Copyright Notice

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-8026, Fax: ++49-711-685-7983
email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Impact of Iub Flow Control on HSDPA System Performance

Marc C. Necker
Institute of Communication Networks and Computer Engineering,
University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Email: necker@ikr.uni-stuttgart.de

Andreas Weber
Alcatel SEL AG, Research and Innovation
Lorenzstr. 10, D-70435 Stuttgart
Email: Andreas.Weber@alcatel.de

***Abstract*— The recently emerging High Speed Downlink Packet Access (HSDPA) enhances conventional WCDMA systems according to the UMTS standard with data rates of up to 14MBit/s in the downlink direction. This is achieved by using adaptive modulation and coding as well as a fast Hybrid Automatic Repeat Request (HARQ) mechanism. This functionality is implemented close to the air interface in the Node B. In addition to the data buffer in the RNC, this requires a second data buffer in the Node B. Consequently, a flow control mechanism is needed which controls the amount of data to be transmitted from the RNC's buffer to the Node B's buffer. The spatial separation of RNC and Node B imposes significant signaling constraints and control dead time limitations to the flow control mechanism. Additionally, due to the time-varying nature of the radio channel, the data rate towards a particular user may be highly variable. In this paper, we study the impact of the flow control on system performance. We will show that it is essential to jointly consider scheduling and flow control in an HSDPA system as the constraints imposed by the flow control may dominate the system performance.**

I. Introduction

In the past years, WCDMA networks based on the UMTS standard have widely been deployed. In these systems, several Node B base stations are connected to a Radio Network Controller (RNC) via the Iub interface. The RNC implements all relevant radio protocols, such as the Radio Link Control (RLC) and the MAC-d, while the Node B is a mere slave device, responsible for the actual physical transmission on the air interface. As the RNC and the attached Node Bs are usually distributed over several sites, the data link between a Node B and the RNC introduces a significant additional delay.

With the evolution of 3G systems, UMTS has been extended by High Speed Downlink Packet Access (HSDPA). HSDPA provides increased data rates in the downlink direction of up to 14MBit/s. It introduces an additional functional layer in the protocol stack, namely the MAC-hs layer. The MAC-hs functionality is implemented in the Node B, which allows a much faster reaction on errors and variations of the channel quality, compared to protocols implemented in the RNC. This allows for fast adaptations of the modulation and coding scheme as well as for a powerful HARQ mechanism [1].

As the HSDPA functionality is distributed, two separate data buffers are required in the RNC and Node B, respectively. Consequently, a data flow control is needed which controls the amount of data to be transmitted from the RNC's buffer to the Node B's buffer. This flow control is typically located in the Node B and signals the amount of data to be transmitted to the RNC. Its goal is to keep the buffer level in the Node B at an adequate level. If the Node B's buffer is too full, the RTT is unnecessarily increased, causing problems to RLC protocols and other higher layer protocols. On the other hand, a minimum buffer level should always be maintained to prevent the buffer from running empty and thus wasting air interface resources.

Due to the time-variant behavior of the radio channel, the data rate on the air interface is highly variable. Additionally, the relatively high protocol delay between a Node B and its corresponding RNC may impose a significant dead time to the flow control loop. As a consequence, the flow control may not be able to react to data rate fluctuations on the air interface as fast as necessary. This issue becomes even more important when using sophisticated scheduling algorithms, which may lead to a rather unpredictable data flow towards each terminal.

Scheduling in mobile communication systems has widely been addressed. A general introduction and in-depth study of scheduling and QoS in HSDPA is provided by Gutierrez in [2]. In [3], Kolding investigates the performance of Proportional Fair (PF) scheduling in HSDPA systems under non-ideal channel condition reporting. In [4], Aniba and Aissa enhance the PF approach to provide fairness, when channel conditions towards different users are heterogeneous. The issue of flow control in HSDPA systems has rarely been addressed so far. In [5], Legg presents an optimized Iub flow control algorithm. In this paper, we study the delay performance of the Iub flow control and highlight related performance problems. We investigate the interplay between the Node B scheduling and the flow control for different mobility and scheduling scenarios. It will become obvious that it is necessary to jointly consider flow control and scheduling in order to evaluate the performance of HSDPA systems. We will show that flow control related signaling constraints have a significant impact on the overall system performance, and we will explore possibilities to improve the performance under the given constraints.

