Performance Evaluation of Internet Applications over the UMTS Radio Interface

Silke Heier, Dirk Heinrichs, Andreas Kemper
Communication Networks, Aachen University of Technology (RWTH), Germany
E-Mail: {she|dih|kem}@comnets.rwth-aachen.de
WWW: http://www.comnets.rwth-aachen.de/˜she

***Abstract*— Universal Mobile Telecommunications System (UMTS) as a 3rd generation mobile telecommunications system is designed to provide high bit data rates to the mobile user. The growing demand of data services in fixed networks is also expected for mobile users. Todays most popular data service is the Internet access. Thus, the mobile Internet access will play a key role to ensure success of UMTS introduction. A typical data service application of the Internet is the *World Wide Web* (WWW) browsing which is running on a *Transmission Control Protocol/Internet Protocol* (TCP/IP) suite. TCP should ensure a reliable end-to-end communication in systems with limited quality of service guarantee. In contrast, *UMTS Terrestrial Radio Access* (UTRA) will grant traffic contracts. In this paper the performance of a WWW browsing session over the UMTS radio interface is examined including TCP/IP and the detailed radio interface protocol stack. A UMTS Radio Interface Simulator (URIS) is presented and simulations have been performed to evaluate quality of service parameters like delay and packet throughput.**

***Index Terms*— UMTS, UTRA, Radio Interface Protocols, WWW Traffic Model, Quality of Service, TCP/IP, UMTS Radio Interface Simulator**

I. Introduction

The main advantages of *Universal Mobile Telecommunications System* (UMTS) are the expected transmission of high bit data rate packet services and the guaranteed quality of service. During the specification of UMTS radio interface protocols significant effort has been pursued to realize a reliable and efficient communication. The nature of the radio link with its higher error rates and the characteristics of packet data services are a challenging task for protocol design. To cope with limited bandwidth, higher delays and error rates, various complex mechanisms of the UMTS protocol stack are used to satisfy mobile users.

This paper demonstrates the performance of UMTS to support mobile Internet access. A typical and demanding Internet application is the *World Wide Web* (WWW) browsing session, since this interactive, asymmetrical and bursty traffic has to be handled efficiently by the radio interface. UMTS provides the *Radio Link Control* (RLC) protocol to support quality of service at the radio interface. The RLC itself realizes a reliable link with *Automatic Repeat Request* (ARQ) mechanisms. Additionally, WWW browsing sessions are supported by *Transmission Control Protocol* (TCP) on an end-to-end error recovery basis. Nevertheless, TCP will not ensure quality of service in terms of packet delay and throughput. This might be applicable for users with access to fixed networks since overprovisioning of bandwidth is feasible in this environment. At the radio interface both, TCP and RLC protocol are using retransmissions to recover lost data packets. This rises the question how both protocols together will meet requirements of the mobile user considering delay and throughput.

The aim of this paper is to present performance results of a WWW browsing session over the UMTS radio interface protocol stack including TCP and Internet Protocol (IP). For evaluation purposes the software based *UMTS Radio Interface Simulator* (URIS) is used. 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. II presents the used simulation methodology and the simulation environment. Sec. III gives a description of the WWW traffic model and the used parameters. Sec. IV and Sec. V illustrate quality of service mechanisms of TCP and RLC protocol. Performance evaluation, simulation scenarios and discussion of simulation results is depicted in Sec. VI and Sec. VII. The paper concludes in Sec. VIII with a summary.

II. Simulation Environment

For this study, the simulation tool 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) [1]. This generic, object oriented library is well suited for telecommunication network simulation purposes and can be used in all event driven, bit accurate simulation environments. The UMTS protocols

TABLE I

MODEL PARAMETERS OF WWW BROWSING SESSION

WWW Parameter	Distribution	Mean	Variance
Session Arrival Rate [h^{-1}]	negative exponential	2.0	—
Pages per Session	geometric	5.0	—
Reading Time between Pages [s]	negative exponential	12.0	—
Objects per Page	geometric	2.5	—
Inter Arrival Time between Objects [s]	negative exponential	0.5	—
Page Request Size [byte]	normal	1136.0	80.0
Object Size [byte]	$\log_2$-Erlang-k	$\log_2 2521 \approx 11.3$	5.4

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* (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.

