Interaction between UMTS MAC Scheduling and TCP Flow Control Mechanisms

Matthias Malkowski, Silke Heier

Aachen University (RWTH), Chair of Communication Networks (ComNets)
E-Mail: {mal|she}@comnets.rwth-aachen.de

***Abstract*— The goal of the *Universal Mobile Telecommunication System* (UMTS) is the delivery of multimedia services to the mobile user. Each different service requires its specific *Quality of Service* (QoS) to satisfy the mobile user. The QoS requirements will be supported by several protocol layers. In this paper, the interaction between the *Medium Access Control* (MAC) scheduling and the *Transmission Control Protocol* (TCP) flow control mechanisms at the UMTS radio interface is presented. Whereas the MAC is responsible to guarantee delay and throughput requirements at the radio interface, TCP realizes an end-to-end flow control. Nevertheless both protocols show dependencies on each other that might reduce the data transmission efficiency. In this paper, the overall performance of Internet applications running over TCP by using different MAC scheduling strategies is discussed. A *UMTS Radio Interface Simulator* (URIS) is used to emulate the standardized UMTS protocol stack and the TCP/IP protocol suite. Simulation results of QoS parameters depict the performance of mobile applications over UMTS.**

***Keywords*—UMTS, Radio Interface Protocols, MAC Scheduler, QoS, WWW Traffic Model, UMTS Simulator**

I. Introduction

The delivery of multimedia services to the mobile user is one of the goals of 3rd generation mobile communication systems. The use of different services at the same time raises the demands for mechanisms to guarantee *Quality of Service* (QoS). To satisfy the mobile user, UMTS provides several *Radio Resource Management* (RRM) strategies. One of these strategies is the scheduling of parallel data flows in the *Medium Access Control* (MAC) layer. Another important mechanism is the retransmission of lost data packets by the *Radio Link Control* (RLC) protocol. Both layers should guarantee a reliable and efficient data transmission over the capacity restricted, unreliable radio link. On the other hand typical Internet applications like web browsing, email or file transfer rely on the TCP/IP protocol stack. Since the classical Internet does not guarantee any QoS, TCP is used for flow control mechanisms and retransmission of lost packets on an end-to-end basis.

This paper will examine the performance of Internet applications if used with different MAC scheduling strategies. Of special concern is the interaction of TCP with the depicted MAC scheduler. Three different MAC scheduling concepts are introduced which should fulfill the QoS requirements in terms of delay and throughput. The concepts will be validated by an *UMTS Radio Interface Simulator* (URIS) that models the radio interface protocols, the TCP/IP protocol stack and the traffic sources.

II. MAC Scheduling Concepts

UMTS supports parallel handling of multiple data streams arising from distinct applications. Applications belong to different service classes (conversational, streaming, interactive, background) which require different QoS demands in terms of delay, jitter, throughput, etc. To support these demands efficiently, the RRM assigns specific transmission parameters to the *Data Link Control* (DLC) layer:

- *Radio Link Control* (RLC) transmission mode,
- Mapping and multiplexing options of logical channels to transport channels (*Radio Bearer Mapping*, RBM),
- *MAC Logical Channel Priorities* (MLP) assigned to every logical channel,
- *Transport Format (Combination) Sets* (TFCS).

