Providing congestion control in the Iub Transport Network for HSDPA

Szilveszter Nádas, Sándor Rácz, Zoltán Nagy
Ericsson Research,
Traffic Analysis and Network Performance Laboratory
H-1037, Laborc u. 1, Budapest, Hungary
Szilveszter.Nadas@ericsson.com

Sándor Molnár
High Speed Networks Laboratory,
Dept. of Telecommunications and Media Informatics
Budapest University of Technology and Economics
H-1117, Magyar tudósok krt. 2, Budapest, Hungary

Abstract—The introduction of High Speed Downlink Packet Access (HSDPA) presents new challenges to be solved in the UMTS Terrestrial Radio Access Network. Bandwidth reservation for HSDPA is not efficient and TCP cannot efficiently resolve a congestion situation because lower layer retransmissions hide the congestion situations from TCP. An HSDPA Flow Control algorithm was introduced by 3GPP to control congestion. This HSDPA Flow Control algorithm was originally intended to control radio scheduler queues in Node B. In this paper we propose an algorithm that also provides congestion control in the transport network. The performance analysis concentrates on transport network limited scenarios and shows that the algorithm can achieve high end-user perceived throughput, while maintaining low delay and loss in the transport network. The analysis also shows the advantages of the newly introduced congestion detection functionality.

I. Introduction

In response to the increased need for higher bitrate and more efficient transmission of packet data over cellular networks, the Wideband Code Division Multiple Access (WCDMA) 3rd Generation Partnership Project (3GPP) Release 5 extended the WCDMA specification with the High Speed Downlink Packet Access (HSDPA) [1]. For HSDPA a new shared downlink transport channel, called High-Speed Downlink Shared Channel (HS-DSCH), is also introduced. This channel is dynamically shared among packet data users, primarily in the time domain. The application of shared channel makes the use of available radio resources in WCDMA more efficient. HSDPA also supports new features that rely on the rapid adaptation of transmission parameters to instantaneous radio conditions. The main principles are: fast link adaptation, fast Hybrid-ARQ (HARQ) with soft-combining and fast channel-dependent scheduling [9].

The main architectural novelty of HSDPA is that the control of radio frame scheduling has been moved from Radio Network Controller (RNC) to Node Bs. While for traditional Dedicated Channel (DCH) traffic fix capacity (e.g. 64 kbps) can be reserved in the access network, for HSDPA, bandwidth reservation per flow is not efficient, because air interface throughput is much higher and fluctuates more. If bandwidth reservation is not used then congestion situation can happen both in the transport network and in the air interface. In the current architecture, TCP cannot efficiently resolve a congestion situation in the access network, because lower layer retransmissions hide the congestion situations from TCP. Thus a flow control function has been introduced to control the data transfer between the RNC and Node B. Originally, the flow control was designed to take only the transmission capabilities of the air interface into account and to limit the latency of layer 2 signaling [2]. However, the increased air interface capacity did not always come with similarly increased Iub transport network capacity in practice. The cost of Iub transport links is still high and is not expected to decrease dramatically [11]. It is a common scenario that the throughput is limited by the capacity available on the Iub transport network links and not by the capacity of the air interface. On these high cost links it is important to maintain high efficiency. It has been identified in 3GPP that HSDPA flow control can resolve also these congestion situations if transport network congestion detection functionality is available. For this purpose new fields were introduced to the Iub HS-DSCH Data frame [2], [4], [5], [6].

Various flow control algorithms are developed for different networks. The most known flow control algorithm is the TCP protocol used mainly in IP networks. TCP is widely investigated and improved. Recent works include improvements based on rate and round trip delay estimation [12], [13]. Many papers discussed flow control in Asynchronous Transfer Mode (ATM) networks where the objective was to utilize the bandwidth not used by traffic carried on Constant Bit Rate (CBR) and Variable Bit Rate (VBR) Virtual Circuits (VC) [10]. These algorithms cannot be directly applied for the HSDPA flow control due to difference in the architectures.

