IP-Packet Service Time Distributions in UMTS Radio Access Networks

Arthur Mutter, Marc C. Necker, Stephan Lück

Abstract— **This work deals with the service time of IP-packets within the UMTS Terrestrial Radio Access Network (UTRAN). The focus is on the influence of the Radio Link Control (RLC) Layer's ARQ-mechanism with respect to the various parametrization possibilities. The service time of IP packets is evaluated by means of pdf and ccdf functions. Beside the discussion of the optimal parameter choice, the service time statistics are linked to TCP performance results. It will be shown that even a good parameter choice may lead to long service times, and that the shortest possible service time does not necessarily lead to the best TCP performance.**

Keywords— **UMTS, WCDMA, ARQ, BEC, TCP**

I. Introduction

Quality of Service (QoS) is a well studied topic. From the technological point of view, it is very well possible to provide QoS in most fixed networks, since, among other things, wireline links are very reliable [1]. In contrast, providing QoS in cellular networks is a much more complicated task. It is well known that a wireless link is unreliable, and the loss probability for a transmitted data frame may be very high. Moreover, the delay on a wireless link in a cellular network is large and on the order of the delay of a trans-Atlantic connection. As a consequence, the application of error correction mechanisms is not a straight-forward task.

The Universal Mobile Telecommunication System (UMTS) is a third generation cellular network based on Wideband Code Division Multiple Access (WCDMA). For data traffic, UMTS applies both Forward Error Correction (FEC) and Backward Error Correction (BEC) to compensate for bit-errors on the wireless link. Even though the UMTS standard foresees sophisticated FEC schemes, such as Turbo-Codes (see for example [2]), the residual loss probability for a data frame on the MAC layer is likely to be on the order of 10-20%. Consequently, an Automatic Repeat Request (ARQ) mechanism is applied to correct these residual errors. ARQ is a well-studied concept and was investigated in a number of publications (see for example [3], or [4] with respect to radio networks).

In UMTS, ARQ takes place within the Radio Link Control (RLC) layer, which is specified in [5]. In contrast to the ARQ schemes applied in second generation cellular systems, [5] defines a vast number of mechanisms in order to detect the loss of data frames and control the request of a retransmission. The impact of these mechanisms on the performance of the commonly used Transmission Control Protocol (TCP) were studied in [8] and [9]. Additionally, the authors investigated the mean delay of the data packets.

In [10], the authors analyze the packet delay in more detail using probability density functions for a number of different scenarios. This was done in a multi-user single-cell environment for both the UMTS Downlink Shared Channel (DSCH) and the UMTS Dedicated Channel (DCH). However, only a generic ARQ scheme was applied, leaving the influence of the powerful UMTS ARQ mechanisms open.

In this paper, we study the impact of the UMTS ARQ parameter selection on the IP-packet service time in detail. We investigate the influence of the parameter selection by means of probability density functions (pdf) and complementary cumulative distribution functions (ccdf) of the IP-packet service time. Additionally, we link these results to the performance of a TCP data connection.

The structure of this paper is as follows: In section II, we present the system model. In section III, we describe the simulation scenario. Section IV presents an analytical approximation of the IP packet service time and discusses the results of the simulation study.

II. System Model

A. Overview

The simulation scenario is shown in Fig. 1. The UMTS User Equipment (UE) is connected to the Radio Network Controller (RNC) of the UTRAN via a Dedicated Channel (DCH) in both up- and downlink direction. The RNC is connected to the Internet via the 3G-SGSN and the 3G-GGSN of the cellular system's core network. Finally, the UE establishes a data connection with a server connected to the Internet. Based on the analysis in [7], the delays given in Fig. 1 were used for all simulations.

The simulation model for this scenario is shown in Fig. 2. It consists of a TCP Reno source which generates IP traffic, the RLC layer, transport and logical channels and the physical layer. The RLC layer is modeled according to the respective standard [5] and

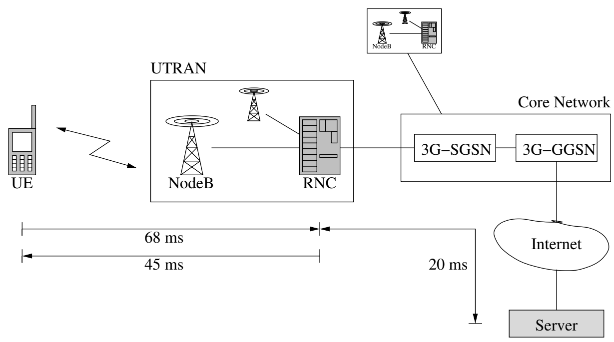

Fig. 1. Simulation scenario

The authors are with the Institute of Communication Networks and Computer Engineering, University of Stuttgart, Germany. E-Mail: welcome@arthur-mutter.de, {necker|lueck}@ikr.uni-stuttgart.de

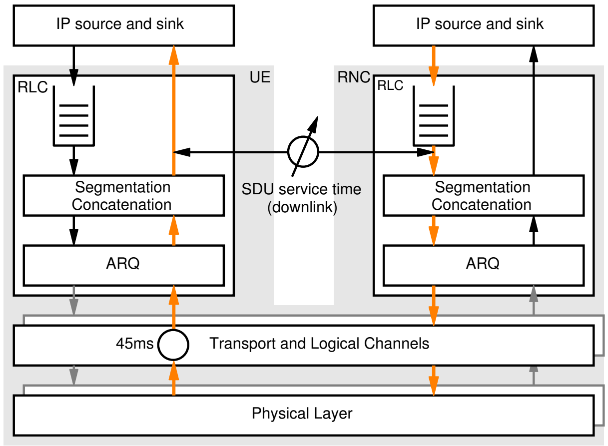

Fig. 2. Simulation model

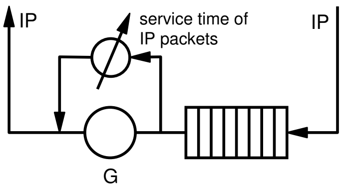

Fig. 3. G/G/1 delay-loss system

is described in section II-C. The modeling of the erroneous physical layer is described in section II-D.

