Copyright Notice

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)

PROTOCOL INTERFERENCE BETWEEN UP- AND DOWNLINK CHANNELS IN HSDPA

Marc C. Necker
Institute of Communication Networks and Computer Engineering
University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Email: marc.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 introduction of High Speed Downlink Packet Access (HSDPA) in conventional WCDMA systems according to the UMTS standard will enable packet-switched broadband services with data rates of several Mbps. Until the deployment of High Speed Uplink Packet Access (HSUPA), the uplink traffic will be handled by conventional Dedicated Channels (DCHs). While the HSDPA downlink shared channel (HS-DSCH) is accompanied by fast feedback signaling channels, all feedback signals related to the uplink DCH, such as ARQ status messages, have to be transmitted via the downlink HS-DSCH. In this paper, we argue that the downlink HS-DSCH is subject to unpredictable delays, causing malicious interference with uplink protocol mechanisms, such as the ARQ. Eventually, we show how this degrades the overall end-to-end performance, and we propose and evaluate mechanisms how to alleviate this problem.**

I. Introduction

In order to facilitate high speed packet-switched data services in third generation WCDMA mobile networks according to the UMTS standard, High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA) are currently being standardized. HSDPA is already in a very mature state, and first commercial systems are expected to be deployed in 2006. In contrast, the development of HSUPA is still in a very early state, and it is expected that all data traffic in the uplink direction has to be handled by regular Dedicated Channels (DCHs) in the near future.

HSDPA achieves high data rates of up to 14 Mbps by means of adaptive modulation and coding, fast scheduling mechanisms and a powerful Hybrid ARQ mechanism [1]. All these mechanisms require a fast reaction to events and changing conditions on the radio link. As a consequence, the HSDPA functionality is realized by an additional layer, namely the MAC-hs layer, which is implemented close to the air interface in the Node B. This layer is underneath the well-known MAC-d and Radio Link Control (RLC) layers, which are implemented in the Radio Network Controller (RNC).

The distribution of functionality between the RNC and the Node B requires two separate data buffers in the RNC and the Node B, respectively. A flow control mechanism is used in order to transfer data from the RNC's buffer to the Node B's buffer. This introduces an additional delay between the RNC and the User Equipment (UE), which increases the Round Trip Time (RTT) of the RLC protocol. In [2], we showed that this delay may be significant and in the order of several hundred milliseconds. Additional delay in the HS-DSCH is introduced by the scheduler in the Node B. Especially in a multi-service scenario, higher priority traffic may temporarily push away lower priority traffic, leading to unpredictable delay spikes for lower priority downlink traffic.

While the HSDPA High Speed Downlink Shared Channel (HS-DSCH) recovers from lost radio frames mostly with its HARQ mechanism on the MAC-hs layer, the uplink DCH solely relies on the RLC ARQ mechanism. RLC layer status reports, which signal the (non-)successful reception of radio blocks in the uplink, have to be transmitted on the HS-DSCH. Due to the effects described above, these status reports may experience a significant delay, which degrades the uplink ARQ mechanism's performance. The delay of user data in the uplink will increase, eventually leading to an increased overall Round Trip Time (RTT) and a degradation of the service performance.

In this paper, we study the delay performance of the HS-DSCH in a multi-service environment and investigate its influence on the uplink DCH performance. Moreover, we study the impact on the overall service performance at the example of a TCP file transfer. We demonstrate that the above described effects cause malicious cross-layer interactions with TCP, and we will describe and evaluate means to alleviate this problem and improve service performance.

Our paper is structured as follows. In section II, we present a detailed description and evaluation of all relevant MAC-hs mechanisms and the resulting problems. In the same section, we describe two alternative approaches to overcome the identified problems. We evaluate the performance of the basic and enhanced HSDPA system in section III and conclude our paper in section IV.