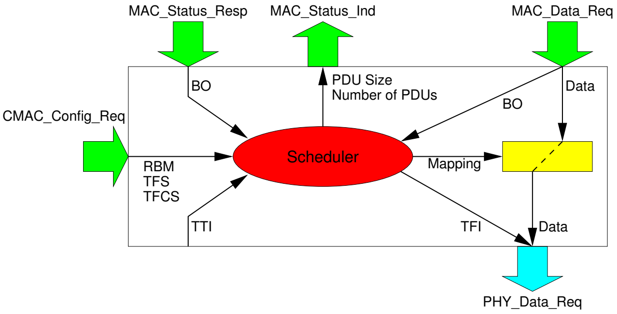


Figure 1. Input and Output Parameter of the MAC Scheduler

Our proposed MAC scheduler uses MLPs to provide priority scheduling. This will guarantee delay and throughput requirements between applications of different QoS classes. The *Transport Format Combination* (TFC) selection as part of the MAC scheduling is performed based on *Buffer Occupancies* (BO) in order to guarantee the required traffic throughput. A *Longest Queue First* (LQF) or a *Queue Length based Weighted Fair Queuing* (QLWFQ) scheduling is used to cover applications of the same priority dedicated to the same QoS class. Fig. 1 depicts the incoming and outgoing parameter of the MAC scheduler which are used for the scheduling process. In our simulation environment full functionality of the MAC layer is emulated in conformance to [1].

III. UMTS Radio Interface Model

The URIS performs capacity and QoS evaluations for various scenarios. The simulator is a pure software solution. The UMTS radio interface protocols are enhanced by a TCP/IP protocol stack. The complex protocols like MAC, RLC and *Packet Data Convergence Protocol* (PDCP) are implemented completely bit accurate in conformance to their specifications. Hence, URIS performs a protocol emulation for performance evaluation.

The RLC protocol provides an *Acknowledged Mode* data transfer that guarantees the packet delivery to the peer entity. *Automatic Repeat Request* (ARQ) mechanisms are applied for correction of transmission errors. A Selective-Repeat ARQ, segmentation/reassembling and concatenation are fully implemented as specified in [2]. The TCP implementation is based on the “Reno” TCP stack and uses the following flow control mechanisms: slow start/congestion avoidance, fast retransmit/fast recovery, delayed acknowledgments and selective acknowledgments [3].

TABLE I. Model Parameters of HTTP Browsing Sessions and FTP Sessions

HTTP Parameter	Distribution	Mean	Variance
Session Arrival Rate [h^{-1}]	negative exponential	30	—
Pages per Session	geometric	5	—
Reading Time between Pages [s]	negative exponential	20	—
Objects per Page	geometric	2.5	—
Inter Arrival Time between Objects [s]	negative exponential	0.5	—
Page Request Size [byte]	normal	1136	80
Object Size [byte]	$\log_2$-Erlang-k	$\log_2 2521 \approx 11.3$	$(\log_2 5)^2 = 5.4$
FTP Parameter	Distribution	Mean	Variance
Session Arrival Rate [h^{-1}]	negative exponential	30	—
Session Size [bytes]	$\log_2$-normal	$\log_2 32768 \approx 15$	$(\log_2 16)^2 \approx 16$
Object Size [bytes]	$\log_2$-normal	$\log_2 3000 \approx 11.55$	$(\log_2 16)^2 \approx 16$
Time between Objects [s]	$\log_{10}$-normal	$\log_{10} 4 \approx 0.6$	$\log_{10} 2.55 \approx 0.4$

To examine the performance of data services like HTTP and FTP, a detailed traffic model is necessary [4]. The parameters of the used traffic models are shown in Tab. I. For both applications the session arrival rate is very high. A highly loaded traffic channel is mandatory to study the effects of scheduling and TCP flow control mechanisms.

IV. Simulation Scenario

The main parameters concerning *Quality of Service* (QoS) of an application are delay and throughput. During the simulations the following measurements were performed:

1) Buffer Occupancy: The amount of data queued in the RLC transmission buffer at the time the scheduler calls the buffer occupancies for transmission planning,
2) TCP Packet Delay: Time from sending a TCP packet to the RLC until correct reception by the TCP receiver,
3) User Data Packet Delay: Time from sending a user data packet until correct reception by the receiver.

Simulations were performed examining the scheduling of one HTTP and one FTP application. The MAC layer multiplexes both applications onto one *Dedicated Transport Channel* (DCH). The assigned transport formats provide a maximum data rate of 67.2 kbit/s. The simulation parameters in conformance to recommendation [5] are shown in Tab. II.

