission link from the old serving cell Cell_o to the new serving cell Cell_n, Algorithm BFS is executed among the UE, Cell_o and Cell_n for frame synchronization. Then Cell_n becomes the serving cell by executing OFC2 and OFC3. In BFS, $N_{\text{sync}} = CR_{\text{UE}} - CS_n$ denotes the number of packet frames that have already received by the UE but have not been removed from Cell_n. These packet frames should be deleted by Cell_n in the frame synchronization algorithm. Fig. 2 illustrates the message flow for Algorithm BFS, and the details are described as follows.

Algorithm BFS (The UE switches the wireless link from the old serving cell Cell_o to the new serving cell Cell_n)

1) *Step BFS1.* When the UE detects that the quality of the wireless link to Cell_o degrades below a threshold, it selects another cell Cell_n in the active set as the new serving cell, and sends a Change_Serving_Cell_Request message to Cell_o through uplink DPCCH. The UE starts the T_{BFS1} timer, and expects to receive a Change_Serving_Cell_Response message from Cell_o before T_{BFS1} expires.
2) *Step BFS2.* When Cell_o receives the Change_Serving_Cell_Request message, it stops high-speed downlink packet frame transmission, and replies the Change_Serving_Cell_Response message to the UE. Cell_o sets $N_{o,\max} \leftarrow N_{o,\max} + w$ (so that (8) holds). At this point, Cell_o becomes a nonserving cell, which executes Algorithm OFC4 to process the next packet frames received from the RNC.
3) *Step BFS3.* Upon receipt of the Change_Serving_Cell_Response message, the UE stops the T_{BFS1} timer. Then through uplink DPCCH, the UE sends a Serving_Cell_Activation message to Cell_n (where CR_{UE} and the cell identity n are specified in this message). The UE starts the T_{BFS2} timer, and expects to receive Serving_Cell_Activation_Response from Cell_n before T_{BFS2} expires.
4) *Step BFS4.* Upon receipt of the Serving_Cell_Activation message, Cell_n becomes the serving cell for the UE by executing Steps BFS5-BFS7 (these steps are for frame synchronization).
5) *Step BFS5.* Cell_n sets $K_n \leftarrow CR_n - CS_n$ and $N_{\text{sync}} \leftarrow CR_{\text{UE}} - CS_n$.
6) *Step BFS6.* If $N_{\text{sync}} \leq K_n$, Cell_n deletes N_{sync} packet frames in the front of the buffer for the UE. Otherwise ($N_{\text{sync}} > K_n$), Cell_n deletes K_n packet frames in the buffer, and deletes the next $N_{\text{sync}} - K_n$ packet frames received from the RNC. Cell_n sets $CS_n \leftarrow CS_n + N_{\text{sync}}$, and sets $N_{n,\max} \leftarrow N_{n,\max} - w$ (so that (8) holds). Then it executes Algorithm OFC2 to process the next packet frames received from the RNC.
7) *Step BFS7.* Through downlink DPCCH, Cell_n assigns an HS-PDSCH to the UE by sending the Serving_Cell_Activation_Response message. After the UE has received this message, it stops the T_{BFS2} timer. Cell_n starts to transmit packet frames to the UE by executing Algorithm OFC3.

At Step BFS3, the Serving_Cell_Activation message is carried through uplink DPCCH. The information that can be carried by uplink DPCCH is limited. When the CR_{UE} value exceeds the size that can be transmitted through an uplink DPCCH, more than one uplink DPCCH transmissions are required to deliver the CR_{UE} value. In Section III, we will propose the NOFC scheme to guarantee single uplink DPCCH transmission.

C. Correctness Proof for Frame Synchronization in BOFC

In BOFC, if $N_{\text{sync}} < 0$ when Step BFS5 is executed, then it implies that a nonserving cell Cell_i (which becomes the serving cell later) has dropped packet frames in its buffer, and these dropped packet frames have not been received by the UE. If so, the dropped packet frames are lost when the UE switches the wireless link to Cell_i. In this section, we prove that in BOFC, $N_{\text{sync}} \geq 0$ (i.e., no packet frame is lost). Let Cell_o and Cell_n be the old and new serving cells in Algorithm BOFC. From Step BFS5, the following relationship holds:

$$N_{\text{sync}} = CR_{\text{UE}} - CS_n. \tag{10}$$

From (6), (10) is written as

$$N_{\text{sync}} = CS_o - CS_n. \tag{11}$$

Thus, to show that no packet frames are lost (i.e., $N_{\text{sync}} \geq 0$), it suffices to prove that

$$CS_o - CS_n \geq 0. \tag{12}$$

Depending on whether Cell_n has dropped packet frames before frame synchronization, we show that the inequality (12) holds in two cases (see *Lemma 1* and *Lemma 2*).