II. HSDPA MAC-hs Mechanisms

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

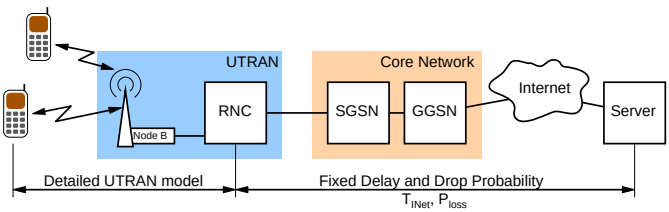

Fig. 1: Architecture of the considered 3G network

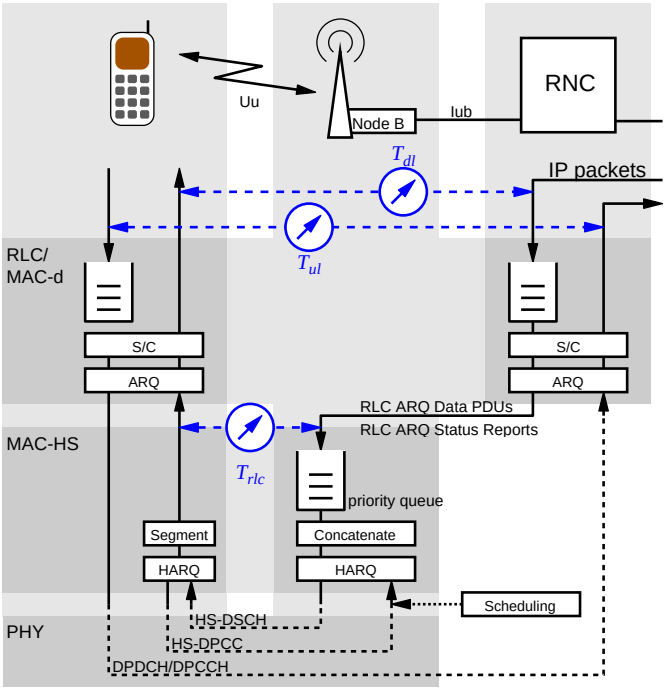

Fig. 2: HSDPA network model for a single user

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 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. HSDPA Protocol Overview

HSDPA offers packet switched communications, where all users are time and code multiplexed onto a single High Speed Downlink Shared Channel (HS-DSCH). The queuing model for a single user is shown in Fig. 2. IP packets arriving at the Radio Network Controller (RNC) are first stored in an input buffer before they are segmented into radio blocks and transmitted to the Node B via the Iub interface. Additionally, the RLC layer may protect the transmitted radio blocks with an ARQ mechanism if the connection is to be operated in Acknowledged Mode (AM). The transmission of data on the Iub-interface is regulated by a flow control mechanism [2].

In the Node B, the radio blocks are stored in *priority queues*, before they are concatenated to larger MAC frames, protected by the HARQ mechanism and finally transmitted on the air interface. In a multi-user scenario, a scheduler in the Node B has to decide which users are allowed to transmit in each scheduling round. Among others, this decision may depend on the current channel state towards each terminal (channel aware scheduling), on the QoS parameters of each connection (QoS aware scheduling), or on a combination of both.

In a multi-service scenario, connections are usually classified into one of several traffic classes with different QoS requirements and individual QoS parameters, such as maximum packet delay or maximum allowed packet loss. The UMTS standard defines four QoS classes [3], which mainly differ in the way they trade off delay sensitivity and reliability. The *conversational class* and the *streaming class* are foreseen to handle real-time traffic with different restrictions on the maximum packet delay bound. In contrast, the *interactive class* and the *background class* do not guarantee any maximum delays but rather emphasize reliable data transport.

A QoS aware scheduler has to give priority to real-time traffic. This can be done in several ways. In [4], we compared several basic approaches to realize service differentiation in HSDPA networks. One of the main conclusions of [4] was that deadline based schedulers are well suitable to meet soft real-time requirements, as for example required by the streaming traffic class, but fail to meet hard real-time requirements needed by the conversational class. As a solution, a combination of static prioritization and a deadline based scheme was proposed, denoted with *static Channel Dependent Earliest Deadline Due (static CD-EDD)*. This scheduling scheme met the QoS-requirements of the real-time traffic classes, while still delivering good performance to non-real-time traffic.