Since we consider a single cell system, the UMTS convergence layer is transparent except for some overhead. This overhead is very small and will be neglected. In this case, a Service Data Unit (SDU) of the RLC layer corresponds to an IP-packet, and the RLC-layer SDU delay essentially is the IP-packet service time of the Radio Access Network. Hence, our model can also be represented by a G/G/1 delay-loss system as shown in Fig. 3, where the server models the service time of an IP-packet. Note that this service time follows a general distribution, and the service time of a packet depends on the service time of the previous packets.

B. Modeling of higher layers

The traffic is generated by a greedy source and fed to the TCP layer, which performs flow control. The TCP stack was configured such that all TCP segments have a constant length of 1500 Bytes. If the waiting time in the transmission queue of the RLC layer is not considered, all delay statistics also apply to general IP-traffic with 1500 Byte long packets.

C. Modeling of RLC layer

C.1 Basic concepts

The RLC realizes a Selective Repeat ARQ scheme with additional features to improve performance. Packets arriving from higher layers (RLC SDUs) are segmented or concatenated to RLC PDUs of fixed length, also called Radio Blocks. The RLC entity transmits the RLC PDUs and receives status reports from its peer entity, which contain positive or negative acknowledgments.

The transmission of status-reports can be sender or receiver driven. Sender driven means, the sender transmits a Poll-PDU to request a status-report. A Poll-PDU is any RLC PDU with the Poll-Bit set in its header. Upon reception of a Poll-PDU the receiver generates a status-report. With receiver driven status-reporting, the receiver decides independently when to send a status-report. This decision is based on timers, detection of missing PDUs and other mechanisms. Hybrid status-reporting, where sender and receiver driven methods are combined, use the advantages of both to achieve better performance.

C.2 Modeling

The RLC layer offers three transfer modes: Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM). We restrict our investigation to AM, since this is the only mode of interest for connecting to the Internet [11]. The main tasks of the RLC layer in AM are:

- segmentation & concatenation of upper layer SDUs
- buffering of packets in the transmission buffer
- performing ARQ functionality

The respective model components are shown in Fig. 2. The 3GPP RLC specification [5] foresees the transmission buffer after segmentation and concatenation. However, from a modeling point of view, it is more convenient to place this buffer before segmentation and concatenation in order to store complete RLC SDUs (i.e. complete IP packets). This does not affect the system behavior, since we assume the functional units within the RLC entity to have zero delay.

D. Modeling of MAC layer

Since we are interested in the impact of the RLC parameter choice, we consider a separate data connection in a single cell system and restrict our analysis to the use of dedicated channels (DCHs) in both up- and downlink direction.

On the MAC-layer, RLC PDUs are grouped into RLC block sets, where the maximum number of RLC PDUs within an RLC block set is $L_{\max}$. A data frame on the MAC layer contains one RLC block set. Its length is specified by the Transmission Time Interval (TTI). In our simulations, we used the reference radio bearers as defined in [6, Chapter 6.10.2.4.1.31], which use a 256kbit-DCH in the downlink and a 64kbit-DCH in the uplink direction. The TTIs are 10ms and 20ms in down- and uplink direction, respectively.

For the performance of the ARQ mechanism, the Round Trip Time (RTT) on the MAC-layer is an important system characteristic. In the following, we will determine the RTT in both uplink and downlink direction. Because of the different TTIs in up- and downlink, we will obtain two different RTTs, one for the path UE–RNC–UE and one for the path RNC–UE–RNC. Note that our simulation system synchronizes the beginning of data frames in downlink and uplink direction, as it is illustrated in Fig. 4.

The path UE–RNC–UE corresponds to the minimum time it takes from a retransmission request of the UE until the reception of the corresponding retransmission. The calculation of $RTT_{\mathrm{UE-RNC-UE}}$ is

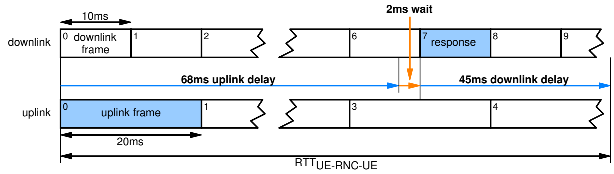

Fig. 4. RTT on the path UE–RNC–UE

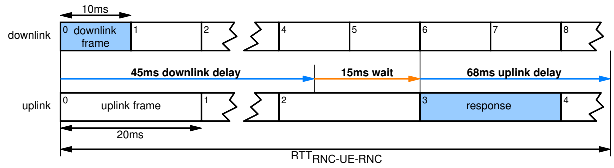

Fig. 5. RTT on the path RNC–UE–RNC

illustrated in Fig. 4: The RNC receives a status-report of the UE 68ms after its transmission. The earliest time for the transmission of a response is the beginning of the next downlink TTI, which is after 2ms:

$$RTT_{\mathrm{UE-RNC-UE}} = \underbrace{68\mathrm{ms}}_{\text{UL delay}} + \underbrace{2\mathrm{ms}}_{\text{wait next TTI}} + \underbrace{45\mathrm{ms}}_{\text{DL delay}} = 115\mathrm{ms}\ . \quad (1)$$