Lemma 1: Suppose that when Cell_o receives the Change_Serving_Cell_Request message in Algorithm BFS (i.e., UE starts to switch the wireless link from Cell_o to Cell_n), Cell_o has delivered CS_o packet frames to the UE, and no packet frames have been dropped by Cell_n. Then Algorithm BFS guarantees that (12) holds; that is, $N_{\text{sync}} = CS_o - CS_n \geq 0$.

Proof: Suppose that when Cell$_o$ receives the Change_Serving_Cell_Request message in Algorithm BFS, Cell$_n$ has not dropped any packet frame; that is, $CS_n = 0$. Since the UE may have received packet frames from Cell$_o$ before it switches to Cell$_n$, it is clear that $CS_o \geq 0$, and $CS_o - CS_n \geq 0$ (i.e., (12) holds). ∎

If Cell$_n$ has dropped packet frames before frame synchronization, we show in *Lemma 2* that the inequality (12) holds if Cell$_o$ always maintains a free buffer space of size w.

Lemma 2: In Algorithm BFS, suppose that Cell$_n$ has dropped packet frames before Cell$_o$ receives the Change_Serving_Cell_Request message. Then $N_{\text{sync}} = CS_o - CS_n \geq 0$ if $N_{o,\max} \leq N_{n,\max} - w$ (i.e., (7) holds).

Proof: Suppose that when Cell$_o$ receives the Change_Serving_Cell_Request message in Algorithm BFS, the RNC has transmitted CS_{RNC} packet frames to the cells in the active set, and CR_o and CR_n packet frames have been received by Cell$_o$ and Cell$_n$, respectively. Let K_o be the number of packet frames queued in the buffer of Cell$_o$.

Since Cell$_n$ has dropped packet frames before it becomes the serving cell (i.e., $CS_n > 0$), from Step OFC4.2, we know that the number of packet frames received by Cell$_n$ is larger than $N_{n,\max}$, and Cell$_n$ has dropped $CR_n - N_{n,\max}$ packet frames. That is

$$CS_n = CR_n - N_{n,\max}. \tag{13}$$

From (2) and (5), we have

$$CS_o = CR_o - K_o. \tag{14}$$

Substrate (13) from (14), we obtain

$$CS_o - CS_n = (CR_o - K_o) - (CR_n - N_{n,\max})$$

or

$$CS_o - CS_n = CR_o - CR_n + N_{n,\max} - K_o. \tag{15}$$

From (7), we have $0 \leq K_o \leq N_{o,\max} \leq N_{n,\max} - w$, (15) is rewritten as

$$CS_o - CS_n \geq CR_o - CR_n + N_{n,\max} - (N_{n,\max} - w)$$

or

$$CS_o - CS_n \geq (CR_o + w) - CR_n. \tag{16}$$

Since the number of the packet frames received by Cell$_n$ is no larger than the number of packet frames sent by the RNC, we have

$$CR_n \leq CS_{\text{RNC}}. \tag{17}$$

Because the flow control with window size w is executed between Cell$_s$ and the RNC (see Steps OFC1.1 and BOF2.1), the CR_o value is bounded by

$$CS_{\text{RNC}} - w \leq CR_o \quad \text{or} \quad CS_{\text{RNC}} \leq CR_o + w. \tag{18}$$

From (17) and (18), we have

$$CR_o + w \geq CS_{\text{RNC}} \geq CR_n$$

which implies

$$(CR_o + w) - CR_n \geq 0. \tag{19}$$

Applying (19) into the right-hand side of (16), we obtain the following inequality:

$$CS_o - CS_n \geq 0.$$

∎

Theorem 1: Suppose that when Cell$_o$ receives the Change_Serving_Cell_Request message in Algorithm BFS, Cell$_o$ has delivered CS_o packet frames to the UE, and CS_n packet frames have been dropped by Cell$_n$. Then (12) (i.e., $N_{\text{sync}} = CS_o - CS_n \geq 0$) always holds.

Proof: If Cell$_n$ has not dropped any packet frames before frame synchronization, then *Lemma 1* shows that (12) holds.