It is well known that channel-aware schedulers may lead to large inter-scheduling gaps, as it was shown for the Proportional Fair (PF) scheduler in [5]. These gaps are due to channel fades belonging to the affected connection and may impose problems to traffic which is sensitive to delay spikes, such as TCP traffic. If a combined QoS- and channel-aware scheduler is used, large inter-scheduling gaps may also be caused by channel-fades of connections with higher priority traffic. In this case, the scheduler needs more air interface resources to meet the QoS requirements of higher priority traffic, leaving no room for background traffic. Compared to the first described effect, the latter effect has gained little attention so far. In the following, we will show by means of simulation, that this effect may be quite severe.

Figure 3 shows the complementary cumulative distribution function (ccdf) of the inter-scheduling time in the Node B for both a streaming traffic connection and a background traffic connection in a multi-service scenario as described later in section III.B. While the streaming user enjoys very small inter-scheduling intervals far below $100\,\mathrm{ms}$ in most cases, the background user suffers from significantly larger inter-scheduling gaps.

The inter-scheduling interval itself is only an indication for the delay T_{rlc} of radio blocks from RNC to UE or the delay of whole IP packets T_{dl}, as these delays are affected by other system aspects as well. In particular, the radio block delay T_{rlc} additionally depends on the number of HARQ retransmissions and on the queuing delay in the Node B. While the delay due to retransmissions is usually below $100\,\mathrm{ms}$, we have shown in [2] that the queuing delay in the Node B can be up to an order of magnitude higher due to effects of the Iub flow control. Fig. 4 plots the ccdf of T_{rlc}, revealing a significant delay tail especially for the background traffic.

Last but not least we have to keep in mind that an IP packet is usually composed of several radio blocks, which amplifies the just described delay effects. Fig. 5 shows the inter arrival time (IAT) of IP packets in the UE, revealing a severe tail of the background traffic's IAT ccdf. Such a tail is the reason

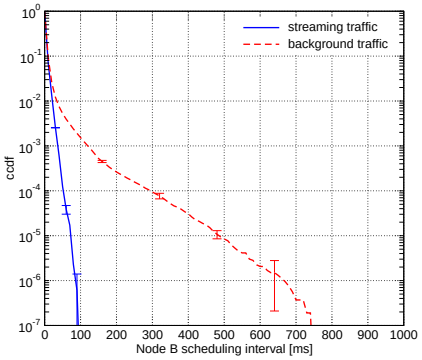

Fig. 3: ccdf of the inter-scheduling time in the Node B

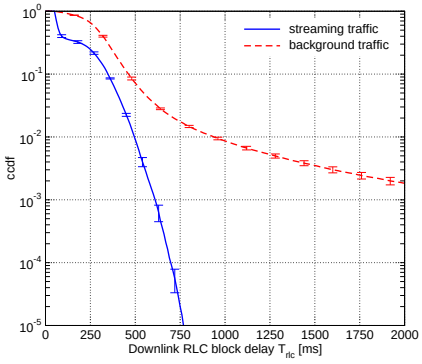

Fig. 4: ccdf of the downlink RLC block delay T_{rlc}

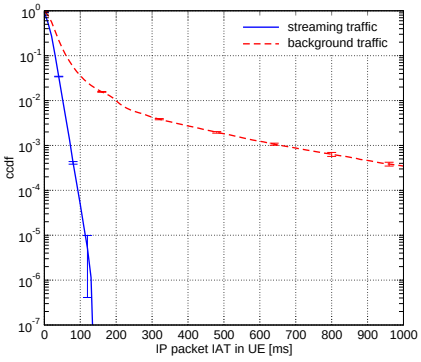

Fig. 5: ccdf of the IAT of IP packets delivered to higher layers at the UE

for delay jitter and may potentially cause timeouts for TCP connections [6].

C. Downlink HS-DSCH and Uplink DCH Interference

While traffic in the downlink is protected from frame losses by the very fast HARQ mechanism, the uplink traffic solely relies on the RLC layer's ARQ mechanism. The status reports of the uplink ARQ, which carry information on successfully and unsuccessfully received radio blocks, have to be transmitted via the downlink HS-DSCH. Consequently, they suffer from the same potentially long radio block delay T_{rlc} described in the previous section. This leads to an increased delay of uplink retransmissions. Eventually, this causes an increased delay in the uplink and an increased Round Trip Time (RTT) experienced by the connection.

