Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Marcel Loetscher

Simulative Performance Optimization for TCP over UMTS

Diploma Thesis DA-2003.19
Winter Term 2002/2003

Tutors: Dr. Thomas Ziegler, Dr. Christoph Mecklenbräuker, Dr. Peter Reichl (ftw. Wien), Dipl.Inf.-Ing. ETH Pascal Kurtansky

Supervisor:
Prof. Dr. Burkhard Stiller

7.3.2003

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Simulative Performance Optimization of TCP over UMTS (Marcel Lötscher)

Diploma Thesis DA-2003.19

Winter Term 2002/2003

Tutors: Dr. Thomas Ziegler, Dr. Christoph Mecklenbräuker, Dr. Peter Reichl (ftw. Wien), Dipl.Inf.-Ing. ETH Pascal Kurtansky

Supervisor: Prof. Dr. Burkhard Stiller

7.3.2003

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Table of Contents

List of Figures

List of Tables

Abbreviations

ACK	Acknowledgement
AM	Acknowledged Mode
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
BLER	Block Error Rate
BMC	Broadcast/multicast control protocol
CDMA	Code Division Multiple Access
CN	Core Network
CRC	Cyclic Redundancy Check
CTCH	Common Traffic Channel (logical channel)
CWND	Congestion Window
DCCH	Dedicated Control Channel (logical channel)
DCH	Dedicated Channel (transport channel)
DPCH	Dedicated Physical Control Channel
DPCH	Dedicated Physical Channel
DPDCH	Dedicated Physical Data Channel
DSCH	Downlink Shared Channel (transport channel)
DTCH	Dedicated Traffic Channel (logical channel)
ETSI	European Telecommunications Standards Institute
FACH	Fast Access Channel
FDD	Frequency Division Duplex
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
IP	Internet Protocol
MAC	Medium Access Control
MAC-b	Medium Access Control - broadcast
MAC-c/sh	Medium Access Control - common/shared
MAC-d	Medium Access Control - dedicated
PDSCH	Physical Downlink Shared Channel
PDU	Protocol Data Unit
PHY	Physical (layer)

PU	Payload Unit
QoS	Quality of Service
RACH	Random Access Channel
RLC	Radio Link Control
RNC	Radio Netork Controller
RR	Round Robin
RRC	Radio Resource Control
RTO	Retransmission Timeout
RTT	Round Trip Time
SACK	Selective Acknowledgement
SAP	Service Access Point
SDU	Service Data Unit
SF	Spreading Factor
SIR	Signal-to-Interference Ratio
SRNC	Serving RNC
TB	Transport Block
TCH	Traffic Channel
TCP	Transmission Control Protocol
TD-CDA	Time Divison CDMA
TDD	Time Division Duplex
TF	Transport Format
TFI	Transport Format Indicator
TM	Transparent Mode
TPC	Transmit Power Control
TTI	Transmission Time Interval
UE	User Equipment
UM	Unacknowledged Mode
UMTS	Universal Mobile Telecommunication System
UTRAN	UMTS Terrestrial Radio Access Network
W-CDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
3G	Third Generation
3GPP	Third Generation Partnership Project

1 Abstract

The number of Internet users and the demand for wireless Internet services are increasing rapidly. Most of the future applications that make use of mobile Internet access will depend on the performance of the Transmission Control Protocol (TCP), a reliable transport protocol for the Internet. Unfortunately, TCP was originally not designed for wireless networks. Generally, wireless networks have many different characteristics compared to fixed networks, as the radio signal is affected by fading, shadowing and interference that also change over time. Furthermore, wireless networks have to deal with a much higher bit error rate (BER), with a lower bandwidth available and with the frequent handovers. Under these circumstances TCP suffers from substantial degradation of performance in the form of poor throughput and long delays. There are many strategies proposed by the research community on how to improve the performance of TCP over wireless links like introducing link-layer retransmission, informing the sender of lost packets or new variants of standard TCP.

Mobile network operators have a strong interest in optimizing TCP performance over UMTS. They need this information for their concept development, for their system design and for their network dimensioning to reach a better utilization of their scare radio resources to minimize the number of UMTS radio access points, and to offer better service quality to customers.

Most existing TCP performance studies that are based on network simulations have used Wireless Local Area Network (WLAN) as the studied wireless network technology. In this thesis however, focus is on the optimization of TCP performance in the future mobile communication standard Universal Mobile Telecommunication System (UMTS) using File Transfer Protocol (FTP) data traffic over the Downlink Shared Channel (DSCH) in the frequency division duplex (FDD) operation mode. For this purpose, an existing implementation of a UMTS module, developed at the University of Rome, has been extended for the network simulator NS-2.

The influence of using different TCP variants, different link layer solutions and parameter settings on the performance of TCP over UMTS have been investigated with the help of the extended UMTS simulator module for NS-2.

To compare the different solutions the mean system goodput and system throughput was measured depending on the system load.

The analysis of the simulation results has shown that RLC acknowledged mode generally performs better than RLC transparent mode for FTP traffic. Furthermore, tight ranges for optimal settings of the RLC transmission window size, the RLC transmission buffer size, the maximum number of allowed link-level retransmission and the poll prohibit timer that prevents from excessive polling were derived from simulation result analysis. With those optimal parameters an acceptable system goodput and system throughput can be achieved for all simulated scenarios in the thesis.

2 Vorwort

In den letzten Jahren ist die Anzahl der Internet Benutzer und die Nachfrage nach drahtlosem Internet Zugang ständig gestiegen. Die meisten zukünftigen Anwendungen, die mobilen Internet Zugang benötigen, werden von der Leistung von TCP (Transmission Control Protocol), einem zuverlässigen Transport Protokoll für das Internet, abhängen. TCP wurde jedoch nicht für drahtlose Netzwerke entworfen. Drahtlose Netzwerke weisen verglichen mit verdrahteten Netzwerken stark unterschiedliche Charakteristiken auf wie die Beeinflussung des Signals durch Fading, Shadowing und Interferenzen. Zudem variieren die Signale in drahtlosen Netzwerken auch zeitlich stark. Drahtlose Netzwerke haben weiter eine viel höhere Bit Fehler Rate (BER), verfügen über kleinere Bandbreiten und müssen zudem in Mobilfunksystemen auch häufig Handovers durchführen. Wegen diesen Umständen leidet TCP unter einer starken Verringerung der Übertragungsrate und grossen Verzögerungen. Zahlreiche Vorschläge seitens der Forschungsgemeinschaft wie man die Übertragungsleistung von TCP über drahtlose Netzwerke optimieren kann, wurden in wissenschaftlichen Papers publiziert. Beispiele sind Mechanismen zur erneuten Übertragung von Datenpacketen auf Link Ebene, Informationssysteme, die den Sender über verloren gegangene Packete informiert oder Neuimplementationen vom TCP Standard.

Mobilfunkbetreiber haben ein starkes Interesse an der Optimierung von TCP für UMTS Dienstleistungen. Sie brauchen diese Informationen für das System Design und die Netzwerkdimensionierung, um eine bessere Auslastung ihrer Betriebsmittel zu erzielen, um die Anzahl der benötigten Basisstationen zu verringern und um Ihren Kunden bessere Quality of Service (QoS) Levels anzubieten.

Die meisten bisherigen Studien auf dem Gebiet der TCP Leistungsoptimierung benützten Wireless Local Area Networks (WLAN) als das zu untersuchende drahtlose Netzwerk. In dieser Diplomarbeit wird hingegen die Leistung von TCP in UMTS (Universal Mobile Telecommunication System) mittels FTP Verkehr über den Downlink Shared Channel (DSCH) im FDD Modus untersucht, wobei TCP sowie UMTS Parameter optimiert werden. Zu diesem Zweck wird eine bereits existierende Implementation eines UMTS Moduls, entwickelt an der Universität Rom, für den Netzwerk Simulator NS-2 modifiziert und erweitert.

Der Einfluss von verschiedenen TCP Versionen, von unterschiedlichen Link Ebenen Lösungen und von verschiedenen TCP und UMTS Parametern auf die Übertragungsleistung von TCP wurde mit Hilfe dieses UMTS simulators untersucht.

Um die verschiedenen Lösungen mitteinander vergleichen zu könne wurde unter Anwendung unterschiedlicher Lastsituationen die jeweils erzielte durchschnittliche Systemdurchsatz gemessen.

Die Analyse der Simulationsergebnisse führte zur Erkenntniss, dass die Benutzung des RLC Acknowledged Mode bei FTP Verkehr der Anwendung von RLC Transparent Mode vorzuziehen ist, da mit RLC Acknowledged Mode bei allen Simulationsszenarien stets bessere durchschnittliche Durchsatzwerte erzielt werden konnten.

Im Weiteren wurden optimale Wertbereiche für das RLC Übertragungsfenster, den RLC Übertragungsbuffer, die maximale Anzahl der erlaubten erneuten Übertragungen von Datenpacketen auf Link Ebene and den Wert des Poll Prohibit Timers der übermässige Polling Anfragen verhindert, berechnet. Bei Benützung dieser optimalen Parameter Einstellungen werden in allen in dieser Arbeit verwendeten Simulatonsszenarien gute Resultate hinsichtlich des Datendurchsatzes erzielt.

3 Introduction

According to some estimation over 90% of Internet traffic is currently transported over the Transmission Control Protocol (TCP). The mobile Internet will support existing applications and, consequently, also the TCP protocol. However, the original TCP protocol [28] dates back to 1981 when wireless networks did not have the same position as they have nowadays. As a consequence, TCP contains certain features that are not very suitable considering the special features of wireless networks. Especially the heart of TCP, namely flow control and retransmission mechanisms, may cause problems over wireless interfaces. These problems originate mainly because the basic TCP assumes that all packet losses are due to network congestion, not bit errors. When this assumption is combined with the rough flow control scheme of TCP, the performance of TCP transmissions over wireless networks can be severely degraded.

But before going into details about how to improve TCP performance in UMTS I would first of all like to clarify what the term “performance” in a mobile communication network means. The term performance of a mobile communication network stands for three individual components: coverage, capacity and quality. In UMTS all of these components are somehow related to each other (see section 4.4.1). This is different to other mobile communication standards like GSM, where the capacity does not depend on the level of coverage. Capacity in FDMA networks is generally limited by the number of available frequencies, in TDMA networks by the number of available time slots, but in CDMA networks (like UMTS) by the amount of interference. Furthermore, some of these three performance measures also depend on the kind of service. Speech quality for example depends on block error rate (BLER<1%); data service quality depends on delay and packet loss rate.

In this performance analysis focus is on the capacity component.

The thesis is organized as follows: first the different ways to obtain capacity measures of UMTS to optimize performance are described. After this, the most important related work in the research area of this thesis are outlined. In chapter 4, I then provide some important background information on the UMTS standard needed to develop a UMTS network simulation module. Chapter 5 describes the performance issues related to TCP traffic on wireless links. In chapter 6 details about the UMTS simulator implementation and the simulation models together with the used simulation parameters are outlined. In chapter 7 simulation results were analyzed and discussed before giving an overall summary and conclusion in chapter 8. Chapter 9 finally gives an overview of possible future work. Technical details of the UMTS protocols and information about the network simulator NS-2 used for simulation appear in the Appendix A, B and C.

3.1 Different approaches of doing performance optimization

There are three different ways of evaluating performance: analyzing data from field trials, analytical investigations and analyzing and evaluation of results from computer simulations. Field trials indicate real life performance but are difficult and expensive to conduct, are often infeasible, and it is often impossible to rerun tests many times with different parameter settings. Analytical investigations on the other side are a quick way to get rough performance estimates. They usually give good understanding, but heavy simplifications are necessary and they can be difficult to set up due to the high complexity of the protocols and the multitude of parameter configurations available. Computer simulation that rely on simplified models of the whole system are another approach to assess the problem. Computer simulations can be helpful during the planning process of a new system or even during the operating process as they can be carried out any time and because generally they are less time consuming and are cheaper than field tests. Computer simulation models are usually also easier to develop than analytical models.

