PERFORMANCE OF INTERNET APPLICATIONS AT THE UMTS RADIO INTERFACE

Silke Heier, Dirk Heinrichs, Andreas Kemper
Communication Networks
Aachen University of Technology
Kopernikusstr. 16
52074 Aachen, Germany

Abstract

Universal Mobile Telecommunications System (UMTS) as a 3rd generation mobile telecommunications system is designed to provide mobile Internet access to the user. In this paper, packet data service applications are examined for performance analysis by simulations. A WWW traffic model is presented as the basis of a WWW load generator. This traffic load generator integrated into an event driven, bit accurate *UMTS Radio Interface Simulator* (URIS) is presented. The simulator is based on a detailed implementation of the standardized UMTS protocol stack extended by the *Transmission Control Protocol* (TCP) and the *Internet Protocol* (IP). Simulation results of quality of service parameters like delay and throughput depict the performance of mobile Internet access over UMTS.

Introduction

In the context of exponential growth of the Internet market, growing demand of data services is also expected for mobile users. Third generation mobile radio networks like *Universal Mobile Telecommunications System* UMTS will provide data services with higher 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. Hence, the mobile Internet access will be one of the key services which should guarantee a successful start of UMTS operation.

This paper will investigate the performance of Internet applications over UMTS from the mobile user point of view. Due to the nature of radio access systems different conditions namely limited bandwidth, higher delays and error rates at the radio interface have to be taken into account compared to fixed network access. Thus, a modeling of the WWW traffic as well as the UMTS radio interface is mandatory for performance evaluation purposes.

In this paper, the UMTS radio interface simulation environment called *UMTS Radio Interface Simulator* (URIS) is presented first. Following, WWW browsing session traffic characteristics are outlined and mapped with identified parameter sets on a WWW traffic model. Simulation results will show which performance in terms of quality of service parameters a mobile user will experience while using Internet applications.

UMTS Radio Interface Model

For this study, the simulation tool URIS was developed. This simulation environment 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) [10]. 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).

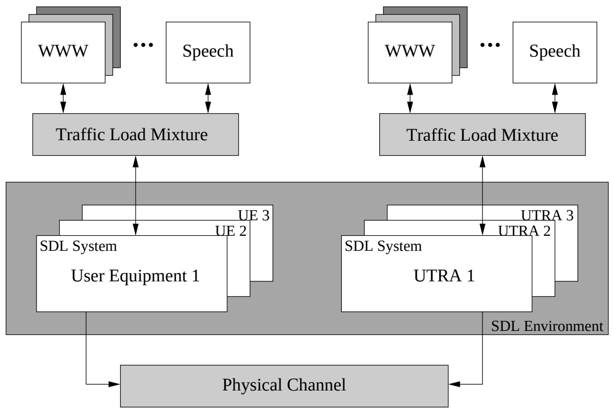


Figure 1: Structure of the URIS Simulator

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 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 *Medium Access Control* (MAC), RLC, *Packet Data Convergence Protocol* (PDCP) and *Radio Resource Control* (RRC) based on UMTS Release 4 and

WWW Parameter	Distribution	Mean	Variance
Session Arrival Rate [h^{-1}]	negative exponential	$2, 20, 40, 60, 80$	—
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	$\log_2 2521 \approx 11.3$	$(\log_2 5)^2 = 5.4$

Table 1: Model Parameters of WWW Browsing Session

TCP, *User Datagram Protocol* (UDP) and IP are specified in SDL. Each layer is represented by an appropriate SDL block which contains the layer specific processes.

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. 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 [8, 4].

The RLC realizes segmentation and retransmission services for both user and control data. The RLC protocol provides three different data transfer services to the adjacent layer: *Transparent data transfer service mode* (TR), *Unacknowledged data transfer service mode* (UM) and *Acknowledged data transfer service 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 [1].

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.

WWW Traffic Model

Surfing the Internet is a WWW browsing session 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 a certain amount of time for studying the information. This time interval is called reading time. Tab. 1 gives an overview of the distributions and their parameters describing the WWW traffic model. The implementation is based on related documentation of [3, 5, 9].