Figure 6 shows two new proposals how to overcome this problem. The figure sketches the downlink part of the model introduced in Fig. 2. Instead of a simple FIFO queue on the MAC-hs layer, the figure shows two alternatives for prioritizing uplink RLC status reports in the Node B. In both cases, two separate FIFO queues are used, where a demultiplexer distributes radio blocks carrying an uplink RLC status report to one queue, and all remaining blocks to the other queue. In the left alternative, both queues form a virtual priority queue (virtual priority queue scheme). For the Node B scheduler, this virtual queue acts like a single priority queue. Within the virtual queue, radio blocks carrying uplink status PDUs are always given priority over all remaining radio blocks.

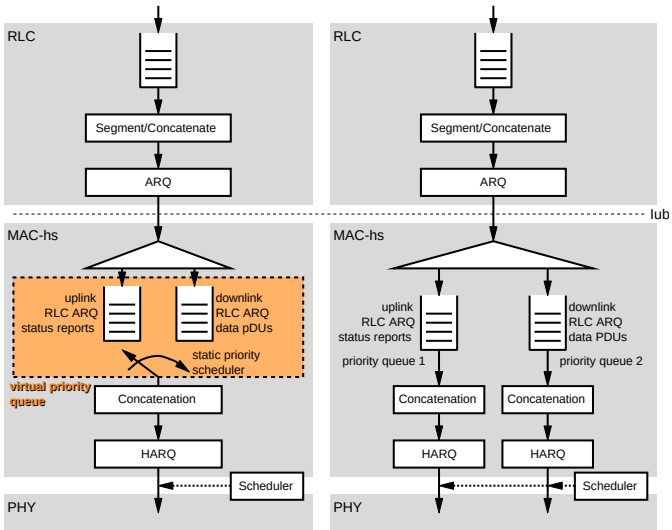

Fig. 6: Prioritization of uplink status reports in the Node B

In the second alternative illustrated on the right side of Fig. 6, the demultiplexer distributes the incoming data blocks into two separate priority queues depending on whether they contain status reports or not (dual priority queue scheme). In contrast to the first alternative, both queues act as separate priority queues towards the Node B scheduler. This allows us to prioritize status reports even over transmissions from other connections (e.g., from real-time connections), which is not possible in the virtual priority queue scheme. For the performance evaluation in section III, we will prioritize the status-reports of the background class over all other transmissions.

Both proposals are technically feasible. The RLC sets the Scheduling Priority Indicator (SPI), which consists of 4 Bit. Consequently, 16 different priorities can be distinguished. The SPI is sent to the Node B together with the corresponding payload. Although the Node B Application Part (NBAP) standard [7] specifies these priorities, it is vendor-specific how the different priorities are handled by the Node B.

D. Influence of Iub flow control

As discussed in [2], signaling delays and measurement inaccuracies of the Iub flow control may lead to long queuing delays in the Node B. If the flow control detects a buffer congestion in the Node B, it will not allow any further radio blocks to be transmitted from the RNC until the congestion has cleared. This will also block uplink RLC status report messages, not allowing them to benefit from the prioritization in the Node B. For this reason, when one of the prioritization schemes is active, we will override the flow control decision in such a way that we will always allow the transmission of uplink RLC status report messages from the RNC to the Node B regardless of the Node B buffer fill level.