The calculation of the RTT in the opposite direction is illustrated in Fig. 5. Since the uplink TTI is twice as large as in downlink direction, $RTT_{\mathrm{RNC-UE-RNC}}$ has a minimum and a maximum value. 45ms after its transmission, the UE receives a data frame from the RNC, which might for example request the transmission of a status report. The earliest transmission time for a response is the beginning of the next TTI in uplink direction, which might be after 5ms or 15ms:

$$RTT_{\mathrm{RNC-UE-RNC}} = \underbrace{45\mathrm{ms}}_{\text{DL delay}} + \underbrace{5(15)\mathrm{ms}}_{\text{wait next TTI}} + \underbrace{68\mathrm{ms}}_{\text{UL delay}} = 118(128)\mathrm{ms}\ . \quad (2)$$

E. Modeling of the physical layer

The physical layer was modeled according to a random drop model, where each transmitted data frame is independently marked as lost with a probability of $P_{\mathrm{loss,DL}}$ and $P_{\mathrm{loss,UL}}$ in the downlink and uplink direction, respectively. $P_{\mathrm{loss,DL}}$ was chosen twice as high as $P_{\mathrm{loss,UL}}$ in order to account for the higher data rate of the downlink channel.

F. Performance metrics

The following metrics were used for the performance evaluation:

- **IP-packet service time:** service time of an IP-packet in downlink direction as defined in section II-A.
- **Throughput:** total throughput on the TCP layer accounting for TCP overhead and all retransmissions.

III. Simulation Scenario

The RLC settings of the three considered schemes are listed in table I. In all schemes, the RLC buffer size

TABLE I

RLC schemes configuration parameters

ARQ scheme	sender	receiver	hybrid
Poll Last PDU in Buffer	on	off	on
Poll Last PDU in Retr. Buf.	on	off	on
Poll-PDU	150	off	150
Poll Periodic	off	off	off
Poll Timer	140 ms	off	140 ms
Poll Window	75%	off	75%
Missing PDU indicator	off	on	off
Status Periodic	2000 ms	600 ms	2000 ms
Timer_Poll_Prohibit	155 ms	off	off
Timer_Status_Prohibit	off	120 ms	120 ms
EPC	off	on	on
MaxDAT	5	5	5
RX & TX Window size	512	512	512

is 50kBytes, and the RLC PDU payload is 40 Bytes according to [6, Chapter 6.10.2.4.1.31]. In particular, we consider one sender-driven, one receiver-driven and one hybrid scheme. Commonly used parameters are *MaxDAT*, RX- and TX Window size. The value *MaxDAT*−1 limits the unsuccessful transmission attempts for a PDU. *TX Window* limits the number of unacknowledged PDUs in the sender. *RX Window* is the reception window at the receiver side and limits the highest sequence number accepted.

In the following, the RLC settings will be discussed.

- **Sender driven scheme:** This scheme only configures sender driven polling functions. A polling function is a function that triggers the transmission of a Poll-PDU. Polling functions react on special events, like the delivery of the last available PDU, expiry of timers or counters. The *Last PDU in Retransmission Buffer* polling function triggers a poll, if the PDU transmitted is the last in the retransmission buffer. It takes care that retransmitted PDUs are acknowledged fast and effectively. The *Last PDU in Buffer* function is similar, but monitors the transmission buffer. The *Poll Timer* function ensures that a status-report is received for each Poll-PDU. It starts a timer whenever a Poll-PDU is transmitted. If this timer expires before a status-report is received, a new Poll-PDU is triggered. In combination with the *Last PDU in Buffer* polling function it protects from deadlocks.

The *Status Periodic* function is a receiver driven function and only activated to prevent deadlocks. This is necessary, because the protection of the *Poll Timer* function works only *MaxDAT*−1 times in case the *TX window* has closed. The timer of the *Status Periodic* function was set to the high value of 2000ms in order to keep its influence on performance as small as possible. The *Poll Window* polling function triggers the transmission of a Poll-PDU whenever the TX window grows beyond the given threshold. This floods the receiver entity with Poll-PDUs to preserve the TX window from closing. Finally, the *Poll Prohibit* function is important, because it limits the maximum poll frequency. Too frequent polling may lead to status-report transmissions with redundant information.

- **Receiver driven scheme:** Here, the receiver decides independently when to transmit a status-report. A very important function is the *Missing PDU indicator*. It triggers the transmission of a status-report

if it detects missing PDUs in the data flow.
As in the sender driven scheme, the *Status Periodic* function prevents from deadlocks. Transmitting all 512 PDUs of the *TX Window* takes 640ms if the maximum of $L_{\max} = 8$ RLC PDUs fits into an RLC Block block. If the frame error ratio (FER) is very small, the *Missing PDU indicator* may not trigger any status-reports for a long time. In this case, the *Status Periodic* function with a timer of smaller than 640ms prevents the *TX Window* from closing. Note that the timer should not be chosen too small in order to not interfere with other RLC mechanisms, which are more rational at high FERs.
The *Status Prohibit* function limits the maximum frequency of status-reports to a fixed value. In contrast, the *Estimated PDU Counter (EPC)* function is an intelligent *Status Prohibit* function. The time for which status-reporting is prohibited is a function of the number of negative acknowledged PDUs in the last status-report. A large amount of negative acknowledgments leads to a long prohibit time. If the timer expires and previously negative acknowledged PDUs are still missing, the transmission of a new status-report is triggered and the *EPC* function is restarted. This leads to a very fast and effective status-reporting.

- **Hybrid driven scheme:** the hybrid scheme uses both sender and receiver driven mechanisms combining the benefits of both. Because it is sufficient to prohibit the transmission of status-reports on the receiver side, the *Poll Prohibit* function was turned off. The *Status Periodic* function was set to the value of 2000ms for the same reason as in the sender driven case.

IV. Performance Evaluation

In the remainder of the paper, we compare the different RLC configuration schemes. In section IV-A we derive an analytical approximation of the IP packet service time. In section IV-B, we investigate the influence of the FER.In section IV-C we study the *Status_Prohibit* function. Finally, we compare this function with the *EPC* function in section IV-D.