Usual simulator approaches model protocols and functions on 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.

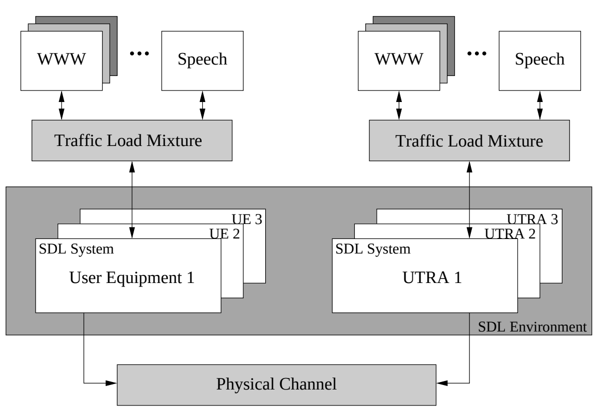


Fig. 1. Structure of the URIS Simulator

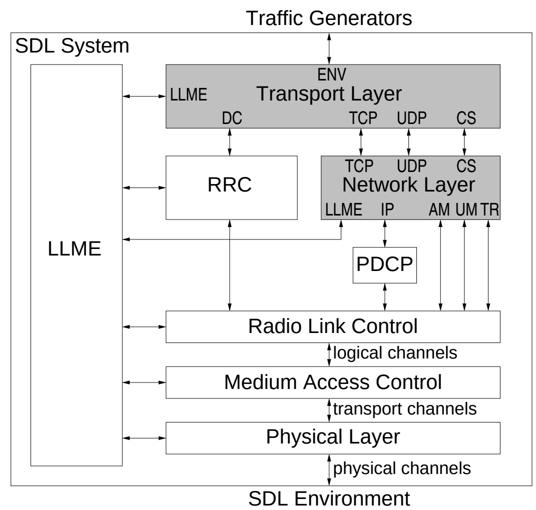


Fig. 2. URIS SDL System Structure

III. WWW TRAFFIC MODEL

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 burstyness 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. I gives an overview of the used WWW traffic model described by parameters and their distributions.

Related documentation can be found in [2–4]. The main part of the later implementation is based on the work of [3, 5].

IV. The TCP/IP Protocol Suite

The modern Internet is mainly based on TCP and IP. In order to provide Internet services to mobile users, 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.

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 [6, 7].

IP is a connectionless 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.

V. 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 *Selective Request ARQ* (SR-ARQ) has been implemented. The SR-ARQ, segmentation/reassembling as well as concatenation are fully implemented as specified in [8].

VI. Simulation Scenarios

The main parameters concerning *Quality of Service* (QoS) which are affected by RLC and TCP/IP, are delay and throughput. Three measurements were made during the simulations:

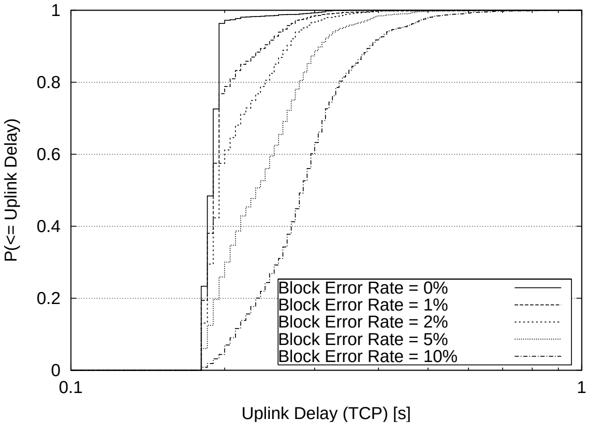
1) Packet Delay: Time from sending a traffic load packet to the protocol stack until correct reception of the packet by the traffic load receiver.
2) Packet Throughput: Size of the traffic load packet divided by the packet delay. The throughput is given in kilobits per second [kbit/s].
3) Retransmissions: Number of retransmissions of RLC PDUs.