This paper is structured as follows. Section II introduces the investigated system and the corresponding model. In section III, the flow control and scheduling algorithms are introduced. Section IV addresses the problems inherent to flow control algorithms in combination with different schedulers and mobility conditions. Finally, section V concludes the paper.

II. System Model

A. System Overview

The basic scenario is shown in Fig. 1. We consider a single-cell environment, where several User Equipments (UEs) connect to the Node B via a High Speed Downlink Shared Channel (HS-DSCH) in the downlink and a dedicated channel (DCH) in the uplink direction. The Node B is connected to the RNC, which itself is connected to the Internet via the 3G-SGSN and 3G-GGSN of the cellular system's core network. The UEs establish a data connection

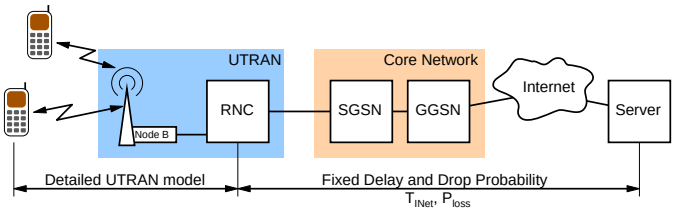

Fig. 1: Architecture of the considered 3G network

with a host in the Internet. The Internet and core network were assumed to introduce a constant delay $T_{\mathrm{INet}} = 20\mathrm{ms}$ in each direction and not lose any IP packets.

B. Simulation Model

All simulations were performed using an event-driven simulation tool, which was implemented using the IKR SimLib [6]. The HSDPA network was modeled with all its relevant RLC, MAC-d and MAC-hs protocols. The physical layer was modeled based on BLER-curves obtained from physical layer simulations including HARQ. Transport formats (TF) on the MAC-hs layer were selected based on the channel quality such that the BLER is 10%. We assumed ideal conditions for the reporting of Channel Quality Indicators (CQI) from the mobile terminals to the Node B, i.e. zero delay, in order to isolate the performance influence of the flow control. The maximum number of MAC-hs retransmissions was limited to $R_{\mathrm{max,hs}}$, and the maximum number of RLC-retransmissions to $R_{\mathrm{max,rlc}} = 10$. The maximum RLC window size was assumed to be unlimited in order to avoid side effects in the results. We neglect the convergence layer, as it only introduces a very small overhead in a single-cell environment.

III. Flow Control and Scheduling

A. RNC / Node B flow control

The flow control process regulates the transfer of RLC blocks from the RNC to the Node B. In order to provide a fair treatment of all data flows, each connection has its own independent flow control process. The general concept is shown in Fig. 2 for one data connection: IP packets arriving at the RNC are first stored in RNC input buffers with one buffer per data connection. The RNC segments and concatenates, respectively, incoming data packets into RLC blocks (S/C). These RLC blocks are protected by the RLC layer's ARQ mechanism and transmitted to the Node B, where they are stored in individual Node B buffers, also known as *HS priority queues*. For our simulations we have chosen the memory large enough to avoid side-effects caused by limited buffer sizes. Furthermore, we assumed that all data flows have the same priority.

The flow control tries to keep track of the dynamic radio channel. It controls the transfer rate R_i of RLC blocks to the Node B so that the waiting time of data blocks in the Node B does not become too large. On the other hand, it tries to provide sufficient RLC blocks so that the Node B buffer never runs empty.

In general, flow control mechanisms are vendor specific. In the following, we present a generic algorithm in order to study some basic performance issues. The goal of our algorithm is to tune the buffer level for every data flow so that a predefined queuing time T_w in the Node B's buffer

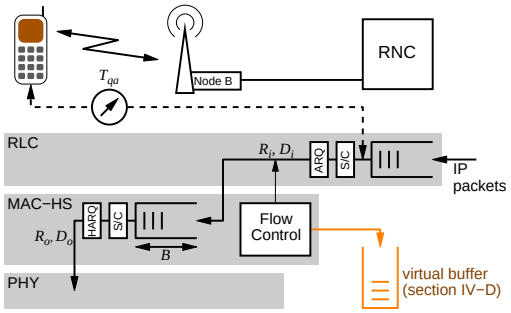

Fig. 2: Flow control overview

is obtained. Consequently, the flow control tries to keep the buffer level B_w of every data flow at:

$$B_w = R_o \cdot T_w \quad , \tag{1}$$

where R_o is the bit rate of RLC frames transmitted for the first time over the radio channel for the considered connection. Hence, R_o corresponds to the data connection's effective channel bit rate. This value has to be measured and, in order to be accurate enough, this measurement value has to be averaged over a certain period of time T_m. However, the longer this measurement period, the more obsolete is this value.