Because there are currently few operating UMTS networks, there are few results publicly available from field tests. Computer simulations are therefore the easiest and best way of evaluating TCP performance over a UMTS network.

3.2 Related work

Many performance evaluations of TCP over wireless networks have already been carried out. In chapter 5 the most important of them are outlined and discussed. However, in the specific field of performance evaluation of TCP over UMTS, not many useful results from performance evaluations are available. It is also very hard to compare the results of different performance evaluations as they depend heavily on the used protocols and parameter settings.

One example of an interesting TCP performance evaluation for UMTS is the network simulations carried out at the Technical University of Crete [42]. Their simulation module is based on statistical values derived from link-layer simulations (see section 6.2.2) performed at Ericsson [45]. In a diploma thesis several TCP variants were evaluated using the network simulator NS-2 (see Appendix C), simulating different traffic scenarios with FTP and Telnet. One of the major conclusions of the thesis was that TCP performance heavily depends on the traffic that is simulated and no TCP variant is an outperformer in all simulation scenarios.

Another interesting performance evaluation of TCP over UMTS was performed at Telia research [50]. A simple model of the UMTS Radio Link Control (RLC) protocol (see section 4.8) was used and packets were randomly marked as erroneous and therefore requiring retransmission. The model was a part of the masters thesis work and is simplistic. It doesn't contain anything more than segmentation of RLC data packets and settings for the Transmission Time Interval (TTI). The UMTS model is implemented as a modification to a normal link. It replaces the internal functionality of the queue in a link to mimic the behavior of the RLC layer.

Furthermore a research group at the Institut National des Telecommunications France had investigated throughput in UMTS using the UMTS module of the OPNET network simulator [48]. They propose a resource scheduling scheme to improve throughput and fairness for non-real-time packet data traffic on the downlink shared channel (DSCH) in UMTS.

Another research group at the Motorola Laboratory in Paris, published a paper about optimizing UMTS link layer parameters using TCP as the transport protocol [63]. Their results provide suggested optimal values for the maximum number of allowed link-level retransmission and the RLC window size of the receivers.

A research group at the University of Texas evaluated TCP performance over UMTS networks by carrying out simulations with different retransmission settings at the Medium Access Control (MAC) layer [47]. This approach is targeted for very delay sensitive applications like real-time applications. To avoid the delay associated with retransmissions at the RLC, a lower layer fast MAC retransmission is introduced in the paper. The maximum number of retransmissions of a certain packet is limited and if this number is reached, the handling of retransmission is handed over to the RLC layer. Thus, retransmission is done on at least two layers (MAC, RLC and even TCP if used as the transport layer protocol).

Another research group from the Northwestern University Evanston, USA, have published together with a group of Motorola a paper about performance analysis on the RLC protocol of UMTS systems [62]. In this paper they explored the impact of different error rates, RLC round trip times, polling mechanisms and RLC transmission window size on the system throughput and goodput. As the result of their investigations they suggest certain optimal values for the transmission window size depending on the error rate of the radio channel. They used a statistical air interface model and performed their simulation with the network simulator OPNET [71].

Furthermore Werner Perndl developed a UMTS module for OMNET at the Technical University of Vienna in his masters thesis to analyze TCP performance on the Downlink Shared Channel (DSCH) in UMTS using different scheduling algorithms [17]. See also section 3.3.1 for more details.

Georg Loeffelmann, another student at the Technical University of Vienna, has also carried out some basic performance evaluation of TCP over UMTS [57]. For his simulations he used the simulation tool Ptolemy Classic that is a set of programs developed at the University of Berkeley. In

the thesis, TCP throughput is evaluated for varied block error rate (BLER) on the transport channel for transport services with data rates from 12.2 kbps up to 2048 kbps.

In addition to these evaluations on high protocol layers there has been also several evaluation conducted on lower layers. A research group at Aalborg University in Denmark carried out some low layer simulations together with a research group at University of Rome in Italy [46]. The evaluation was targeted to real-time applications and tried to improve the performance without increasing the transmission delay. It was shown that the proposed strategy can effectively exploited to remove the local retransmissions at link layer.

Different to all these approaches is the solution proposed in [29]. This paper proposes a fairly complex, but easy to use analytical model, which allows performance investigations of TCP file transfer over a wireless link including link layer retransmissions, in particular for UMTS.

Beside papers and theses, there are many other resources available that show results from performance investigations in UMTS, like the Internet-Draft of the Network Working Group of the Internet Engineering Task Force (IETF). This Internet-Draft is available on the Internet and provides a set of recommendations for configuring parameters for protocols used to support TCP connections over 2.5G and 3G wireless networks like UMTS [41]. Using an appropriate large Window Size (Sender & Receiver), an increased Initial Window (Sender) compared to settings in wired networks, the use of Selective Acknowledgments (Sender & Receiver), the use of Explicit Congestion Notification (Sender, Receiver & Intermediate Routers) and not to use IP Header Compression, is proposed.

3.3 Implementation basis for the UMTS module

In this section the two UMTS simulator implementations that were used as basis implementations for the final UMTS simulator implementation are briefly sketched.

3.3.1 UMTS module for OMNET developed at Technical University of Vienna

As already mentioned in section 3.2, Werner Perndl has developed a simulator written in C++ for OMNET [17]. The original plan was to adapt this software to NS-2. Therefore a fundamental design draft for an implementation of a UMTS simulator module for NS-2 has been developed (see Appendix E) including long lists of pseudo codes that describe the behavior of the UMTS protocols. However, an advanced external implementation of such a UMTS simulator module (see next section) was provided at a later stage of my thesis work, I have then only partly continued adapting Werner Perndl's code for my implementation. The scheduling algorithms developed for simulations in OMNET would be interesting for future work (see also section 9.3).

3.3.2 UMTS module for NS-2 developed at University of Rome

Afredo Todini and Francesco Vacirca, two Ph. D. students of the Department of Telecommunications Engineering at the University of Rome, Italy, have developed a series of modules for the simulation of the UMTS radio interface (both TDD and FDD) for their masters theses. They made modifications to the existing wireless model in NS-2 and added new modules like the NOAH routing algorithm (see Appendix 6.4.1), a new RLC layer (see section 6.4.3) and a new MAC layer (see section 6.4.4). More details about the implementation appear in sections 6.4 and 6.5 and in the references [67] and [68].

3.4 Summary and outlook

In this introduction chapter I have defined the expression "performance optimization", then presented some important related research, and sketched the basis implementation of the final UMTS simulator. The next chapter gives some background information about UMTS relevant to build the UMTS simulator.

4 Universal Mobile Telecommunication System (UMTS)

With the rise of the information society, users of data and telecommunication services nowadays expect that the same services are available to them wherever they are, be it at the office, at home or on holidays abroad. Present wireless or mobile systems are constrained in terms of transmission rates and their ability to offer real multimedia services.

UMTS is about to change this by introducing a large number of promising new services like location-based services and multimedia services. UMTS, also referred to as wideband code division multiple access (W-CDMA), is one of the major 3G mobile communications technologies that are being standardized within the framework of International Telecommunication Union (ITU). The standard is being developed by the Third Generation Partnership Project (3GPP), a joint venture of several organizations including the European Telecommunications Standards Institute (ETSI) and the Association of Radio Industry Business (ARIB) of Japan. UMTS will offer global radio coverage and world-wide roaming. For that purpose the UMTS Radio Access Network (URAN) will be built in a hierarchical way in layers of varying coverage, called hierarchical cell structure (HCS) and depicted in Figure 4-1. A higher layer will cover a larger geographical area than a lower layer. In the top layer there will be satellites covering the whole planet; the lower layers form the UMTS terrestrial radio access network UTRAN. They are divided into macro-, micro- and picolayer. Each layer is divided into cells. The lower the hierarchical level, the smaller the cells. Smaller cells allow for a higher user-density. Macro cells guarantee a continuous coverage for moving mobiles, while pico cells are necessary to achieve good spectrum efficiency and high capacity of hot spot areas (e.g. airports, railway stations). In every layer the achievable data rates are significantly different. Macro cells will achieve a minimum data rate of 144 kbps at a maximum mobile speed of 500 km/h, micro cells a data rate of around 384 kbps at a maximum mobile speed of 120 km/h and pico cells finally a data rate of 2 Mbit/s at a maximum mobile speed of 10 km/h. For the simulation I use a single macro cell scenario with slow moving mobile handsets (speed around 5 km/h).

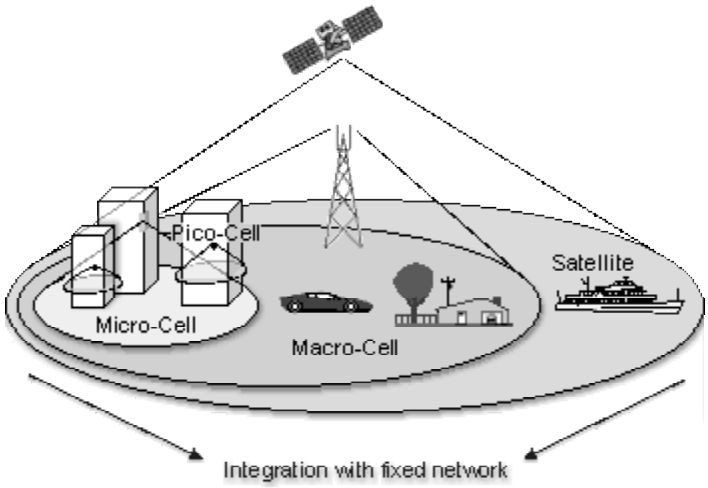

Figure 4-1: The Hierarchical cell structure of UMTS

3GPP proposes two operation modes: The Frequency Division Duplex (FDD) and the Time Division Duplex (TDD) mode. The FDD mode uses a different multiple access method (W-CDMA) than the TDD mode (TD-CDMA Time Division CDMA). This decision was not only for technological reasons but it is a political compromise between different groups in ETSI (European Telecommunication Standards Institute) [64].

In this thesis focus is on the FDD mode, because standardization for this mode is more advanced and because I am mainly interested in simulating macro cell scenarios.

4.1 UMTS services

The mobile communication industry is currently going through a period of upheaval, moving away from offering pure voice and text communication services towards a much richer form of multimedia mobile communication services. The driving force behind UMTS will be the availability of such attractive applications and services that persuade customers to spend money on them. Technology alone will not generate revenue, applications must translate technology into a useful service that subscribers are willing to pay for. Some of the key functions of these applications will be localization (e.g. localization of the own position, searching for other people and maps showing how to get to a certain place), monitoring (e.g. monitoring stock quotes), communication (e.g. e-mail), personalization and connectivity. These applications are designed for the user to “save time” by enhancing job efficiency, to “kill time” by entertaining, to “inform better” or to provide more security by alarm notification and emergency location. Countless applications and services suggestions exist because of the enormous number of contexts in which people work at live. A detailed list of examples of these applications is shown in [56].

Services in UMTS can be roughly categorized into four different traffic classes. Of those four classes two are real-time traffic classes (conversational and streaming) and two are non real-time classes (interactive and background). In my simulations I focus on non real-time classes.