Simulations of a WWW browsing session running on TCP/IP in RLC AM were executed for uplink page requests and page downloads on the downlink. The assumption has been made that a *Dedicated Channel* (DCH) of 64 kbit/s is dedicated to the mobile user. This is a comparable channel capacity to a fixed narrowband ISDN access. The corresponding RLC configuration for this channel capacity is taken from [9] and are shown in Tab. II.

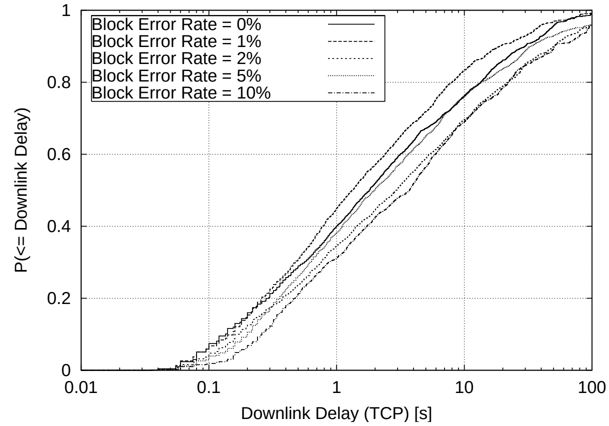
The cumulative distribution function of the measured packet delay and the complementary cumulative distribution function of the measured packet throughput have been calculated. Additionally the number of retransmissions of each RLC packet was counted and the probability function has been calculated. Measurements have been made for increasing transport block error rates from 0% up to 10%. A transport block error is a missing RLC PDU caused by not recoverable errors in the physical layer. It is the residual error after error detection, correction and checking by the physical layer. Tab. II shows the parameters of RLC and TCP/IP used for these simulations.

VII. Simulation Results

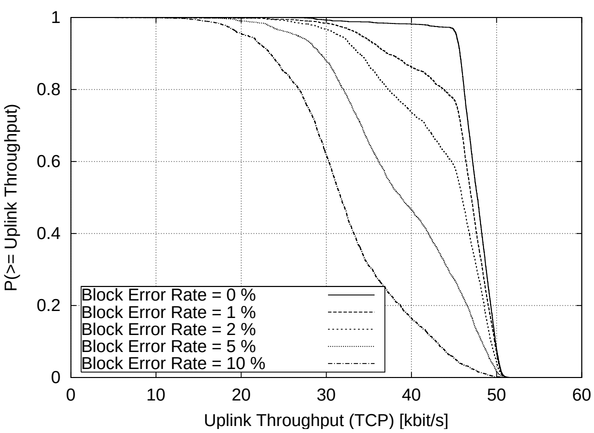
The simulation results are given in Fig. 3. The delay of single traffic load packets in the uplink is shown in Fig. 3(a). The correlation between delay and block error rate follows the expectation: Higher error rates cause higher delays. With a block error rate between 0% and 10%, 90% of all data packets will have a delay of less than 0.4 seconds. The constant delay of 0.18 seconds is caused by the uplink page request size. The mean size of a page request is 1134 bytes. A typical TCP packet carries 512 bytes only. In consequence tree TCP packets are needed for a single page request. Each TCP/IP packet needs a header overhead of 40 bytes. Since a single RLC PDU carries 38 bytes of payload data, 34 RLC PDUs are needed to transfer the page request. For that amount of data nine *Transmission Time Intervals* (TTIs) or 0.18 seconds are needed to transmit a page request in an errorfree scenario. The corresponding throughput values shown in Fig. 3(c) are pointing out that error rates between 0% and 2% will cause a minimum throughput of 35 kbit/s for 90% of all transmitted data. The