The flow control tries to compensate the difference between the desired buffer level B_w and the actual level B within a period of T_u/α, with $0 < \alpha \leq 1$. As soon as the flow control is aware of the new measurement values, a new transfer rate R_i is calculated which is in use for the next update period:

$$R_i = \max\left(0, R_o + \alpha \frac{B_w - B}{T_u}\right) \quad . \tag{2}$$

Consequently, for every data flow during the next update period T_u, the RNC may transfer a maximum data volume D_i:

$$D_i = R_i \cdot T_u \quad . \tag{3}$$

This is performed by issuing resource grants to the RNC in regular intervals of $TTI_{\mathrm{RLC}} = 10\mathrm{ms}$, as illustrated in Fig. 3. These resource grants are sent periodically with an update interval period of T_u. That is, the Node B signals T_u/TTI_{RLC} resource grants to the RNC at the end of each update interval (cmp. Fig. 3). The update interval has to be small enough to allow the flow control to accurately follow the channel dynamics. On the other hand, it has to be large enough to keep the signaling load between RNC and Node B at a reasonable level. Due to protocol delays, the resource grants are not used instantaneously but with a certain delay T_p, also known as *dead time*.

Up to now, our generic algorithm works fine in case of

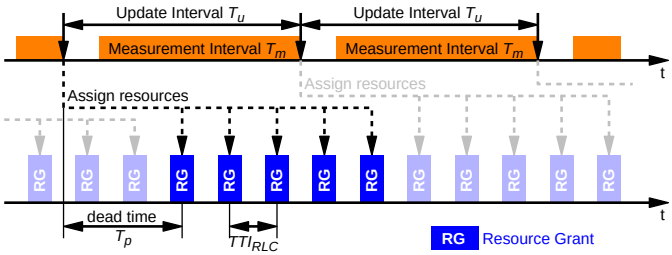

Fig. 3: Flow control timing

a static data flow when there is always enough data to be transmitted. In order to enable the algorithm to work in case of a beginning data transfer or in the case of data transfer interruptions we have to refine the measurement of R_o.

In case of a starting data transfer and in case of interruptions longer than T_u we do not have any measurements. Instead, a default value $R_o = R_{\mathrm{def}}$ is applied. Alike, in case of short transfer interruptions we have to consider the potential data rate, i.e. the data rate in case the buffer is not empty, rather than the real data rate:

$$R_o = \begin{cases} \frac{D_o}{T_b} & \text{if } T_b > 0 \\ R_{\mathrm{def}} & \text{otherwise} \end{cases} . \quad (4)$$

D_o is the data volume transferred during the last update interval T_u, and T_b the total amount of time during the measurement period T_m during which the data buffer was not empty.

In addition to the results given in this paper, we performed simulation runs in order to find optimal values of the parameters involved in the flow control [7]. We chose $T_w = 100$ms, $T_m = 50$ms, $R_{\mathrm{def}} = 100$kbit/s and $\alpha = 1.0$.

B. Node B scheduling

In the Node B, a Round Robin (RR) and a Proportional Fair (PF) [8] scheduler were considered. While the RR scheduler equally serves all users in a cyclic manner, the PF scheduler bases its scheduling decision on the current channel conditions towards each user. As of such, the RR scheduler produces a smoother data flow towards each user compared to the PF scheduler, which may serve users in bursts when their channel condition is above average. Note that the PF scheduler essentially behaves like a RR scheduler if no mobility is considered.

IV. Performance Evaluation

A. Simulation Scenario

We consider five independent terminals which perform bulk data transfer in the downlink. TCP NewReno with window scaling was used as transport protocol. Terminal mobility was modeled taking into account both slow and fast fading. We define three scenarios. In the *StaticRR* scenario, RR scheduling is used. The terminals do not move and exhibit identical channel conditions. The *MobileRR* scenario differs in that all mobiles move at a speed of $v = 30$km/h. They periodically experience the same slow fading profile, where each mobile starts at a different position of the profile in order to obtain independent channel conditions in-between the mobiles. The same mobility conditions were assumed in the *MobilePF* scenario, where the RR scheduler was substituted by a PF scheduler.