4.2 UMTS network architecture

Figure 4-2 illustrates the UMTS network architecture. It is composed of the Core Network (CN) and the UMTS Terrestrial Radio Access Network (UTRAN). The Core Network is responsible for call routing and data connections to external networks. The UTRAN, which performs the terrestrial mobile radio access to User Equipments (UE) and that hides all mobile access activities from the CN, is composed of several Radio Network Subsystems (RNS), each one including a Radio Network Controller (RNC) and several Node Bs (the 3G term for Base Station). The RNC plays a very important role in the radio network and autonomously handles all functions of the Radio Resource Management (RRM). RNC interfaces the core network via the Iu interface and uses Iub to control one Node B. The Iur interface between RNCs allows soft handover between RNCs. Serving control functions such as admission, RRC connection to the User Equipment (UE), congestion and handover/macro diversity are managed entirely by a single serving RNC (SRNC). The term controlling RNC (CRNC) is used to define the RNC that controls the logical resources of its UTRAN access points. The main task of Node B is the conversion of data to and from the radio interface, called Uu interface, including forward error correction (FEC), rate adaptation, W-CDMA spreading/ despreading, and quadrature phase shift keying (QPSK) modulation on the air interface. It measures the quality and strength of the connection and determines the frame error rate (FER), transmitting these data to the RNC as a measurement report for handover and macro diversity. The Node B is equivalent to the GSM base station and is the Asynchronous Transfer Mode (ATM) termination point that is connected to the RNC via the Iub-interface (see Appendix B.1). Node B performs the air interface processing, which includes channel coding, interleaving, rate adaption and spreading. The Node B is also responsible for softer handovers and inner closed-loop power control. A single Node B can support both FDD and TDD modes. Finally, the User Equipment (UE), based on the same principles as the GSM mobile station, can connect via the radio interface Uu to a Node B. It consists of the Mobile Termination (MT), which performs the radio transmission, the Terminal Equipment (TE) and the UMTS Subscriber Identity Module (USIM). The TE enables end-to-end applications and the USIM provides subscriber identity, authentication and encryption keys [64].

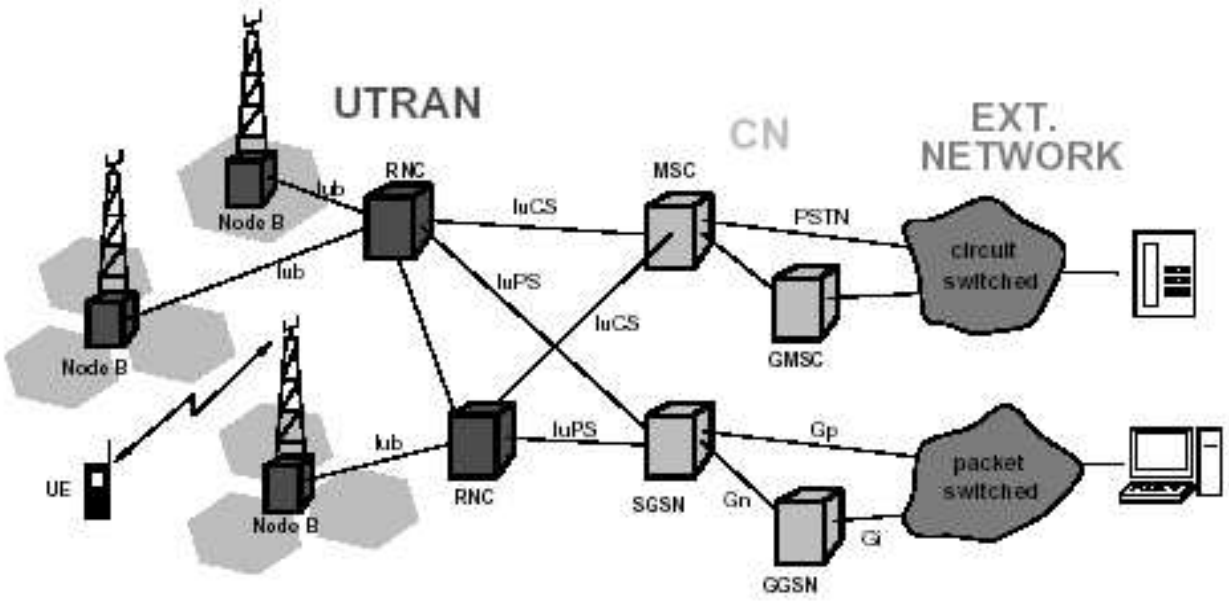

Figure 4-2: UMTS architecture [64]

4.3 UMTS multiple radio access

The basis for any mobile system is its air interface design, and particularly the way the common transmission medium is shared among the users. A multiple access scheme defines how the radio spectrum is divided into channels, and how the channels separate the different users of the system. W-CDMA is the multiple access method selected by ETSI as the basis for UMTS air interface technology. In general, there are three resources for radio systems, frequency, time and power. Division by frequency, so that each pair of communicators is allocated part of the spectrum for all of the time, results in Frequency Division Multiple Access (FDMA). Division by time, so that each pair of communicators is allocated all (or at least a large part) of the spectrum for part of the time results in Time Division Multiple Access (TDMA). In Code Division Multiple Access (CDMA), every communicator will be allocated the entire spectrum all of the time. CDMA uses codes to identify connections. All users interfere with each other and the shared resources is power in this case. A simple receiver then would use a correlator to de-spread the wanted signal, which is passed through a narrow bandpass filter. Unwanted signals will not be de-spread and will not pass through the filter. Figure 4-3 shows an overview of the three different multiple access schemes used in mobile communication networks [64].

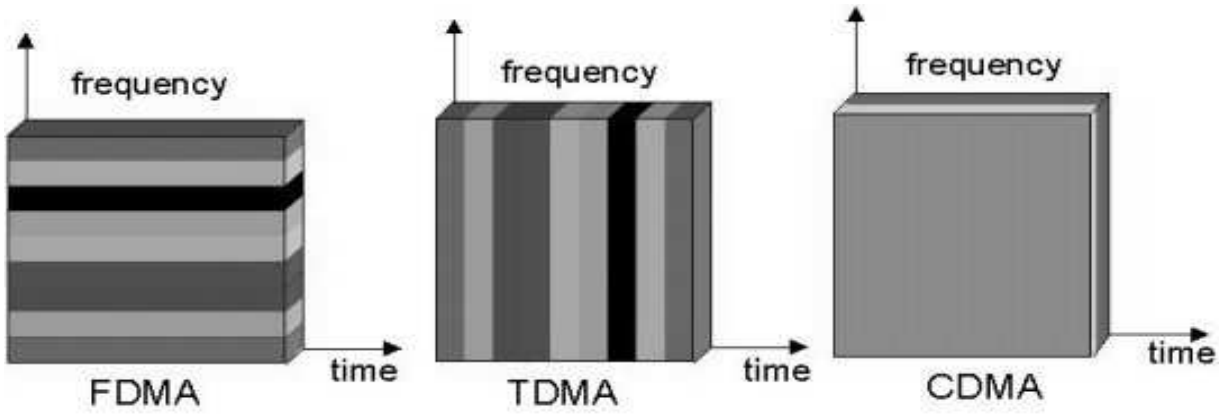

Figure 4-3: Multiple access schemes: FDMA, TDMA and CDMA

4.4 Radio Resource Management (RRM)

Radio Resource Management (RRM) is a set of algorithms that control the usage of W-CDMA resources. The network operator has to define the goal functions of the RRM algorithm like expected throughput maximization, maximization of the expected revenues or minimizing the packet delays. RRM functions can be implemented in many different ways, having an impact on the overall system efficiency and on the operator infrastructure cost, so that RRM strategies will play an important role in the UMTS scenario. RRM is implemented in the User Equipment, the Node B and the RNC. In second-generation networks, RRM is based on hard limits, which means that a fixed number of channels are allocated to the users. Soft capacity, flexible services and bursty traffic make the RRM of third generation systems more complex than in second-generation systems. Some examples of RRM algorithms are: Handover Control, Admission Control, Code Management, Load Control and Power Control, which I outline in the following subsection. RRM functionality is aimed to guarantee QoS, to offer high capacity and to maintain the planned coverage area. Thus, RRM optimization is an important part of UMTS when trying to achieve efficient performance of the radio access network. The UTRAN Service controlled by the Radio Resource Control (RRC, see section 4.10) protocol is used by the RRM algorithms to deliver information over the radio channel.

4.4.1 Power Control

UMTS uses W-CDMA, which is an interference limited multiple access system when Rake receivers are used for detection. Because all users share the same frequency band, internal interference generated by the system is the most significant factor in determining system capacity and call quality. The transmit power for each user must be reduced to limit interference. However, the power should be enough to maintain the required SIR (Signal-to-Interference Ratio) for a satisfactory call quality. Maximum capacity is achieved when SIR of every user is at the minimum level needed for an acceptable channel performance. The transmission power is a controllable radio resource highly related to the network capacity. For example, delay sensitive services with stringent bit error rate (BER) requirements can be accommodated by increasing their transmit powers so as to increase their signal-to-interface ratio (SIR, see section 4.14) resulting in a lower bit error rate.

Power control, which is employed on a connection basis, provides protection against shadowing that causes variation in the received signal strength. The protection is given by controlling the power of the users to be the minimum required to maintain a given Signal-to-Interference Ratio (SIR) for the required level of performance. In this way, each user contributes to the interference of the least extent possible. An important fact is that the downlink output power at the Node B is a common resource, shared by all served users. Therefore, the more power is needed for one user, the less power is available to serve the others. Due to interference, building penetration, obstacles like hills and similar things, different users will require different output power levels. Unfortunately, improving the quality of the received signal by adjusting the transmitter power is not a trivial problem. Assuming that a user u1 with a low signal-to-interface ratio increases its transmitter power. By doing this, the SIR of u1 is momentarily increased. However, the increased transmitting power level of u1 will also increase the interference in the other links in the system. Other users (u2, u3, u4...) will suffer from increased interference and become unsatisfied with the quality of the received signal. This causes the User Equipment to increase their own transmission power levels and this in turns degrades again the SIR of user u1 and the whole process can start again from the beginning.

Every user would ideally transmit and receive at the same power level, but because of the different location of each user and because the received power of one user reduces as the user population in the cell increases, this is not practical.

Power control can be divided into open loop power control (OLPC) and closed loop power control (CLPC).

Open Loop Power Control (OLPC)

The RRM mechanism uses OLPC for setting the initial power level of the User Equipment before any radio connections have been established and the CLPC can be used. The setting of the power levels are based on estimates of the propagation loss, which RRM obtains by measuring the received signal strength of a channel without power control at the receiver.

Closed Loop Power Control (CLPC)

CLPC can be classified into inner loop power control and outer loop power control.

RRM uses the inner loop power control for adjusting the transmitted power of the User Equipments in every time slot. This is done by measuring the SIR that represents the quality of the received signal, compare it with a target SIR, which is set by the Radio Resource Control (RRC) layer, and adjust the power according to it. Such a measure-command-react cycle is executed 1500 times per second and is therefore fast enough to prevent any power imbalance among all the downlink signals received at the User Equipment up to speeds of approximately 70 km/h.

RRM uses the outer loop power control to maintain a certain quality in terms of Block Error Rate (BLER). To achieve a certain Quality of Service (QoS) level of a moving user that experiences channel condition variations the desired SIR target must be adjusted continuously (every few seconds). This is achieved by comparing a measured BLER value (BLER is estimated by performing Cyclic Redundancy Checks and then counting the detected block errors) with a BLER target (depending on the service) and using the difference to regulate the SIR target used by the CLPC. The outer loop power control operates on a frame basis [51].

4.5 Radio Interface Protocol Architecture

The radio interface of UMTS consists of a number of layers, where each layer is responsible for a specific part of the overall radio access network functionality. Layer 1, also known as the physical layer, provides functionality related to physical processing such as channel coding and interleaving, multiplexing, data modulation, spreading to the chip rate and so on. Layer 2 comprises two sub-layers, the Medium Access Control (MAC) and the Radio Link Control (RLC). MAC provides functionality for scheduling and subsequent mapping to transport channels. MAC also provides dynamic selection of the instantaneous transport format for each transport channel. The protocols in Layer 3, also called the network layer, are called Radio Resource Control (RRC) protocol, the Packet Data Convergence Protocol (PDCP) and the Broadcast / Multicast Control (BMC) protocol. The network layer and RLC are divided into control and user planes. Control planes are used to control a link or a connection, user planes are used to transparently transmit user data from the higher layers. PDCP and BMC exist in the user plane only. There is no difference between RLC instances in control and user planes. Figure 4-4 provides an overview of the UMTS radio interface protocols [64].