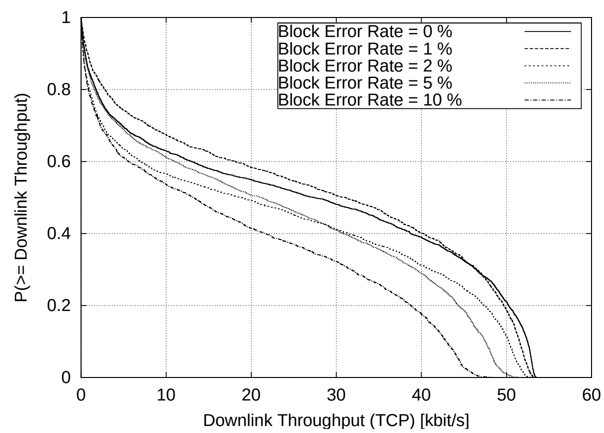
(a) Uplink Traffic Load Packet Delay



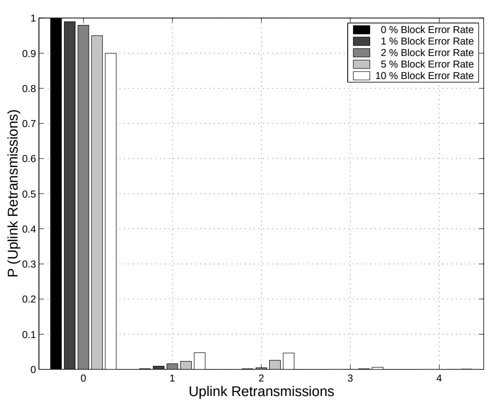
(b) Downlink Traffic Load Packet Delay



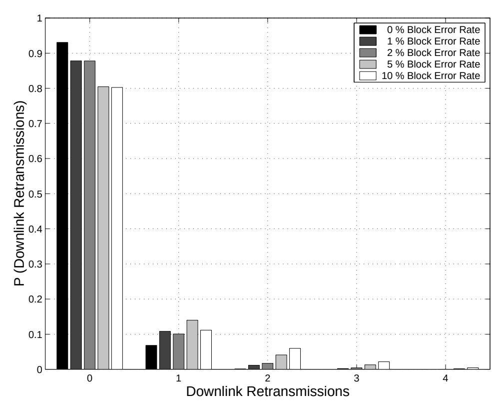
(c) Uplink Traffic Load Packet Throughput



(d) Downlink Traffic Load Packet Throughput



(e) Uplink Number of Traffic Load Packet Retransmissions



(f) Downlink Number of Traffic Load Packet Retransmissions

Fig. 3. Simulation Results for Traffic Load Packet Delay and Throughput and RLC Packet Retransmissions

TABLE II

SIMULATION PARAMETERS

Parameter	Value
TTI Length	0.02 s
PDUs per TTI	4
Bytes per RLC PDU	40
Dedicated Channel Capacity	64 kbit/s
RLC TxWinSize	1024 PDUs
Max. No. of PDU Retransmissions	40
Poll Timer	0.1 s
Poll Prohibit Timer	0.13 s
Status Prohibit Timer	0.1 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%, 2%, 5%, 10%

theoretical maximum throughput rate of 64 kbit/s will not be reached since it is reduced by the TCP/IP/RLC protocol overhead and padding. An upper limit of 51.4 kbit/s is reached, which is 80% of the available channel capacity. Fig. 3(e) shows the probability of retransmissions of RLC PDUs. With increasing error rates even more RLC PDUs must be retransmitted.