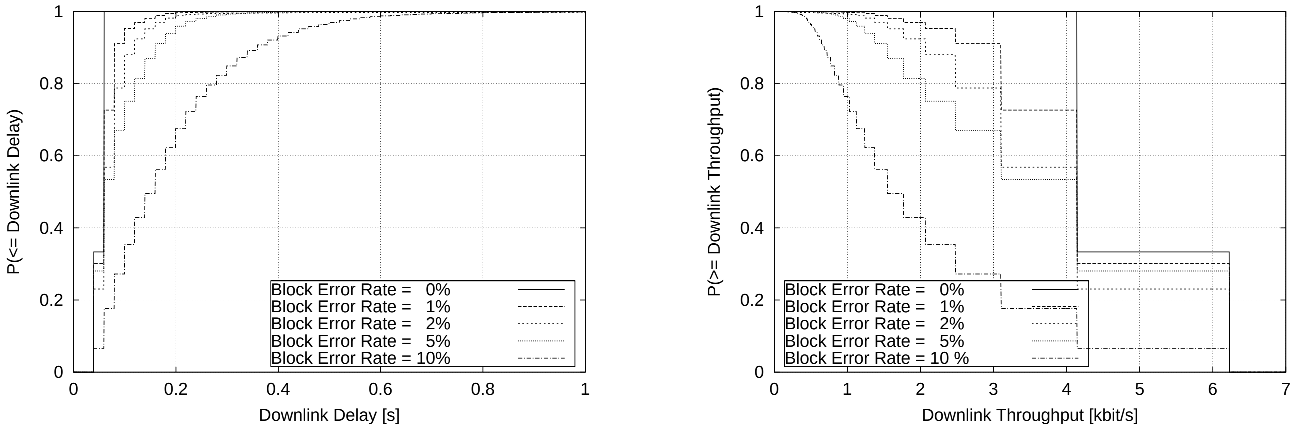
Simulation Scenarios

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

1. Packet Delay: Time from sending a traffic load packet to the 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].

Traffic Generator	CBR	WWW
Session Data Rate [kbit/s]	12.4	Table 1
TTI Length [s]	0.02	0.02
DCH Capacity [kbit/s]	20	64
RLC TxWinSize [PDU]	1024	1024
Status Prohibit Timer [s]	0.1	0.1
Poll Prohibit Timer [s]	0.13	0.13
Poll Timer [s]	0.1	1.0
Max. TCP Segment [byte]	-	512
Max. TCP Window [kbyte]	-	16
Min./Max. TCP RTO [s]	-	3 / 64
Block Error Rate	0, 1, 3, 5, 10%	

Table 2: Simulation Parameters



(a) CBR Packet Delay

(b) CBR Packet Throughput

Figure 2: Simulation Results for Constant Bit Rate

Simulations of *Constant Bit Rate* (CBR) and E-Mail/WWW browsing sessions running on TCP/IP in RLC AM were executed. The assumption has been made that a *Dedicated Channel* (DCH) of 20 kbit/s for CBR and 64 kbit/s for E-Mail/WWW traffic is dedicated to the mobile user. The corresponding RLC configuration for this channel capacity is taken from [2] and is shown together with the TCP parameters in Tab. 2.

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 after error detection, correction and checking by the physical layer.

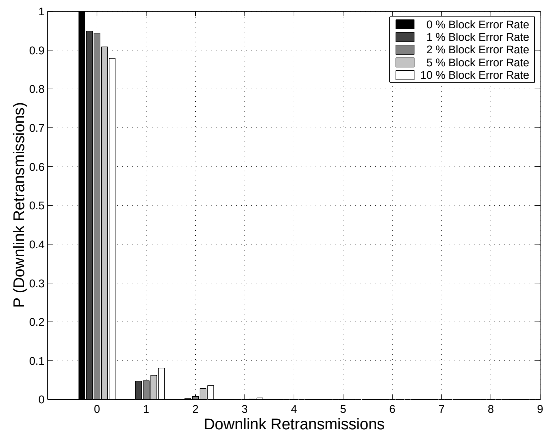


Figure 3: CBR RLC PDU Retransmissions

Constant Bit Rate Simulation Results

The results of the CBR simulations are given in Fig. 2. Fig. 2(a) shows the user data packet delay. As can be seen, the minimum delay is 0.04 s. 0.02 s are needed by the lower layers (MAC and PHY) to transmit the data to the receiver. The remaining 0.02 s are queuing time in the RLC layer.

