Quality of Service Scheduling for UMTS

Silke Heier, Matthias Malkowski
Communication Networks
Aachen University
Kopernikusstr. 16
52074 Aachen, Germany
E-Mail: {she|mal}@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. In this paper, a scheduling algorithm for the Medium Access Control (MAC) is presented that satisfies these QoS requirements. The main task of a scheduler is the mapping of logical channels to appropriate transport channels in accordance to service requirements. Considering packet based services, the traffic characteristic is very dynamic due to its interactive and bursty nature. In consequence, a highly dynamic and flexible scheduling is required since the scarce radio resource should be used most efficient. The presented paper introduces a scheduler with dynamic channel type switching and Transport Format (TF) selection in accordance to the service QoS requirements. To validate the scheduling concept a typical mobile user application mix is examined for performance analysis by simulations.

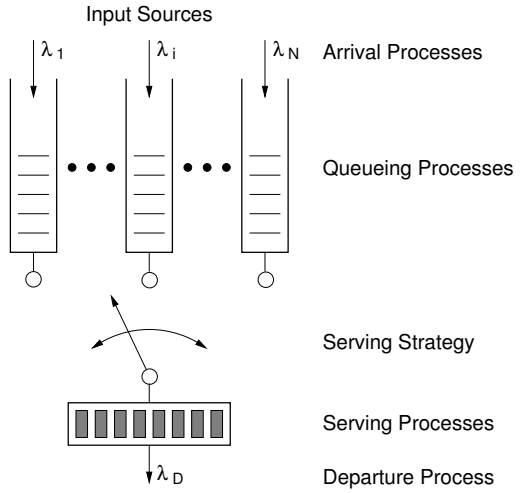


Figure 1: Input Queued Scheduling System

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 kbps in rural areas, 384 kbps in hot-spots and up to 2 Mbps in indoor scenarios. The use of several different services at the same time raises the demands for mechanisms to guarantee 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 benefit of the proposed concept is the highly dynamic switching of logical channels to transport channels on behalf of the actual load situation. 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. Simulation results will show which performance a mobile user will experience while executing a typical mobile user application mix.

II. MAC Scheduling Concept

UMTS supports parallel handling of multiple data streams arising from various applications. Applications belong to service classes (conversational, streaming, interactive, background) which require different QoS demands in terms of bit error rate, delay, throughput, etc. The MAC layer is responsible for the scheduling process considering the QoS requirements of each application. First it is outlined how queuing theory will be realized with the elements of the UMTS protocol stack (Fig. 1).

The arrival process determines the way packets are delivered to the queuing system by input sources. A typical parameter is the arrival rate λ_i. In our simulator the input sources correspond to the load generators. Each load generator has different distribution functions for packet size, the number of packets and the time the packets arrive. These distribution functions describe an arrival process that is specific to each type of traffic.

The queuing process describes how packets are stored and delivered for further processing. The queuing takes place in the *Radio Link Control* (RLC) layer of the UMTS protocol stack. The queuing processes in the RLC layer operate with a *First In First Out* (FIFO) scheme. The packet that is put into the queue first is delivered first, too. In the RLC the FIFO queues can be configured to be a loss system. To each packet a lifetime can be assigned. If the lifetime of the first packet in the queue is exceeded it is simply removed (discarded) from the queue.

The serving or scheduling strategy is responsible for the selection of the queue that should be processed next. In UMTS a static external priority, the *MAC Logical Channel Priority* (MLP), is assigned to each logical channel (i.e. each RLC queue). Since these priorities can not be dynamically changed, several algorithms remain that could be applied for the serving strategy. They can be divided into serving strategies that are capable of scheduling between queues of different priorities and those that require queues of the same priority. The following algorithms are able to schedule between queues of different priorities:

- Strict Priority: One designated queue is served if all queues of a higher priority are served before, i.e. all queues with a higher priority are empty.
- Rate-Controlled Priority: Every priority class has a given nominal rate. The serving strategy fetches as many packets from the queues as claimed by the rate. Only if lower prioritized queues are empty or have less packets queued than could be sent more packets as the nominal rate are transmitted. This serving method has the advantage that no queues starve due to the volume of high priority traffic.

