UMTS MEDIUM ACCESS CONTROL QUALITY OF SERVICE SCHEDULING

Silke Heier, Christian Ellerbrock, Matthias Malkowski

Aachen University of Technology (RWTH), Communication Networks (ComNets),
Kopernikusstr. 16, 52074 Aachen, Germany, {she|ell|mal}@comnets.rwth-aachen.de

Abstract — The goal of *Universal Mobile Telecommunications 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. In this paper, a scheduling concept for the Medium Access Control is presented that satisfies the QoS requirements. To validate the scheduling concept a typical mobile user application mix is examined for performance analysis by simulations. A *UMTS Radio Interface Simulator* (URIS) is used that is based on an emulation of the standardized UMTS protocol stack extended by the *Transmission Control Protocol* (TCP) and the *Internet Protocol* (IP). Simulation results of QoS parameters like delay and throughput depict the performance of mobile applications over UMTS.

Keywords — UMTS Radio Interface Protocols, WWW Traffic Model, MAC Scheduler, QoS, TCP/IP, UMTS Radio Interface Simulator

I. INTRODUCTION

The delivery of multimedia services to the mobile user is one of the goals of 3rd generation mobile communication systems. UMTS will provide data services with data rates of up to 144 kbit/s in rural areas, 384 kbit/s in hot-spots and up to 2 Mbit/s in indoor scenarios. The use of several different services at the same time raises the demands for mechanisms to guarantee *Quality of Service* (QoS) for the applications. To satisfy the mobile user, UMTS provides several Radio Resource Management strategies. One of these strategies is the scheduling of parallel data flows in the *Medium Access Control* (MAC) layer.

This paper will introduce a MAC scheduling concept that is able to fulfill the QoS requirements in terms of error rate, delay, jitter and throughput. The concept will be validated by simulations of an typical application mix. Therefore, an *UMTS Radio Interface Simulator* (URIS) is used that models the radio interface protocols as well as the traffic sources.

In this paper, the MAC scheduling task is outlined first. After presenting our scheduling concept, the URIS is presented, followed by a short overview of HTTP and FTP traffic modeling. Simulation results will show which performance in terms of QoS parameters a mobile user will experience while executing a typical mobile user application mix.

II. THE MAC SCHEDULING CONCEPT

UMTS supports the parallel handling of multiple data streams arising from distinct applications. Applications belong to different service classes (conversational, streaming, interactive, background) which all require different QoS demands in terms of bit error rate, delay, jitter, throughput, etc.. To support these demands efficiently, the different layers are assigned specific transmission parameters by the *Radio Resource Management* (RRM). For the *Data Link Control* (DLC) layer these are in particular:

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

Our proposed MAC scheduler uses MLPs to provide a priority scheduling between applications of different QoS classes. This will guarantee the strong delay requirements of applications of the conversational and streaming class. TFC selection as part of the MAC scheduling is performed based on the buffer occupancies of the applications in order to guarantee the required traffic throughput. A *Longest Queue First* (LQF) and a Random scheduling is used to cover applications of the same priority dedicated to the conversational/streaming and interactive/background class. The diagram in Fig. 1 depicts the processing of the MAC scheduler. In our simulation environment the full functionality of the MAC layer is emulated in conformance to [1].

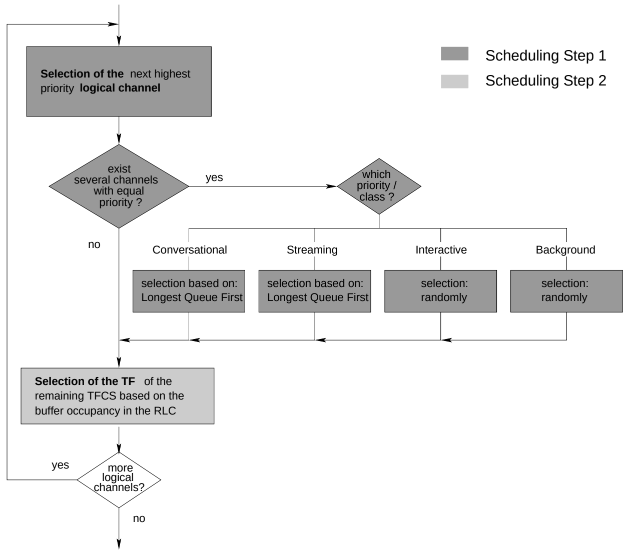


Fig. 1. Proceeding of the MAC Scheduler

Table 1
Model Parameters of HTTP Browsing Sessions and FTP Sessions

HTTP Parameter	Distribution	Mean	Variance
Session Arrival Rate [h^{-1}]	negative exponential	2	—
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	2	—
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	$\log_{10} 4 \approx 0.6$	$\log_{10} 2.55 \approx 0.4$

III. UMTS Radio Interface Model

The URIS was developed to perform capacity and QoS evaluations for various scenarios. The simulator is a pure software solution in the programming language C++. The UMTS protocols at the radio interface enhanced by a TCP/IP protocol stack were specified with the *Specification and Description Language* (SDL).

