Quality of Service of Internet Applications over the UMTS Radio Interface

Silke Heier, Andreas Kemper, Sebastian Gräbner, Jan-Oliver Rock
Communication Networks
RWTH Aachen, Germany
{she|kem|sgr|jor}@comnets.rwth-aachen.de
http://www.comnets.rwth-aachen.de/~{she|kem}

Abstract The continuing evolution of exponential growth of the Internet market and growing demand of data services in fixed networks is also expected for mobile users. *Universal Mobile Telecommunications System* (UMTS) as the 3rd generation mobile system is designed to provide high bit data rates to the user. The mobile Internet access will play a key role to ensure success of UMTS introduction. Typical Internet applications are running on a *Transmission Control Protocol/Internet Protocol* (TCP/IP) stack which should ensure a reliable end-to-end communication in systems with limited quality of service guarantee. In contrast, most protocols of radio access systems will grant traffic contracts. This raises the question if the benefits of TCP might be a drawback for the overall performance of the radio interface with its own quality of service management mechanisms. The aim of this paper is to examine the interaction between TCP and the UMTS *Radio Link Control* (RLC) protocol. Of special concern is the interaction of TCP retransmissions with the RLC *Automatic Repeat Request* (ARQ) mechanisms. Simulations have been performed to evaluate the performance of the implemented protocol stack with and without TCP. Simulation results of quality of service parameters like delay and packet throughput depict the performance of mobile Internet access over UMTS.

Keywords: UMTS, UMTS Radio Access, Radio Interface Protocols, WWW, Quality of Service, TCP/IP, RLC

1. Introduction

The aim of *Universal Mobile Telecommunications System* (UMTS) is the provisioning of high bit rate data services to the mobile user. Hence, typical Internet applications like *World Wide Web* (WWW) browsing will migrate from fixed access network systems to the mobile environment. Normally these applications are running on a *Transmission Control Protocol/Internet Protocol* (TCP/IP) protocol stack which guarantees a reliable end-to-end communication. Nevertheless, these protocols will not ensure quality of service in terms of delay and throughput. In most cases the customer faces best-effort resource management strategies in fixed networks. This might be applicable since over-provisioning of bandwidth is feasible in fixed networks.

Due to the nature of radio links, the radio interface protocols have to cope with limited bandwidth, higher delays and error rates. Therefore, the UMTS radio interface protocols provide various and highly complex functions and mechanisms to realize a reliable communication. Traffic contracts and quality of service requirements are supported if desired.

This paper demonstrates the opportunities of the UMTS *Radio Link Control* (RLC) protocol to support quality of service. RLC provides a reliable link with ARQ mechanisms. Both, TCP and RLC protocol are using retransmissions to recover lost data packets. Running an Internet application over UMTS rises the question if it is necessary to use TCP. On the one hand side, protocol overhead burdens the radio interface with its limited bandwidth. On the other hand side, TCP retransmissions and timeouts may interfere with the *Automatic Repeat Request* (ARQ) of the RLC protocol. Another opportunity is running a TCP application over an unacknowledged RLC connection. Since TCP is not designed for radio access, it has to be shown how it will perform in an environment with high error rates and delays.

The aim of this paper is to present performance results with the help of a UMTS Radio Interface Simulator. The performance evaluation considers the detailed UMTS protocol implementation as well as the traffic load generation for WWW browsing sessions including TCP/IP modeling. Thus, it is feasible to present the performance of a TCP application in an unreliable mobile environment. The simulation results will show the performance in terms of quality of service parameters a mobile user will experience while surfing the Internet.

Following this introduction, Sec. 2 gives a description of the traffic model. Furthermore it presents the used simulation methodology and the simulation environment. Sec. 3 and Sec. 4 illustrate the quality of service mechanisms of the TCP and the RLC protocol. Performance evaluation, simulation scenarios and discussion of simulation results is depicted in Sec. 5 and Sec. 6. The paper concludes in Sec. 7 with a summary.