In [15], [16], [17], the authors addressed HSDPA flow control. It is a common assumption in these papers that Iub transport network capacity is not limiting. These flow controls are optimized only for efficient use of the air interface. In [14], the authors introduced a transport network overload control algorithm for Best-Effort DCH traffic. They showed that already in the case of DCHs, this improves transport network utilization.

We propose a rate-based flow control algorithm that also supports scenarios where the Iub transport network is the limitation factor. This algorithm also uses the Iub transport network congestion detection functionality standardized by

3GPP.

The rest of the paper is structured as follows. The next section gives a system overview. Section III describes the proposed HSDPA flow control algorithm. The performance of this flow control algorithm is evaluated in Section IV. Finally, Section V concludes the paper.

II. SYSTEM OVERVIEW

The nodes and protocol layers involved in the HSDPA Flow Control (FC) are depicted in Fig. 1 [1]. The figure also shows the location of the FC related functionalities. The task of the FC is to regulate the transfer of Medium Access Control-d Packet Data Units (MAC-d PDUs) from Serving RNC (SRNC) to Node B, more precisely to regulate the transfer of data from the Radio Link Control (RLC) buffer in RNC to the MAC-hs buffer in Node B, called Priority Queue (PQ). In the rest of the article *flow* denotes these MAC-d PDU flows.

The FC is responsible for handling two potential bottlenecks: the Iub Transport Network (TN) and the air interface (Uu). Typically the TN bottleneck is a single link between RNC and Node B, where all flows of the same Node B share the same TN bottleneck buffer and TN capacity. These flows can utilize the remaining TN capacity from high priority traffic (e.g. DCH). Each flow belonging to the same cell shares Uu resources but each flow has a dedicated PQ in the Node B. A Node B contains one ore more cells. FC is responsible for efficient use of these changing TN and Uu bottlenecks. FC must maintain high end-user throughput while maintaining low end-to-end delay for delay sensitive applications (e.g. gaming over best effort HSDPA). The delay target for MAC-d PDUs is typically smaller than 100 ms.

MAC-hs is responsible for link adaptation, HARQ and channel-dependent air interface scheduling. RLC Acknowledged Mode (AM) is used to compensate HARQ failures, to provide seamless HS-DSCH handover and efficient channel switching to/from DCH [7]. RLC in AM retransmits all PDUs that were lost in TN or Uu. Thus TCP cannot detect congestion in TN or Uu unless the congestion causes serious RLC protocol problem. Consequently, TCP cannot control congestion on these interfaces and a system specific congestion control solution is needed.

Packet loss and the resulting RLC retransmission shall be minimized because it significantly increases the delay variation. The TN delay shall be kept low due to delay sensitive applications over HSDPA and to minimize control loop delay for FC and RLC. High delay in the PQs for non-delay sensitive applications is not a problem for the end-user, but it is still a problem for RLC protocol performance and during handover when all PDUs waiting in a PQ are dropped. For delay sensitive applications, the PQs usually store small amount of data because these applications usually generate low load.

FC related data frames and control frames are standardized in [2]. HS-DSCH Iub data frames (DF) contain the user data and transmit information about the amount of user data waiting in the RNC, called User Buffer Size (UBS), and contain information for congestion detection, the Frame Sequence Number (FSN) and the Delay Reference Time (DRT). Based on UBS, Uu related information and the output of congestion detection the FC algorithm in the Node B decides how many MAC-d PDUs can be transmitted from the RNC for a given flow. This is reported to the SRNC using the HS-DSCH Capacity Allocation Control Frame (CA). The CA defines maximum MAC-d PDU length, HS-DSCH Credits, HS-DSCH Interval and HS-DSCH Repetition Period values. The MAC-d shaping in SRNC ensures that within a given *Interval* not more than *Credits* PDUs are sent. *Repetition Period* defines how many times the *Interval* and the *Credits* are repeated. A newly received CA overrides the old one. Setting the *Repetition Period* unlimited allows to define an allowed shaping rate, e.g. max PDU length 42 octets, Credits 4 and Interval 10 ms settings define shaping rate of 134.4 kbps.