The software architecture of the URIS simulator is shown in Fig. 2. The simulator consists of various traffic generators and a traffic load mixture unit which is used to adjust scenarios with desired load mixtures. The physical channel module models transmission of bursts in radio frames on the radio interface. This includes discarding of erroneous bursts depending on the error model.

The complex protocols like MAC, RLC, *Packet Data Convergence Protocol* (PDCP) and *Radio Resource Control* (RRC) are implemented completely bit accurate in conformance to their specifications. Hence URIS uses a protocol emulation for performance evaluation.

The RLC protocol provides three different data transfer services: *Transparent Mode* (TR), *Unacknowledged Mode* (UM) and *Acknowledged Mode* (AM). The AM transmits higher layer data packets with guaranteed delivery to the peer entity. Therefore, it is mainly used for traffic of interactive class or background class type and is well suited for Internet applications. *Automatic Repeat Request* (ARQ) mechanisms are applied for correction of transmission errors. A Selective-Repeat ARQ has been implemented for AM. Segmentation/reassembling, concatenation and ARQ are fully implemented as specified in [2].

The modern Internet is mainly based on TCP and IP. The TCP implementation realized in URIS is based on the so called "Reno" TCP stack and uses the following flow control mechanisms: slow start/congestion avoidance, fast retransmit/fast recovery, delayed acknowledgments and selective acknowledgments [3].

To examine the performance of data services like HTTP and FTP, a detailed traffic model is necessary (Tab. 1)[4]. A typical WWW browsing session consists of a sequence of page requests. These pages contain a number of objects with a dedicated object size each. After the page has been entirely received at the User Equipment, the user is consuming a certain amount of time for studying the information. If a FTP session is started, first the overall transmission size is determined. Then the first object is transferred, followed by an interval with no activity. After this interval the next object is transmitted. This process is repeated until the determined transmission size is reached.

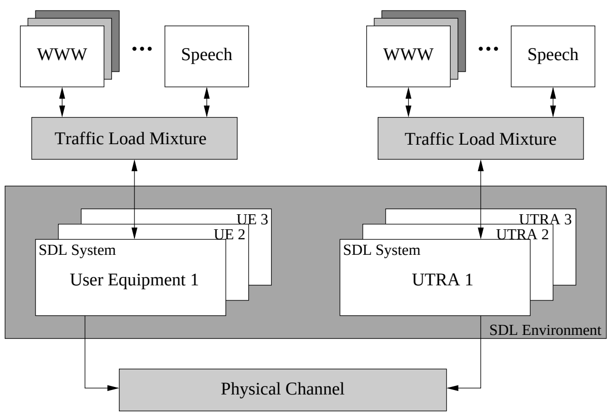
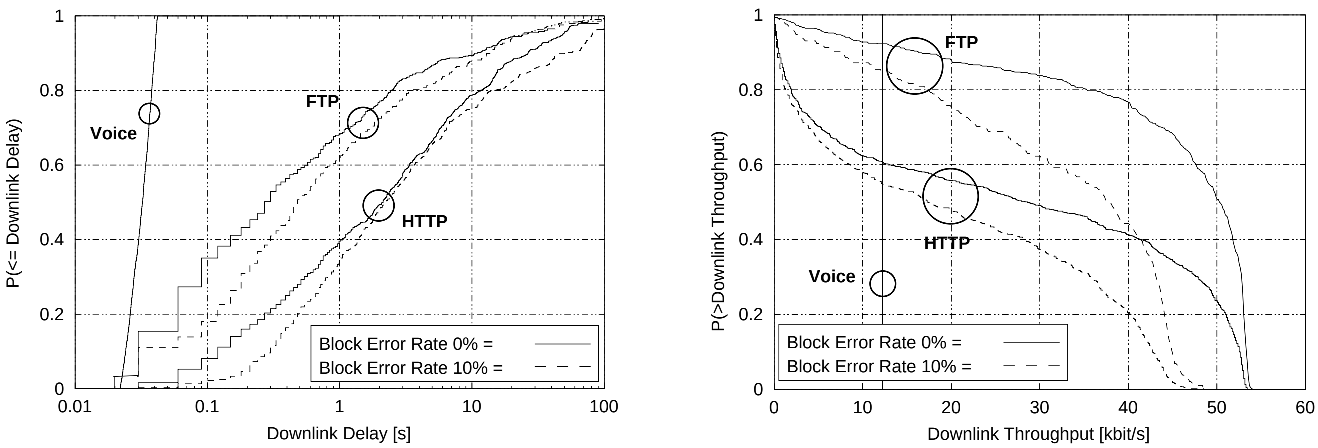