B. Impact of scheduling and mobility

We first consider ideal flow control conditions, i.e. T_p is zero and the update period is short, i.e. $T_u = 10$ms. Figures 4–6 plot the complementary cumulative distribution function (ccdf) of the queue adjusted IP packet delay T_{qa} for the three considered scenarios and different values of $R_{\mathrm{max,hs}}$. T_{qa} is measured as shown in Fig. 2. It does not take into account the RNC queuing delay, which is mainly determined by the traffic source behavior. In the static scenario, the channel remains constant for all transmission attempts of a data block. Consequently, the loss probability is 10% for the first and virtually zero for the second transmission attempt due to the HARQ mechanism, which explains the optimal curves for $R_{\mathrm{max,hs}} \geq 1$. In contrast, in the mobile scenarios, the loss probability is significantly larger than zero for retransmissions, as the channel quality changes in-between transmission attempts. As all transmissions have to be performed with the same TF, retransmissions may experience a channel which does not provide the quality necessary for the selected TF. This eventually leads to RLC layer retransmissions, which explains why, after a certain minimum delay, the ccdf of the delay drops down to the beginning of a heavy distribution tail.

On average, the effective TCP throughput in the MobileRR scenario is 32.5 kByte/s, compared to 47.6 kByte/s in the MobilePF scenario. This implies a gain of approximately 1.5 of the PF scheduler over the RR scheduler, which goes well along with the findings in [3]. It is interesting to observe that the PF scheduler leads to longer delays compared to the RR scheduler, even though it achieves a higher data rate. This is due to the fact that the PF scheduler tries to exploit good channel conditions and selects TFs with relatively high data rates. This eventually leads to a less predictable data rate, higher loss probabilities for retransmissions compared to the RR case, and consequently to longer delays.

C. Update period and dead time

In a real system, the dead time T_p is likely to be on the order of 10 to 30ms. Alike, the update period is expected to be larger than 10ms. Figures 7–9 show the ccdf of T_{qa} for an ideal dead time of $T_p = 0$ms and different update periods. In the StaticRR scenario, a longer update period does not worsen the performance. This is reasonable, since the data rate towards a particular user is constant. Consequently, the data rate R_i assigned by the flow control is constant as well, and the length of the update period is of little importance.

In contrast, the mobile scenarios lead to a highly time-variant data rate towards each terminal. Consequently, the adjusted data rate R_i becomes less accurate as the update period increases, and the flow control may transfer too much data to the Node B buffer in time periods, where the wireless link's actual data rate is already much smaller. Due to this effect, we expect the delay performance to worsen as the update period increases. Consider the MobileRR case in Fig. 8. As soon as the update period increases, the tail of the delay ccdf becomes significantly larger. For $T_u = 30$ and 50ms, there is virtually no difference in the delay ccdf for $R_{\mathrm{max,hs}} = 2$ and $R_{\mathrm{max,hs}} = 4$. The delay performance worsens in the case of a PF scheduler, as shown in Fig. 9, since the PF scheduler produces a more variable data flow (cmp. section III-B).

The update period can be kept small by spending more signaling overhead. Alternatively, it can be kept variable, adaptive to the channel characteristic and traffic volume. In contrast, the dead time T_p cannot easily be influenced, as it mainly constitutes of signaling delay and processing overhead.

Figures 10–12 show the ccdf of the delay T_{qa} for the three scenarios and a short update period $T_u = 10$ms. As before, the

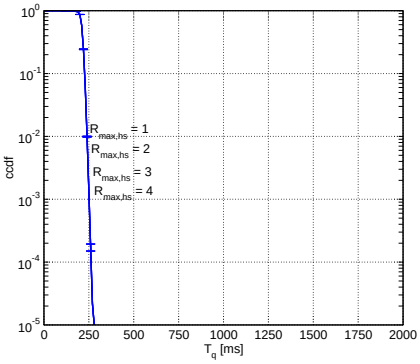

Fig. 4: Influence of $R_{\max,\mathrm{hs}}$ on the delay T_{qa}, StaticRR scenario, $T_p = 0$ms, $T_u = 10$ms

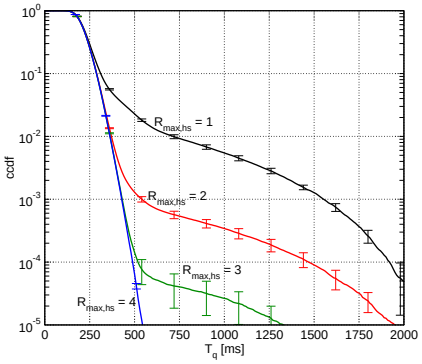

Fig. 5: Influence of $R_{\max,\mathrm{hs}}$ on the delay T_{qa}, MobileRR scenario, $T_p = 0$ms, $T_u = 10$ms