The FC algorithm itself is not standardized, each vendor can implement its own algorithm. For the algorithm it is allowed to use all information available in the Node B. We aggregate information in different levels. In Fig. 1 we indicated our Node B and cell level aggregations, the use of Air Interface Scheduling and MAC-hs PQ information.

Originally, only transport protocol specific congestion detection techniques were possible to use in the FC. In case of ATM/AAL2 based transport, the partly received AAL2 SSCS SDUs can be detected and used as congestion information (see AAL2 protocol details in [8]). In case of IP/UDP solution Explicit Congestion Notification (ECN) can be used.

To improve congestion detection and create a general congestion detection framework, the possibility of transport protocol independent congestion detection was proposed (partly by the authors of this article) and added to the standardized DFs (namely FSN and DRT [4], [5], [6]).

III. FLOW CONTROL ALGORITHM DESCRIPTION

In this section, we introduce a rate-based FC solution, which includes per flow part, per cell and per Node B level aggregated information. A rate-based solution is chosen because it generates moderate CA load and there is no acknowledgement for DFs or CAs. High CA load should be avoided because it requires high processing power from the nodes. The DF
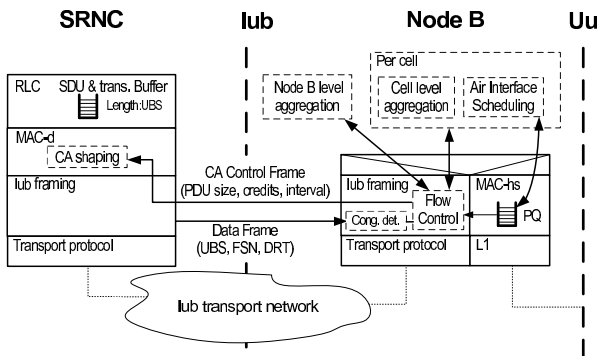

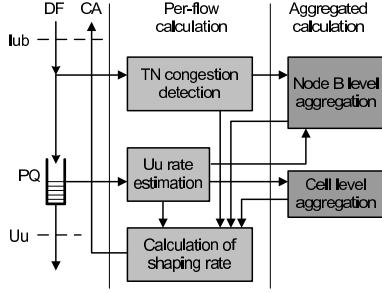Fig. 1. Protocol stack and flow control architecture

Fig. 2. Flow control architecture in Node B

and CA acknowledgement would be needed for using a TCP-like window based solution. A credit based solution would generate high CA load and would be sensitive for loss in TN. The architecture of the FC algorithm in Node B is depicted in Fig. 2. The per flow part is responsible for the fast reaction to a congestion situation in TN or long PQ delays. The cell level aggregation estimates the frequency of scheduling a PQ in the given cell. The Node B (or TN) level aggregation approximates the available TN capacity for HSDPA and distributes it among the flows. The different aggregation levels of the algorithm are evaluated on different timescale because the speed of adaptation to the different bottlenecks should be in line with the speed by which the available bottleneck capacity is changed.

The notations introduced in this section reflect the different aggregation levels and the type of the variables. Variables without superscript denote per flow variables, while superscript $cell$ denotes cell level and $NodeB$ means Node B level aggregation. The used variables are of type rate (R), coefficient (Q), buffer size (b), counter (N) and time (t).

A. Flow states

If the UBS field in the DFs is zero for more than 100 ms, then the state of the flow changes to inactive. In inactive state the FC sends a CA indicating 32 kbps bitrate and the flow is not included in the aggregated calculations. Thus in practice very low bitrate flows will not be affected by FC. If a DF with UBS larger than zero is received in the Node B, then a new bitrate is calculated and a CA is sent immediately.