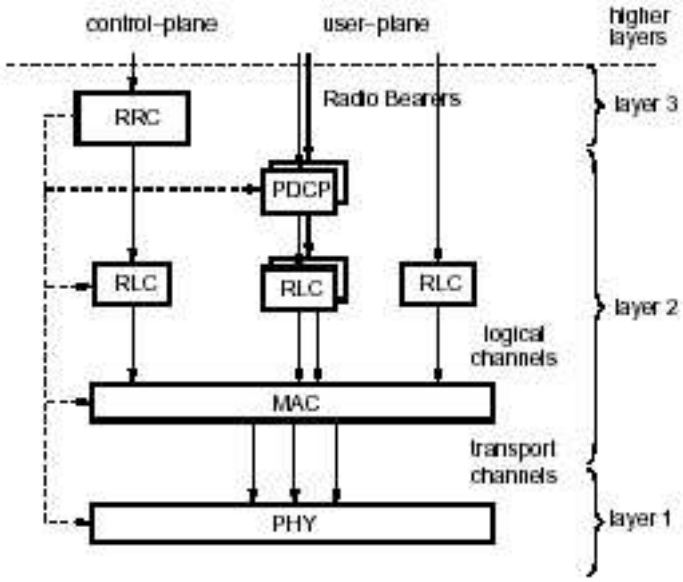

Figure 4-4: Radio interface protocol architecture [1]

Figure 4-5 and Figure 4-6 respectively show the termination points of each protocol for the downlink shared channel DSCH (see section 4.6.2) in control and user planes. Note that there is only the physical layer implemented in the Node B. Communications of higher layers are handled in the RNCs. Each layer provides services at its Service Access Points (SAPs). A service is defined by a set of service primitives (operations) that a layer provides to upper layer(s) (see section 4.12).The protocols that are all assigned to a certain layer are all based on the ISO/OSI reference model.

In the next sections each protocol is presented and its functionality is outlined.

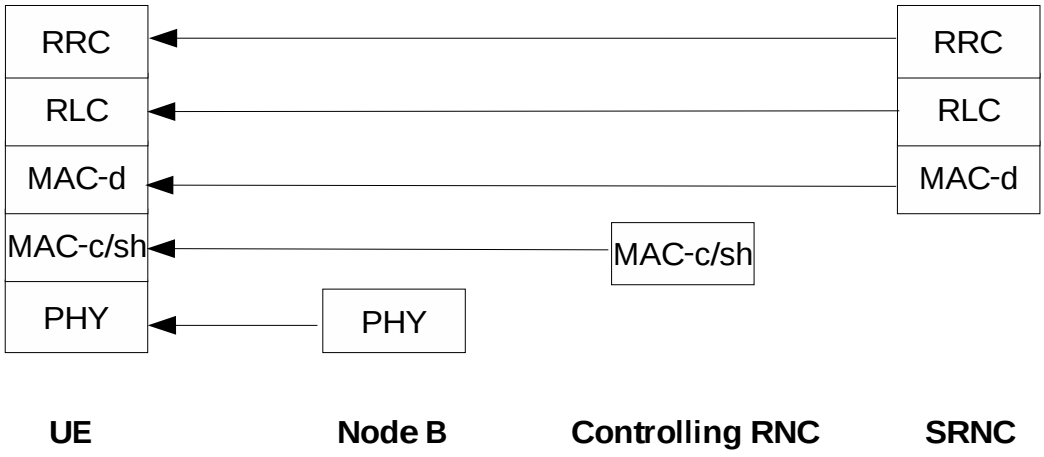

Figure 4-5: Protocol termination points for DSCH, control plane [1]

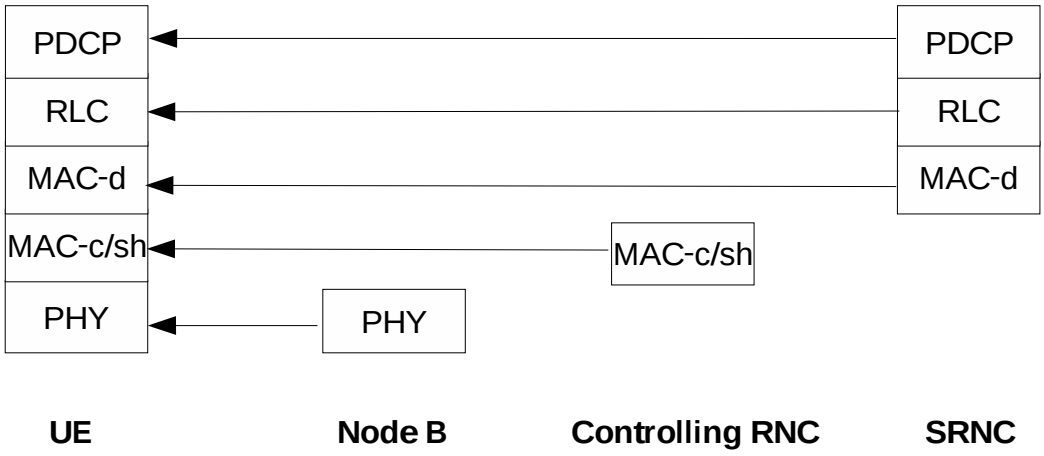

Figure 4-6: Protocol termination points for DSCH, user plane [1]

4.6 Physical (PHY) layer

The physical layer offers information transfer services through the air interface to the MAC layer and higher layers by means of the transport channels (see section 4.6.2). The physical layer is responsible for the Forward Error Coding (FEC), encoding and decoding of transport channels, power control, measurements and indication to upper layers (e.g. BER), error detection on transport channels (CRC), multiplexing and demultiplexing of transport channels, rate matching, modulation/ demodulation, spreading/despreading of physical channels, frequency and time synchronization. In this research I focus on the FDD (Frequency Division Duplex) operation mode for the physical layer. The FDD mode for UTRAN uses W-CDMA with direct sequence spreading. Uplink and downlink use different frequencies. Frequencies between 1920 and 1980 MHz are used for the uplink and between 2110 to 2170 MHz for the downlink [64].

4.6.1 Physical channels

Physical channels are defined by a specific carrier frequency, scrambling code and spreading code. Physical channels typically consists of a three-layer structure of superframes, radio frames (see Figure 4-7 and Figure 4-8), and time slots. Depending on the symbol rate of the physical channel, the configuration of radio frames or time slots varies. In the following subsections I outline the two physical channels, the Downlink Dedicated Physical Channel (DPCH) and the Physical Downlink Shared Channel (PDSCH) that are of importance for the simulation.

Downlink Dedicated Physical Channel (DPCH)

There is only one type of dedicated physical channel in the downlink, called the downlink Dedicated Physical Channel (DPCH).

A downlink dedicated physical channel (DPCH) contains two components: the Dedicated Physical Data Channel (DPDCH) and the Dedicated Physical Control Channel (DPCCH). The DPCH transmits in 10ms radio frames, each of which is subdivided into 15 time slots, for the purposes of closed loop power control. Each time slot time multiplexes the DPDCH and DPCCH as depicted in Figure 4-7. The length of each field in the slot is determined by the slot format.

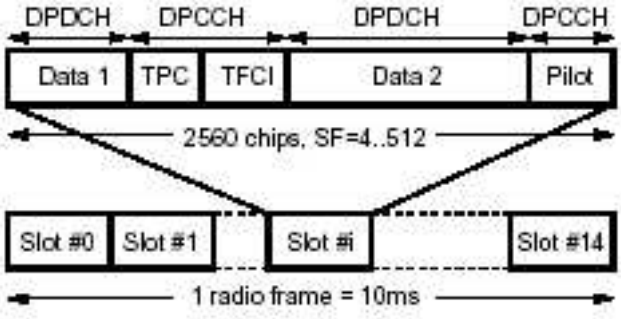

Figure 4-7: DPCH frame structure [6]

Physical Downlink Shared Channel (PDSCH)

Data of the shared downlink are transferred over the physical channel PDSCH (Physical Downlink Shared Channel). Figure 4-8 illustrates the PDSCH frame structure. A PDSCH is always associated with a downlink DPCH. Each user which shares a PDSCH requires therefore an active downlink DPCH.

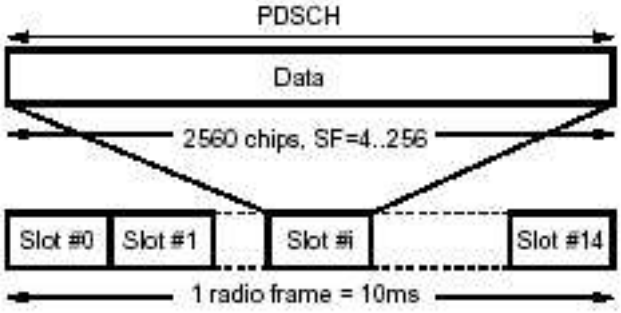

Figure 4-8: PDSCH frame structure [6]

4.6.2 Transport channels

Transport channels constitute the interface by which the MAC communicates with the physical layer. In this subsection I only discuss those transport channels that are of importance for packet switched data traffic in the downlink, as this is the focus of the thesis.

Transport channels can be categorized into two groups, common transport channels and dedicated transport channels.

The difference between them is that dedicated channels, which are identified by code and frequency, can only be used by one user. Common channels on the other hand are shared among several users. To distinguish between the users on the Common channels they are addressed additionally by an inband identification flag.

Transport channels are unidirectional and described between the MAC and the physical layer.

For the downlink direction there is only one dedicated transport channel type, called the Dedicated Channel (DCH). A DCH, which operates bi-directional, is assigned to one UE only. Looking at common transport channels for the downlink data transmission, we find the Forward Access Channel (FACH) and the Downlink Shared Channel (DSCH). The Forward Access Channel (FACH) is used for transmission of relatively small amount of data. The Downlink Shared Channel (DSCH) is similar to the FACH, but it supports the use of fast power control as well as variable bit rate on a

frame-by-frame basis; it may also use a variable spreading factor on a frame-by-frame basis. The DSCH is shared by several UEs carrying dedicated control information and data. The DSCH is a time-shared code resource (DSCH can be allocated with 10ms resolution to the different users, see Figure 4-9), which enables fast allocation, fast transport channel switching and high bitrate transmission. As a pure data channel, it is always associated with a downlink DCH. It is mainly used to transfer bursty high bitrate packet data traffic in the downlink and to save orthogonal downlink codes. A number of UE with relatively low activities can share a single channelization code and gain from statistical multiplexing. The non-bursty active periods of packet transmission on the other side are typically carried on the DCH [66], [40].

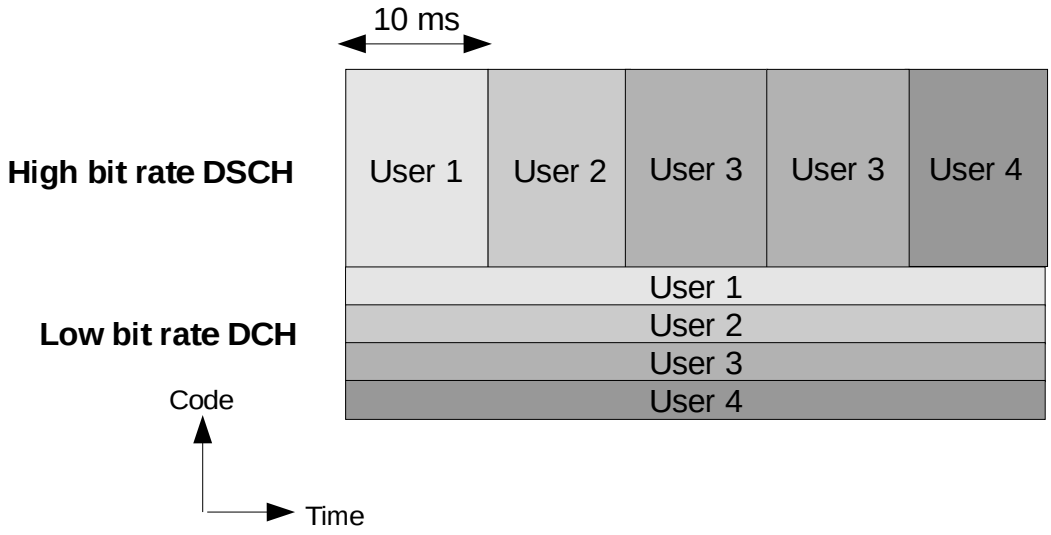

Figure 4-9: Utilization of DSCH and DCH

The switching process from the DCH to the high data rate DSCH is controlled by the packet scheduler and triggered once the RLC buffers get overfilled and queuing takes place due to high rate of packet arrival, large size packets or the increasing number of users. DSCH can therefore not only increase system performance but also decreases blocking. A DSCH can be allocated for one UE at a time for the duration of a scheduling period, following the longest queue first algorithm.