If Cell$_n$ has dropped any packet frames before frame synchronization, *Lemma 2* shows that (12) holds if (7) holds. Since Steps OFC2.2 and 3.2 guarantee that (7) holds, $CS_o - CS_n \geq 0$ always holds. ∎

In *Lemma 2*, we show that if a nonserving cell maintains w more buffer slots than the serving cell, then no packet frames will be lost during frame synchronization. One may question if we really need to maintain so many extra buffer slots in the nonserving cell, i.e., can we set

$$N_{o,\max} = N_{n,\max} - w + i, \quad \text{where} \quad i \geq 1. \tag{20}$$

In the following lemma, we show that if (20) holds, then packet frames may be lost.

Lemma 3: If $N_{o,\max} > N_{n,\max} - w$ during downlink packet frame transmission, then (12) may not hold; i.e., $CS_o - CS_n < 0$.

Proof: Suppose that Cell$_o$ maintains a buffer of size $N_{o,\max} = N_{n,\max} - w + i$ where $1 \leq i$. Consider the scenario that the transmission speed from the RNC to Cell$_n$ is much faster than that to Cell$_o$. In this case, all packet frames in the current window sent by the RNC have been received by Cell$_n$, but none of them was received by Cell$_o$, i.e.,

$$CR_n = CS_{\text{RNC}} = CR_o + w. \tag{21}$$

Furthermore, if the transmission speed from the RNC to Cell$_o$ is much faster than that from Cell$_o$ to the UE, then it is possible that

$$K_o = N_{o,\max} = N_{n,\max} - w + i. \tag{22}$$

By applying (22) and (21) into (15), we obtain the following equations:

$$\begin{aligned} CS_o - CS_n &= CR_o - CR_n + N_{n,\max} - K_o \\ &= CR_o - (CR_o + w) + N_{n,\max} \\ &\quad - (N_{n,\max} - w + i) \\ &= -i < 0 \end{aligned}$$

and i packet frames have been lost after frame synchronization. ∎

III. NOFC Scheme

In this section, we propose the NOFC scheme, and then show how NOFC can combine with BOFC to take advantages of both

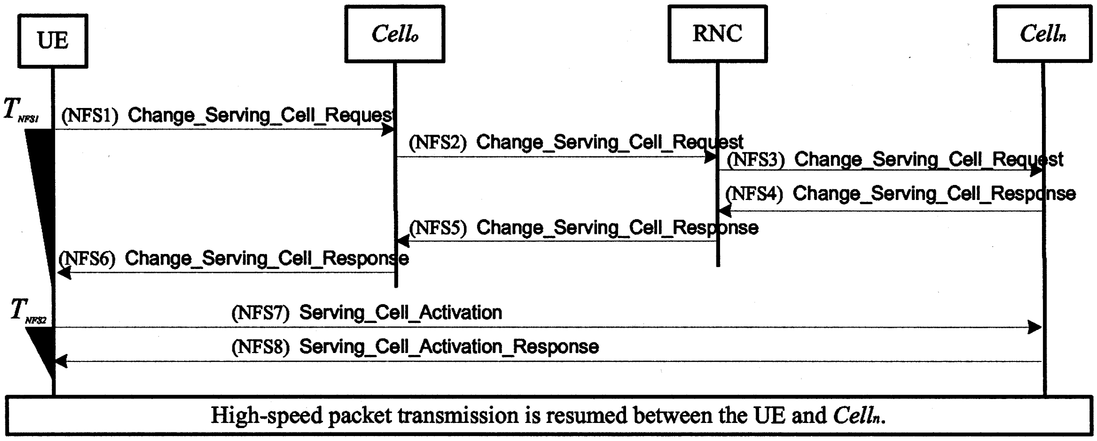


Fig. 3. Message flow for Algorithm NFS.

schemes. This combined OFC (COFC) scheme can be further improved by *counter reset*, where the counter values are offset to smaller values so that the messages required for frame synchronization can be reduced.

In the NOFC scheme, the information (i.e., the CS_o value) needed for frame synchronization is delivered from the old serving cell Cell_o to the new serving cell Cell_n through the RNC. Like BOFC, NOFC utilizes Algorithms OFC1–4 for packet frame transmissions from the RNC to the cells, and from the serving cell to the UE.

A. Network-Based Frame Synchronization (NFS) Algorithm

Fig. 3 illustrates the message flow for the network-based frame synchronization algorithm NFS, and the details are described as follows.