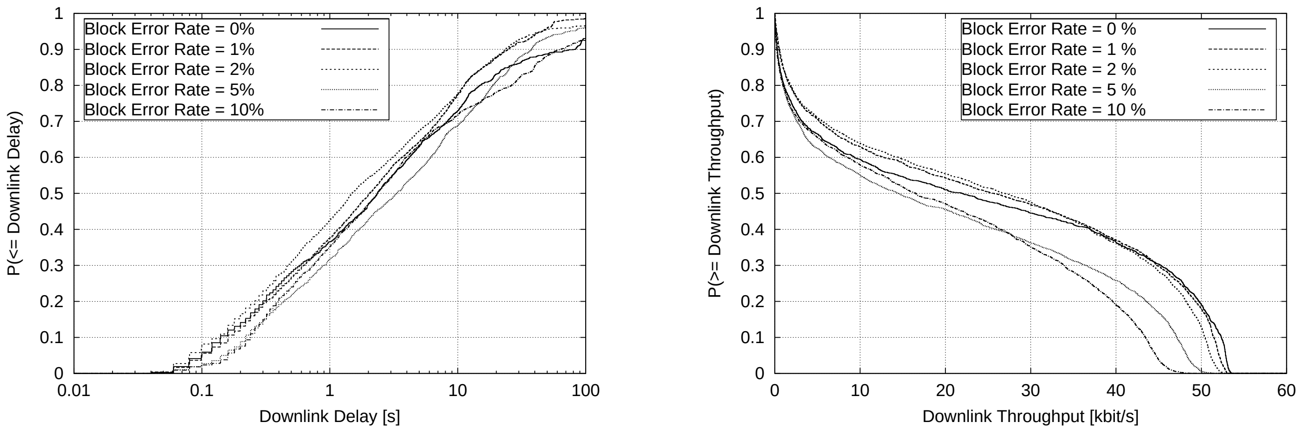
For a block error rate of zero percent, all packets have the same delay of 0.04 s. At a block error rate of 5 %, 90 % of all data packets need less than 0.16 s to be transmitted, at a block error rate of 1 % less than 0.08 s.

Fig. 2(b) shows the user data packet throughput. The maximum throughput of 6.2 kbit/s seems to be bad for a 20 kbit/s channel capacity. But the minimum delay for transmitting a data packet from sender to receiver is 0.04 s and the size of a packet is 248 bits. The maximum throughput is calculated related to one data packet as:

$$\frac{DataPacketSize}{TransferDelay} = \frac{248\,bit}{0.04\,s} = 6.2\,kbit/s$$

In Fig. 3 the efficient usage of the available channel capacity can be seen. With an error rate of 0 % there are no retransmissions, all PDUs are transmitted only once. With an error rate of 10 %, almost 88 % of all PDUs are transmitted only once. 8 % are transmitted twice and 4 % are transmitted three times or more.

This measurements have shown, that the channel capacity of 20 kbit/s is sufficient. Additionally to the source data rate of



(a) WWW Traffic Load Packet Delay

(b) WWW Traffic Load Packet Throughput

Figure 4: Simulation Results for WWW 64 kbit/s

12.4 kbit/s, 1.2 kbit/s is needed for the RLC protocol header, 1.5 kbit/s for 10 % retransmissions of PDUs, 3 kbit/s for acknowledgments and the rest for padding.

E-Mail Simulation Results

The simulations with traffic load packets of fixed size were made to give an idea of the time a user has to wait for downloading an E-Mail or downloading a file per FTP. Packet sizes of 10 kbyte were used to model typical text E-Mails. Packet sizes of 100 kbyte or 1 Mbyte model the download of E-Mails with bigger attachments. A channel capacity of 64 kbit/s was used.