A. Analytical approximation of IP-packet service time

In this section we determine the minimum IP-packet service time if no data frames are lost. Additionally, we approximate the service time if one RLC block set is lost for the hybrid and receiver driven schemes.

The number of RLC PDUs an IP-packet is segmented into is

$$N_{PDU} = \frac{1500\text{Byte}}{40\text{Byte / PDU}} = 37.5\text{PDUs} \quad . \tag{3}$$

A maximum of $L_{\max} = 8$ RLC PDUs can be transmitted within one TTI in the downlink direction. Consequently, the necessary number of TTIs is:

$$N_{\text{TTI}} = \frac{N_{\text{PDU}}}{L_{\max}} = \frac{37.5\text{PDUs}}{8\text{PDUs / TTI}} = 4.69 \quad . \tag{4}$$

Since the TTI in downlink is 10ms, the minimum time necessary for transmission and reception of a whole IP-packet is:

$$T_{\min} = 45\text{ms} + \lceil 4.69 - 1 \rceil \cdot 10\text{ms} = 85\text{ms} \tag{5}$$

If RLC block sets are lost during transmission, the service time of an IP-packet can only be approximated. This is due to the many RLC mechanisms that impact the service time. In the following, we will approximate the best and worst case service time of an IP-packet if only one RLC block set is lost, which is detected by the *Missing PDU indicator*.

In the best case, the first of the IP-packet's 5 RLC block sets is lost (Fig. 6). After one TTI, the *Missing PDU indicator* function in the receiver recognizes the loss. If not prohibited by the *Status Prohibit* mechanism, a status report is generated immediately. This status report will be transmitted after 5ms when the next TTI in uplink direction starts (cmp. Fig. 5). The sender receives the status report and retransmits the requested PDUs, which in the best case fit into one TTI. These will be transmitted after 2ms when the next TTI in downlink direction starts. Consequently, if one RLC block set is lost, the best case service time T_{best} for an IP-packet yields to

$$\begin{aligned} T_{\text{best}} &= \underbrace{45\text{ms}}_{\text{DL delay}} + \underbrace{10\text{ms}}_{\text{missing PDU}} + \underbrace{5\text{ms}}_{\text{wait next TTI}} + \\ &\quad \underbrace{68\text{ms}}_{\text{UL delay}} + \underbrace{2\text{ms}}_{\text{wait next TTI}} + \underbrace{45\text{ms}}_{\text{DL delay}} \\ &= 175\text{ms} \quad . \end{aligned} \tag{6}$$

The worst case differs from the best case in four points. First, the 5th RLC block set is lost (cmp. Fig. 6), which means that there is an additional delay of 4 TTIs corresponding to 40ms. Second, the transmission of a status-report is prohibited and cannot be made until the expiry of the timer *Timer_Status_Prohibit*, which is 120ms in the receiver driven and the hybrid scheme. Third, the status report has to wait 15ms instead of 5ms until the next uplink TTI starts (cmp. section II-D). Last but not least, the requested PDUs do not fit into one RLC block set due to other PDUs that need to be transmitted first. In other words, some of the requested PDUs have to be transmitted with the next RLC block set in the next TTI, and there is an additional delay of one TTI. Hence, the worst case delay T_{worst} if one RLC block set is lost is:

$$\begin{aligned} T_{\text{worst}} &= T_{\text{best}} + \underbrace{40\text{ms}}_{\text{last TTI lost}} + \underbrace{10\text{ms}}_{\text{wait next TTI}} + \\ &\quad \underbrace{120\text{ms}}_{\text{status proh.}} + \underbrace{10\text{ms}}_{2^{\text{nd}}\text{ TTI}} = 355\text{ms} \quad . \end{aligned} \tag{7}$$

Note that T_{worst} is only an approximation under the assumption that the RLC layer responds perfectly

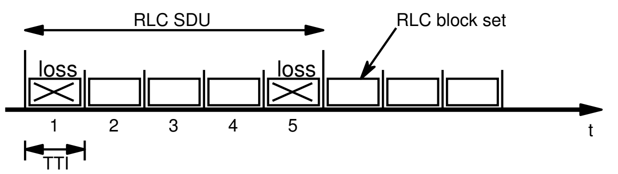

Fig. 6. Best and worst case if one RLC block set of an RLC SDU is lost.

to the data loss. The actual worst case service time under any circumstances is much longer, for example if the RLC block set including the status-report is lost during transmission. In any case, T_{worst} is a useful estimate for the following performance evaluation.

Finally, we define T_{avg} as the average of T_{best} and T_{worst}:

$$T_{\text{avg}} = \frac{T_{\text{worst}} + T_{\text{best}}}{2} = 265\text{ms} \quad . \tag{8}$$

B. Impact of the Frame Error Ratio

In this section, we discuss the impact of the FER on the IP-packet service time. In particular, we consider four different FERs in the downlink direction of $P_{\text{loss,DL}} = 0.01, 0.10, 0.20$ and 0.30. As mentioned above, the FER in uplink direction was set to $P_{\text{loss,UL}} = \frac{1}{2} P_{\text{loss,DL}}$ Fig. 7 - 11 show the pdfs of the IP-packet service time for the considered RLC schemes.

B.1 General observations

All diagrams show a first sharp peak around 90ms, which corresponds to the minimum IP-packet service time $T_{\min} = 85$ms as derived in eq. (5). Since all IP-packets have a constant length, the service time cannot be smaller than $T_{\min}$. This sharp peak decreases in height with increasing FER. Additionally, the notch after the peak increases. The notch is caused by SDUs whose PDUs were received error free but were delayed by higher priority traffic, which mainly consists of retransmissions. The following peaks in the diagrams mark the arrivals of retransmissions which allowed SDUs to be completed and delivered to upper layers. Because of the decrease in data rate the peaks get smoother towards higher service times.