One of the major drawbacks with DSCH is that it is transmitted from only one access point. Due to this, it does not benefit from macro-diversity (refers to the condition that several radio links are active at the same time) as the DCH users do. Macro-diversity is for example needed to perform soft handovers. When using soft handover the radio links are added and removed in a way that the UE always keeps at least one radio link to the UTRAN.

The smallest entity of traffic that can be transmitted through a transport channel is a Transport Block (TB). A Transport Block is the basic data unit exchanged between the Physical layer and the MAC layer. A Transport Block Set is then a collection of transport blocks which are sent over a given transport channel at a given instant. Once in a certain period of time, called Transmission Time Interval (TTI), a given number of TBs will be delivered to the physical layer. The Transmission Time Interval (TTI) is determined by the interleaving scheme in operation on the given transport channel. In my simulations I use a fix value of 10ms for the TTI.

Each transport channel has associated with it a Transport Format Set. This is a set of Transport Formats, where each Transport Format defines the coding, interleaving and mapping onto physical channels.

4.7 Medium Access Control (MAC) layer

The Medium Access Control (MAC) layer is specified in [5]. It is located on top of the physical layer and handles the scheduling of radio bearers with different Quality of Service (QoS= requirements. The main services of the MAC layer are unacknowledged data transfer of MAC Service Data Units (SDUs) between peer MAC entities, reallocation of radio resources and MAC parameters, and reporting of measurements (e.g. traffic volume) to the RRC layer. Important functions to provide these services are mapping of logical channels (describe what is transported) onto appropriate transport channels (describe how data is transported), including dynamic switching between different transport channels, selection of appropriate transport format for each transport channel depending on instantaneous source rate and monitoring the dedicated logical channels to provide traffic volume and transmission quality indication reports to the RRC layer. RRC can then switch the connection of logical channels to different types of transport channels (common or dedicated).

4.7.1 Logical channels

Logical channels are the MAC layer services to the RLC layer. A logical channel is an information stream dedicated to the transfer of a specific type of information over the radio interface. Logical channels that are used to transmit information of the control plane are called control channels. The other channels used to transmit in the user plane are called traffic channels.There are eight different logical channels of which the two most important ones for the simulation are the Dedicated Traffic Channel (DTCH) and the Common Traffic Channel (CTCH). The DTCH is a point-to-point channel dedicated to one UE for transfer of user information. The CTCH is used for broadcasting messages to all or at least one group of UEs.

4.7.2 Channel mapping

After introducing the three different kind of channels (physical, transport and logical channels) this section illustrates how these channels are mapped together. Transport channels are mapped directly to their corresponding physical channels as depicted in Figure 4-10. The Downlink Shared Channel (DSCH) for example is directly mapped onto the physical Downlink Shared Channel (PDSCH). The DSCH in turn was mapped to by the MAC layer from the logical channel DTCH.

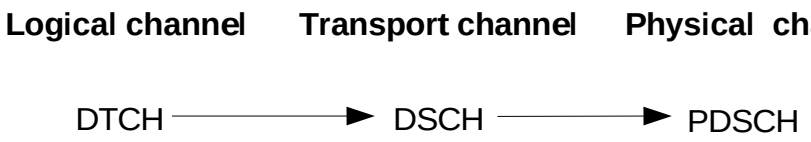

Figure 4-10: Mapping of logical, transport and physical channel in the shared downlink

4.7.3 Medium Access Control (MAC) entities

MAC entities perform several functions on both UE and UTRAN side like mapping between logical and transport channels, selection of transport format for each transport channel depending on instantaneous source rate, priority handling data flows of one UE and between data flows of several users on the DSCH, multiplexing of higher layer data into transport blocks delivered to the physical layer on common transport channels and vice versa, and many more.

The following three different entities are defined for FDD mode:

- **Medium Access Control - broadcast (MAC-b)**
  Identifies the MAC entity that handles the broadcast channel (BCH). On this channel, information about the system is broadcasted into the entire cell. There is one MAC-b entity in each UE and one MAC-b entity in the UTRAN located in the Node B for each cell.
  This entity is not of importance for the simulation and will not be discussed in more details.

- **Medium Access Control - common/shared (MAC-c/sh)**
  Identifies the MAC entity that handles the common and shared transport channels (e.g. FACH, DSCH). There is one such entity in each UE and one for each cell in the UTRAN located in the CRNC (not in the Node B) for each cell.
- **Medium Access Control - dedicated (MAC-d)**
  Is responsible for handling dedicated logical channels and dedicated transport channels allocated to a UE. There is one MAC-d entity in the UE and one MAC-d entity in the UTRAN (SRNC) for each UE. It is also responsible for selecting a Transport Format Indicator (TFI) and Transport Format Combination Indicator (TFCI) to be used in each Transmission Time Interval (TTI) when dynamically sharing resources between bearers supported by a UE.

4.7.4 Data flow through the MAC layer

When a MAC SDU enters the MAC from the logical channel, headers are added according to the protocol and then sent to the transport channel. Depending on multiplexing definitions and mapping to either common or dedicated channels, MAC may add a MAC header (see Appendix A.1) to the RLC Protocol Data Unit (PDU). A complete MAC PDU accords with a Transport Block (TB, see section 4.6.2), which is delivered to the physical layer for transmission.

I try to illustrate the data flow through the MAC layer with an example. When a MAC SDU received from logical channel DTCH/DCCH entering the MAC-d, the first thing is to choose the type of transport channel. In this example, that data will be mapped to FACH. Then, the multiplexing unit will add the C/T to the MAC SDU to indicate the logical channel where the data originates. Then the MAC layer sets the priority of the data. The next step is to pass the data to MAC-c/sh. In MAC-c/sh, the entity adds the UE-id, UE type and target channel type field (TCTF) to the data to form a transport block or MAC PDU. Then the scheduler decides, which PDUs should be send and sends this PDU to the transport channel FACH.

4.7.5 Scheduling in UMTS

Assigning radio resources in a smart way will become one of the key issues in future mobile networks like UMTS. New applications require specific characteristics from the assigned resources to work properly (e.g. real-time services) and systems providing QoS guarantees will occur. Scheduling algorithms try to meet the requirements of the users on one hand, and on the other they try to optimize the utilized resources. Therefore, scheduling policies have a big impact on the performance as for example “fairness” reduces system throughput.

The scheduler categorizes all applications into two groups, the best-effort and the QoS services. Now it is up to the scheduling algorithm how to guarantee the QoS requirements of the prioritized QoS services and to share the remaining resources equally among all remaining best-effort applications the same time. When using code division scheduling on the other hand, the capacity has usually to be shared among a large number of radio bearers, allocating low bit rate simultaneously for each use.

In general, a scheduling algorithm must satisfy three, sometimes contradictory, requirements [8]:

Performance bounds, fairness and protection and ease of implementation. The main objectives of the scheduler is therefore to integrate traffic sources with different transmission rates, priorities, delays and packet loss requirements optimizing the channel utilization.

There are basically two different packet scheduling methods available, the time division scheduling and the code division scheduling. When using the time division scheduling, the capacity is allocated to one or very few radio bearers at a time. The allocated bit rate can be very high and the time needed to transfer the data in the buffer is short. The Downlink Shared Channel in UMTS is an example where time division scheduling is used. In UMTS scheduling of data segments is done in the RLC layer, where the data segments are stored in the transmission buffer. However, the scheduler itself is located in the MAC layer. The scheduling in UMTS is a resource allocation function closely connected to the transport format selection. During communication the MAC

scheduler selects the appropriate transport format within an assigned transport format set for each active transport channel depending on source rate and radio resource limitations. The selection can be done on a 10ms frame basis or slower. Depending on the selected transport format one or more transport blocks can be transmitted.

For the simulations a Round Robin (RR) scheduler has been used (see also section 6.4.4).

4.8 Radio Link Control (RLC) layer

The Radio Link Control (RLC) protocol is specified in [4]. It provides logical link control services over the radio interface like segmentation/reassembly of variable-length upper layer PDUs into/from RLC Payload Units (PU), concatenation and retransmission. It provides packet buffering service to upper layer protocols for incoming and outgoing packets. Information from the CRC error detection at the physical layer can be used for error handling. The functions of RLC cover both the control plane and the user plane of UMTS. Since my focus is on the user bearer traffic performance, the control plane functions, such as ciphering and recovery are not included in the simulation model. RLC protocol entities are located in the UE and the RNC. There may be several simultaneous RLC links per UE. Each link is identified by a radio bearer Id. It is possible to operate RLC in three different data transfer modes, the Transparent Mode (TM), the Acknowledged Mode (AM) and the Unacknowledged Mode (UM). Which mode is used is defined by RRC. In my simulations I concentrate on the Transparent and the Acknowledge Mode.

In the following I outline the different RLC modes standardized in the UMTS specification. More technical details about the RLC layer appear in the Appendix A.

4.8.1 Transparent Mode (TM)

In this mode, upper layer PDUs are transmitted to the RLC layer without adding any protocol information. This means that although information from the error detection at the physical layer can be used, this service does not provide a reliable link. RLC SDUs must have a fixed size that is a multiple of a RLC protocol data units (PDU) size. RLC SDUs can be therefore segmented into RLC PDUs of equal size. If segmentation has been enabled, all the transparent mode data PDUs carrying one RLC SDU are sent in the same transmission time interval (TTI), and no segment from another RLC SDU is sent in this TTI. If segmentation has not been configured by upper layers, then more than one RLC SDU can be sent in one TTI by placing one RLC SDU in one TMP PDU. This means that all transparent mode data PDUs in one TTI must be of equal length. The peer entity that has to reassembly the pieces does not need any information from the header to do so, as the different PDUs were sent in different time slots. With this any segmented PUs can be assigned uniquely to a SDU. In Figure 4-11 a simplified RLC TM entity is shown.

Using the Transparent Mode performs well with delay sensitive services (e.g. speech, audio or video) as those services don’t make use of retransmissions on low level layers. To make sure that no old data unit is delivered, TM provides a dropping mechanism that prevents delivery of already expired PDUs.

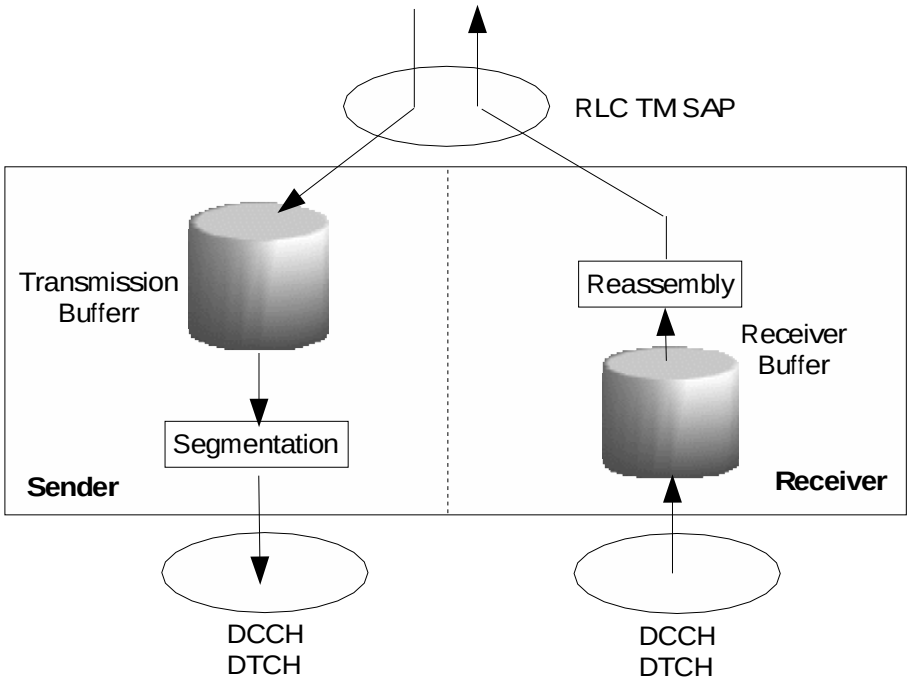

Figure 4-11: RLC TM entity

4.8.2 Unacknowledged Mode (UM)

This service transmits upper layer PDUs without guaranteeing delivery to the RLC peer entity. However, if a loss of PDUs is detected, higher layers are informed by RLC. Duplicated or corrupted PDUs received from the MAC sublayer are discarded and the loss is signalled. Figure 4-12 depicts a simplified RLC UM entity.