Packet Size $[kbyte]$	BLER $[\%]$	Delay $[s]$	Throughput $[kbit/s]$
10	0	1.70	47.06
10	1	1.70	47.06
10	5	2.20	36.36
10	10	2.14	37.38
100	0	15.20	52.63
100	1	15.26	52.42
100	5	16.14	49.56
100	10	19.16	41.75
1000	0	151.00	52.98
1000	1	151.04	52.97
1000	5	165.42	48.36
1000	10	182.26	43.89

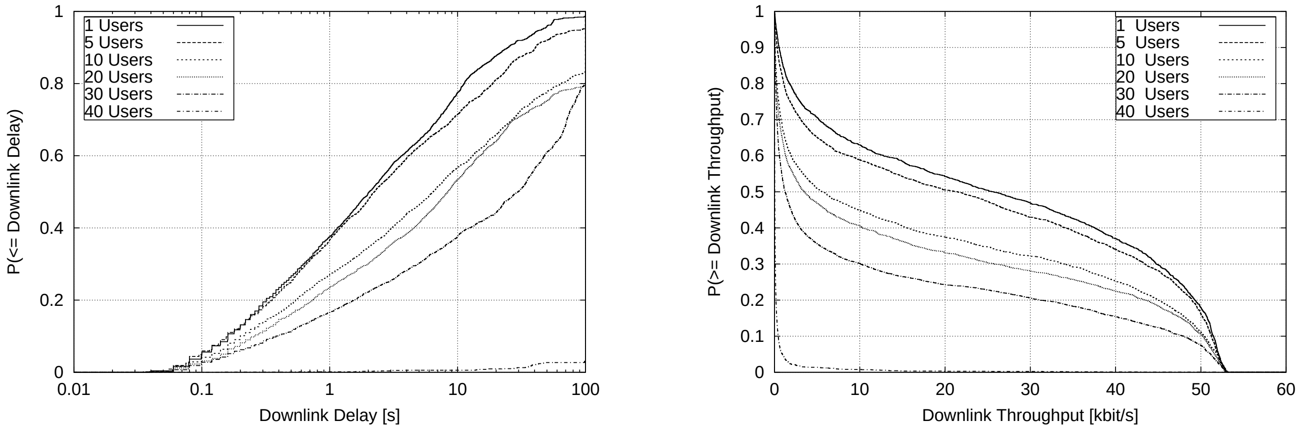
Table 3: E-Mail Simulation Results

The results given in Tab. 3 show the delay and throughput dependent on packet size and block error rate. Included into this measurements is the TCP connection establishment, since it is part of the necessary protocol overhead using these services.

As can be seen the necessary transmission time for small traffic load packets of 10 kbytes is neglectable. Delays of 2 s can be easily accepted. The throughput is relatively low compared to the reached throughput with bigger packets because a fixed time for the TCP connection establishment is necessary, which lowers the effective throughput for the whole transfer. Higher error rates of 10 % result in increasing delays up to doubled values. With bigger load packets the throughput increases continuously because the time for TCP connection establishment is getting less important for the results. With big load packets, higher block error rates have less impact on the results than with small ones, but with a block error rate of 10 % the throughput decreases by approximately 20 %. The maximum throughput reaches 52.98 kbit/s of the 64 kbit/s channel.

WWW Browsing Simulation Results

The results for two WWW browsing sessions per hour on a 64 kbit/s channel can not be interpreted like the CBR scenario. Since the total amount of data of a requested page is characterized by a high variance, this traffic load leads to much higher delays (Fig. 4), which possibly reach a scale of 10 s to 100 s. The minimum delay for user data packets is 0.04 s. This value can only be reached by small packets, which can be transferred within one *Transmission Time Interval* (TTI) and without queuing time. But since most of the user data packets are too big being transferred during one TTI, the probability for such low delays is very small.



(a) Shared Channel WWW Traffic Load Packet Delay

(b) Shared Channel WWW Traffic Load Packet Throughput

Figure 5: Simulation Results for WWW 64 kbit/s

The vast majority of data has a certain queuing time, leading to higher delays and lower throughput. 50 % of all page objects will have a delay of less than 1.5 to 3.5 seconds.

Fig. 4(b) shows that a maximum throughput of 53.6 kbit/s is reached, but only 20 % of all data packets reach a throughput of 40 kbit/s and above. The lowest throughput values of up to 10 kbit/s are caused by very small data packets and by data packets with a long queuing time.

The maximum throughput reaches 84 % of the available channel capacity, the rest is overhead and padding generated by TCP/IP and RLC layer. Related to the maximum reachable performance considering RLC and TCP header, a channel usage of 97.6 % is reached.