III. Performance Evaluation

A. Simulation Model

The HSDPA network was modeled with all its relevant RLC, MAC-d and MAC-hs protocols. The physical layer was mod-

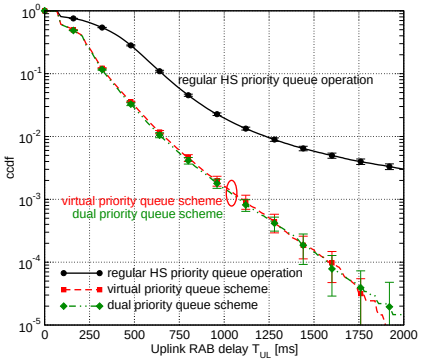

Fig. 7: ccdf of the uplink delay T_{UL} for an uplink BLER of $P_{\mathrm{L,UL}} = 0.2$

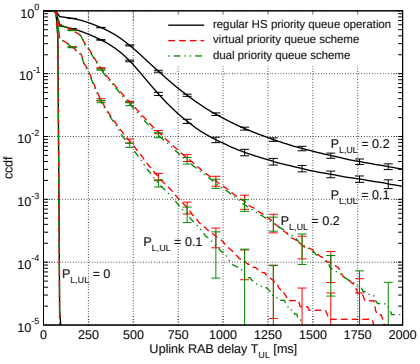

Fig. 8: ccdf of the uplink delay T_{UL} for different values of $P_{\mathrm{L,UL}}$

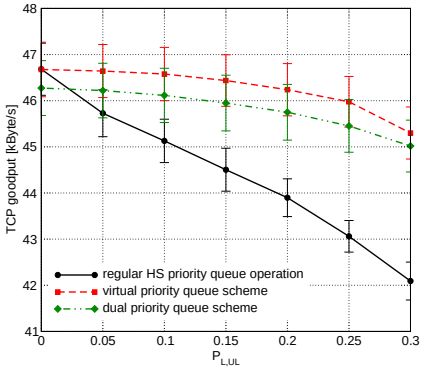

Fig. 9: TCP goodput over $P_{\mathrm{l,UL}}$ for the different Node B queuing options

eled including the HARQ, based on BLER-curves obtained from physical layer simulations. Transport Formats (TF) on the MAC-hs layer were selected based on the channel quality such that the BLER is 10% for the first transmission. The physical channels towards all mobiles were modeled with a variable path loss, including slow fading and fast fading. We assumed ideal conditions for the reporting of Channel Quality Indicators (CQI) from the UEs to the Node B, i.e. zero delay and error-free feedback, in order to isolate the performance influence of the scheduling mechanisms. Alike, the Iub flow control between the RNC and the Node B was assumed to operate with no dead time and short update periods in order to avoid side effects from the flow control [2].

The maximum number of MAC-hs retransmissions was limited to $R_{\mathrm{max,hs}}$, and the maximum number of RLC-retransmissions to $R_{\mathrm{max,rlc}} = 10$. We optimized the RLC layer's ARQ parameters such that it delivered good throughput for streaming and background traffic (see section II.B) while limiting the overhead due to unnecessary retransmissions and status reports. We neglect the convergence layer, as it only introduces a very small overhead in a single-cell environment.

B. Traffic Scenario

We reuse the scenario from [4], where we selected one typical application for three of the four QoS classes. In particular, we consider gaming traffic for the conversational class, streaming video for the streaming class, and FTP download for the background class. Gaming traffic was modeled according to the model presented in [8]. Since we consider the downlink direction, we increased the data rate over that from [8], where only the uplink was considered, to obtain a ten times higher load with a mean data rate of about 2.5 kBps. Streaming video traffic was modeled by a constant rate source with a data rate of 39 kBps and a packet size of 576 Bytes. The FTP traffic model was based on a greedy traffic source. In accordance with recent measurements [9], the underlying TCP sender and receiver pair employed the NewReno algorithm with a limited receiver advertised window of 64 kByte.

For our scenario, we consider ten UEs with one active data flow each, moving at a velocity of $v = 30$ km/h. The ten UEs break down into five UEs running a gaming application, two UEs running a streaming application, and three UEs performing an FTP download. The RLC layer is configured in AM for the gaming and FTP flows, and in Unacknowledged Mode (UM) for the streaming traffic. A timer mechanism deletes all packets in the streaming flows' RLC input queues with a waiting time larger than 2 s. In the Node B, the already mentioned static CD-EDD scheduling approach [4] was used.