The UMTS specification [1] prescribes a strict priority scheduling for radio bearers. For the queues that have the

same priority many algorithms may be applied. For our serving strategy we selected:

- Longest Queue First (LQF): The LQF algorithm uses the occupancy of the queues in the RLC layer which is a default parameter of the respective primitives. Because this algorithm aims at keeping all queues small the LQF algorithm has a big advantage if the space in the queues is limited. The average delay is also relatively low since the big queues are usually those that create high delay values. Nevertheless, the LQF algorithm is not fair since services that generate a high traffic load can block the processing of other queues.
- Queue Length based Weighted Fair Queuing (QLWFQ): The available channel capacity is split between the different queues according to their queue length. Compared to the LQF algorithm this algorithm has the advantage that no queues starve due to other filled queues.

The number of serving processes determine how many packets can be transmitted in parallel. This number can be derived from the selected TF. The number of *Transport Blocks* (TB) of a TF directly corresponds to the number of serving processes. The transmission duration of one packet is constant, so the processing time of each serving process is always the *Transmission Time Interval* (TTI) length of the regarded transport channel.

The departure process describes the rate of packets leaving the queuing system. The number of packets which leave the queuing system each TTI is determined by the selected TF. The rate of these packets can be calculated with the following equation:

$$\lambda_D = \frac{\text{Number of Transport Blocks}}{\text{TTI Length}}$$

In the UMTS protocol stack specific transmission parameters are assigned 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 (*Radio Bearer Mapping*, RBM),
- *MAC Logical Channel Priorities* (MLP) assigned to every logical channel,
- *Transport Format (Combination) Sets* (TFS, TFCS).

The RBM consists of mapping options for logical channels. Up to eight options may coexist for each logical channel. Every mapping option contains a transport channel identification, the allowed TFs and the MLP.

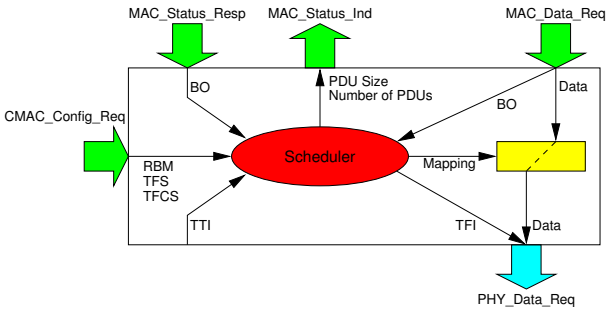


Figure 2: MAC Scheduler

Our proposed MAC scheduler uses MLPs to provide a priority scheduling between applications of different QoS classes. This will guarantee delay requirements of applications of the conversational and streaming class. TFC selection as part of the MAC scheduling is performed based on buffer occupancies in order to guarantee the required traffic throughput. A LQF and a QLWFQ scheduling is used to cover applications of the same priority. The scheduling is triggered by the TTI. The selected mappings of logical channels to transport channels and the TFs can be changed every TTI. Fig. 2 depicts incoming and outgoing parameters of the MAC scheduler. In our simulation environment the full functionality of the MAC layer is emulated in conformance to [1].

III. MAC Scheduling Algorithm

The proposed scheduler compares all possible scheduling options with predefined rules to find the best one. Since not all scheduling options are always available, the scheduler starts with finding valid ones. Therefore, the solution space is reduced to solutions that are valid and can be used in the next TTI. The solution space is spanned by mapping options of logical channels and TFCs of transport channels that are multiplexed on a *Coded Composite Transport Channel* (CCTrCH):

$$\prod_{i=1}^{m} a_i \cdot \prod_{j=1}^{n} b_j$$

where m is the number of logical channels, a_i is the number of mapping options for logical channel i, n is the number of TFCS and b_j is the number of TFC in the TFCS with the number j. Since the scheduler has to compare these solutions the complexity of the algorithm is:

$$O(a_1 a_2 ... a_m b_1 b_2 ... b_n)$$

This notation can be simplified by taking the geometric mean of the products:

$$O(\bar{a}^m \bar{b}^n)$$

Internally the solution space is spanned by a tree where the leaves in the lowest level are valid solutions of the scheduling problem. The branches that are near the root of the scheduling tree correspond to mapping options. Each level refers to one logical channel. The branches near the leaves of the tree distinguish between applicable TFCs. Here, each level is in relation to one TFCS. Fig. 3 shows a simplified scheduling tree. Reasons for leaves that are not in the lowest layer of the tree are mapping options or TFC that are not valid at scheduling time. A backtracking algorithm with the exponential complexity mentioned above is used to find valid leaves of the tree.

