Overflow Control for UMTS High-Speed Downlink Packet Access

Phone Lin[1], Yi-Bing Lin[2], and Imrich Chlamtac[3]

[1]Dept. of C.S.I.E., National Taiwan University, Taipei, Taiwan, R.O.C.

[2]Dept. of C.S.I.E., National Chaio Tung University, Hsinchu, Taiwan, R.O.C.

[3]Center for Advanced Telecommunications Systems and Services, The University of Texas at Dallas, U.S.A.

Abstract— **This paper proposes overflow control scheme 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. 3GPP specification TR 25.950 proposes HSDPA. In HSDPA, the UE only selects one cell (referred to as the serving cell) in the active set for high-speed downlink transmission. In this mechanism, 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 non-serving 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 non-serving cells do not send packet frames to the UE, their buffers may overflow. In this paper, we propose the scheme to address the buffer overflow issue. Our scheme guarantees that when the buffer of a non-serving 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.**

I. INTRODUCTION

Universal Mobile Telecommunication System (UMTS) [6] is a third generation system proposed by the 3rd 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 services. 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 [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 active set is below the threshold, then the cell is removed from the active set. At most three cells are allowed to be 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 an UE may increase the overall interference within an UTRAN, and thus the data transmission rate decreases.

3GPP TR 25.950 [4] proposes an approach to support *High-Speed Downlink Packet Access* (HSDPA) [4], [2], [3], where an 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*

This work was supported in part by BenQ Corp. and the NSC, ROC, under the contract number NSC90-2213-E-002-148.

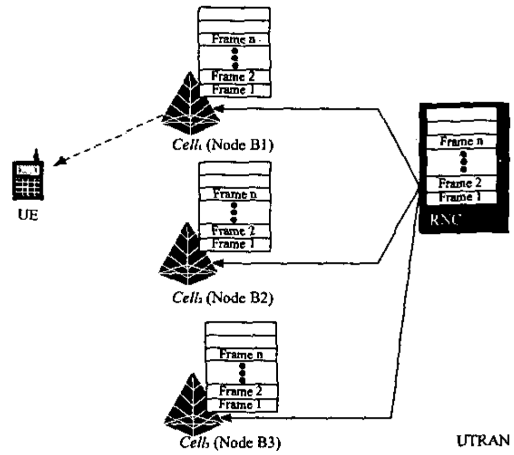

Fig. 1. The network architecture of HSDPA

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. Bit rate performance analysis of the downlink transmission for HSDPA is out of the scope of this paper, and will be treated in a separate report. Figure 1 illustrates the network architecture of HSDPA with the active set $\{Cell_1, Cell_2, Cell_3\}$ 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 non-serving cell, the packet frames are queued in a buffer. The *Stop-And-Wait Hybrid ARQ* (SAW-Hybrid ARQ) [7] 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 non-serving cell may be full, and a mechanism is required to avoid buffer overflow at that non-serving 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 non-serving 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 a overflow control scheme *Basic Overflow Control* (BOFC), and prove the correctness of the scheme. In BOFC, the information needed for frame synchronization is carried by the uplink DPCCHs. Our scheme guarantees that when the buffer of a non-serving 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. The Basic Overflow Control (BOFC) Scheme

This section presents the Basic Overflow Control (BOFC) scheme. 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 non-serving 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. The 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 the cell $Cell_i$ 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 non-serving 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 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.

Case OFC1.1. If $CS_{RNC} \bmod w \neq 0$, the packet frames of the current window are not transmitted completely, and the RNC continues to send the next packet frame to $Cell_i$.

Case OFC1.2. If $CS_{RNC} \bmod w = 0$, 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.

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

$$WI_s \leftarrow CR_s \bmod w$$

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, *Stop-And-Wait Hybrid ARQ* (SAW-Hybrid ARQ) [7] 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 radio interface. The UE replies an ACK or a 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, it replies the status of transmission to $Cell_s$. There are two cases:

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.