The performance of the sender driven scheme with respect to the service time is always worse compared to the other two schemes. This is because the sender driven scheme usually polls the receiver in regular intervals and cannot directly react on lost PDUs. Hence, the receiver has to wait longer for the retransmission of missing PDUs.

B.2 $P_{\text{loss,DL}} = 0.01$

As it can be seen in Fig. 7 the peak at $T_{\min}$ is very high. This is because nearly no PDU is lost and almost every SDU can immediately be delivered to higher layers at the receiver side.

The receiver driven scheme shows a significant peak between 700ms and 800ms. This effect can be explained as follows. Usually, the *TX Window* is kept open by the *Missing PDU indicator*. However, if the FER is very small, this trigger is useless, since very few PDUs are lost. Consequently, it is the responsibility of the *Status Periodic* function to prevent the *TX Window* from closing. If one of these status-reports in uplink direction is lost, the *TX Window* may close. In this case, the system has to wait 600ms for the next periodic status-report, leading to the higher service time of at least $T_{\min} + 600$ms for some few IP-packets.

Instead of using the *Timer_Status_Periodic* mechanism, the problem that receiver driven schemes experience at low FERs could also be solved by configuring

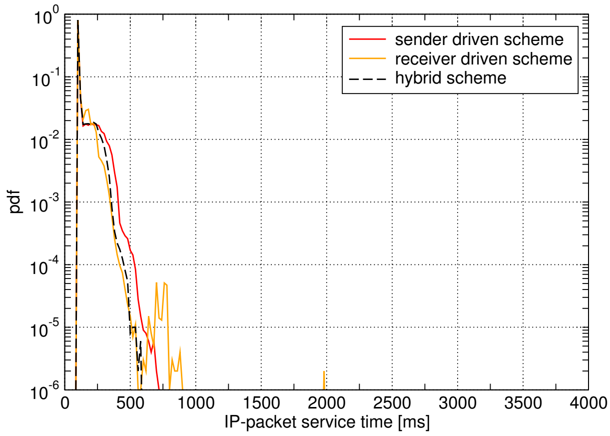

Fig. 7. pdf of the IP-packet service time, $P_{\text{loss,DL}} = 0.01$

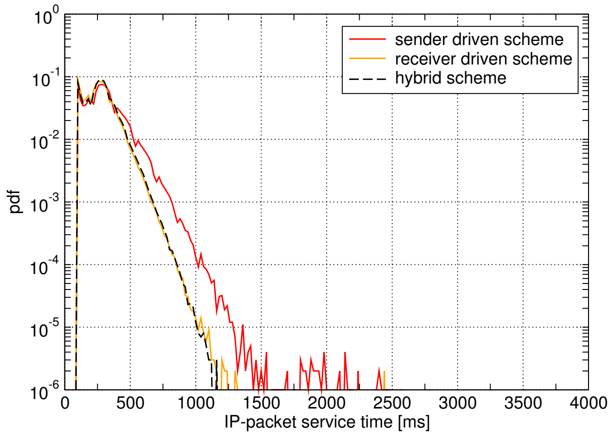

Fig. 8. pdf of the IP-packet service time, $P_{\text{loss,DL}} = 0.1$

the *Poll Window* mechanism on the sender side, as it was done in [8]. This was not done here, since it is a sender driven mechanism. Also, the just described effect is of significance only at very low FERs. Simulations showed that at higher FERs ($P_{\text{loss,DL}} \geq 0.08$) the service time pdfs of the receiver driven scheme with and without the *Poll Window* function are identical.

B.3 $P_{\text{loss,DL}} = 0.10$

Figure 8 shows the same results for an FER of 0.1. Compared to $P_{\text{loss,DL}} = 0.01$, the first peak at $T_{\min}$ is an order of magnitude smaller. We can also observe a second peak around 260ms. Because of the random loss of RLC blocks belonging to a particular IP-packet the peak is smooth. It represents the service time of IP-packets, which needed one retransmission for one or more of their respective RLC blocks. This agrees with our analytical approximation of $T_{\text{avg}} = 265$ms.

From about 600ms to 1300ms, the sender driven scheme shows small spikes in its service time pdf. These have a distance of 160ms, corresponding to the value of the *Poll Prohibit* timer of 155ms[1]. This means, 155ms is the shortest interval with which retransmissions can be received with the sender driven scheme. Hybrid and receiver driven scheme show no

[1] Note that due to the downlink TTI of 10ms, the timer will only be effective at the beginning of the next TTI after its expiry