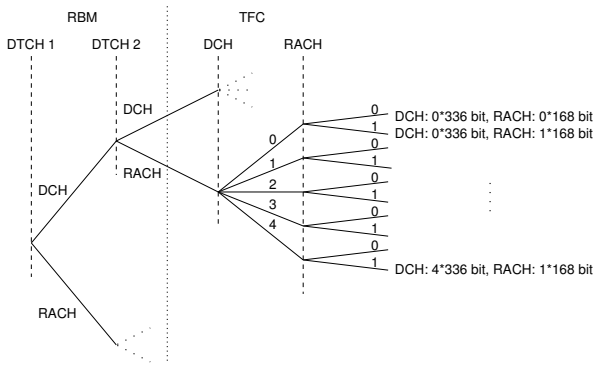


Figure 3: Scheduling Tree

An algorithm with a non-polynomial complexity is usually not acceptable. For typical UMTS cases this circumstance is less critical since most of the variables are bounded to low values. For example, if a user equipment in FDD mode is considered, there is only one uplink CCTrCH so n is one. Although there may be a lot of logical channels but the number of valid mapping options is typically one because of a single CCTrCH. Hence, the variables a_i are set to one which leads to a linear complexity of

$$O(b_1).$$

A. Scheduling Example

In this section an exemplary scheduling illustrates the operation of the scheduler. The scheduling is structured into four parts.

1) Restriction of RBM: The restriction of possible mapping options is the first part of the backtracking algorithm. Fig. 4 gives an example of mapping options for one logical channel. Here, DTCH 1 can either be mapped onto FACH, DSCH, DCH 1 or DCH 2. The mapping to DCH 1 was selected during the last TTI. Mappings cannot be switched if the considered channel is in an enduring TTI. Hence, this fact prevents the scheduler from selecting DCH 2. If DCH 2 would be the channel currently selected, the scheduler could not switch the mapping at all because of the ongoing 20 ms TTI. Further reasons of invalid mapping options are given by the RRC protocol. In our example a mapping to the FACH is not allowed because the user equipment is in *CELL_DCH* state.

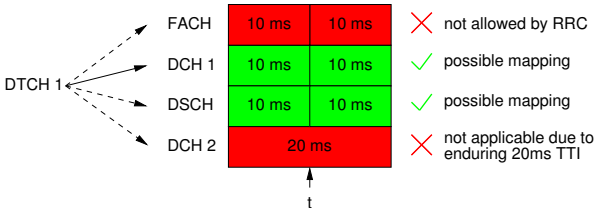


Figure 4: Mapping Options

2) Restriction of the TFCS: The second part of the backtracking algorithm is to find the valid TFCs. Fig. 5 shows a TFCS with TFCs for two transport channels. DCH 1 is a 256 kbps data channel and DCH 2 is a 12.2 kbps channel used for voice communication. Several reasons limit the selection of TFCs. Mapping options can not be changed in an enduring TTI. Therefore the TFs of DCH 2 can not be changed and half of the TFCS (TFC 0-4) can not be selected. The mapping selection may also restrict the TFCS. Every mapping option has a list of allowed TFs. The TFC that contains not allowed TFs have to be ignored (TFC 9). Beside these restrictions, single TFC may not be allowed for selection because of external reasons, e.g. transmission conditions (TFC 7-8). The restrictions are configured by the RRM.

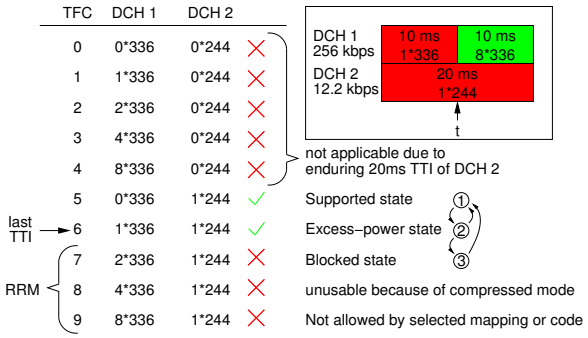


Figure 5: Transport Format Combination Set

3) Multiplexing: For all scheduling options found by the backtracking algorithm the scheduler calculates a possible multiplexing. An example is shown in Fig. 6 for one transport channel. The selected TF would allow the transmission of one 336 bit TB.