Algorithm NFS (The UE switches the wireless link from Cell_o to Cell_n)

1) *Step NFS1.* Similar to Step BFS1, Change_Serving_Cell_Request message is sent from the UE to Cell_o. The cell identity n is specified in this message. The UE sets the T_{NFS1} timer, and expects to receive a Change_Serving_Cell_Response message from Cell_o before this timer expires.
2) *Step NFS2.* When Cell_o receives the Change_Serving_Cell_Request message, it stops downlink packet frame transmission to UE, and sends a Change_Serving_Cell_Request message to the RNC, where Cell_o and cell identity n are specified in this message. At this point, Cell_o becomes a nonserving cell, which sets $N_{o,\max} \leftarrow N_{o,\max} + w$ and executes Algorithm OFC4 to process the next packet frames received from the RNC.
3) *Step NFS3.* Upon receipt of Change_Serving_Cell_request, the RNC forwards this message to Cell_n.
4) *Steps NFS4 and NFS5.* Cell_n computes $N_{\text{sync}} \leftarrow CS_o - CS_n$, where CS_o is specified in the Change_Serving_Cell_Request message. Cell_n deletes the first N_{sync} packet frames in the buffer, and replies Cell_o a Change_Serving_Cell_Response message through the RNC.
5) *Step NFS6.* Cell_o forward Change_Serving_Cell_Response to the UE.
6) *Step NFS7.* When the Change_Serving_Cell_Response message is received, the UE stops the T_{NFS1} timer. Then it sends a Serving_Cell_Activation message to Cell_n through uplink DPCCH. The UE starts timer T_{NFS2}, and before this timer expires, it expects to receive the Serving_Cell_Activation_Response message from Cell_n.
7) *Step NFS8.* Through downlink DPCCH channel, Cell_n assigns a HS-PDSCH to the UE by sending the Serving_Cell_Activation_Response message. Cell_n sets $N_{n,\max} \leftarrow N_{n,\max} - w$ and starts to transmit packet frames to the UE by executing Algorithm OFC3. After the UE receives this message, it stops T_{NFS2}.

As we mentioned before, the NOFC scheme is similar to the BOFC scheme, except that the information for frame synchronization is sent from Cell_o to Cell_n through the RNC. Therefore, correctness of NOFC is also proved by that in Section II-C.

B. COFC Scheme

From Figs. 2 and 3, it is clear that if the CR_{UE} value can be fitted in one DPCCH, then the number of messages exchanged in BFS is less than that in NFS. On the other hand, if multiple DPCCHs are required to deliver CR_{UE} in BFS, then the cost of BFS is higher than that of NFS (because the DPCCH delay is anticipated longer and less reliable than that of message delivery in Steps NFS2–5 of NFS). Therefore, it is desirable to combine BFS with NFS to take advantages of both schemes. This new scheme is referred to as the COFC scheme. In the frame synchronization algorithm CFS of the COFC scheme (see Fig. 4), Cell_o determines whether BFS or NFS should be exercised when it receives the Change_Serving_Cell_Request message. Let N^* be the maximum value of the number that can be carried through one uplink DPCCH transmission. If CS_o (which is the same as CR_{UE}) is less than N^*, then Cell_o triggers to execute BFS (i.e., Step BFS2 is performed; cf., Step CFS6 in Fig. 4). Otherwise, Cell_o executes NFS by sending the CS_o value to Cell_n through messages in Steps NFS2–5 (see CFS2–5

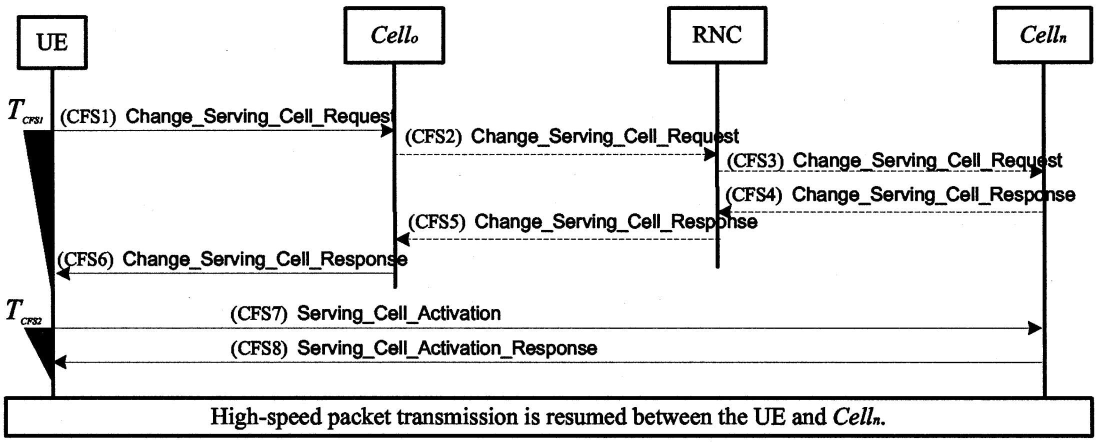


Fig. 4. Message flow for Algorithm CFS.