C. Uplink delay

We first investigate the uplink IP packet delay T_{UL} for the different prioritization schemes. Figure 7 plots the ccdf of T_{UL} for an uplink block error probability of $P_{\mathrm{L,UL}}$ of 0.2. Shown are three ccdfs for regular HS priority queue operation and for the two alternative schemes described in section II.C. While the difference between the ccdfs of the two described schemes is only minor and within the error bars, the gain compared to the regular HS priority queue operation is significant.

Fig. 8 shows the same ccdfs for different uplink block error probabilities $P_{\mathrm{L,UL}}$. The minimum delay can be observed for $P_{\mathrm{L,UL}} = 0$. For larger $P_{\mathrm{L,UL}}$ the delay quickly increases, while the general behavior of the alternatives remains the same.

D. TCP background performance

As the second performance metric, we study the total goodput achieved by a single background TCP connection in the multi-service scenario. Figure 9 plots the TCP goodput over the uplink block error probability $P_{\mathrm{L,UL}}$. In principle, the only data to be transmitted in the uplink are the TCP acknowledgments. With a gross data rate of 144 kbps, the uplink DCH has more than enough capacity to carry these acknowledgments, even at a block error probability of $P_{\mathrm{L,UL}} = 0.3$.

From Fig. 9 we can see that the TCP goodput decreases almost linearly with $P_{\mathrm{L,UL}}$ for regular HS priority queue operation. This is mainly caused by two effects. The first reason is that the average bandwidth delay product of the system is in the order of the maximum TCP sender transmit window size of 64 kByte. For certain time periods, the bandwidth delay product may be smaller or larger, where the latter case will lead to an underutilized data link and a lower TCP throughput.

The second effect is illustrated in Fig. 10, where the number of TCP timeouts per transmitted Maximum Segment Size

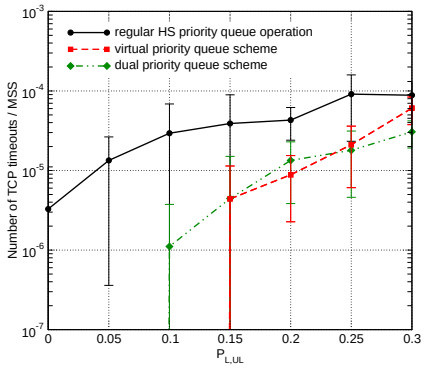

Fig. 10: Number of TCP timeouts per MSS over $P_{\mathrm{L,UL}}$

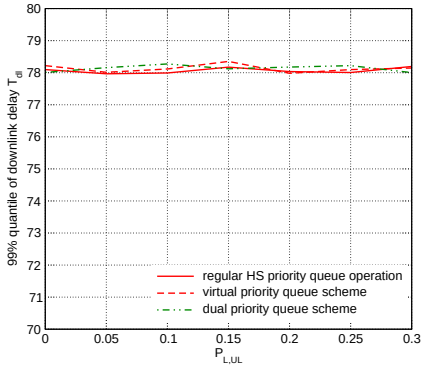

Fig. 11: 99% quantile of gaming traffic downlink delay T_{dl}, $P_{\mathrm{L,UL}} = 0.2$

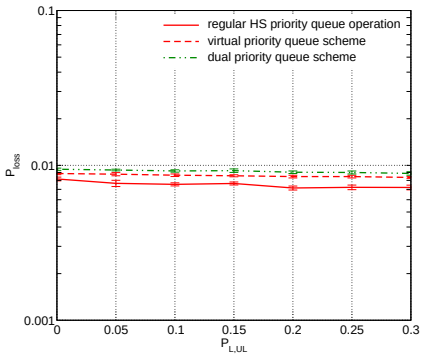

Fig. 12: P_{loss} of IP packets within a streaming connection, $P_{\mathrm{L,UL}} = 0.2$

(MSS)[1] is plotted over $P_{\mathrm{L,UL}}$. The chart reveals an increasing amount of TCP timeouts as $P_{\mathrm{L,UL}}$ increases. However, we should note that the number of TCP timeouts is fairly low. Hence, this is the less severe effect impacting TCP performance.