Finally it can be estimated how long the user has to wait to retrieve a whole WWW page at a radio interface. The waiting time is dependent on the page content by means of number of page objects and size of the objects. In our traffic model a mean page consists of 2.5 objects. Assuming that 50 % of all objects have a delay of less than 2 seconds, a page retrieval of 2.5 objects needs 5 seconds for that delay probability. A worst case estimation considers the case of a WWW page with several Mbytes content. With a probability of 95 % these big objects will have less than 30 seconds delay. Again, considering a mean page of 2.5 objects, the page delay will rise to 1 minute 15 seconds in a worst case scenario.

Shared Channel WWW Browsing Simulation Results

UMTS provides the opportunity to share a transport channel between a number of users. In the downlink a *Downlink Shared Channel* (DSCH) can be used in *Frequency Division Duplex* (FDD) mode. Since every user needs reading times during a WWW browsing session the question arises how many users might share a channel without seamless service degradation by taking the advantage of statistical multiplexing. With this simulation the possibility and limitations of sharing a channel between a different number of users has been evaluated. To model the traffic generated by several users, the session arrival rate has been increased. For one single user a session arrival rate of two sessions per hour, for ten users 20 sessions per hour etc. has been assumed (Tab. 3). The channel rate was set to 64 kbit/s and the block error rate was set to 1 %. Since all WWW browsing sessions have the same transmission priority, a first-come-first-serve strategy was used to send data packets. This is the common strategy for Internet applications using TCP/IP even in a fixed, wired environment. Simulation results of other scheduling strategies can be found in [6].

The result of the downlink measurements shows major differences between the various traffic load situations. Minimum and maximum delay and throughput do not show a variation, because there are situations where a single user is able to use the whole channel capacity. But it can be seen, that the probabilities of low delay decreases and of higher delays increases. Corresponding results can be seen in the throughput measurements. Five users do not face a major impact on the performance, but with ten to 30 users the performance decreases noticeably. With 40 users the system is overloaded, the channel data rate is not sufficient to transmit the generated data.

Even for this scenario it can be estimated how long a single user has to wait to retrieve a whole WWW page at a radio interface. Assuming up to five users and that 50 % of all objects have a delay of less than 2 seconds, a page retrieval of 2.5 objects needs 5 seconds for that delay probability. In consequence, five users will not note any service degradation by sharing the channel capacity. The result for up to 30 users is quite different. For 10 users 50 % of all objects will have a delay of less than 7 seconds, for 20 users less than 9 seconds and for 30 users less than 30 seconds. Calculated for a whole page of 2.5 objects, the users have to wait for a page retrieval 17.5 seconds, 22.5 seconds or 75 seconds for a shared channel of 10, 20 or 30 mobile users. As a result, the user faces a remarkable service degradation by sharing the channel among 30 users.

Conclusions

This paper presents a simulator which can be used to model the protocol stack and service behavior at the UMTS radio interface in a realistic manner. Due to the detailed and standard related implementation, the simulator can be used to evaluate the performance to be expected by real systems. In a second step the simulator can be used to optimize performance or to develop advanced protocol algorithms.

Running Internet applications over TCP and RLC AM will be an ordinary scenario during the introduction of UMTS. First terminals will rely on standard applications which will be adopted from the fixed world. Hence, 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.

The simulations with multiple users have shown, that it is possible to share a transport channel between up to 30 users. Thereby it has been utilized that never all users are using the channel simultaneously. Nevertheless there are overlapping request and download periods of different users, resulting in higher delays and lower throughput.

It has to be mentioned that effects of the core network are not included in our simulations. The overall performance will get worse since TCP as a end-to-end protocol will experience additional delays and congestion in the core network. The mobile user will face a reduced throughput and longer waiting times for WWW page downloads than in fixed networks.

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] 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.

[2] 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.

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

[5] Victor S. Frost, Benjamin Melamed. *Traffic Modeling For Telecommunications Networks*. IEEE Communications Magazine, March 1994. pp. 70–81.

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

[7] Harri Holma, Antti Toskala, Hrsg. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. John Wiley & Sons Ltd., March 2001.

[8] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow. *Selective Acknowledgement Options*. RFC, Internet Engeneering Task Force, April 1996.

[9] 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.

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

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