The RLC layer adds a sequence number to every RLC PDU to guarantee data integrity of PDUs of upper layers. Only PDUs that were received in the right order are forwarded. The Unacknowledged Mode provides means for segmentation, concatenation and padding (see Figure 4-14).

In my simulation scenarios I do not simulate RLC UM and leave it to future work (see section 9.8).

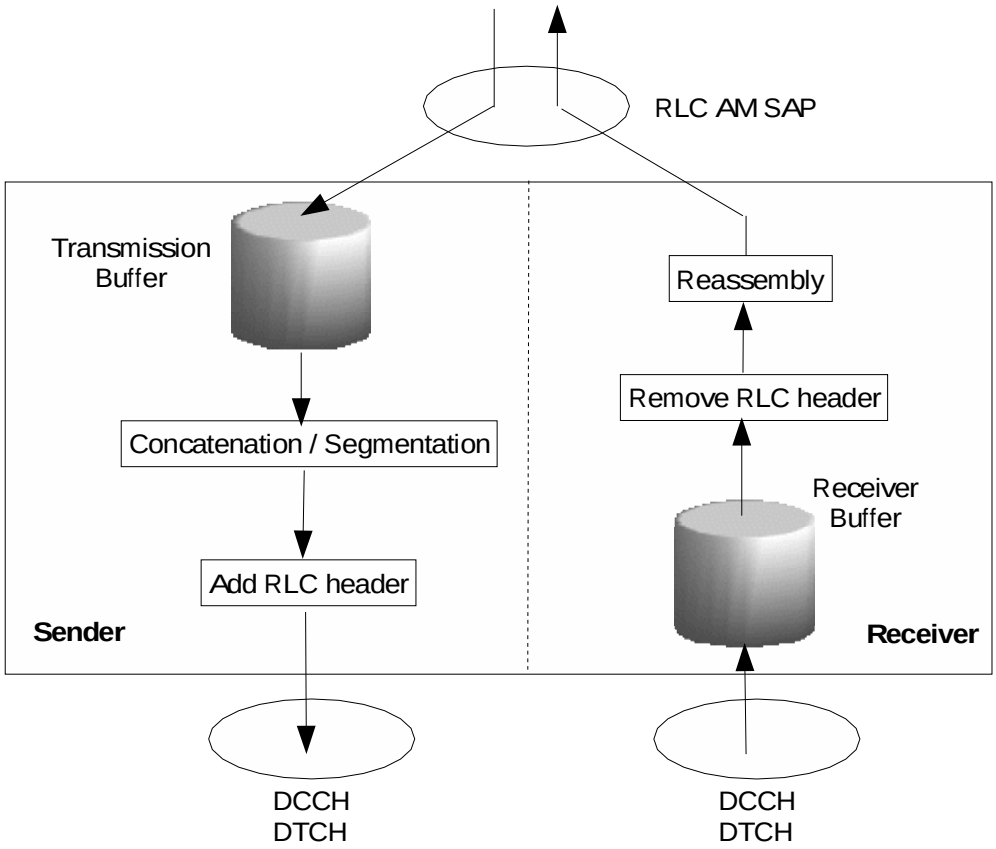

Figure 4-12: RLC UM entity

4.8.3 Acknowledged Mode (AM)

In contrast to the other modes, the Acknowledge Mode provides a quite a reliable link to the RLC peer entity with a negligibly small probability of undetected errors and a low level of loss for the upper layer traffic. However, retransmission of PDUs by automatic repeat request (ARQ) introduces latency and delay jitter to the data flow. This is why TCP sees the underlying network as a network with a relatively large bandwidth delay product (BDP) when using RLC AM. Reliability of data transfer using RLC is achieved by using automatic repeat request (ARQ) where the receiver signalize erroneous transmission to the sender that will retransmit the data. All data being retransmitted are given a higher priority than those who are about to be sent for the first time and are therefore resent immediately after an error indication.

There are three types of ARQ schemes, stop-and-wait (SAW), go-back-N (GBN) and selective repeat (SR). RLC can use any one of the retransmission schemes by configuring the transmission window and acknowledgement format appropriately. ARQs are implemented using status reports. The mechanism for triggering status reports is not clearly defined in the standards and leaves the manufacturers freedom in implementing it. Probably the easiest and best way is to send a status report whenever a received PDU is erroneous and to bring the sender to poll periodically after a certain amount of RLC SDU sent.

Figure 4-13 provides an overview of the RLC AM entity. Notice that this diagram shows only the parts that are relevant for the simulation.

The tradeoff between quality and delay of RLC can be controlled by RRC through setting an appropriate number of allowed retransmissions provided by RLC. Therefore, RLC AM can be described as an ARQ that can be configured for either high persistence or low persistence. The

RLC AM can also enforce in-sequence delivery of upper layer PDUs and is providing flow control by controlling the data rate with which the peer RLC transmitting entity may send data.

AM is mainly used for interactive or background services and is well suited for Internet applications as those services require an error free delivery of the data.

In the following subsections I outline the most important functions of RLC in the sender and the receiver.

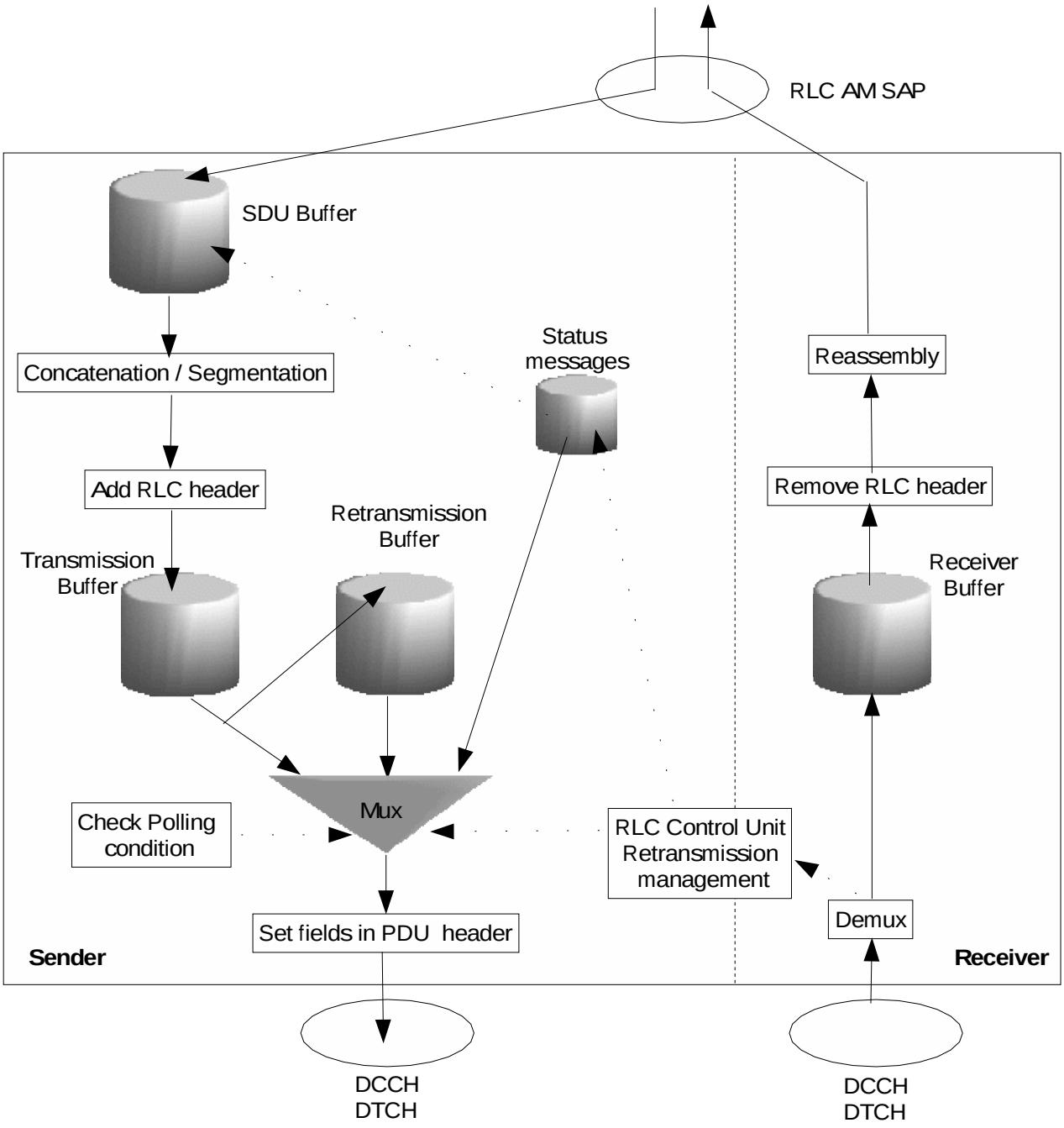

Figure 4-13: RLC AM entity [4]

RLC AM functions of the transmitter in the UTRAN side

After receiving RLC SDUs, those SDUs are segmented and if possible concatenated to earlier PDUs (see Figure 4-14) into RLC payload units (PU) of fixed size. These PUs are then stored in the

transmission buffer and the retransmission buffer. To be able to reassemble the RLC SDUs at the RLC peer entity a header with additional information is added to the PU and padding is performed. Before this RLC PDU is being transmitted to the logical channel, the ciphering process is applied. However, note that I do not simulate ciphering and deciphering of data. When PDUs are requested by the MAC layer for transmission, a multiplexer decides which PDUs have to be delivered to the MAC layer and schedules the time of transmission. One or several PDUs may be transmitted in each Transmission Time Interval (TTI) and MAC decides how many PDUs shall be transmitted in each TTI. This means that the TTI defines how often data should be sent from the RLC layer. The amount of data on the other hand that can be transmitted during one TTI is defined by the bandwidth and the length of the TTI.

The receiving part of the transmitter on the other side will receive regularly acknowledgements from the receiver, indicating already received packets, which can then be deleted from the retransmission buffer.

Another very important function of the transmitter is to poll Status Reports from the receiver. Several polling triggering mechanisms have been defined to do this (see Appendix A.7).

RLC AM functions of the receiver in the UTRAN side

At the receiving side, the PDUs are received through the logical channel DTCH from the MAC layer. Firstly, status PDUs are forwarded to the RLC control unit. The receiver acknowledges successful reception or requests retransmission of the missing AM data PDUs by sending one or more STATUS PDUs to the AM RLC peer entity. When the peer entity receives an acknowledgement, the PDUs will be deleted from the Retransmission queue. When it receives a negative acknowledgement, it will retransmit the PDU.
Secondly, the PDU is stored in the receiver buffer until the complete SDU has been received. When finally all PDUs of one upper layer SDU have been received, headers are removed, the SDU is reassembled and the SDU is delivered to upper layers. If RLC is not able to deliver the data correctly, the upper layer will get a notification.