2. Simulation Environment

A WWW browsing session is a typical application which is running on the TCP/IP protocol stack. A WWW traffic model is necessary for simulative examinations of the performance of data services of mobile radio networks. 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. During a page request, several packets for each object may be generated which means that the page request constitutes of a bursty sequence of packets. The bursty-Ness during the page request is a characteristic feature of packet transmission. After the page has entirely arrived at the terminal, the user is consuming certain amount of time for studying the information. This time interval is called reading time. Tab. 1 gives an overview of the used WWW traffic model described by parameters and their distributions.

Related documentation can be found in Frost and Melamed, 1994; Arlitt and Williamson, 1995; Paxson, 1994. The main part of the later implementation is based on the work of Arlitt and Williamson, 1995.

For this study the simulation tool *UMTS Radio Interface Simulator* (URIS) was developed at the Chair of Communication Networks. This simulation environment is used to investigate, optimize and develop features of the radio interface protocol stack. In addition, it offers the opportunity of capacity and quality of service evaluation by simulations of various scenarios. The simulator is a pure software solution in the programming language C++. The simulation model is implemented with the help of a powerful C++ class library which was developed by the Chair of Communication Networks and is called the *SDL Performance Evaluation Tool Class Library* (SPEETCL) Steppler and Lott, 1997. This generic, object oriented library is well suited for telecommunication network simulation purposes and can be used in event driven, bit accurate simulation environments. The UMTS protocols at the radio interface enhanced by a TCP/IP protocol stack were specified with the *Specification and Description Language* (SDL). To generate an executable out of the SDL phrase notation and the C++ library, a SDL2SPEETCL code generator is used.

The software architecture of the URIS simulator is shown in Fig. 1. Up to now 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 the transmission of bursts in radio frames on the radio interface. This includes discarding of erroneous bursts depending on the error model.

The core of the simulator are the modules *User Equipment* (UE) and *UMTS Terrestrial Radio Access* (UTRA) which are built formally similar. Each UE and UTRA is implemented as an SDL system which contains the protocol implementation of the layers. Fig. 2 gives an overview of the protocol structure in a generic SDL system. The complex protocols like *Medium Access Control*

Table 1. Model Parameters of WWW Browsing Session

WWW Parameter	Distribution	Mean	Variance
Session Arrival Rate [h^{-1}]	neg. exp.	2.0	—
Pages per Session	geometric	20.0	—
Reading Time between Pages [s]	neg. exp.	33.0	—
Objects per Page	geometric	2.5	—
Inter Arrival Time between Objects [s]	neg. exp.	0.5	—
Page Request Size [byte]	normal	1136.0	80.0
Object Size [byte]	$\log_2$-Erlang-k	≈ 11.3	5.4