in Fig. 4). Then it performs Step NFS6, that is the same as Step BFS2. After the UE has received the Change_Serving_Cell_Response message from Cell$_o$, it exchanges Serving_Cell_Activation and Serving_Cell_Activation_Response message pair with Cell$_n$ (Steps CFS7 and CFS8). In the Serving_Cell_Activation message, if $CR_{\mathrm{UE}} < N^*$, then the CR_{UE} value is included. If $CR_{\mathrm{UE}} \geq N^*$, then the value 0 is included. When Cell$_n$ receives the Serving_Cell_Activation message from the UE, it checks whether it has already received the CS_o value from Cell$_o$. If so, it ignores the CR_{UE} value (which is 0 in this case) received from the UE. Otherwise, it uses the received CR_{UE} value to compute N_{sync}. The CFS message flow illustrated in Fig. 4 is basically the same as NFS in Fig. 3, except that the messages sent in Steps CFS2–5 are dashed, which means that these steps may or may not be executed. The timers T_{CFS1} and T_{CFS2} are the same as T_{NFS1} and T_{NFS2}, respectively.

C. COFC With Counter Reset

Consider a HSDPA transmission session. Suppose that J frame synchronizations are performed in this session (i.e., the UE switches serving cells J times). After the jth frame synchronization ($1 \leq j \leq J$), more than N^* packet frames have been transmitted. Then after the jth frame synchronization, four more message exchanges are required when the CFS algorithm is exercised. To reduce the possibility of these extra message exchanges, we can substract the counters CR_{UE}, CS_i, CR_i and CS_{RNC} by a number θ if $CR_{\mathrm{UE}} \geq N^*$ when frame synchronization occurs. We slightly modify the steps in CFS to accommodate this counter reset action. At Step CFS2, if $CS_o \geq N^*$ (i.e., $CR_{\mathrm{UE}} \geq N^*$), then Cell$_o$ first determines the offset value θ

$$\theta \leftarrow \left\lfloor \frac{CS_o}{w} \right\rfloor w. \tag{23}$$

In *Lemma 4*, we will show that the θ value computed in (23) guarantees that after counter reset, flow control between the RNC and the serving cell operates correctly. This θ value is delivered to the RNC and then to Cell$_n$ through the Change_Serving_Cell_Request message. If there are other nonserving cells Cell$_t$ in the active set, then the RNC also forwards the θ value to Cell$_t$. Upon receipt of the θ value, both Cell$_n$ and Cell$_t$ take the following action. Let $i = n$ or t. Cell$_i$ deletes $N_{\mathrm{sync}} = CS_o - CS_i$ packet frames in the buffer, and modifies the CS_i and CR_i values as follows:

$$CS_i \leftarrow (CS_i + N_{\mathrm{sync}}) - \theta \quad \text{and} \quad CR_i \leftarrow CR_i - \theta.$$

Upon receipt of the Change_Serving_Cell_Response messages from Cell$_n$ (see Steps CFS4–8 in Fig. 5), the RNC, Cell$_o$, and the UE modify their counters as follows:

$$CS_{\mathrm{RNC}} \leftarrow CS_{\mathrm{RNC}} - \theta, CS_o \leftarrow CS_o - \theta,$$
$$CR_o \leftarrow CR_o - \theta, \text{ and } CR_{\mathrm{UE}} \leftarrow CR_{\mathrm{UE}} - \theta.$$

Note that the θ value must be a multiple of w [as computed in (23)]. This restriction is required so that flow control between the RNC and the serving cell operates correctly after counter reset. More precisely, the above restriction ensures that the module action performed on CS_{RNC} (see Cases OFC1.1 and OFC1.2) are not affected by the reset operation, as proven in the following lemma.