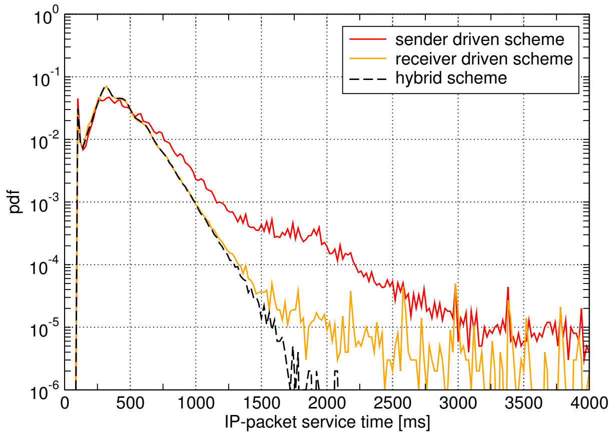

Fig. 9. pdf of the IP-packet service time, $P_{\text{loss,DL}} = 0.2$

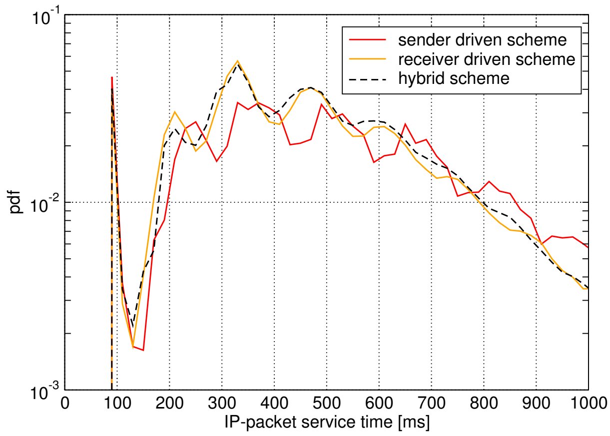

Fig. 11. Zoom into the pdf of Fig. 10

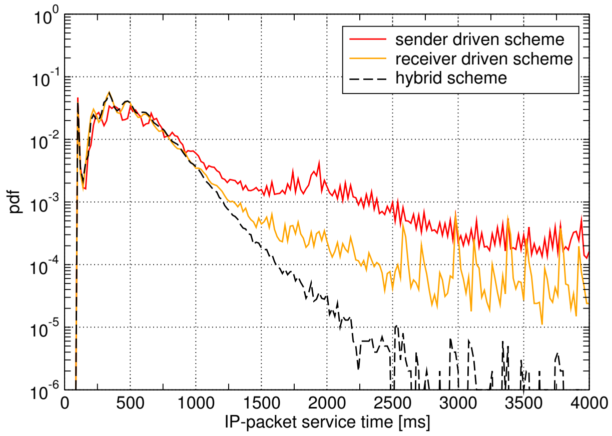

Fig. 10. pdf of the IP-packet service time, $P_{\text{loss,DL}} = 0.3$

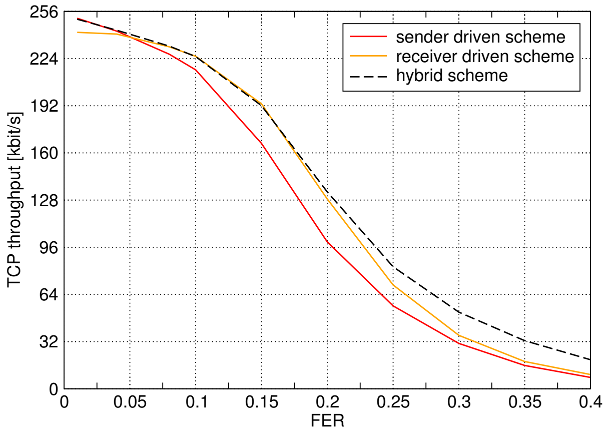

Fig. 12. TCP Throughput

spikes, because the *EPC* function keeps the time period within which the transmission of status-reports is prohibited, variable.

B.4 $P_{\text{loss,DL}} = 0.20$

When comparing the service time pdfs for an FER of 0.20 in Fig. 9 with those for lower FERs, we can observe that the first peak at 90ms decreased and the notch became bigger. This means fewer IP-packets were received without the loss of a corresponding RLC block set. Beside the first and second peak, we can also observe a very smooth third peak.

The second peak is at about 300ms, which is slightly higher than the analytical approximation $T_{\text{avg}} = 265\text{ms}$. This difference can easily be explained with the reduced effective data rate at higher FERs, which is due to bandwidth consuming retransmissions. The spikes observed for the sender driven scheme at $P_{\text{loss,DL}} = 0.1$ in Fig. 8 turned into "waves". The reason for this is the higher number of necessary retransmissions, which smoothens the peaks.

B.5 $P_{\text{loss,DL}} = 0.30$

Figures 10 and 11 show the behavior at a FER of 0.30. Regarding the height of the first peak and the following deep notch, very few IP-packets arrive at the receiver without any error. The previously observed wave characteristic gets stronger, because more and more retransmissions are necessary for completely receiving an IP-packet.

The second peak is made up of IP-packets needing just one retransmission. Obviously, at this high FER, most SDUs need more than one retransmission. Because of the decreasing effective data rate the peaks get not only smoother but also the distance in-between gets wider. The first three peaks are spaced 120ms apart, which nicely corresponds to $RTT_{\text{UE−RNC−UE}}$. All following peaks are spaced 140ms and higher. The peaks of the sender driven scheme are spaced further apart, since it does not respond to lost PDUs as fast as the other two schemes.

B.6 Summary

To conclude our study with respect to the FER, Fig. 12 shows the throughput of a TCP connection over the FER for the three considered RLC schemes. The throughput chart confirms the results of the service time analysis. In general, the hybrid scheme provides the best performance with respect to both IP-packet service time and TCP throughput. For an FER between 0.05 and 0.15, the TCP throughput is almost the same with the hybrid and the receiver driven scheme. In other FER regions, the receiver driven scheme performs worse, which agrees with the service time analysis. Alike, for a very small FER, the sender driven scheme delivers the same TCP throughput as the hybrid scheme. However, with increasing FER, the throughput quickly drops below that of the hybrid and the receiver driven scheme.

It also became obvious that large service times on the order of half a second are common at FERs of 20% or more. At these FERs, all distributions have a heavy tail, where the hybrid scheme still shows the best behavior. In any case, IP packets might be delayed by several seconds, which may impose problems on transport protocols like TCP [11].

It can finally be stated that sender driven mechanisms by itself are not recommendable. They lack the information of PDU losses and cannot react on actual channel events. On the other hand, receiver driven schemes are only useful if periodic status reporting is done with an interval of about the RTT. Moreover, a receiver driven scheme has no information about the sender's state, such as the transmission buffer occupancy. The best choice is a combination into a hybrid scheme. This allows the immediate reaction on a frame loss at the receiver side and at the same time control the ARQ based on the sender's state.