Simulation results are given for the different scheduling strategies. First, priority scheduling is examined where HTTP is assigned a higher priority compared to FTP. In this scenario FTP is a background traffic. When HTTP and FTP are assigned to the same QoS class/priority two different scheduling strategies are examined. For a fair treatment of both data flows a LQF and a QLWFQ strategy are simulated. While LQF prefers the service with the highest source data rate, QLWFQ splits the available channel capacity concerning the ratio of the source data rates.

TABLE II. Simulation Parameters

Traffic Generator	HTTP/FTP
TTI Length [s]	0.02
Transport Format Set [bit]	0x336, 1x336, 2x336, 4x336
Max. MAC Data Rate [kbit/s]	67.2
MLP	HTTP: 2, FTP: 3
RLC Mode	AM
Max. TCP Segment [byte]	512
Max. TCP Window [kbyte]	16
Min./Max. TCP RTO [s]	3 / 64

The complementary cumulative distribution function of the measured buffer occupancies, TCP delay and user data packet delay have been calculated. Measurements have been performed for an error free physical channel. Studies of RLC interaction with TCP can be found in [6–8].

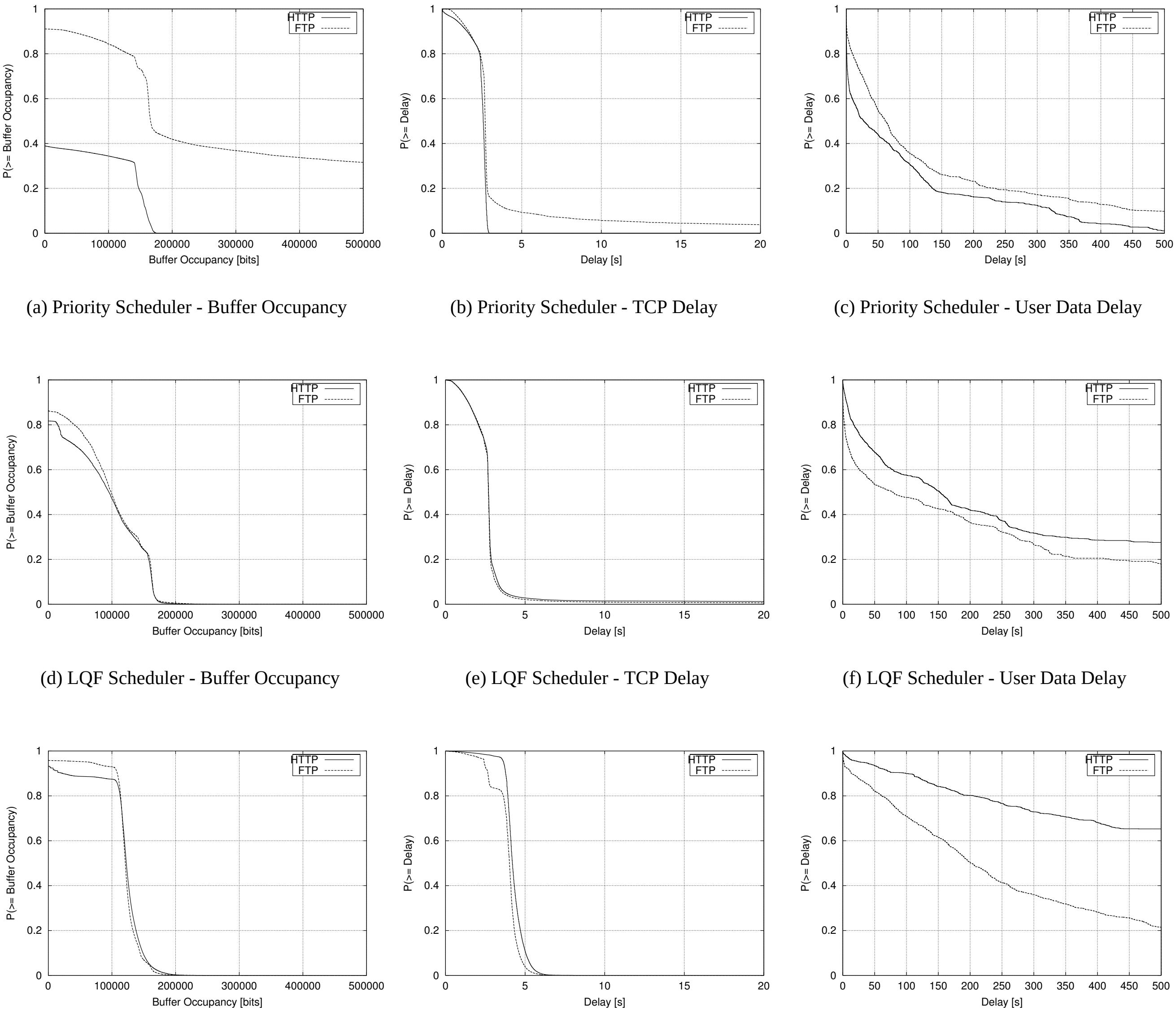
V. Simulation Results - Buffer Occupancy