Lemma 4: In CFS, the following relationship holds:

$$CS_{\mathrm{RNC}} \bmod w = \left(CS_{\mathrm{RNC}} - \left\lfloor \frac{CS_o}{w} \right\rfloor w \right) \bmod w. \tag{24}$$

Proof: Since a packet frame is sent from the RNC to Cell$_o$ before it is sent from Cell$_o$ to the UE, we have

$$CS_{\mathrm{RNC}} \geq CS_o \quad \text{or} \quad \left\lfloor \frac{CS_{\mathrm{RNC}}}{w} \right\rfloor \geq \left\lfloor \frac{CS_o}{w} \right\rfloor. \tag{25}$$

From (25), there exists a nonnegative integer b such that

$$\left\lfloor \frac{CS_{\mathrm{RNC}}}{w} \right\rfloor - \left\lfloor \frac{CS_o}{w} \right\rfloor = b. \tag{26}$$

Let

$$CS_{\mathrm{RNC}} = \left\lfloor \frac{CS_{\mathrm{RNC}}}{w} \right\rfloor w + a \tag{27}$$

where $0 \leq a < w$. Equation (27) implies that

$$CS_{\mathrm{RNC}} \bmod w = a. \tag{28}$$

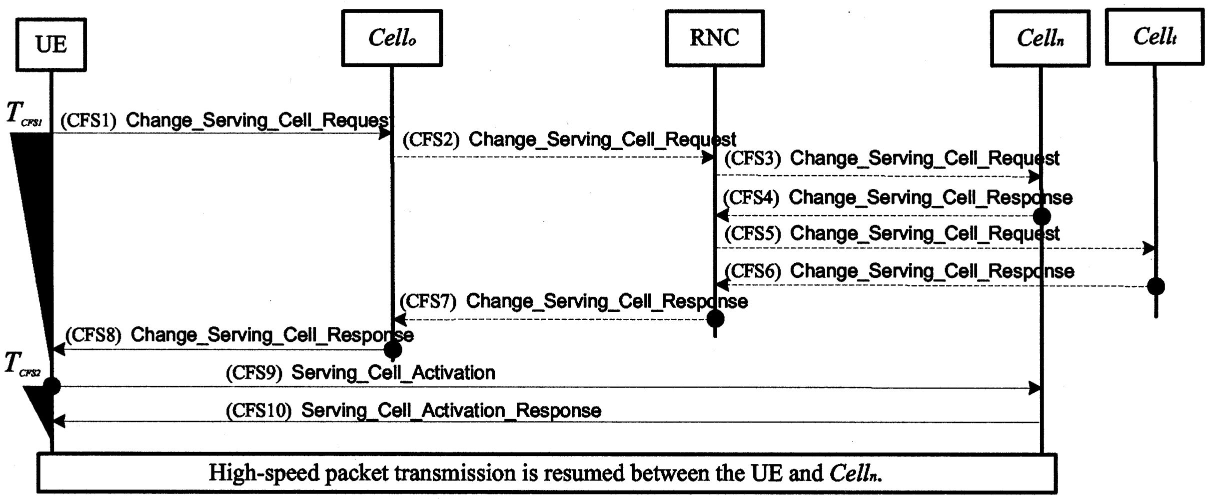


Fig. 5. Message flow for Algorithm CFS with Counter Reset (the counters are reset at the "•" points in the time lines).

Then, from (27) and (26), we have

$$\begin{aligned}&\left(CS_{\text{RNC}} - \left\lfloor \frac{CS_o}{w} \right\rfloor w\right) \bmod w\\ &= \left(\left\lfloor \frac{CS_{\text{RNC}}}{w}\right\rfloor w + a - \left\lfloor\frac{CS_o}{w}\right\rfloor w\right) \bmod w\\ &= (bw+a) \bmod w\\ &= a. \qquad (29)\end{aligned}$$

From (28) and (29), the relationship (24) holds. ∎

Since all counters CS_{RNC}, CS_i, CR_i (for all Cell$_i$ in the active set) and CR_{UE} are subtracted by the same value θ, frame synchronization still functions correctly in CFS, which can also be proved by that in Section II-C.

IV. CONCLUSIONS

This paper described OFC schemes to support the UMTS HSDPA mechanism specified in 3GPP TR 25.950. We first introduced the HSDPA, and then discussed the buffer overflow issue not addressed in 3GPP TR 25.950. To resolve this issue, we proposed three OFC schemes: BOFC; NOFC; and COFC. In BOFC, the information needed for frame synchronization is carried by the uplink DPCCH. When the size of frame synchronization information exceeds the capacity of an uplink DPCCH, this information must be carried through multiple uplink DPCCH transmissions. To avoid multiple HSDPA transmissions, we proposed NOFC to guarantee one uplink DPCCH transmission for frame synchronization through message exchange between the old and new serving cells. Then we proposed COFC that combines BOFC and NOFC. In this scheme, the old serving cell decides whether to transmit frame synchronization information through the network or uplink DPCCH. This paper described the procedures required for overflow control and frame synchronization for HSDPA. Our schemes guarantee that when the buffer of a nonserving cell is full, the previously received packet frames in the buffer can be safely dropped, and after the UE has switched wireless link to the new serving cell, no packet frames are lost. We also provided the correctness proof for the proposed schemes.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers. Their comments have significantly improved the quality of this paper.