B. TN congestion detection

The TN congestion detection part of the algorithm is performed whenever a DF arrives to the Node B. There are three different congestion detection methods, namely:

- *Destroyed Frame Detection (DFD).* Due to the fact that ATM cells (not the DFs) could be lost in the TN, it is possible that only a part of a DF is lost. When the segmented DF is reassembled it is possible to detect that a part of the DF was lost.
- *FSN gap detection.* The 4 bit FSN in the DF can be used to detect missing DFs (which were fully lost).
- *Dynamic Delay Detection (DDD).* The DRT field in the DF contains the value of a reference counter in the RNC

when the DF was sent. The DRT is compared to a similar reference counter in Node B when the DF is received. The difference between the two counters increases when the TN buffer is built up. Congestion is detected when this difference increases too much compared to the minimum difference.

In case of large TN buffers DFD and FSN cannot provide efficient congestion detection because protocol problems occur before the TN buffer becomes full. DDD provides congestion detection in this case and it decreases TN delay and loss also in case of small TN buffers.

C. Calculation of shaping rate

The shaping rate is calculated for every active flow separately, once per 100 ms. The 100 ms value is a compromise between fast reaction, low CA frequency and low calculation complexity. The time of this calculation in every 100 ms is called tick time, which is not synchronized for the different flows.

The FC distributes the estimated TN capacity among the flows proportional to their estimated Uu rate. Thus a newly arriving flow gets its share of the TN capacity, while the rate of ongoing flows is reduced. The short term TN congestions are taken into account by a coefficient. The calculated rate is translated to CA format and sent to the SRNC.

The estimated TN link capacity ($R_{estTN}^{Node\ B}$) and the sum of estimated Uu capacities (R_{estUu}^{NodeB}) are provided by the Node B level aggregation. The estimated Uu rate (R_{estUu}) is provided by the cell level aggregation. The FC calculates the shaping rate as follows:

$$R_{calc} = \min\left(\frac{R_{estUu}}{R_{estUu}^{NodeB}} \cdot R_{estTN}^{NodeB}, R_{estUu}\right) \cdot Q_{Iub}, \quad (1)$$

where Q_{Iub} is a coefficient that reacts fast on a detected congestion. The Q_{Iub} is set to 0.5 if TN congestion has been detected since the last tick, otherwise it is increased by 0.05. The maximum value for Q_{Iub} is 1.

D. Node B level aggregation

The purpose of the Node B level aggregation is to estimate the available TN capacity. The value of R_{estTN}^{NodeB} is updated once per second. It is based on the number of TN congestion detected by the per flow ticks of the different flows during the last second, called $N_{IubCong}^{NodeB}$. As there is one tick in every 100 ms per flow, one flow can contribute at most ten flags during the evaluation period.

The R_{estTN}^{NodeB} is limited by a pre-configured minimum and maximum rate (called R_{minHS}^{NodeB} and R_{maxHS}^{NodeB}), which must be configured according to the TN configuration. If $N_{IubCong}^{NodeB}$ is greater than or equal to a predefined threshold (e.g. five) or the number of active flows (N_{flow}^{NodeB}), then the $R_{estTN}^{Node\ B}$ is decreased by 2% of R_{maxHS}^{NodeB}. If the $R_{estTN}^{Node\ B}$ was not decreased in the last ten seconds then it is increased by 1% of R_{maxHS}^{NodeB} every second. The constants are determined

by considering how frequently and to which extent the TN capacity typically changes.

The R_{estUu}^{NodeB} equals to the sum of R_{estUu} for all flows in the Node B. This variable is used for distribution of R_{estTN}^{NodeB} among the flows.