The buffer occupancy illustrates the processing of scheduling strategies. On the left hand side of Fig. 2 simulation results for the dequeuing of offered traffic are shown. Rectangular curve shapes indicate that buffers are dequeued very fast. The TCP transmission window size causes a maximum buffer size of around about 140 kbit. The transmission window carries 32 TCP segments of 512 bytes. A TCP/IP header information adds 40 bytes to each TCP segment. In result, a whole TCP window causes 141312 bits in the RLC buffer.

It can be stated, that the TCP layer has major impact on the traffic characteristic. The windowing mechanism provides a flow control which shapes the traffic of the data source. In result, the scheduler in the MAC layer is not able to estimate source data rate and characteristic of the application in a high load scenario by the buffer occupancies of the RLC layer. The transmission planning of the MAC scheduler depends on TCP shaped traffic since TCP buffer occupancies are hidden for the MAC. Nevertheless, the MAC layer is unable to adapt the Transport Formats to the source data rate since TCP delivers a traffic shaped data flow.

In case of priority scheduling (Fig. 2(a)) the distribution of the buffer occupancy for HTTP is lower than for FTP. Due to a higher priority of HTTP traffic, FTP traffic can not influence this distribution. The background FTP transmission is blocked by the HTTP application. This is the case in 40% of the time, so the buffer occupancy for the FTP traffic gets exceedingly high due to retransmissions triggered by the TCP layer. A maximum buffer occupancy of 6.4 Mbit was measured for this scenario.

Fig. 2(d) shows the buffer occupancy for the LQF scheduling. Since both applications have the same priority, the LQF algorithm aims to keep both buffer occupancies equal. The maximum buffer occupancy has a slightly higher maximum than a whole TCP window. This is caused by a few TCP retransmissions on the downlink triggered by retransmission timeouts. This scheduling strategy shows a slower dequeuing process since the curve is not rectangular. The slanting decline of the curve is caused by retransmission timeouts and



(a) Priority Scheduler - Buffer Occupancy

(b) Priority Scheduler - TCP Delay

(c) Priority Scheduler - User Data Delay

(d) LQF Scheduler - Buffer Occupancy

(e) LQF Scheduler - TCP Delay

(f) LQF Scheduler - User Data Delay

(g) QLWFQ Scheduler - Buffer Occupancy

(h) QLWFQ Scheduler - TCP Delay

(i) QLWFQ Scheduler - User Data Delay

Figure 2. Simulation Results

the resulting congestion avoidance of the TCP protocol. By retrieving a retransmission timeout, TCP assumes that there has been a congestion. TCP reduces the transmitting window and starts its congestion avoidance mechanism. The TCP transmit window size increases slowly which results in an reduced buffer occupancy in the RLC layer.