During the multiplexing, scheduling strategies are applied. Since logical channels have different MLPs and priority scheduling is applied, the multiplexing function assigns available TBs to the logical channel with the highest priority and a non-empty buffer, here DTCH 2. Because DTCH 2 has a BO of 500 bits all 332 bits of a RLC SDU are accounted as user data that could be transmitted. If the buffer had contained less bits than to be transmitted the lost space would be accounted as padding.

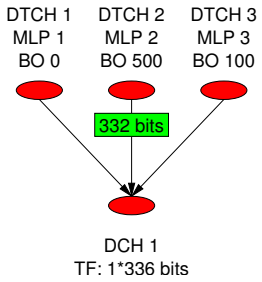


Figure 6: Multiplexing for different Priorities

If all queues are empty and there are still TB available, the unused space of these TB is also accounted as padding. MAC header overhead is accounted negatively, too.

Fig. 7 shows another multiplexing situation. Three logical channels which have equal priorities are mapped onto the same transport channel. Therefore the LQF algorithm is used that selects the logical channel with the highest BO. In the example this is DTCH 2. If there are more TBs available after this selection, the LQF algorithm starts from the beginning and selects the queue which has the next highest BO. The accounting of data that would be transmitted and the calculation of the amount of padding is the same as for priority scheduling. If the QLWFQ algorithm is applied, the number of PDUs that would be fetched from each queue is directly proportional to the actual BO.

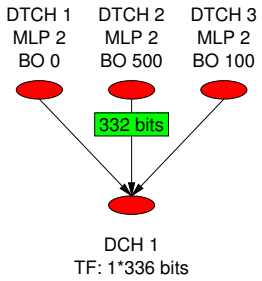


Figure 7: Multiplexing for the same Priority

4) Rating of Scheduling: After the scheduler found a valid set of scheduling options and performed the multiplexing, the scheduling solution is compared to the last one found. The following criteria, ordered by their importance, are applied for this comparison:

1) more bits of a higher priority,
2) less padding required.

Tab. 1 gives an example of the benchmarking of some scheduling solutions. Solution *B* is better compared to solution *A* because there are more bits of the higher priority transmitted. For the same reason solution *C* is preferred before solution *B* despite there is more padding generated. In solution *D* there is one unused TB less transmitted, so criterion two gives a positive benchmarking for this solution. At last solution *E* is considered. This solution has even less unused padding space because of another TF on one of the transport channels.

Solution	MLP 1	MLP 2	Padding	Ranking
A	0	332	332	5.
B	332	0	332	4.
C	332	100	564	3.
D	332	100	232	2.
E	332	100	64	1.

Table 1: Rating of Scheduling Solutions

With the procedure described in the last paragraphs the best scheduling solution regarding the criteria above remains after the backtracking algorithm has terminated.

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	$\log_{10} 4 \approx 0.6$	$\log_{10} 2.55 \approx 0.4$

Table 2: Model Parameters of HTTP Browsing Sessions and FTP Sessions

By changing the multiplexing function or the criteria in the benchmarking function completely different scheduling algorithms can be implemented very quickly.

IV. Simulation Results

To examine the MAC scheduler performance for data services like HTTP and FTP, traffic models according to Tab. 2 are used [2]. For both applications the session arrival rate is high in order to simulate a high load scenario. For performance analysis of the scheduling concept a multiplexing scenario of HTTP and FTP is chosen. The transport formats that are assigned to the transport channel provide a maximum data rate of 67.2 kbps. The simulation parameters shown in Tab. 3 are configured due to recommendations taken from [3].