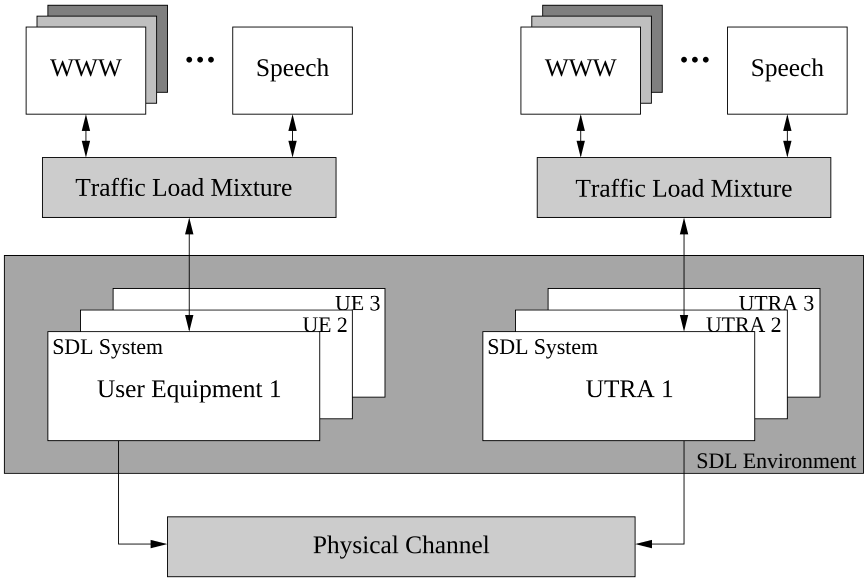


Figure 1. Structure of the URIS Simulator

(MAC), RLC, *Packet Data Convergence Protocol* (PDCP) and *Radio Resource Control* (RRC) based on UMTS Release 4 and the TCP/IP and *User Datagram Protocol* (UDP)/IP are specified in SDL with object oriented methods. For simulation purposes, the TCP/IP protocol stack can be easily bypassed by switching the transport and the network layer to transparent transmission.

Usual simulator approaches model protocols and functions on the basis of abstractions and simplifications. The aim of URIS is a detailed, bit accurate implementation of the standardized protocols. This offers the opportunity to determine the performance of UMTS in a realistic manner.

3. The TCP/IP Protocol Suite

The modern Internet is mainly based on TCP and IP. In order to provide Internet services to the mobile user, the TCP/IP protocol suite has to be adapted to the UMTS protocols. The UMTS radio interface protocol stack offers this functionality by providing the PDCP.

TCP provides the overlying protocols with a reliable connection. The following services are offered:

- Retransmission of lost packets,
- Recovery from out-of-order delivery,
- Flow control.

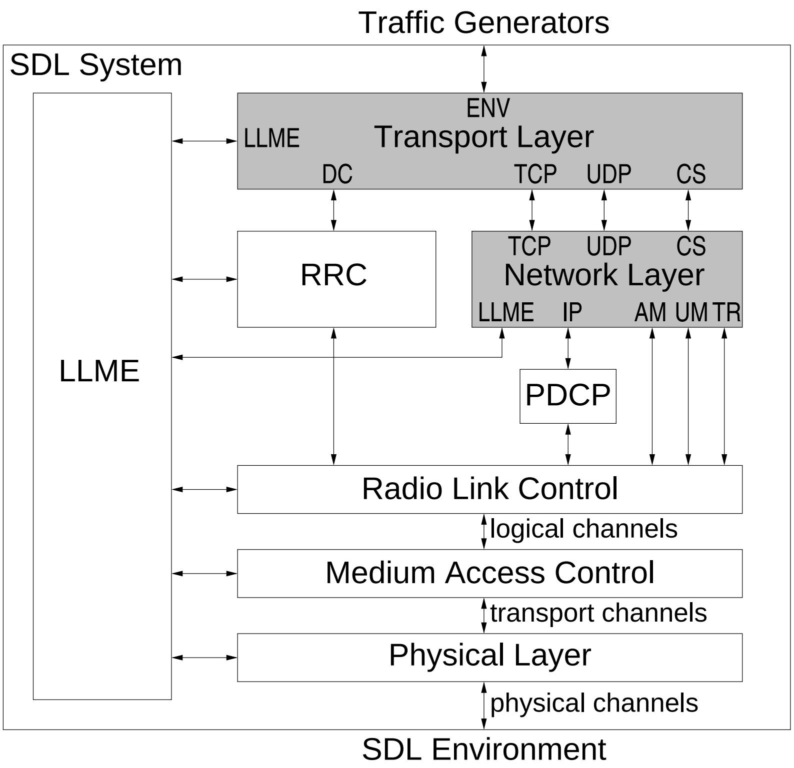


Figure 2. URIS SDL System Structure

The TCP implementation realized in URIS is based on the so called “Reno” TCP stack and uses the following flow control mechanisms:

- Slow start and congestion avoidance,
- Fast retransmit and fast recovery,
- Delayed acknowledgments,
- Selective acknowledgments Mathis et al., 1996; Fall and Floyd, 1996.