The QLWFQ algorithm (Fig. 2(g)) has a limited maximum buffer occupancy, too. Since both buffers are filled with 100 kbits in 90% of the time, the scheduler will transmit two transport blocks for each application at each transmission time interval since the ratio of the buffer occupancies is one half. In consequence, each application will experience half of the overall channel capacity most of the time. The TCP protocol detects congestion for both applications and reduces its transmission window accordingly. This can be seen at the declension of the buffer occupancies curves between 100 kbits til 200 kbits.

VI. Simulation Results - TCP Packet Delay

The TCP packet delay is directly related to the buffer occupancies and the scheduling technique. At least each TCP packet will face a maximum delay that is needed to transfer a whole TCP window of 141312 bits. The data rate for TCP segments is 62.4 kbit/s.

$$\text{TCP Delay} = \frac{\text{Buffer Occupancy}}{\text{Data Rate}} = \frac{141312\ \text{bits}}{62.4\ \text{kbps}} = 2.265\ \text{s}$$

In Fig. 2(b) the TCP packet delay for the priority scheduling is shown. The delay of the higher prioritized HTTP traffic shows that 80% of the TCP segments are delayed less than 2.9 seconds. Higher delays above 2.3 seconds are related to RLC and TCP acknowledgments for the uplink direction. The delay of the lower prioritized FTP traffic is significantly increased while blocked by the higher prioritized HTTP traffic. This blocking takes place for approximately 20% of the FTP TCP segments. These segments are delayed more than three seconds which triggers a retransmission timeout in the TCP layer. Because the TCP layer tries to send retransmissions for the blocked FTP segments, additional load is generated which increases the TCP packet delay again. The TCP segment delay for FTP increases so much, that no delay requirement could be guaranteed.

Equal priority is given to the traffic flows during LQF scheduling. The maximum TCP delay gets very high since each traffic type blocks the other one depending on the buffer occupancy situation. As the LQF algorithm tries to keep queue length equal, the mean TCP packet delay is low and similar for HTTP and FTP traffic. For 5% of the time, the delay is greater than three seconds which will cause retransmissions by the TCP layer. But the retransmissions will not cause unlimited delays since no traffic flow is blocked totally and both will get transmission capacity from time to time.

The result for the QLWFQ algorithm is shown in Fig. 2(h). The QLWFQ strategy tries to give capacity to both traffic flows. This results in a low maximum delay because the transmission of HTTP or FTP is rarely completely blocked. But 85% of the TCP segments notice a delay greater than 3s. As a result both TCP packet flows will face retransmissions of TCP segments due to retransmission timeouts.

VII. Simulation Results - User Data Packet Delay

The user data packet delay is the perceived delay for HTTP page objects and FTP downloads. The delays are high because of the high load scenario. In case of the priority scheduling (Fig. 2(c)) delays of WWW page objects are smaller than delays of FTP downloads. Running FTP as a background service extents the download times noticeable.

The LQF strategy has negative impact on both delay distributions (Fig. 2(f)). Because the traffic type with the shorter queue gets blocked by the other one, TCP retransmissions occur. Since unnecessary TCP retransmissions burden the radio interface, new user data packets have to wait until retransmissions are transfered correctly. The delays of HTTP page objects are higher because the FTP load generator produces more and bigger data packets than HTTP. In result the FTP buffer occupancy in the RLC is higher which causes even more transmission capacity assignment by the LQF for the FTP traffic. To sum it up, it can be noticed that the LQF strategy prefers the data flow which generates the highest load. Other data flows can use the remaining channel capacity.