E. Uu rate estimation

To predict the Uu rate of a flow (R_{estUu}), the possible peak Uu rate of the flow (R_{peakUu}) is divided by the average number of competing flows (N_{flow}^{cell}). For long PQs the estimated values are further reduced. The R_{peakUu} is reported by the radio scheduler.

$$R_{estUu} = Q_{t_{PQ}} \cdot \frac{R_{peakUu}}{N_{flow}^{cell}}, \qquad (2)$$

where $Q_{t_{PQ}}$ is a coefficient calculated from the estimated time to serve all PDUs in the PQ (t_{PQ}). The time t_{PQ} is calculated as follows

$$t_{PQ} = \frac{b_{PQ}}{\frac{R_{peakUu}}{N_{flow}^{cell}}}, \qquad (3)$$

where b_{PQ} is the length of the PQ (in bits). The value of $Q_{t_{PQ}}$ is set to keep the delay in an appropriate level. Its value is one if t_{PQ} is smaller than 50 ms, it is zero if t_{PQ} is larger than 150 ms and it decreases linearly if t_{PQ} is between 50 ms and 150 ms.

The HSDPA flow control solutions described in [15], [16], [17] could also be used for the Uu part of the algorithm.

IV. PERFORMANCE ANALYSIS

We implemented the FC algorithm introduced in Section III in a WCDMA protocol simulator. It contains the introduced HSDPA related protocol functions, namely TCP/IP, RLC, MAC-d, HS-DSCH Iub framing and MAC-hs. The ATM/AAL2 TN is modeled as a link of fix capacity with a finite buffer, and fix downlink propagation delay (3.6 Mbps, 432 kbit and 10 ms). The radio environment consists of standard models for distance attenuation, shadow fading and multipath fading, based on 3GPP typical urban channel model, see [3]. The radio scheduler in Node B uses round robin scheduling scheme. The radio network used by the simulator consists of an RNC and a Node B with 3 cells.

As the novelty of our method is the introduction of TN congestion control, we concentrate our analysis on TN limited cases. We implemented a traffic model, which can load the TN and it is simple enough to evaluate the system behavior in detail. This model has three parameters: number of users attached to the Node B, object size downloaded by the users and the mean of reading time which is the gap between two consecutive downloads of the same user (5 sec). The users are equally distributed among the cells.

We use the following performance measures to investigate the performance and potential protocol problems: average total TCP throughput, average DF delay in TN, average MAC-d PDU loss ratio in TN, number of TCP timeouts per user and number of RLC resets per user. The simulations were run for 600 sec to evaluate these measures. Notice that due to the
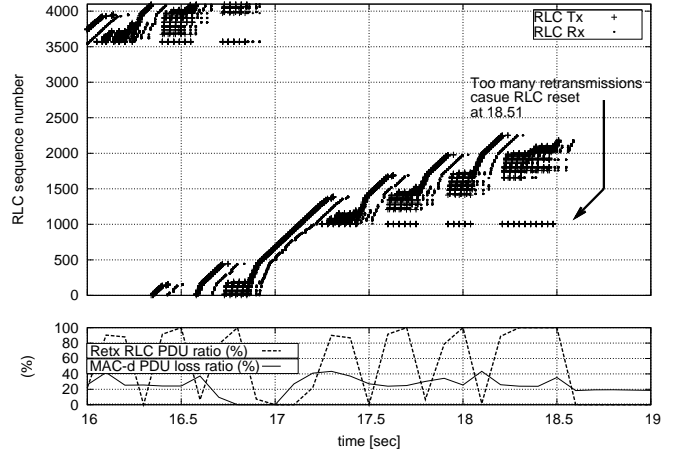

Fig. 3.   RLC protocol behavior with no flow control

protocol overheads the maximum achievable TCP throughput is 2.7 Mbps in our scenario.

If the system is TN limited, then the lack of FC causes serious protocol problems. In this case, the TN bottleneck buffer is typically full causing high TN delay and loss ratio. This causes several RLC retransmissions, RLC and TCP protocol problems resulting much lower TCP throughput.