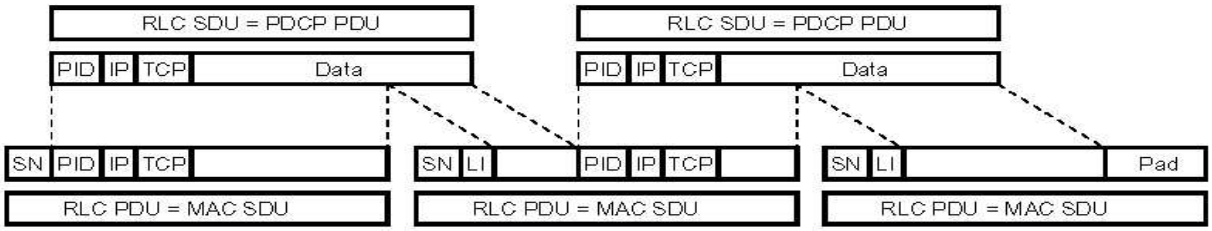

Figure 4-14: RLC segmentation and concatenation

4.9 Packet Data Convergence Protocol (PDCP) layer

PDCP protocol entities are located in the UE and the RNC. This protocol exists only in the user plane and only for services for packet switched services. One main function of the protocol is transmitting and receiving of network layer data units. UMTS supports several network layer protocols like IPv4 and IPv6. Protocol transparency is also provided to hide details of the used protocol from the user.

Another main function of PDCP is to perform IP header compression and decompression. This is mainly because IP was originally designed for fixed networks where the overhead is not a big problem. However, in radio networks like UMTS, radio resources are scarce and IP header compression is crucial to increase channel efficiency, especially if small packets are sent frequently. The method used for the compression (see [35] for details) is specific for the combination of network

and transport protocol (e.g. TCP/IP or UDP/IP). Which algorithm and parameters are used is negotiated by RRC for each PDCP entity during connection establishment.

4.10 Radio Resource Control (RRC) layer

The RRC protocol is responsible for reliable connections between UE and UTRAN by handling the control plane signalling of layer 3. It acts as a management entity and configures and controls the operations of all lower layers. The RRC protocol handles a large number of signaling tasks. One of the most important service offered by the RRC protocol is the assignment, reconfiguration and release of radio resources. Once the radio resource has been allocated, the RRC uses measurements from the lower layers (performed on the UE) to monitor the resources (also control of the requested QoS) and reconfigures them to optimize the system. Another important task that RRC handles is the broadcasting of system information, including both, data originated from the UTRAN and core network data.

A UE can either have no or only one RRC connection. This RRC connection establishment procedure can only be initiated by the UE sending a RRC connection request message to the radio access network. When connected, the UE has been assigned a Radio Network Temporary Identity (RNTI) to be used as its own identity on common transport channels.

4.11 Broadcast / Multicast Control (BMC) layer

This layer handles broadcast and multicast messages. SMS Cell Broadcast Service is one of the major services using this layer. This protocol layer is not important for my simulations and therefore it was not implemented for the simulator.

4.12 Interactions between the different layers

Each layer has two interfaces, the peer-to-peer interfaces that I have discussed in the previous subsections and the service interface. The service interface defines the primitives (operations for logical exchange of information and control between two layers) that a layer provides to the layer above it. The common primitives are indication (Ind), request (Req), response (Resp) and confirmation (Conf).

Figure 4-15 illustrates the primitives between the different protocol layers.

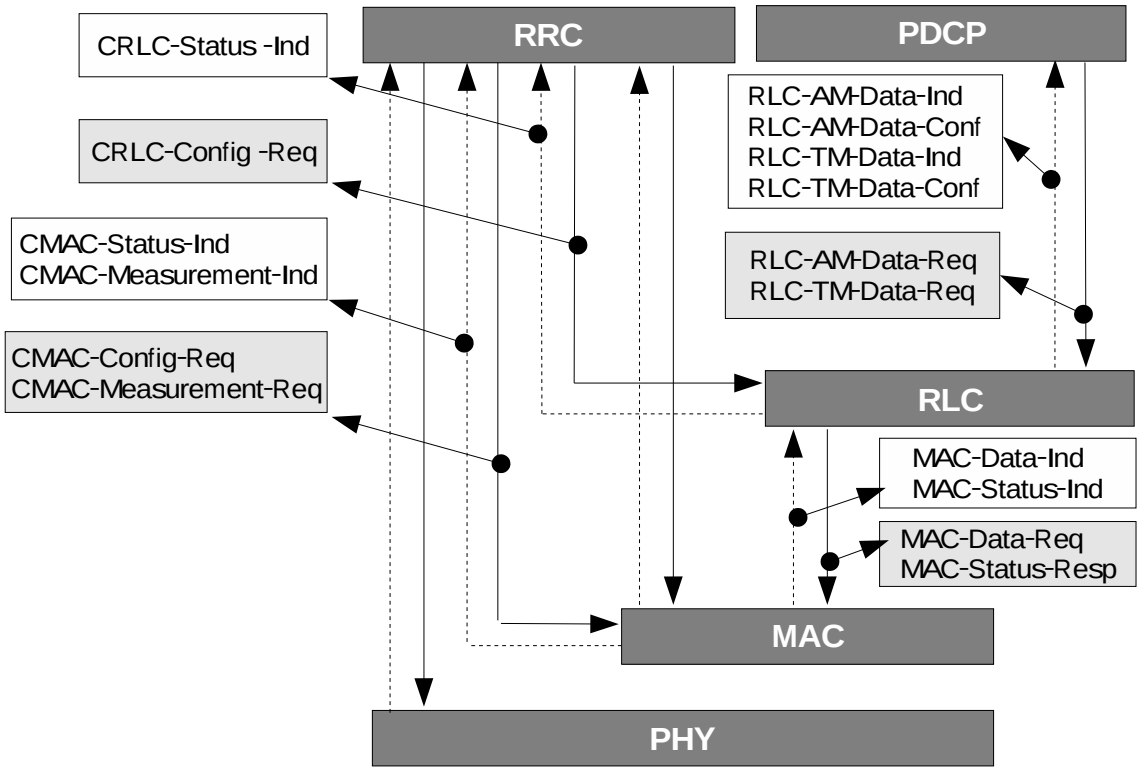

Figure 4-15: The primitives between the different protocol layers

More details about protocol layer interactions appear in Appendix B.

4.13 Radio wave propagation and interference model

It is the nature of radio wave propagation that makes the cellular mobile communication possible. However, it is also the nature of radio wave propagation that makes it difficult to model and/or predict the complicated phenomenon of the attenuation of radio waves. One of the most accepted and commonly used quality measures in the literature is the Signal-to-Interference Ratio (SIR, see section 4.14). The SIR is strongly dependent on the received power and thus related to power control (see section 4.4.1). In this section I provide some general background information about radio propagation and interference models in wireless networks. In the next section I will outline a SIR based propagation model in a macro cellular environment for the physical layer of the UMTS radio transmission modeled in the simulator used to conduct my simulations [52].

Radio reception quality can exhibit very large fluctuations in space and time. Sources of these fluctuations include changes in distance between receiver (User Equipment in the downlink) and transmitter (Node B in the downlink), and changes in environment or terrain and activity of other transmitters. Together these physical conditions produce four main effects that can be modeled as path loss, shadow fading, fast fading and interference all outlined in this section. The following equation can be used to calculate the transmission loss g, where $P(r)$ is the path loss, $LogSF(t)$ the correlated shadow fading and L_{FF} the fast fading.

$$g = P(r) \times LogSF(t) \times L_{FF}$$

4.13.1 Path loss

Path loss is defined as the ratio of received power over transmitted power for a give propagation path, and is usually a function of propagation distance. On a long time average, the observed power at the receiver depends mainly on the distance to the transmitter. Most electromagnetic waves in wireless systems propagate through environments more complex than free space, where they are reflected, scattered, and diffracted by walls, terrain, buildings, and other objects. Unfortunately, it is very hard to calculate the ultimative details of propagation in complex environments like mobile communication networks are residing in. An exact estimate of the path loss in UMTS is not available. However, as empirically observed overall trends show, path loss relates the average received power to the distance between the Node B and the User Equipment according to the general inverse propagation law $P(r) = A \times r^{-\alpha}$, where r is the distance between the User Equipment and the Node B, α a constant value called path loss exponent and A a constant. Users that moves towards the edge of the cell with their User Equipment will therefore experience a rapidly decreasing path loss.

Typical values for a city environment in UMTS are 200 to 300 meters for r, around -30 dB for A and 2-4 for the path loss exponent α, where 2 corresponds to free space propagation, and 4 to urban environments with high buildings [52].

4.13.2 Shadow fading

Shadow fading is the slowly varying component of propagation caused by obstructions from obstacles like hills, buildings or walls between the User Equipment and the Node B. As the User Equipment moves through the radio network the signal strength will fluctuate randomly due to the relative change in position of the obstacles between the User Equipment and the Node B and therefore the time factor needs to be included in the equation. This is called spatially correlated shadow fading.

Shadow fading is modeled by a multiplicative log-normal random distribution with zero-mean Gaussian random variable and a standard deviation σ. The standard deviation σ ranges from 4 to 12 dB, depending on the environment [7]. The following equation shows the calculation of the

$$\mathrm{LogSF}(t) = a \times \mathrm{LogSF}(t - dt) + \sqrt{1 - a^2} \times X$$

$$a = \exp\left(-0.5 \times \frac{d}{d_c}\right)$$

shadow fading, where X is a random variable with a log-normal distribution with 0 dB mean and σ dB standard deviation, d_c stands for the correlation distance and the parameter d denotes the distance covered by the User Equipment within the time interval dt depending on its velocity v. A typical value of d_c is 11, the time step dt is assumed to be 10 ms and for the user velocity a value of 3 km/h up to 5 km/h is adequate for a pedestrian scenario simulation [17].

Based on path loss alone the received signal power at a fixed distance from the transmitter should be constant. However, shadow fading causes the received signal power at equal distances from the transmitter to be different, since some locations have more severe shadow fading than others. Thus, to ensure that the received SIR requirements are met at a given distance from the transmitter, the transmit power must be increased to compensate for severe shadow fading at some locations. This power increase imposes additional burdens on the transmitter battery and causes additional interference to other users in the same frequency band as discussed in section 4.4.

4.13.3 Fast fading

The phenomenon of fast fading is related as the fluctuation of the amplitude of a radio signal due to the relative motion between transmitter and receiver. The fluctuation due to fast fading are much

faster than fluctuations due to shadow fading. In our simulator we do not consider effects from fast fading.

4.13.4 Interference

Interference is one of the major limiting factors in the performance of cellular mobile radio systems like UMTS. It is an additive disturbance to the received signal caused by the activity of other transmitters sharing the same radio channel. Unlike thermal noise, which can be overcome by increasing the transmitted power, increasing the transmission power cannot combat the interference. This is because increasing the subscriber transmission power will increase the interference for other subscribers (see Figure 4-2 for more details about this).

In the downlink, users are separated by orthogonal codes. Without multipath propagation the orthogonality between the codes remains and there would be only interference from neighboring cells. However, due to the high delay spread in macro cells, orthogonality is lost and the User Equipment will observe signals of other users within the same cell as interference. This fact is described by the orthogonality factor γ, which is typically set to 0.4 in macro cellular environments (see also the SIR equation in section 4.14) [7], [69].

In UMTS cells will be heavily overlapped and the coexisting transmissions will cause interference both in the operating cell (intra-cell interference) and in the neighboring cells (inter-cell interference). This can lead to an intra-cell interference of about 80% of the total interference.

4.13.5 Thermal noise

The thermal noise can be modeled as a zero mean additive white Gaussian noise (AWGN) and can be calculated from the following equation: $N = K \times T \times B \times F$ where K is the Boltzmann constant (1.38×10^{-23} Joules/Kelvin), T is the system temperature (290 K), B is the channel bandwidth in Hz and F is the noise figure.

In UMTS, the thermal noise power is around -103 dBm [7], which is a much smaller value than the value of the total interference.