C. Impact of Status-Prohibit Timer

It is usually desirable to maximize the throughput and at the same time minimize the packet delay. In a receiver based scheme, the IP-packet service time can be reduced by decreasing the RLC timer *Timer_Status_Prohibit*. As a consequence, status reports can be sent more frequently, thus leading to lower service times. This effect can be seen in Fig. 13 and 14, which show the pdf and the ccdf of the IP-packet service time, respectively, for the receiver driven scheme and different values of *Timer_Status_Prohibit*.

Additionally, Fig. 17 shows the TCP throughput depending on the value of *Timer_Status_Prohibit*. The maximum throughput is achieved for a timer value of 155ms. For smaller values, status reports may be sent more than once per RTT. This may lead to retransmission requests for RLC blocks, where a retransmission is still ongoing. This causes duplicate data transmission and a lower user throughput. It can be avoided by properly choosing the value of *Timer_Status_Prohibit*. Alternatively, the sender could implement a mechanism that avoids the retransmission of RLC blocks within one RTT after their last transmission. The 3GPP RLC specification [5] does not deal with this issue. Since it would add complexity to the transmitter, this option was not taken into account.

For higher values of the timer, the TCP throughput decreases, since the *TX Window* closes more frequently due to the lack of status reports. In this case, data transmission may stall until a new status-report is triggered by the *Status Periodic* function, which reopens the *TX Window* as described before.

A second effect leading to a decrease in TCP throughput is caused by the in-order delivery of IP-packets in the UE and the RNC. Assume that a frame loss occurs during the transmission of an IP-packet. Even if all successive IP-packets are transmitted correctly, they cannot be delivered to upper layers until the correct reception of the first IP packet. As soon as the retransmission for the first IP-packet has successfully been completed, a burst of IP-packets will be delivered to upper layers. In conjunction with TCP, this effect causes *ACK compression* [12], which decreases the throughput.

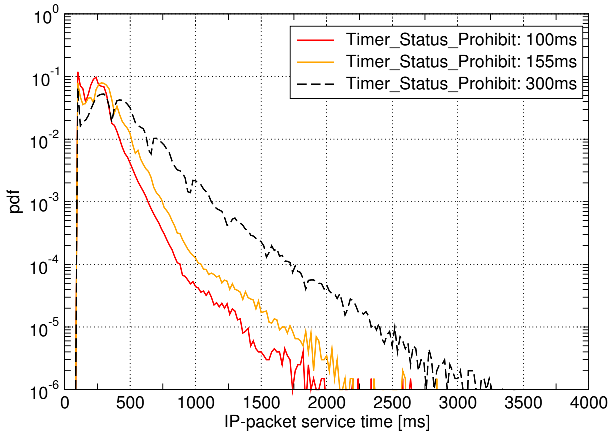

Fig. 13. pdf of the IP-packet service time, $P_{\mathrm{loss,DL}} = 0.1$

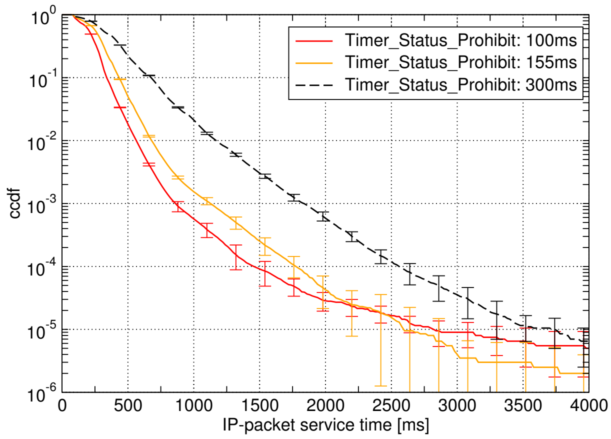

Fig. 14. ccdf of the IP-packet service time, $P_{\mathrm{loss,DL}} = 0.1$

D. Comparison of EPC and Status Prohibit function

The *EPC* and *Status Prohibit* functions are very similar. The difference is that *EPC* does not prohibit status-report transmission for a fixed time, but for the RTT estimate plus the time it takes the AMD-PDUs requested with the last status-report to be retransmitted. *EPC* only prohibits status-report transmission if at least one AMD-PDU was acknowledged negatively with the last status-report. Additionally, it takes care of transmitting a new status-report if not all requested AMD-PDUs were received. In the following, both functions will be compared. Simulations were done with either activated *EPC* or activated *Status Prohibit*. All other parameters were configured as shown in table I for the receiver driven scheme.

Figures 15 and 16 show the pdf and the ccdf of the IP packet service time with activated EPC and for different values of the RTT estimate, respectively. When comparing the ccdfs of the *Timer_Status_Prohibit* and the *EPC* mechanism in Fig. 14 and Fig. 16, it can be seen, that the EPC function provides a significantly lower service time. This even holds if the prohibit time of the EPC function is higher than the prohibit time of the *Status Prohibit* function.

Fig. 17 shows the TCP throughput depending on both the *Timer_Status_Prohibit* timer and the RTT estimate. The maximum throughput is achieved with an RTT estimate of 120ms which corresponds to the calculated $RTT_{\mathrm{UE-RNC-UE}}$. Further, the TCP throughput curve for EPC is very similar to the *Status Prohibit* curve, but shifted to the left. This is because the *EPC* function prohibits status-report transmission for longer than the RTT estimate, like mentioned before. Summarized it can be said, that the *Status Prohibit* function is always inferior to the *EPC* mechanism.

V. Conclusion

We investigated the parameter choice of the ARQ mechanism in the RLC layer of UMTS. In particular, we compared parameter settings for sender driven, receiver driven and hybrid schemes. Simulations showed that hybrid ARQ schemes, which combine the mechanisms of sender and receiver driven schemes, deliver the best performance with respect to IP packet service time and TCP throughput.