Fig. 2. Structure of the URIS Simulator

IV. Simulation Scenario

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

1) User Data Packet Delay: Time from sending a user data packet until the correct reception by the receiver,
2) User Data Packet Throughput: Size of the user data packet divided by the user data delay.



(a) User Data Packet Delay

(b) User Data Packet Throughput

Fig. 3. Simulation Results - Stand-Alone Application

Simulations were performed examining the interaction of three different applications. One voice application characterized by an on-off CBR traffic generator, one HTTP and one FTP application. The MAC layer maps these applications to separate transport channels. The transport formats that are assigned to the transport channels provide a maximum data rate of 67.6 kbit/s for HTTP and FTP sessions and 13.3 kbit/s for voice. They are configured due to recommendations taken from [5]. The simulation parameter are shown in Tab. 2.

The cumulative distribution function of the measured user data packet delay and the complementary cumulative distribution function of the measured user data packet throughput have been calculated. Additionally the number of transport blocks that are transferred during one *Transmission Time Interval* (TTI) were counted. Measurements have been performed for transport block error rates of 0% and 10%.

Table 2
Simulation Parameters

Traffic Generator	Voice	HTTP	FTP
Source Data Rate [kbit/s]	12.2	Table 1	Table 2
TTI Length [s]	0.02	0.02	0.02
Transport Format Set [bit]	0x266	0x338	0x338
	1x266	1x338	1x338
		2x338	2x338
		4x338	4x338
Max. MAC Data Rate [kbit/s]	13.3	67.6	67.6
MLP	5	6	6
RLC Mode	TR	AM	AM
Max. TCP Segment [byte]	-	512	512
Max. TCP Window [kbyte]	-	16	16
Min./Max. TCP RTO [s]	-	3 / 64	3 / 64
Block Error Rate	0, 10%		

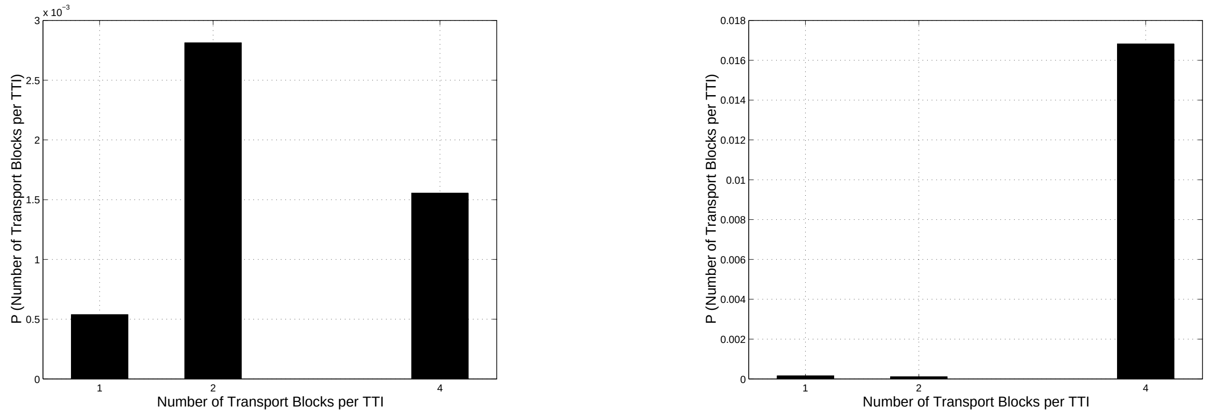
V. SIMULATION RESULTS - STAND-ALONE APPLICATION

Fig. 3 shows the simulation results when only one single application is running on the protocol stack, e.g. no other application runs in parallel. The user data packet delays of the applications are given in Fig. 3(a). As can be seen, the voice application shows the lowest delays of 20 ms up to 40 ms. Due to the configuration of the CBR generator all data packets can be transferred during one TTI. The transparent RLC mode leads to the same characteristics for delay and throughput considering a block error rate of 0% and 10%. Unlike for the HTTP and FTP applications erroneous user data packets are not retransmitted again.