To illustrate the potential protocol problems when no FC is used, we investigate a scenario with 5 users and 10 Mbyte object size. Fig. 3 shows the RLC trace for one of these users. The figure on top shows the time when an RLC PDU of a given sequence number is sent from RNC (RLC Tx, plus) and when it is received in Node B (RLC Rx, dot) for the selected user. The MAC-d PDU loss ratio in the TN (solid line) and the retransmitted RLC PDU ratio (dashed line) is depicted in the bottom figure. Notice that a MAC-d PDU carries exactly one RLC PDU. The retransmitted RLC PDU ratio compares the number of retransmitted RLC PDUs (which are the result of RLC retransmission) and the total number of transmitted RLC PDUs. It can be seen on the figure how the increase of loss in TN causes several RLC retransmissions. At time 18.51 an RLC reset happens because of too many retransmissions of the same RLC PDU. The achieved TCP throughput during the whole simulation was 1.2 Mbps, the number of TCP timeouts was 15 per user caused by 15 RLC resets per user.

Even if we limit the CA shaping rate of the flows to a fixed value (e.g. 800 kbps/flow) the system will work only slightly better despite the fact that this setting in case of 5 users does not allow extreme TN overload. In the same scenario as above the achieved TCP throughput was 1.7 Mbps, the number of TCP timeouts was 8 per user caused by 7 RLC resets per user.

For the flow controlled case, we compared the simulation results with and without using DDD. Fig. 4 compares the performance for different number of users and two different object sizes.

The simulations show that the proposed FC can maintain high TCP throughput and reasonable TN delay and loss. The
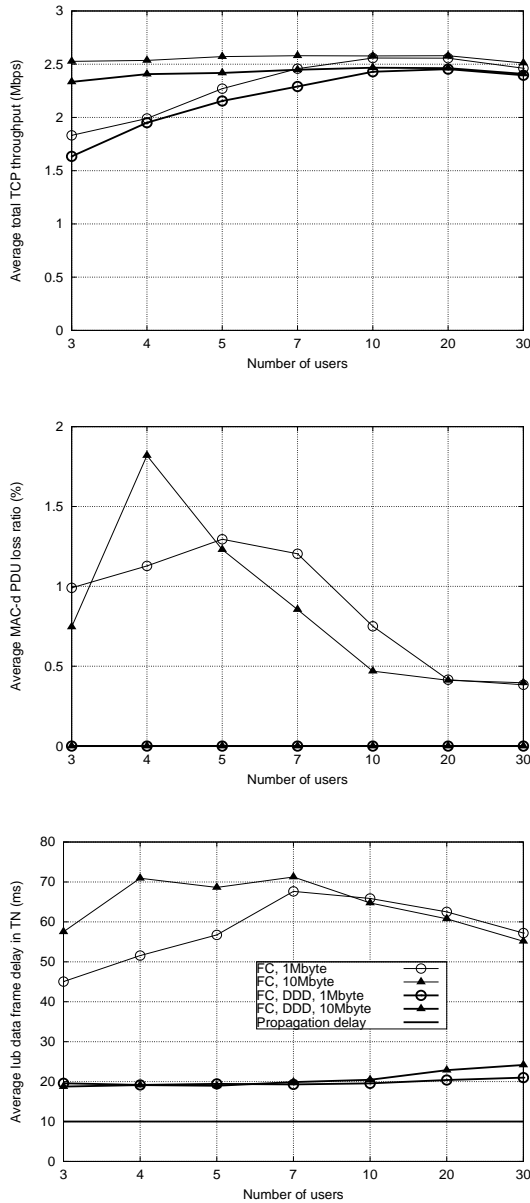
Fig. 4.   TCP throughput, MAC-d PDU loss ratio and DF delay

V. Conclusions

An HSDPA Iub Flow Control algorithm utilizing several levels of aggregation has been proposed. The need for Iub transport network congestion detection was shown and the Iub transport network congestion detection techniques were described. It was shown by simulation that the proposed algorithm using the introduced congestion detection techniques can maintain high transport network utilization while also keeping the delay and loss in the transport network low. It is shown that the introduction of dynamic delay based congestion detection can further decrease the delay and eliminate the loss, while maintaining only slightly less TCP throughput. The solution has been compared to a scenario without flow control and it has been shown that the lack of flow control causes serious performance degradation in the system when the Iub transport network capacity is limiting the throughput.