IP is a connection-less and unreliable protocol that operates on a best-effort basis. This means that IP packets may be lost, out-of-order or even duplicated without IP handling these situations. This has to be done by higher layers. In URIS, the IP protocol implementation currently performs data encapsulation.

4. Radio Link Control Protocol

The RLC realizes segmentation and retransmission services for both user and control data. The RLC protocol provides three different data transfer services:

- *Transparent data transfer service mode* (TR),
- *Unacknowledged data transfer service Mode* (UM),
- *Acknowledged data transfer service Mode* (AM).

The TR mode is an unidirectional service typically used for broadcast or paging services, where it is not necessary to guarantee an error-free transmission. A use for transmission of streaming data (e.g. audio or video) is also feasible, especially if real-time transmission is more important than error reliability. A dropping mechanism prevents delivery of already expired *Protocol Data Units* (PDUs).

The UM is an unidirectional service typically used for streaming applications (streaming class) where it is not necessary to guarantee an error-free transmission. Voice over IP (conversational class) is also a feasible service conveyed by the UM. The UM transmits higher layer data packets without guaranteeing delivery to the peer entity. By using a sequence-number check function in the receiving entity, the UM is capable of detecting missing RLC PDUs, but error recovery is not performed. A dropping mechanism prevents transmission of already expired PDUs.

The AM transmits higher layer data packets with guaranteed delivery to the peer entity. Therefore it is mainly used for traffic of the interactive class or background class type. ARQ mechanisms are applied for correction of transmission errors. It is possible for the higher layers to request a transmission confirmation from the AM. A Go-Back-N ARQ has been implemented as a first approach for the AM. Segmentation and reassembling as well as concatenation are fully implemented as specified in 25.322, 2001.

5. Simulation Scenarios

The main parameters concerning *Quality of Service* (QoS), which are affected by RLC and TCP/IP, are delay and throughput. Two measurements are made during the simulation in both the RLC layer and the transport layer:

1 Packet Delay: time from sending a traffic load packet to the respective layer until correct reception of the packet by the traffic load receiver,

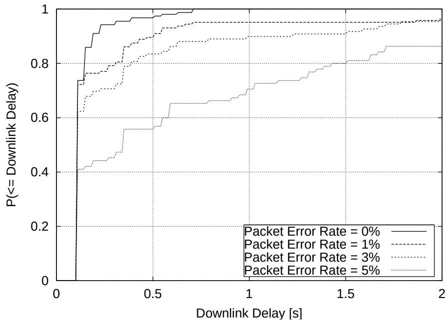
2 Packet Throughput: size of the traffic load packet divided by the packet delay. The throughput is given in bytes per second [byte/s].

Three types of simulation scenarios have been executed:

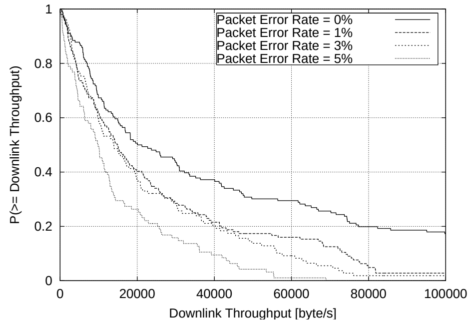
1 RLC AM without TCP/IP,

2 TCP/IP with RLC AM,

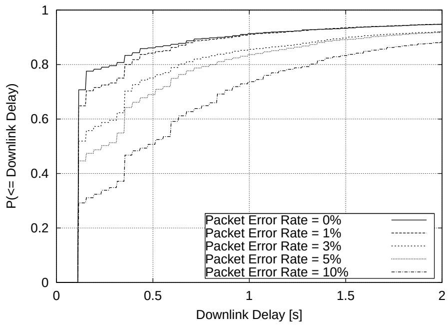
3 TCP/IP with RLC UM.