The modern Internet is based on TCP/IP. The TCP implementation realized in URIS is the so called "Reno" TCP stack. The protocols of the radio interface like MAC, RLC and *Packet Data Convergence Protocol* (PDCP) are implemented completely bit accurate in conformance to their specifications. Hence, URIS uses a protocol emulation for performance evaluation. Please refer to [4–6] for a detailed description of the simulator, the load generators and the radio interface protocol emulation. Even other simulation results are shown and discussed in these papers.

Here the *Buffer Occupancies* (BO) are shown to illustrate the dequeuing process of the scheduling strategies. The BO is defined as amount of data queued in the RLC transmission buffer at the time the scheduler requests the BO for transmission planning. Both RLC control information as well as TCP packets that have to be retransmitted are included in the transmission buffer.

In case of priority scheduling (Fig. 8(a)) the distribution of the buffer occupancy for HTTP indicate that the buffers are dequeued very fast. Since a higher priority was assigned to HTTP traffic, FTP traffic does not influence this distribution. Due to the priority assignment, 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. Due to the blocked FTP traffic, HTTP objects have a higher throughput than FTP objects (Fig. 8(b)). The maximum user data throughput achieves 57.8 kbps which is 86% of the transport channel capacity. The rest is protocol overhead.

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

Table 3: Simulation Parameters

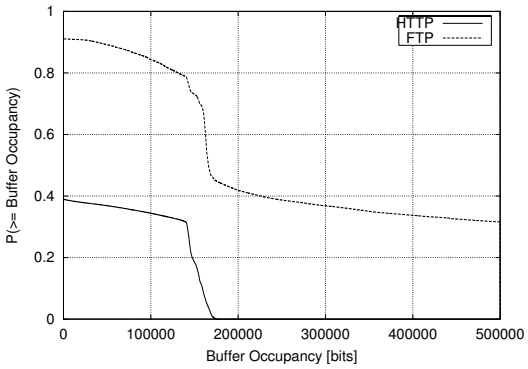
Fig. 8(c) shows the buffer occupancy distributions for the LQF scheduling. Since both applications have the same priority the LQF algorithm aims at keeping both buffer occupancies equal and as low as possible. This scheduling strategy shows a slower dequeuing process since the curves are not rectangular. The slanting decline of the curves is caused by TCP retransmission timeout and the resulting congestion avoidance of the TCP protocol. TCP reduces the transmitting window and starts its congestion avoidance mechanism. The TCP transmit window size increase slowly which results in an reduced buffer occupancy in the RLC layer. The throughput in Fig. 8(d) shows that the LQF algorithm prefers the application which generates most of the traffic, here FTP. Nevertheless both services block each other and TCP reacts with its congestion avoidance algorithms. Due to that both application suffer in terms of throughput since TCP triggers retransmissions which burdens the radio interface. Ordinary user data packets experience long waiting times until the retransmissions are transfered correctly.

The QLWFQ algorithm (Fig. 8(e)) has a limited maximum BO, too. Since both buffers are equally filled with around about 100 kbit in 90% of the time each application will mostly experience half of the overall channel capacity. TCP detects congestion for both applications and reduces its transmission window accordingly. This can be seen at the declension of the BO curves between 100 kbit til 200 kbit. Fig. 8(f) shows the user data throughput. Since the channel capacity is split between both services the capacity assignment of the scheduler changes very quickly especially if new objects have to be transfered. TCP is not able to set its parameters like retransmission timeout and round-trip delay correctly. TCP detects many congestions but its congestion algorithms are to slow to follow the dynamic capacity assignment. A huge amount of TCP retransmission burdens the radio interface that increases TCP delays significantly.

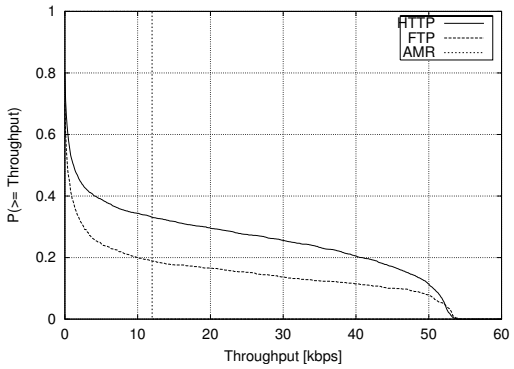
V. Conclusion

Running an application mix will be an ordinary scenario during the introduction of UMTS. If the available scarce bandwidth resources shall be shared between applications, a dynamic and efficient MAC scheduling is required to cope with the bursty nature of Internet applications. Depending on the assigned TFCS by the RRM our proposed MAC scheduler will always select the most efficient solution in accordance to the required QoS.