The virtual priority queue scheme and the dual priority scheme both achieve a significant improvement of the TCP goodput. Compared to the regular HSDPA operation, a much smaller decrease of the TCP goodput can be observed with increasing $P_{\mathrm{L,UL}}$. The reason is the mitigation of the two above described effects, which is confirmed by the delay analysis of section III.C and the number of TCP timeouts in Fig. 10.

When comparing both investigated alternatives for prioritization on the MAC-HS layer, we can record that a prioritization of uplink RLC status reports within a connection improves the system performance, while an additional prioritization over other connections is slightly worse. The reason is that a prioritization over other connections overrides the channel-aware scheduling term of the static CD-EDD scheduler, leading to a less efficient channel utilization.

Last but not least it is important to observe the behavior of the higher priority traffic as we improve the performance of the background traffic. In accordance with [4], we compare the 99% quantile of the downlink delay T_{dl} for the gaming traffic. For the streaming traffic, we compare the loss probability P_{loss} of IP packets due to buffer overflows, transmission errors or deadline violations within the radio access network (RAN).

Both metrics are shown in Fig. 11 and 12, respectively. The quantile of the gaming traffic shows no degradation of the gaming traffic's QoS at all. Due to the static prioritization of the gaming traffic, only the dual priority queue scheme has the potential to harm the gaming traffic's QoS at all. For the streaming traffic, the loss probability P_{loss} shows a minor increase by approximately 0.1% for the preferred virtual priority queue scheme. This degradation is due to the increased amount of data in the background class, and can be observed for both discussed HS priority queue schemes.

It is important to note that the described problems could also be alleviated by increasing the maximum TCP transmission window size by the use of TCP window scaling [10]. However, this is a parameter that cannot be influenced by the RAN, and it is expected that it will still take some time until TCP window scaling is widely deployed (cmp. measurements in [9]), while the presented RAN-based solution can easily be deployed by the operator.

[1]The MSS was set to 1500 Bytes.

IV. Conclusion

In this paper, we investigated the interference between the downlink HS-DSCH and the uplink DCH in a multi-service HSDPA environment. We showed that downlink delay spikes negatively affect the performance of the uplink DCH and presented two prioritization schemes to overcome this problem. In order to evaluate the performance of the original and the enhanced systems, we performed a cross-layer analysis with gaming, streaming and TCP traffic. As a main conclusion we note that the proposed prioritization schemes can significantly improve the performance of background traffic while having only a minor impact on higher priority traffic.

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] M. C. Necker and A. Weber, "Impact of Iub flow control on HSDPA system performance," in *Proc. 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005)*, Berlin, Germany, September 2005.

[3] 3GPP TS 23.107, *Quality of Service (QoS) concept and architecture (Release 6)*, 3rd Generation Partnership Project, June 2005.

[4] M. C. Necker, "A comparison of scheduling mechanisms for service class differentiation in HSDPA networks," *International Journal of Electronics & Communications*, vol. 60, no. 2, pp. 136–141, Feb. 2006.

[5] T. E. Klein, K. K. Leung, and H. Zheng, "Enhanced scheduling algorithms for improved TCP performance in wireless IP networks," in *Proc. IEEE GlobeCom*, Dallas, TX, December 2004.

[6] M. Scharf, M. C. Necker, and B. Gloss, "The sensitivity of TCP to sudden delay variations in mobile networks," in *Proceedings of the 3rd IFIP-TC6 Networking Conference, Lecture Notes in Computer Science (LNCS) 3042*, Athens, Greece, May 2004, pp. 76–87.

[7] 3GPP TS 25.433, *UTRAN Iub Interface NBAP Signalling (Release 6)*, 3rd Generation Partnership Project, December 2005.

[8] R. Bangun and E. Dutkiewicz, "Modelling multi-player games traffic," in *Proc. International Conference on Information Technology: Coding and Computing (ITCC 2000)*, 2000.

[9] A. Medina, M. Allman, and S. Floyd, "Measuring the evolution of transport protocols in the internet," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 2, pp. 37–52, April 2005.

[10] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," IETF, RFC 1323, May 1992.