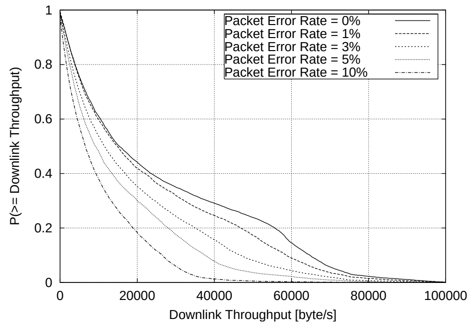
(a) Traffic Load Packet Delay with RLC AM and without TCP/IP



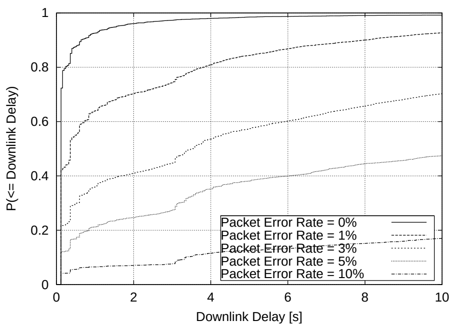
(b) Traffic Load Packet Throughput with RLC AM and without TCP/IP



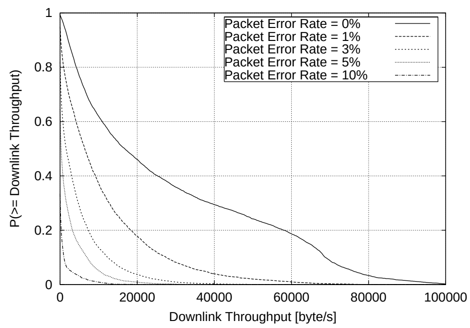
(c) Traffic Load Packet Delay with RLC AM and TCP/IP



(d) Traffic Load Packet Throughput with RLC AM and TCP/IP



(e) Traffic Load Packet Delay with RLC UM and TCP/IP



(f) Traffic Load Packet Throughput with RLC UM and TCP/IP

Figure 3. Simulation Results for Traffic Load Packet Delay and Throughput

The cumulative distribution function of the measured packet delay and the complementary cumulative distribution function of the measured packet throughput have been calculated. Measurements have been made for increasing packet error rates from 0% up to 10%. Tab. 2 shows the parameters of RLC and TCP/IP used for these simulations.

6. Simulation Results

The results of the simulations are given in Fig. 3. As can be seen a minimum delay is always present, caused by the constant transmission delay of the physical layer. Packet error rates of 0% represent the ideally achievable throughput. The throughput is not limited by the underlying dedicated channel capacity since the offered load is always below 2 Mbit/s.

Set 1: RLC Acknowledged Mode without TCP/IP

The first scenario evaluates the capabilities of the implemented Go-Back-N ARQ of the RLC AM without interference of the TCP/IP protocols. The data packets generated by the load generator are sent directly to the RLC protocol. The results of this scenario are shown in Fig. 3(a) and Fig. 3(b). The distinct steps of the delay curves in size of one *Transmission Time Interval* (TTI) length are caused by the retransmission requests.

As expected, the packet delay and packet throughput degrade while the packet error rates increase. At packet error rates up to 3%, 90% of the pack-

Table 2. Simulation Parameters

Parameter	Value
TTI Length	0.04 s
Dedicated Channel Capacity	260 kbyte/s
RLC TxWinSize	1024 PDUs
Max. No. of PDU Retransmissions	40
Status Prohibit Timer	0.08 s
Poll Timer	1.0 s
Maximum TCP Segment Size	512 byte
Maximum TCP Send Window	16 kbyte
Min. TCP Retransmission Timeout	3 s
Max. TCP Retransmission Timeout	64 s
Delayed Acknowledgment	Not used
Selective Acknowledgment	Not used
Header Compression	Not used
Packet Error Rate	0%, 1%, 3%, 5%, 10%

ets have a delay of less than 1 s. The throughput is approximately 2/3 of the throughput at 0% error rate. For a 5% packet error rate, the Go-Back-N mechanism is no longer able to provide sufficient data throughput. Due to the unnecessary Go-Back-N retransmissions of correctly received packets even more transmission errors are caused and congestion occurs within the RLC protocol.