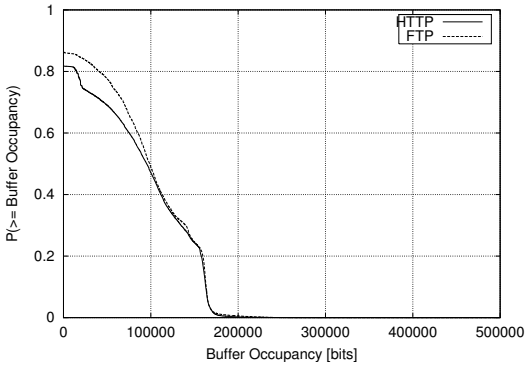
Nevertheless simulations have shown that TCP cannot cope with a quick MAC scheduling. TCP's flow control



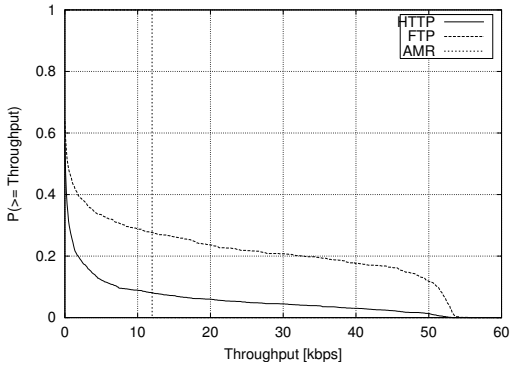
(a) Priority Scheduler - Buffer Occupancy



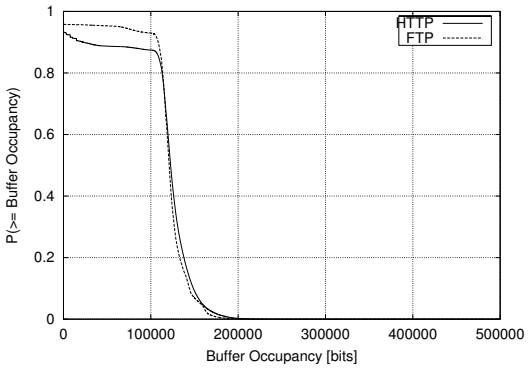
(b) Priority Scheduler - User Data Throughput



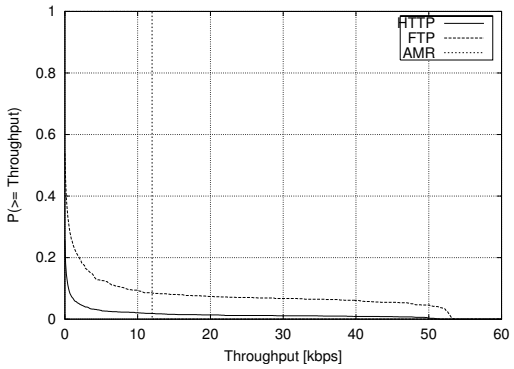
(c) LQF Scheduler - Buffer Occupancy



(d) LQF Scheduler - User Data Throughput



(e) QLWFQ Scheduler - Buffer Occupancy



(f) QLWFQ Scheduler - User Data Throughput

Figure 8: Simulation Results

mechanisms are too slow for a dynamic radio link. The guaranteed capacity of the UMTS radio interface is not used efficiently since TCP burdens the radio link with unnecessary retransmissions or leaves capacity unused. An adaptation of TCP for the mobile world is mandatory to guarantee an efficient use of the radio link.

References

[1] 3GPP TS 25.321, "Medium Access Control (MAC) Protocol Specification," Technical Specification V4.1.0, Release 4, 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, June 2001.

[2] Victor S. Frost and Benjamin Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, March 1994, pp. 70–81.

[3] 3GPP TS 34.108, "Common Test Environments for User Equipment (UE) Conformance Testing," Technical Specification V4.0.0, Release 4, 3rd Generation Partnership Project, June 2001.

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

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

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