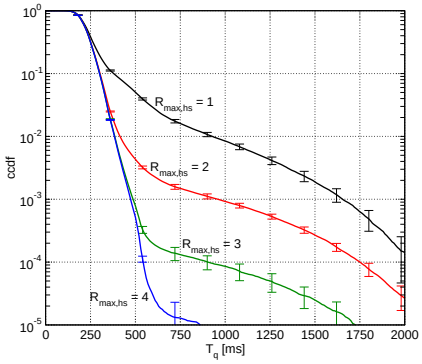

Fig. 6: Influence of $R_{\max,\mathrm{hs}}$ on the delay T_{qa}, MobilePF scenario, $T_p = 0$ms, $T_u = 10$ms

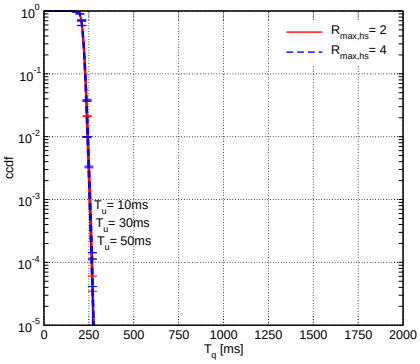

Fig. 7: Influence of T_u on the delay T_{qa}, StaticRR scenario, $T_p = 0$ms

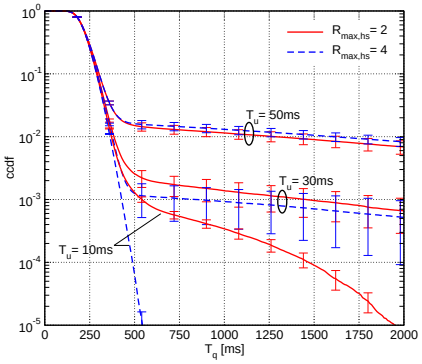

Fig. 8: Influence of T_u on the delay T_{qa}, MobileRR scenario, $T_p = 0$ms

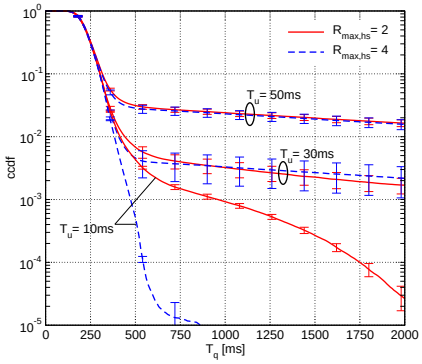

Fig. 9: Influence of T_u on the delay T_{qa}, MobilePF scenario, $T_p = 0$ms

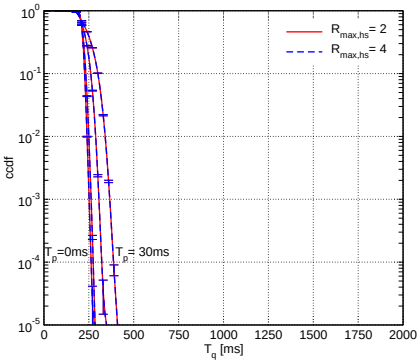

Fig. 10: Influence of T_p on the delay T_{qa}, StaticRR scenario, $T_u = 10$ms

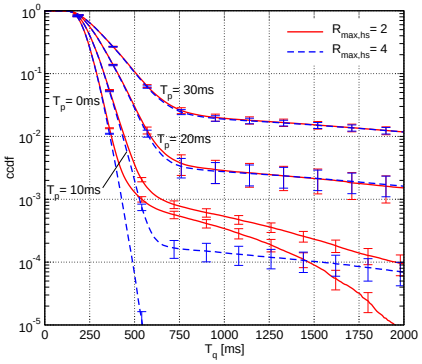

Fig. 11: Influence of T_p on the delay T_{qa}, MobileRR scenario, $T_u = 10$ms

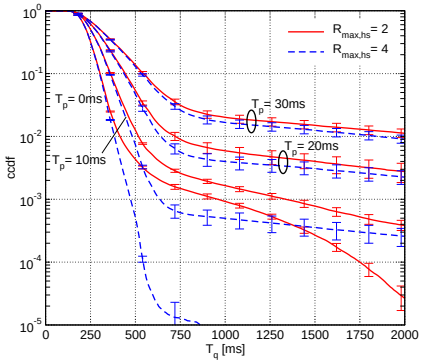

Fig. 12: Influence of T_p on the delay T_{qa}, MobilePF scenario, $T_u = 10$ms

StaticRR scenario proves well-tempered with respect to non-optimal parameters. The reason is again the constant data rate towards each user. In contrast, the MobileRR scenario shows a significant increase in the delay distribution's tail even for a very small dead time of 10ms. The tail becomes dramatic as the dead time increases to 20ms or higher. In these dead time regions, there is almost no difference in the delay distribution as $R_{\max,\mathrm{hs}}$ is increased from 2 to 4. Similar to a large update period, an increasing dead time may lead to the flow control transferring too much data into the Node B buffer, resulting in the large distribution tail. Again, the delay performance worsens as the RR scheduler is substituted with a PF scheduler for the same reasons detailed above.

Figure 13 illustrates the effect of the dead time by plotting the Node B buffer level B and the waiting time of the last transmitted MAC-hs PDU over the time. The figure contains the curves for a dead time of $T_p = 0$ms and $= 30$ms, respectively. The graph unveils that for $T_p = 0$ms, the waiting time of MAC-hs PDUs stays close to the target value of $T_w = 100$ms. On the other hand, a large dead time of $T_p = 30$ms results in strong buffer level fluctuations and strong raises in the buffer fill level, which eventually lead to delay spikes.