The traffic of HTTP and FTP sessions lead to much higher delays. Small delays can only be reached by small packets that can be transferred within few TTIs and without vital queuing times. But since most of the user data packets are too big to be transferred during one TTI, the probability for such low delays is very small. Differences between HTTP and FTP result from the different traffic source characteristic.

Fig. 3(b) presents the throughput of user data packets for each application. Voice shows constant throughput of 12.2 kbit/s due to the configured CBR generator. The maximum throughput for HTTP and FTP of 54 kbit/s reaches 80% of the available channel capacity, the rest is overhead and padding generated by TCP/IP, RLC and MAC layer. More detailed performance results of Internet applications can be found in [6–8]

Fig. 4(a) shows the probability for the transmitted number of transport blocks per TTI in the HTTP uplink, chosen by the MAC scheduler. Besides page requests that mostly need four transport blocks per TTI, in particular the *Automatic Repeat Request* (ARQ) mechanism in the RLC layer is responsible for the traffic where only one or two transport blocks per TTI are transmitted. Due to the high load on the downlink and only few status PDUs resulting of the ARQ mechanism, an "on-off" characteristic can be examined (Fig. 4(b)).



(a) Uplink

(b) Downlink

Fig. 4. Stand-Alone Application - Number of Transport Blocks per TTI, 0% BER, HTTP

VI. Simulation Results - Application Mix

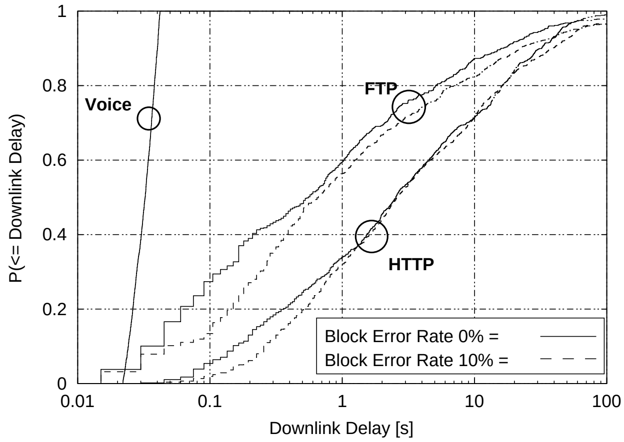
In the next simulation series three applications are running simultaneously. The respective transport channels were multiplexed in the physical layer onto one *Coded Composite Transport Channel* (CCTrCH). All possible transport format combinations were configured by the RRM in the way that they do not outrange an overall data rate of 67.6 kbit/s per user. Tab. 3 lists all transport format combinations that could be chosen by the MAC scheduler.
Fig. 5(a) and Fig. 5(b) present user data packet delay and throughput of the three applications. Voice as the prioritized service shows the same characteristics for delay and throughput as in simulations with no interference from other applications. The guaranteed bit rate of the voice application leads to higher delays for HTTP and FTP applications. 80% of all user data packets for HTTP objects are transferred in 20 seconds or less, but only 10 s or less were needed in the stand-alone scenario. Considering that a HTTP download consists of several objects, the overall time that is needed to download a WWW page increases heavily.
The influence of the prioritized voice application can easily be identified in the distribution functions of the throughput in Fig. 5(b). The precise breaks at half of the maximum throughput for HTTP and FTP applications result from the gradation of transport formats for the respective transport channels. If the mobile user speaks, the remaining data rate allows at most use of 2x338 bits per 20 ms for either HTTP or FTP. This represents exactly half of the maximum data rate. The maximum data rate itself can only be achieved if the mobile user does not speak and one of the remaining applications is not active.
The influence of the prioritized voice application can also be identified considering the number of transport blocks chosen by the MAC scheduler for the transport channel of the HTTP application (Fig. 5(c) and 5(d)). While for the uplink the propositions between the maximum number and the remaining number of transport blocks have changed, the "on-off" characteristic that had been observed in the downlink is no longer present.