We also found that, despite all efforts on the RLC layer, a significant amount of IP-packets may suffer from a delay on the order of seconds. This is mainly caused by the large RTT of the system and may impact the performance of transport protocols. To solve this problem, it is necessary to reduce the RTT, as it is done in future extensions to UMTS, such as HSDPA.

Acknowledgements

The research work of this paper was done in cooperation with Alcatel Research and Innovation Department, Stuttgart.[2]

References

[1] L. Burgstahler, K. Dolzer, C. Hauser, J. Jähnert, S. Junghans, C. Macián, W. Payer: "Beyond Technology: The Missing Pieces for QoS Success", in *Proc. ACM SIGCOMM 2003 Workshops*, Karlsruhe, 2003.

[2] 3GPP TS 25.212: "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD) (Release 5)".

[3] S. Lin, D. J. Costello, Jr., M. J. Miller: "Automatic-Repeat-Request Error-Control Schemes", *IEEE Communications Mag.*, Vol. 22, Iss. 12, December 1984, pp. 5-17.

[4] R. A. Comroe, D. J. Costello, Jr.: "ARQ Schemes for Data Transmission in Mobile Radio Systems" *IEEE Journal on Selected Areas in Communications*, Vol. 2 , Iss. 4, July 1984, pp. 472–481.

[5] 3GPP TS 25.322: "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Link Control (RLC) protocoll specification (Release 5)".

[6] 3GPP TS 34.108: "3rd Generation Partnership Project; Technical Specification Group Terminals; Common test environments for User Equipment (UE) (Release 4)"

[7] 3GPP TS 25.853: "3rd Generation Partnership Project; Technical Specification Group (TSG) RAN; Delay Budget within the Access Stratum (Release 1999)"

[8] M. Rossi, L. Scaranari, M. Zorzi: "On the UMTS RLC Parameters Setting and their Impact on Higher Layers Performance", in *Proc. IEEE VTC 2003-Fall*, Orlando, Florida, October 2003

[2]Alcatel SEL AG, Research & Innovation, Lorenzstr. 10, 70435 Stuttgart, Germany. Contact: Andreas Weber (Andreas.Weber@alcatel.de).

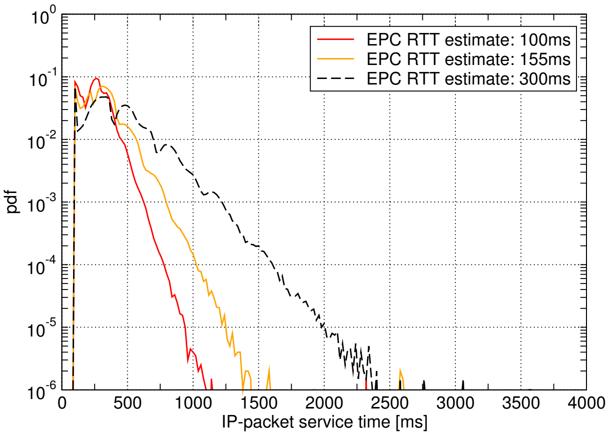

Fig. 15. pdf of the IP-packet service time, $P_{\mathrm{loss,DL}} = 0.1$

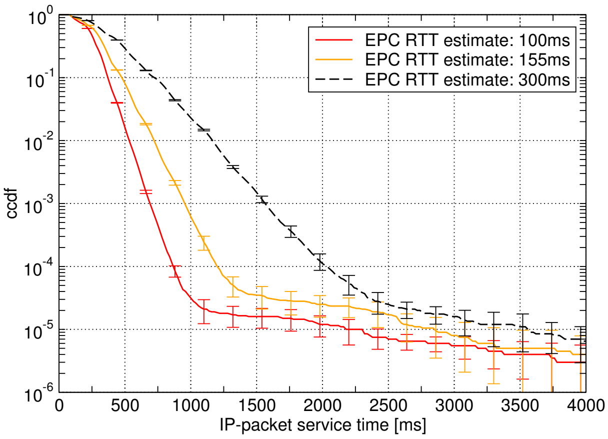

Fig. 16. ccdf of the IP-packet service time, $P_{\mathrm{loss,DL}} = 0.1$

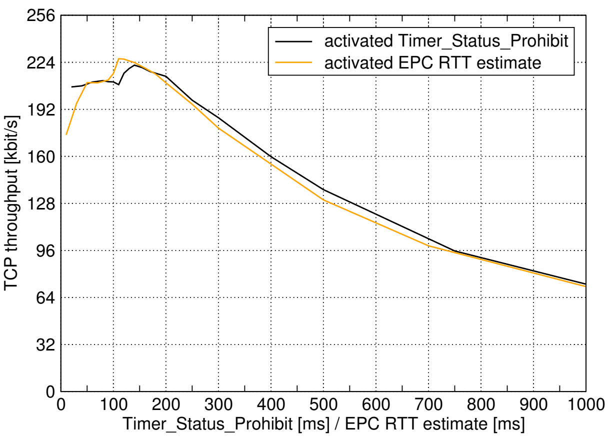

Fig. 17. TCP throughput, $P_{\mathrm{loss,DL}} = 0.1$

[9] Q. Zhang, H.-J. Su: "Performance of UMTS Radio Link Control", in *Proc. IEEE ICC*, Vol. 5, pp. 3346-3350, April 2002.

[10] A. Mate, C. Caldera, M. Rinne: "Performance of the Packet Traffic on the Downlink Shared Channel in a WCDMA cell", in *Proc. IEEE ICT 2001*, Bucharest, Romania, June 2001

[11] R. Ludwig: "Eliminating Inefficient Cross-Layer Interactions in Wireless Networking", PhD thesis, RWTH Aachen, April 2000.

[12] L. Zhang, S. Shenker, D. Clark: "Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic", *in Proc. ACM SIGCOMM*, pp. 133–147, 1991.