Fig. 3(b) and Fig. 3(d) show the results for the downlink. The delays are remarkable higher due to the higher packet size of a WWW object. Single WWW objects of high packet size need up to 100 seconds for transmission. The minimum delay of 0.04 seconds can only be reached by small user data packets. For block error rates of 5% and 10% the corresponding throughput is lower since every page object is more often affected by errors which results in time consuming retransmissions at the RLC layer (see Fig. 3(f)). Even more delay is caused by queue waiting times. While the transmission of the first object of a WWW page is started, an immediately following object has to wait until the previous one has been transmitted. Additional delays are caused by the behavior of TCP, especially due to its polling and flow control mechanisms. The corresponding object throughput reaches a maximum of 53.6 kbit/s which is 84% of the available channel capacity. This value is slightly better than in the uplink due to minor padding necessity for large user packets.

Fig. 3(f) shows the probability of retransmissions. Remarkable are the probabilities for once and twice retransmissions at an error rate of 0%. Retransmissions are not expected, because no PDUs get lost. These retransmitted PDUs are used by the sender for polling purposes.

VIII. CONCLUSIONS

The simulation results show, that surfing in the Internet is possible but the mobile user will suffer from higher waiting times compared to fixed Internet access. Delays are caused by higher block error rates which have to be recovered by time consuming RLC and TCP retransmissions.

The simulation results give the time a user has to wait for a WWW object. Since a mean WWW page consists of 2.5 objects (Table I) the user has to wait around about 4 minutes (or 250 seconds) in a worst case scenario. This means that the user requested a page which contains 2.5 very large objects. Most WWW pages contain a mixture of object sizes. The simulation results show that the user will wait less than 2 seconds for 50% of all object sizes. In consequence a user will face a mean waiting time for a whole WWW page with a mean of 2.5 objects of around about 5 seconds.

Additional effects resulting from congestion, delays and lost packets in the core network are not regarded. For an end-to-end performance simulation the behavior of the fixed network nodes have to be considered. Nevertheless the UMTS radio interface is the bottleneck of the data transmission which will have the most impact concerning the overall contribution in terms of end-to-end transmission delays.

ACKNOWLEDGMENTS

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

[1] Matthias Steppler and Matthias Lott, “SPEET - SDL Performance Evaluation Tool,” *8th SDL Forum ’97*, September 1997.

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

[3] Martin F. Arlitt and Carey L. Williamson, “A Synthetic Workload Model for Internet Mosaic Traffic,” 1995, Proceedings of the 1995 Summer Computer Simulation Conference, Ottawa, Canada, July, pp. 24–26.

[4] Vern Paxson, “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*, March 1994.

[5] Holger Cornelsen, “Entwicklung und Implementierung von Lastgeneratoren für multimediale Sprach-, Audio- und Paketdatenanwendungen,” M.S. thesis, Chair of Communication Networks, RWTH Aachen University of Technology, Nov. 1996.

[6] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “Selective Acknowledgement Options,” RFC, Internet Engeneering Task Force, April 1996.

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

[8] 3GPP TS 25.322, “Radio Link Control (RLC) Protocol Specification,” Technical Specification V4.1.0, Release 4, 3rd Generation Partnerschip Project, Technical Specification Group Radio Access Network, June 2001.

[9] 3GPP TS 34.108, “Common Test Environments for User Equipment (UE) Conformance Testing,” Technical Specification V3.4.0, Release 1999, 3rd Generation Partnerschip Project, June 2001.

[10] Harri Holma and Antti Toskala, Eds., *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, John Wiley & Sons Ltd., Mar. 2001.

[11] B. Walke, *Mobile Radio Networks, 2nd Edition*, John Wiley & Sons Ltd., Chichester, England, 2 edition, 2001.

[12] B. Walke, Mac Peter Althoff, and Peter Seidenberg, *UMTS - Ein Kurs*, J. Schlembach Fachverlag, Weil der Stadt, Germany, 2001.