Table 3
Transport Format Combination Set for the Application Mix

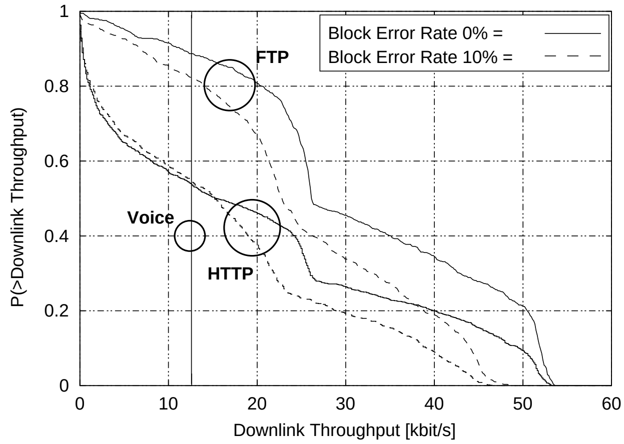
TFC	Transport Format		
	Voice $[kbit/s]$	HTTP $[kbit/s]$	FTP $[kbit/s]$
0	0	0	0
1	0	0	16.9
2	0	0	33.8
3	0	0	67.6
4	0	16.9	0
5	0	16.9	16.9
6	0	16.9	33.8
7	0	33.8	0
8	0	33.8	16.9
9	0	33.8	33.8
10	0	67.6	0
11	13.3	0	0
12	13.3	0	16.9
13	13.3	0	33.8
14	13.3	16.9	0
15	13.3	16.9	16.9
16	13.3	16.9	33.8
17	13.3	33.8	0
18	13.3	33.8	16.9

VII. Conclusion

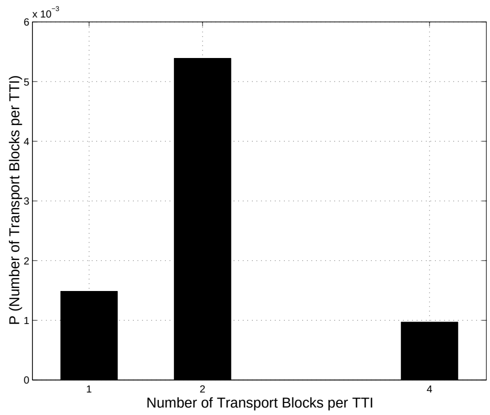
Running an application mix like examined in this paper will be an ordinary scenario during the introduction of UMTS. If the available scarce bandwidth resources shall be shared between applications, traffic adjusted transport format combinations have to be configured.
Simulations have shown that the proposed scheduler guarantees the strong delay and throughput constraints of the voice application. From the voice application point of view the radio interface behaves like a channel with fixed reservations.



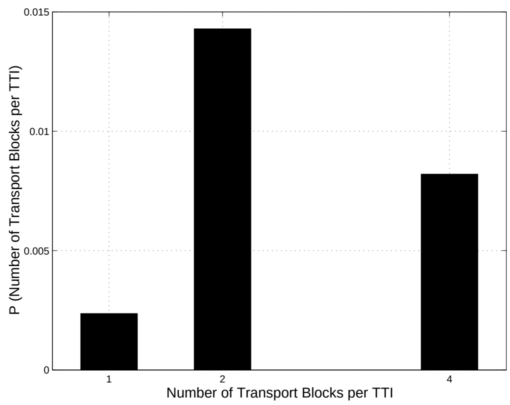
(a) User Data Packet Delay



(b) User Data Packet Throughput



(c) No. of Transport Blocks/TTI, 0% BER, HTTP, Uplink



(d) No. of Transport Blocks/TTI, 0% BER, HTTP, Downlink

Fig. 5. Simulation Results - Application Mix

For the applications of the interactive class appropriate transport formats were chosen. The scarce resource of capacity is efficiently used since during speakers silent times remaining capacity can be used by other applications in a very flexible manner.

Nevertheless efficiency depends on the available transport format sets and their granularity. Depending on the service characteristic and their actual mixture the RRM has the challenging task to compose appropriate transport formats combination sets. They have to be adapted and updated whenever load conditions or traffic mixtures will change.

References

[1] 3GPP TS 25.321, "Medium Access Control (MAC) Protocol Specification," Tech. Rep., 3rd Generation Partnerschip Project, Technical Specification Group Radio Access Network, 2001.

[2] 3GPP TS 25.322, "Radio Link Control (RLC) Protocol Specification," Tech. Rep., 3rd Generation Partnerschip Project, Technical Specification Group Radio Access Network, 2001.

[3] M. Allman, V. Paxson, and W. Richard Stevens, Eds., *RFC 2581: TCP Congestion Control*, Internet Engeneering Task Force, April 1999.

[4] Martin F. Arlitt and Carey L. Williamson, "A Synthetic Workload Model for Internet Mosaic Traffic", *Summer Computer Simulation Conference*, University of Saskatchewan, Ottawa, Canada, 1995.

[5] 3GPP TS 34.108, "Common Test Environments for User Equipment (UE) Conformance Testing," Tech. Rep., 3rd Generation Partnerschip Project, 2001.

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

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

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