The large distribution tails may stall the RNC-Node B data transfer for a relatively long time period. In other words, the service process seen by the RNC data queue may be very bursty. Such a behavior imposes the risk of packet losses at the RNC and Node B queues. Moreover, it may have a severe impact on the performance of the RLC protocol. Altogether, this may have a negative impact on higher protocol layers and brings up the risk of TCP timeouts. In our considered scenarios, we did not observe a significant number of time-outs, and the TCP throughput did not depend on the flow

control parameters, though this might be an issue in more adverse scenarios.

D. Enhancement of the Flow Control Algorithm

In this section, we develop two effective mechanisms to improve the performance of the flow control in mobile environments. In particular, we will try to compensate two effects.

First, as the flow control assigns a resource grant to the RNC, we know that the granted amount of data will arrive at the Node B after the dead time T_p.During this dead time, the flow control keeps monitoring the buffer level B in the Node B, but does not take into account the granted but not yet received data. Hence, too much further data is requested from the Node B. We can tackle this issue by introducing a virtual buffer, as shown in Fig. 2. If the flow control requests a certain amount of data from the RNC, it adds the same amount of virtual data to the virtual buffer, where it remains for the duration of the dead time. The fill level of the virtual buffer is then added to the measurement value B of the actual Node B's buffer level.

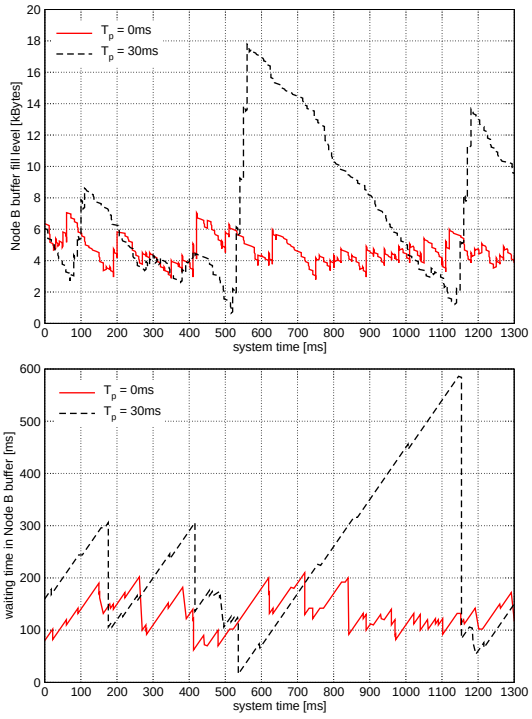

Fig. 13: Node B buffer level (top) and waiting time of last transmitted MAC-hs PDU (bottom), MobileRR scenario, $T_u = 10$ms