To avoid unnecessary retransmissions and congestion a Selective-Reject-ARQ (SR-ARQ) mechanism should be used. The SR-ARQ is currently under study. It is expected that the performance will be better but a Go-Back-N ARQ as a first approach is sufficient to study the general behavior.

Set 2: TCP/IP with RLC Acknowledged Mode

The RLC AM has been used to investigate the interaction of the TCP protocol with the RLC ARQ. Fig. 3(c) shows the results of the measured TCP delay for several packet error rates. The distinct steps of multiple TTIs are caused by retransmissions on RLC protocol level. As expected, the delay imposed by the TCP protocol and the underlying layers increases for higher packet error rates. At packet error rates up to 5%, 90% of the packets have a delay of less than 1.5 s. Fig. 3(d) shows the respective throughput. As a result of higher delays, lower data throughput is achieved.

Compared to the first scenario, the delays of traffic load packets are higher and the peak throughput is lower. This is caused by two effects. First, TCP adds protocol overhead which results in a higher gross data rate that has to be transmitted over the radio interface. Second, the window mechanism of TCP realizes a flow control of bursty WWW traffic where packets have to wait since the send window is closed. Overhead and window waiting times produce additional delays and lower throughput. Even if the radio link layer guarantees a higher peak throughput (see Fig. 3(b)) the TCP flow control mechanism is too slow to use the guaranteed bandwidth efficiently. RLC radio bearer capacity is left unused since the TCP send window is closed due to bursty traffic.

An interesting effect has been observed at high loss rates (10%). Due to the heavy losses on the radio link, the Go-Back-N ARQ mechanism of the RLC must perform many retransmissions. In the mean time the *Retransmission Timeout* (RTO) for the transmitted TCP segments expires which leads to a retransmission by the TCP protocol. In this case the delays get even worse since data packets will be send twice on different protocol layers which increases the traffic unnecessarily. As a result, duplicate TCP packets will arrive at the TCP receiver since the original packets are delivered by the RLC ARQ first and second by TCP retransmission. Resulting duplicate acknowledgments are the reason why TCP's fast retransmit algorithm is triggered. By mistake, TCP assumes congestion and reduces the throughput which results in unused but guaranteed RLC radio bearer capacity.

Set 3: TCP/IP with RLC Unacknowledged Mode

The RLC UM has been used to evaluate the performance of TCP's retransmission mechanisms. Fig. 3(e) depict the TCP delay which shows higher delays with growing packet error rates. It can be noted that the steps of the curves mark the expiration of a TCP RTO. In all simulations a minimum RTO of three seconds has been used. It is obvious that the retransmission mechanism of TCP is far too slow to handle the heavy packet losses of a radio link. Fig. 3(f) gives the respective throughput which clearly shows the heavily reduced throughput for higher error rates. Even if the radio link layer guarantees high radio bearer capacity the TCP retransmission mechanism is too slow to use the guaranteed bandwidth efficiently. RLC radio bearer capacity is left unused since the TCP send window is closed due to bursty traffic, missing acknowledgments and retransmission timeouts.

7. Conclusion

This paper comprises simulative examinations of the performance of WWW surfing over the UMTS radio interface. The simulation results show that a WWW session using TCP and the RLC UM is not feasible to satisfy a mobile user. If packets are lost the TCP protocol infers that there must be congestion in the network between the two peers. In consequence, a TCP retransmission leads to a decrease of the send window size which results in less throughput. This assumption is correct in wired environments but TCP is unable to handle the unreliable radio link of mobile users due to the high error rates and TCP's slow error handling. Very high delays and small throughput lead to unsatisfied mobile users. Improvements may be archived by using the TCP SACK option which allows the receiver to convey more information about its state to the sender, thus increasing the efficiency of the retransmission strategy.