Acknowledgment

References

[1] 3GPP TS 25.308 V6.3.0, High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2 (Release 6), 2004-12
[2] 3GPP TS 25.435 V6.3.0, UTRAN Iub Interface User Plane Protocols for Common Transport Channel data streams (Release 6), 2005-09
[3] 3GPP TS 25.943 V6.0.0, Deployment aspects (Release 6), 2004-12
[4] 3GPP TSG-RAN3 meeting #43, Tdoc R3-041115, Introduction of an HS-DSCH Data Frame "Frame Number Sequence" field, *Prague, Czech Republic, 16th-20th August 2004*
[5] 3GPP TSG-RAN3 meeting #43, Tdoc R3-041118, Introduction of an HS-DSCH Data Frame "Delay RNC reference Time (DRT)" extension, *Prague, Czech Republic, 16th-20th August 2004*
[6] 3GPP TSG-RAN WG3 Meeting #47, Tdoc R3-050763, Transport Network Congestion Detection and Control, *Athens, Greece, 9th-13th May 2005*
[7] 3GPP TS 25.322 V6.9.0 Radio Link Control (RLC) protocol specification (Release 6), 2006-09
[8] Segmentation and Reassembly Service Specific Convergence Sublayer for the AAL type 2, *ITU-T Recommendation I.366.1*
[9] S. Parkvall et al., WCDMA evolved – High-speed packet-data services, *Ericsson Review,* no. 02, 2003
[10] Raj Jain, Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey, *Computer Networks and ISDN Systems, pp. 1723-1738, Vol. 28, No. 13, February 1995*
[11] Ana-Beln Garca, et al., ATM Transport between UMTS Base Stations and Controllers: Supporting Topology and Dimensioning Decisions. *IEEE Proc. PIMRC September 2004, pp. 2176-2180*
[12] L. Brakmo, S. O'Malley, and L. Peterson, TCP Vegas: New techniques for congestion detection and avoidance. *In Proceedings of the SIGCOMM '94 Symposium (Aug. 1994) pp 24-35*
[13] David X. Wei, Cheng Jin, Steven H. Low and Sanjay Hegde, FAST TCP: motivation, architecture, algorithms, performance, *IEEE/ACM Trans. on Networking,* in press
[14] M. Sagfors, V. Virkki, T. Kuningas, Overload Control of Best-Effort Traffic in the UTRAN Transport Network *in Proc. IEEE VTC 2006-Spring*
[15] P. J. Legg, Optimised Iub Flow control for UMTS HSDPA, *in Proc. IEEE VTC 2005-Spring*
[16] Marc C. Necker, Andreas Weber, Impact of Iub Flow Control on HSDPA System Performance, *IEEE Proc. PIMRC Sept. 2005, pp. 1703-1707*
[17] Marc C. Necker, Andreas Weber, Parameter Selection for HSDPA Iub Flow Control. *Proceedings of the 2nd International Symposium on Wireless Communication Systems, Siena, 2005*

usage of DDD decreases the delay and completely eliminates the loss in TN, while maintaining only slightly less TCP throughput. When the FC works it prevents the RLC resets and TCP timeouts in both cases and improves the performance.

Further simulations with very long TN buffers show that FC without DDD could not maintain this high TCP throughput. In this case, only DFD and FSN gap detection is available for TN congestion detection and protocol problems happen before the TN buffer is full and a DF is lost. Consequently, FC without DDD cannot prevent protocol problems when the TN buffer is very large, because congestion information is not available in time. FC with DDD can maintain the same performance as in Fig. 4 also in case of vary large TN buffers, because DDD works the same in this case.