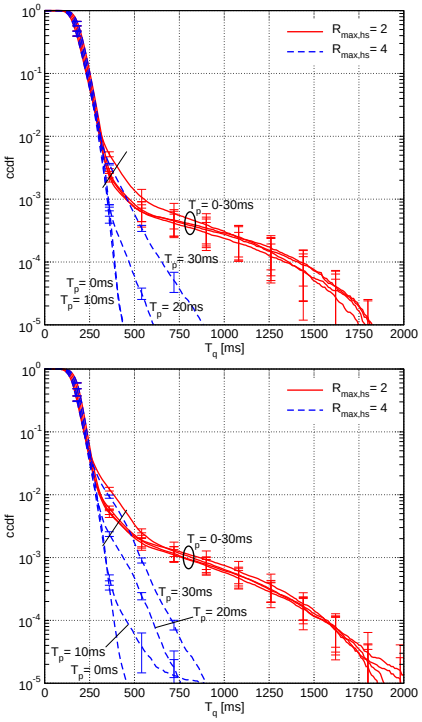

Fig. 14: Enhanced flow control, MobileRR (top) and MobilePF (bottom), $T_u = 10$ms

The second effect which we consider results from the fluctuating data rate towards each terminal. We improve the behavior by limiting the value of R_o in the equations of section III-A to a certain maximum value. For the maximum, we chose the moving average $R_{\mathrm{o,max}}$ of R_o, which is recalculated once within each update period T_u:

$$R_{\mathrm{o,max_{new}}} = (1-\beta)R_{\mathrm{o,max_{old}}} + \beta R_o \qquad (5)$$

This prevents a short period of good channel conditions with a high data rate to flood the buffer. At the same time, it emphasizes the buffer fill level dependent dynamics of the flow control algorithm.

Figure 14 plots the delay ccdf for the MobileRR scenario, if the above discussed enhancements are activated, with $\beta = 0.01$. Consequently, R_o follows only slow fading. The graph unveils a significant improvement in the delay distribution, even for large dead times T_p. The TCP throughput remains unchanged, which indicates, that the improved delay distribution does not go along with buffer under-runs. A similar improvement can be observed for the MobilePF scenario.

V. Conclusion

In this paper, we demonstrated that the flow control mechanism between the RNC and the Node B in an HSDPA system has a significant impact on the performance of the system. Especially in scenarios with fluctuating data rates towards each mobile terminal, the effect of the flow control may be dominating. We showed that the IP packet delay characteristic deteriorates as the control loop's dead time and update period become larger and presented mechanisms to counteract the malicious effect of the dead time. Further studies need to be done to get a deeper understanding of the flow control and the relevant system parameters. Additionally, cross-layer interactions with higher layer protocols and services need to be explored in order to study the user-perceived performance impact, such as the page loading times with web services.

References

[1] R. A. Comroe and D. J. Costello, Jr., "ARQ schemes for data transmission in mobile radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 4, pp. 472–481, July 1984.

[2] P. J. A. Gutiérrez, "Packet scheduling and quality of service in HSDPA," Ph.D. dissertation, Aalborg University, Department of Communication Technology Institute of Electronic Systems, Aalborg University Niels Jernes Vej 12, DK-9220 Aalborg ?st, Denmark, October 2003.

[3] T. Kolding, "Link and system performance aspects of proportional fair scheduling in WCDMA/HSDPA," in *Proc. Vehicular Technology Conference 2003 (VTC 2003-Fall)*, vol. 3, October 2003, pp. 1717–1722.

[4] G. Aniba and S. Aissa, "Adaptive proportional fairness for packet scheduling in HSDPA," in *Global Telecommunications Conference (GLOBECOM 2004)*, vol. 6, December 2004, pp. 4033–4037.

[5] P. J. Legg, "Optimised Iub flow control for UMTS HSDPA," in *Proc. IEEE Vehicular Technology Conference (VTC 2005-Spring)*, Stockholm, Sweden, June 2005.

[6] *IKR Simulation Library*. [Online]. Available: http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/

[7] M. C. Necker and A. Weber, "Parameter selection for HSDPA Iub flow control," in *Proc. 2nd International Symposium on Wireless Communication Systems (ISWCS 2005)*, Siena, Italy, September 2005.

[8] N. Challa and H. Cam, "Cost-aware downlink scheduling of shared channels for cellular networks with relays," in *Proc. IEEE International Performance Computing and Communications Conference (IPCCC)*, Phoenix, AZ, April 2004.