Running a WWW session over a standard TCP and the RLC AM will be an ordinary scenario during the introduction of UMTS. First UEs will rely on standard application solutions which will be adopted from the fixed world. Hence, TCP will run end-to-end including the radio interface. This paper shows that this solution is applicable to satisfy the mobile user but the performance suffers since TCP mechanisms will not efficiently use the guaranteed QoS of the UMTS radio bearers in terms of delay and throughput. It has to be mentioned that effects of the core network are not included in our simulations. This is not only due to currently missing simulation capabilities but also partly unknown structure of the core network. While it is hard to model the behaviour of packets traveling along the internet, it can't even be assumed that the entire core network is IP-based. Thus it can be concluded that the overall performance will get worse since TCP as an end-to-end protocol will experience additional delays and congestion in the core network. Using heavily disturbed radio links the mobile user will face a reduced throughput and longer waiting times for WWW page downloads than in fixed networks. This relies on the weak TCP performance in a mobile environment and can not be influenced by assignment of radio bearers with higher capacity.

The simulation results of a WWW session running without TCP/IP in RLC AM show the best performance. As a result of all simulations, future UEs are suggested where typical packet data applications should run without a standard TCP at the radio interface. The plain UMTS radio interface protocol stack offers reliable radio bearers with sufficient QoS for WWW browsing sessions. To face the requirements of the Internet world, TCP might terminate or start running end-to-end in the UMTS core network.

Biographies

Silke Heier and Andreas Kemper are Ph.D students at the Chair of Communication Networks. Together with her students, including Sebastian Gräbner and Jan-Oliver Rock, Mrs. Heier provided significant improvements to the UMTS protocol simulator URIS. While she is almost finished with her degree, mainly her students implemented layers two and three under her supervision. Andreas Kemper is her successor with respect to further developments on the simulator. Due to the current status of implementation, his focus is mainly on the development of physical layer and radio resource control (RRC) routines. The horizon in development appears to be the coupling of this protocol simulator with a system-level simulator to provide more detailed information on channel quality, resulting for instance from propagation and interference situation.

Acknowledgment

The authors would like to thank Prof. B. Walke of the Chair of Communication Networks for his support and friendly advice to this work.

References

25.322, G. T. (2001). Radio Link Control (RLC) Protocol Specification. Technical Specification V4.1.0, Release 4, 3rd Generation Partnerschip Project, Technical Specification Group Radio Access Network.

Arlitt, M. F. and Williamson, C. L. (1995). A Synthetic Workload Model for Internet Mosaic Traffic. Proceedings of the 1995 Summer Computer Simulation Conference, Ottawa, Canada, July, pp. 24–26.

Fall, K. and Floyd, S. (1996). Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *Lawrence Berkeley National Laboratory*.

Frost, V. S. and Melamed, B. (1994). Traffic Modeling For Telecommunications Networks. *IEEE Communications Magazine*. pp. 70–81.

Gräbner, S. (2001). Simulative Performance Evaluation of the UMTS Radio Link Control Protocol. Master's thesis, Chair of Communication Networks, RWTH Aachen University of Technology.

Holma, H. and Toskala, A., editors (2001). *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. John Wiley & Sons Ltd.

Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A. (1996). Selective Acknowledgement Options. RFC, Internet Engeneering Task Force.

Paxson, V. (1994). Empirically-Derived Analytic Models of Wide-Area TCP Connections. *IEEE/ ACM Transactions on Networking, 2 (4), pp. 316–336, August 1994; ftp://ftp.ee.lbl.gov/papers/WAN-TCP-models.ps.Z.*

Rock, J.-O. (2001). Simulative Performance Evaluation of IP based Services over UMTS. Master's thesis, Chair of Communication Networks, RWTH Aachen University of Technology.

Steppler, M. and Lott, M. (1997). SPEET - SDL Performance Evaluation Tool. *8th SDL Forum '97*.

Walke, B. (2001). *Mobile Radio Networks, 2nd Edition*. John Wiley & Sons Ltd., Chichester, England.