REFERENCES

[1] "Technical Specification Group Radio Access Network; Working Group 2; Radio Interface Protocol Architecture," 3GPP (Third Generation Partnership Project), Tech. Spec. 3G TS 25.301 ver. 3.4.0 (2000–03), 2000.
[2] "Technical Specification Group Radio Access Network; High-Speed Downlink Packet Access; Overall UTRAN Description; Release 5," 3GPP (Third Generation Partnership Project), 3G TR 25.855 ver. 5.0.0 (2001–09), 2001.
[3] "Technical Specification Group Radio Access Network; Physical Layer Aspects of UTRA High Speed Downlink Packet Access. Release 4," 3GPP (Third Generation Partnership Project), 3G TR 25.848 ver. 4.0.0 (2001–03), 2001.
[4] "Technical Specification Group Radio Access Network; UTRA High-Speed Downlink Packet Access; Release 4," 3GPP (Third Generation Partnership Project), 3G TR 25.950 ver. 4.0.0 (2001–03), 2001.
[5] "Technical Specification Group Radio Access Network; UTRAN Iub Interface: General Aspects and Principles; Release 4," 3GPP (Third Generation Partnership Project), Tech. Spec. 3G TS 25.430 ver. 4.1.0 (2001–06), 2001.
[6] "Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2," 3GPP (Third Generation Partnership Project), Tech. Spec. 3G TS 23.060 ver. 4.1.0 (2001–06), 2001.
[7] A. Das, F. Khan, A. Sampath, and H.-J. Su, "Performance of hybrid ARQ for high speed downlink packet access in UMTS," in *Proc. IEEE VTC'01-Fall*, vol. 4, 2001, pp. 2133–2137.
[8] P. Frenger, S. Parkvall, and E. Dahlman, "Performance comparison of HARQ with chase combining and incremental redundancy for HSDPA," in *Proc. IEEE VTC'01-Fall*, vol. 3, 2001, pp. 1829–1833.
[9] H. Holma and A. Toskala, *WCDMA for UMTS*. New York: Wiley, 2000.
[10] H. Honkasalo, K. Pehkonen, M. T. Niemi, and A. T. Leino, "WCDMA and WLAN for 3G and beyond," *IEEE Wireless Commun. Mag.*, vol. 9, pp. 14–19, Feb. 2002.
[11] W. S. Jeon, D. G. Jeong, and B. Kim, "Design of packet transmission scheduler for high-speed downlink packet access systems," in *Proc. IEEE VTC'02-Spring*, vol. 3, 2002, pp. 1125–1129.
[12] T. Kawamura, K. Higuchi, Y. Kishiyama, and M. Sawahashi, "Comparison between multipath interference canceller and chip equalizer in HSDPA in multipath channel," in *Proc. IEEE VTC'02-Spring*, vol. 1, 2002, pp. 459–463.
[13] P. Lin, Y.-B. Lin, and I. Chlamtac, "Modeling frame synchronization for UMTS high-speed downlink packet access," *IEEE Trans. Veh. Technol.*, to be published.

[14] Y.-B. Lin and I. Chlamtac, *Wireless and Mobile Network Architectures*. New York: Wiley, 2001.

[15] R. Love, A. Ghosh, R. Nikides, L. Jalloul, M. Cudak, and B. Classon, "High-speed downlink packet access performance," in *Proc. IEEE VTC'01-Spring*, vol. 3, 2001, pp. 2234–2238.

[16] "ARQ Technique for HSDPA," Lucent Technologies, Rep. R2A010 021.

[17] A. Morimoto, S. Abeta, and M. Sawahashi, "Performance of fast cell selection coupled with fast packet schedulling in high-speed downlink packet access," *IEICE Trans. Commun.*, vol. E85-B, no. 11, pp. 2021–2031, 2002.

Phone Lin (M'02) received the BSCSIE degree and Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C., in 1996 and 2001, respectively.