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

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

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 to send 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) in Case OFC3.1 and (4) in Case OFC3.2, it is clear that the following relationship holds

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

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

For a non-serving 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 III, 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

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

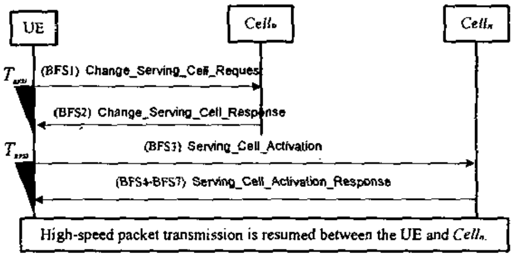

Fig. 2. The message flow for Algorithm BFS

Algorithm OFC4 (exercised by a non-serving cell $Cell_i$)

Step OFC4.1. When $Cell_i$ receives a packet frame from the RNC, it sets $K_i \leftarrow CR_i - CS_i$.

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 $N_{i,max}$.

Step OFC4.3. $Cell_i$ adds the received packet frame at the tail of the buffer, and increments CR_i by one.

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

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

B. The Basic Frame Synchronization (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, $N_{sync} = CR_{UE} - CS_n$ 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 procedure. Figure 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 $Cell_n$)

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.

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 non-serving cell, which executes Algorithm OFC4 to process the next packet frames received from the RNC.

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.

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).

Step BFS5. $Cell_n$ sets $K_n \leftarrow CR_n - CS_n$ and $N_{sync} \leftarrow CR_{UE} - CS_n$.

Step BFS6. If $N_{sync} \leq K_n$, $Cell_n$del etes N_{sync} packet frames in the front of the buffer for the UE. Otherwise ($N_{sync} > K_n$), $Cell_n$ deletes K_n packet frames in the buffer, and deletes the next $N_{sync} - K_n$ packet frames received from the RNC. $Cell_n$ sets $CS_n \leftarrow CS_n + N_{sync}$, 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.

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.

III. Correctness Proof for Frame Synchronization in BOFC

In this section, we prove that the BOFC scheme functions correctly. In BOFC, if $N_{sync} < 0$ when Step BFS5 is executed, then it implies that a non-serving cell $Cell_i$(wh ich 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, $N_{sync} \geq 0$ (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.

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

From (6), (10) is written as

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

Thus to show that no packet frames are lost (i.e., $N_{sync} \geq 0$), 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 $N_{sync} = CS_o - CS_n \geq 0$.

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 $N_{sync} = CS_o - CS_n \geq 0$ 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 re-written 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

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

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

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

From (17) and (18), we have

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

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., $N_{sync} = CS_o - CS_n \geq 0$) 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 non-serving 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 non-serving 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.,

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

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

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

and i packet frames have been lost after frame synchronization.

∎

IV. Conclusions

This paper described overflow control scheme to support 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 the overflow control scheme BOFC. In BOFC, the information needed for frame synchronization is carried by the uplink DPCCHs. This paper described the procedures required for overflow control and frame synchronization for HSDPA. Our scheme guarantee that when the buffer of a non-serving 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 scheme.

References

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Working Group 2; Radio Interface Protocol Architecture. Technical Specification 3G TS 25.301 version 3.4.0 (2000-03). 2000.

[2] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; High Speed Downlink Packet Access; Overall UTRAN Description; Release 5. Technical Report 3G TR 25.855 version 5.0.0 (2001-09). 2001.

[3] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical Layer Aspects of UTRA High Speed Downlink Packet Access; Release 4. Technical Report 3G TR 25.848 version 4.0.0 (2001-03). 2001.

[4] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRA High Speed Downlink Packet Access; Release 4. Technical Report 3G TR 25.950 version 4.0.0 (2001-03). 2001.

[5] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Iub Interface: General Aspects and Principles; Release 4. Technical Specification 3G TS 25.430 version 4.1.0 (2001-06). 2001.

[6] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Descripton; Stage 2. Technical Specification 3G TS 23.060 version 4.1.0 (2001-06). 2001.

[7] Lucent. ARQ Technique for HSDPA. Technical Report R2A010021, Lucent.