To prevent such an unfair sharing of the capacity, the QLWFQ strategy was examined. Each data flow should get capacity assigned proportional to its load characteristic. But the QLWFQ experiences worst delay measurements (Fig. 2(i)). No delay restrictions can be fulfilled by this strategy because of the negative impact of the TCP interaction. Because the capacity assignment of the scheduler changes very quickly in accordance to the actual buffer ratios, the TCP flow control mechanisms are too slow to adapt their parameter correctly. Most of the time both data flows will cause same buffer occupancies. In that case the throughput will be half of the channel capacity for both data flows. In times where one application is not sending, the other application could use the whole channel capacity. But the measurements show that this is not the case since the congestion avoidance mechanism of the TCP rises the size of the send window too slow. The buffer occupancies in Fig. 2(g) illustrate that the TCP send window is reduced since they are filled with 100 kbit most of the time but the data amount of a maximum TCP window can be reviewed for only 5% of the time. Additionly TCP segment delays are very high and a huge amount of TCP retransmissions is triggered. Since these unnecessary TCP retransmissions burden the radio interface, ordinary user data packets experience long waiting times until retransmissions are transfered correctly.

VIII. Conclusion

Running an application mix over TCP will be an ordinary scenario during the introduction of UMTS. First terminals will rely on standard applications which will be adopted from the fixed world. Hence, Reno-TCP will run end-to-end including the radio interface. This paper shows that it is applicable to satisfy the mobile user but performance suffers since TCP mechanisms will not efficiently use the guaranteed QoS of the radio bearers in terms of delay and throughput.

One aim of UMTS is the efficient use of scarce bandwidth resources. If the radio link shall be shared efficiently between applications, appropriate scheduling strategies have to be introduced. The planning horizon of the MAC scheduler is 10 ms up to 80 ms. The simulations have shown that TCP can not cope with such a quick adaptation. TCPs flow control mechanisms are too slow for a dynamic radio link. On the other hand side TCP has major impact on the MAC scheduler. For a detailed planning strategy it is worthwhile to know the most actual load situation of the data sources. The buffer occupancies of the RLC are not meaningful since TCP realizes a traffic shaping. In result, the guaranteed capacity of the UMTS radio interface is not used efficiently since TCP burdens the radio interface with unnecessary retransmissions or leaves capacity unused since the TCP window is too slow to follow the changing radio interface conditions dynamically.

To enhance the overall performance several solutions are possible. First the RLC can use the discard function to remove all data packets with a live time above the retransmission timeout of TCP. This data will be retransmitted by the TCP anyway. Another way is the improvement of TCP for mobile users. The flow control mechanisms of TCP like slow start, fast recovery and congestion avoidance have to be parameterized to cope with the dynamic of the changing radio link conditions. Concerning the MAC scheduler an interlayer communication will be useful to determine the actual load situation of the traffic sources. An efficient planning can only be performed if the load characteristic of the traffic source is known without any traffic shaping in between.

References

[1] 3GPP TS 25.321, "Medium Access Control Protocol Specification," Technical Specification V4.1.0, 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, June 2001.

[2] 3GPP TS 25.322, "Radio Link Control (RLC) Protocol Specification," Technical Specification V4.1.0, 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, June 2001.

[3] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *Lawrence Berkeley National Laboratory*, July 1996.

[4] Victor S. Frost and Benjamin Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, March 1994.

[5] 3GPP TS 34.108, "Common Test Environments for User Equipment (UE) Conformance Testing," Technical Specification V4.0.0, 3rd Generation Partnership Project, June 2001.

[6] S. Heier, D. Heinrichs, and A. Kemper, "Performance Evaluation of Internet Applications over the UMTS Radio Interface," Birmingham AL, US, May 2002, VTC Spring 2002 - The IEEE Semiannual Vehicular Technology Conference on Connecting the Mobile World.

[7] S. Heier, D. Heinrichs, and A. Kemper, "IP based Services at the UMTS Radio Interface," London, UK, May 2002, 3G 2002 - Third International Conference on 3G Mobile Communication Tecnologies.

[8] S. Heier, D. Heinrichs, and A. Kemper, "Performance of Internet Applications at the UMTS Radio Interface," San Francisco, US, May 2002, 3Gwireless 2002 - 2002 International Conference on Third Generation Wireless and Beyond.