In 2001, he was appointed as an Assistant Professor in the Department of Computer Science and Information Engineering (CSIE), National Taiwan University, Taipei, Taiwan, R.O.C. His current research interests include personal communications services, wireless Internet, and performance modeling.

Dr. Lin is a Guest Editor for IEEE WIRELESS COMMUNICATIONS Special Issue on Mobility and Resource Management.

Yi-Bing Lin (M'96-SM'96-F'03) received the BSEE degree from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1983, and the Ph.D. degree in computer science from the University of Washington, Seattle, in 1990.

From 1990 to 1995, he was with the Applied Research Area at Bell Communications Research (Bellcore), Morristown, NJ. In 1995, he was appointed as a Professor in the Department of Computer Science and Information Engineering (CSIE), National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C. In 1996, he was appointed as Deputy Director of Microelectronics and Information Systems Research Center, NCTU. During 1997–1999, he was elected as Chairman of CSIE, NCTU. His current research interests include design and analysis of personal communications services network, mobile computing, distributed simulation, and performance modeling. He has published over 150 journal articles and more than 200 conference papers. He is the co-author (with Imrich Chlamtac) of the book *Wireless and Mobile Network Architecture* (New York: Wiley, 2001). He is an Adjunct Research Fellow of Academia Sinica, Beijing, China, and is Chair Professor of Providence University, Providence, RI. He also serves as consultant to many telecommunications companies, including FarEasTone and Chung Hwa Telecom.

Dr. Lin is a Senior Technical Editor of *IEEE Network*, an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, an Associate Editor of IEEE Communications Survey and Tutorials, an Editor of *IEEE Personal Communications Magazine*, an Editor of *Computer Networks*, an Area Editor of *ACM Mobile Computing and Communication Review*, a Columnist for *ACM Simulation Digest*, an Editor of *International Journal of Communications Systems*, an Editor of *ACM/Baltzer Wireless Networks*, an Editor of *Computer Simulation Modeling and Analysis*, an Editor of the *Journal of Information Science and Engineering*, Program Chair for the 8th Workshop on Distributed and Parallel Simulation, General Chair for the 9th Workshop on Distributed and Parallel Simulation, Program Chair for the 2nd International Mobile Computing Conference, Guest Editor for the ACM/Baltzer MONET special issue on Personal Communications, a Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS special issue on Mobile Computing, a Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS special issue on Wireless Internet, and a Guest Editor for the *IEEE Communications Magazine* special issue on Active, Programmable, and Mobile Code Networking. He received the 1998, 2000, and 2002 Outstanding Research Awards from National Science Council, R.O.C., and the 1998 Outstanding Youth Electrical Engineer Award from CIEE, R.O.C. He also received the NCTU Outstanding Teaching Award in 2002. He is an ACM Fellow.

Imrich Chlamtac (M'86–SM'86–F'93) received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis.

Since 1997, he holds the Distinguished Endowed Chair in Telecommunications at the University of Texas at Dallas (UTD), Richardson, on joint CS and EE appointment. He holds the titles of Sackler Professor at Tel Aviv University, Israel, Bruno Kessler Honorary Professor at the University of Trento, Italy, and University Professor at the Technical University of Budapest, Hungary. Currently on leave from UTD, he is President of CreateNet, an international nonprofit research organization based in Europe, consisting of over two dozen universities, research centers, and member companies, including the University of Trento, Politecnico di Torino, Technical University of Berlin, Technion—The Israel Institute of Technology, the Technical University of Budapest, ICT-IRST, CNR, Microsoft, HP, Nokia, EM, Lucent, Marconi, and other industry leaders. He has published over 300 refereed articles and is the co-author of the first textbook on Local Area Networks (Lexington Books, 1981, 1982, 1984) and of *Mobile and Wireless Networks, Protocols and Services* (New York: Wiley, 2000), an Amazon.com bestseller and *IEEE Network Magazine* Editor's choice. He is the founding Editor-in-Chief of the *ACM-URSI-Baltzer Wireless Networks (WINET)*, the *ACM-Baltzer Mobile Networking and Nomadic Applications (MONET)* journals, and the *SPIE/Kluwer Optical Networks (ONM) Magazine*. He is the co-founder and past President of CONSIP and BCN corporations.

Dr. Chlamtac is a Fellow of the ACM Society, a Fulbright Scholar, and an IEEE Distinguished Lecturer. He is the winner of the 2001 ACM Sigmobile annual award and the IEEE ComSoc TCPC 2002 award for contributions to wireless and mobile networks, and recipient of multiple best paper awards in the areas of wireless and optical networks.