4.14 Propagation model used in my simulations

In this section I outline a SIR based propagation model in a macro cellular environment for the physical layer of the UMTS radio transmission modeled in the simulator used for my simulations [67], [14].

In UMTS the average Signal-to-Interface Ratio (SIR) per Transmission Time Interval (TTI) at the mobile receiver can be defined using the following equation

$$\overline{SIR_i} = \frac{(g_{ii} \times p_i)}{\sum_{j=1;(j \neq i)}^{Q} g_{ij} \times p_j + P_{other} + n_i}$$

where

- SIR_i is the Signal-to-Interference Ratio at the receiver side (at the antenna connector of the mobile terminal) for the i^{th} active link
- p_i is the transmit power at the Node B used for serving the i^{th} link, g_{ii} the path loss, shadow-fading and fast fading model for the simulated environment for serving the i^{th} link
- g_{ij} is the cross-interference coefficient from the i^{th} link to the j^{th} link in the same cell at an instant time (includes the effects of path loss and shadow-fading, the orthogonality factor for the simulated environment and the spreading factors of the i^{th} and j^{th} link)

- P_{other} is the received inter-cell interference (interference caused by other cells)
- n_i is the received thermal noise power at the receiver i, which can be modelled with constant value in the simulation.

Fast power control loop in the simulation model are modelled ideal and therefore the path loss, shadow fading and fast fading (this is only appropriate for slowly moving UE like in my simulation scenarios, see section 6.7) in g_{ii} are compensated ideally by controlling the transmit power p_i up to the maximum available transmit power. The product $(g_{ii} \times p_i)$ can be then modeled with constant value in the simulation. However, notice that the total received power of the own cell

$$\sum_{j=1;(j\neq i)}^{Q} g_{ij} \times p_j$$

is going to be variable during the simulation.

Notice that in the ideal case of orthogonal links, the cross-interference coefficient would be equal to zero ($g_{i(j)} = 0$) for all $i \neq j$. The SIR_i is assumed to be constant in a frame, but may vary from frame to frame. From link-level simulations (see also section 6.2.2) we will receive a non-linear function f(.) that shows the coded_BER versus the SIR curve. We then finally only need to calculate the Block Erasure Rate (BLER) for packets via another non-linear function g(.) which maps the coded_BER to a BLER [14], [53].

$$BLER = g(SIR) = 1 - (1 - \overline{BER(10 \times \log(SIR)_{10})})^{Lengthofblock}$$

4.15 Summary and outlook

In this chapter I have presented some background information about UMTS that is needed to develop a UMTS module for the network simulator NS-2. In the following chapter I will outline some important issues in using TCP over wireless networks and will present the most important different approaches to sending TCP data over UMTS.

5 Performance issues of TCP over wireless networks

Many proposals for improving the performance of the Transmission Control Protocol (TCP) over wireless networks like UMTS have been published the last few years like [20], [21], [26],[31], [38], [39] and many more. However, some are of experimental character and will not fulfill the requirements of the industry. Realistic proposals can be categorized into the following groups: TCP end-to-end, split-connection and link-level solutions. TCP end-to-end solutions try to optimize current implementations of TCP versions. They especially address the issue that the nature of wireless links is significantly different from the nature of wired links. Many TCP parameter modifications have been proposed, especially for congestion control, to improve performance over wireless links. TCP split-connection solutions try to cope with the problems arising in wireless networks by splitting up the connection into a wireless part where protocols are used that were specially designed for wireless data transmission, and into a wired part that makes use of current established TCP protocol versions. Finally, the third solution introduces services, usually provided in the transport layer, into the link layer. By providing retransmission (automatic repeat request, ARQ) and forward error correction (FEC), link layer solution can shield the effects of the lossy wireless link partly from TCP [21].

In the following sections I first discuss some issues important to TCP in general, then outline important modifications to the original TCP versions proposed by researchers, then introduce split connection approaches, and finally present some promising link layer proposals.

5.1 Characteristics of wireless links

Wireless links are characterized by a very high and continuously varying bit error rate (BER). This high BER together with a large round trip time (RTT) and the fact that often only small sized packets are exchanged over wireless links are maybe the most important factors that limit the utilization of a wireless link.

Another major problem that often occurs in wireless networks is variable bandwidth in the connection between sender and receiver due to interference, mobility and QoS. Mobile clients can even become temporarily disconnected or need to perform a handover. Notice that due to the complexity of this issue, I will not consider this problem in my thesis and leave it to future work.

5.2 Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) provides a reliable, connection-oriented, byte stream, point-to-point transport layer service [28]. It has been the dominant and most thoroughly tested reliable transport layer protocol ever. Nowadays around 90% of the Internet traffic uses TCP as its transport layer protocol [72]. Because of its popularity many research papers about the protocol and its optimization proposals of it have been published, i.e. [10], [18], [22], [25], [31].

In the near future more and more services running over TCP will be offered in high-speed wired and wireless environments. However, the new high-speed environments like UMTS exceed the range for which TCP was initially designed, tested and tuned. TCP shows some undesirable patterns of behavior in the context of wireless networks. As a consequence more active research in the field of TCP over wireless networks (WLAN, UMTS) is in progress.

Probably the most challenging area will be the TCP congestion control in wireless networks as most current TCP implementations are designed for wired networks that rely on packet loss being a good indicator for network congestion. A congested network element is indeed a likely reason for a packet loss in wired networks with stationary hosts, but not in wireless networks.

Parameters	Description
Limited capacity	• Compared to the wired networks, wireless network only offers rather limited data rates.
Long Round Trip Times	• In general, wireless media exhibit longer latency delays than wired ones. This affects TCP throughput and increases the interactive delays perceived by the user.
Random losses	• Wireless links suffer from a high bit error rate that causes packet corruption, for example due to fading channels, shadowing, etc.
User mobility	• Wireless networks enable the user to move around. When a user is moving from one cell to another a handoff needs to be performed that requires the exchange of extra data. In addition to that frequent disconnection will occur due to temporary bad link quality.
Short Flows	• Most services offered in wireless systems include the transmission of rather small amounts of data. This means that when the application layer protocol opens a TCP connection for the transfer, that there is a very large probability that the whole transfer is completed while the TCP sender is still in the slow start phase. Therefore the TCP connection never manages to fully utilize the available bandwidth.
Power consumption	• For battery operated devices like UMTS UE's power consumption is a very important aspect. Typically, communication over a wireless medium consumes more battery power than CPU processing.

Table 5-1: Parameters that affect the performance of TCP in a wired-cum-wireless environment

While wired networks offer a virtually error free transmission medium, the error characteristics of wireless links cannot be ignored as wireless link errors tend to be very frequent and bursty. Wireless links are also highly sensitive to direction of propagation, fading (mobility) and general interference (e.g. interference of other users) that are much more likely the cause of a packet loss than congestion. Unfortunately these frequent errors trigger congestion control measures at the source and severely degrade performance due to non appropriate reaction (unnecessary reduction of the congestion window) of most TCP variants, discussed later in this chapter.

Table 5-1 provides a list of the most important parameters that affect the performance of TCP in a combined wired and wireless environment like UMTS.

The motivation behind TCP was to add reliability on top of an unreliable IP network. The original TCP incorporated a sliding window mechanism, which, in conjunction with packet acknowledgements and segment sequence numbers, guaranteed a reliable data transmission as well as flow control (to avoid overflowing the receiver's buffer) [27].

TCP makes use of the sliding window technique that allows the source to transmit a given number of segments before receiving an acknowledgement. After an acknowledgement is received, the window "slides" to allow one or more segments to be transmitted. In standard TCP cumulative ACKs are used that indicate the last correctly received segment with its sequence number.

In the beginning of the Internet, network congestion did not constitute a focus of concern due to the limited number of interconnected hosts. As the number of hosts increased, congestion problems became more evident and in 1988 a widely accepted congestion control algorithm was finally suggested [9]. After TCP Tahoe (see section 5.5.1), TCP Reno (see section 5.5.2) was the first major enhancement of the original paper introducing a slow start mechanism (see section 5.3.1) and a new way of calculating the retransmission timeout. Subsequently these new variants like TCP NewReno (see section 5.5.3) and TCP Vegas (see section 5.5.4) were proposed. All these papers have in common that they assume that TCP packets are transferred over a wired link with a negligible small Bit Error Rate (BER). Consequently, most standard TCP versions do not try to

detect the nature of the errors and do therefore not distinguish between losses due to congestion and losses due to bit errors on the link. Instead of tracking the error source and acting depending on what caused the error, they minimize the congestion window (cwnd) significantly any time an error occurred. This reduction of the congestion window size results in reduced utilization of the available bandwidth, and a long retransmit timeout period (RTO) results in long idling time before retransmission, thereby degrading the performance in terms of end-to-end throughput. In wireless networks this distinction between packet losses due to congestion or due to the lossy nature of wireless links is essential.

The next section discusses this issue in more detail and introduces methods how to cope with this problem.

5.3 TCP Congestion control in wireless networks

TCP congestion control addresses the problem when too many sources sending too much data, that is too fast for the network to handle. Congestion control is therefore slightly different from flow control, that tries to avoid an overflow of the receiver's buffer. Congestion control is well-known as one of the major problems in TCP and well analyzed in the literature.

The key idea behind congestion control is to avoid unnecessary retransmissions in order to achieve a better throughput/goodput. TCP utilizes acknowledgements to pace the transmission of segments and interprets timeout events as indicating congestion. The transmitter implements a congestion window (cwnd) that is defined as the maximum number of transmitted but still unacknowledged segments permitted.

The following subsections will introduce the different mechanisms in congestion control in the various versions of TCP. Note that not all components are implemented in all protocol variants.

5.3.1 TCP slow start

The aim of the slow start algorithm is to test the available capacity of the network, particularly at the connection establishment, and to reach the optimum rate of transmission as quickly as possible.

Slow start is used at the beginning of a TCP connection and in certain instances after congestion is detected. When a new connection is established with a host on another network, the congestion window is initialized to a certain amount of segments (used to be one but a bigger number of segments are now often suggested [41]). Each time an ACK is received, the congestion window is increased by one segment. The sender can transmit up to the minimum of the congestion window and the advertised window. The congestion window is control imposed by the sender, while the advertised window is control imposed by the receiver. The former is based on the sender's assessment of network congestion; the latter is related to the amount of available buffer space at the receiver for this connection. Assuming that the initial congestion window size is one segment the slow start algorithm works the following way. The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of those two segments is acknowledged, the congestion window is increased to four. This provides an exponential growth typically sending one ACK for every two segments that it receives. Notice that only the congestion window size changes during TCP slow start. The advertised window size remains the same.

5.3.2 TCP congestion avoidance

As soon as the congestion window reaches the slow start threshold (ssthresh), the slow start algorithm hands over the congestion window (cwnd) control to the congestion avoidance algorithm. In this mode, the primary objective is to maintain high throughput without causing congestion. If TCP detects segment loss, it assumes that congestion has been detected. As a corrective action, TCP reduces its data flow rate by reducing the congestion window (cwnd). After reducing cwnd, TCP goes back to slow start. In congestion avoidance mode further increase of the transmission

rate is only linear by increasing the congestion window by one per Round Trip Time (RTT). This is a linear growth of cwnd, compared to slow start's exponential growth. The increase in cwnd should be at most one segment each round-trip time (regardless how many ACKs are received in that RTT), whereas slow start increments cwnd by the number of ACKs received in a round-trip time. Linear increase of the congestion window continues until a timeout at the sender occurs due to a missing acknowledgement, or until the sender detects a gap in the transmitted data because of continuous acknowledgements for the same packet. In either case the sender sets the congestion threshold to half of the current congestion window. The congestion window itself is set to one segment and the sender starts sending a single segment. The exponential growth as described above starts once more up to the new congestion threshold, the window grows in linear fashion.

5.3.3 Fast retransmit / Fast recovery

The fast retransmit and the fast recovery mechanism allow TCP to detect and recover from segment drops more effectively than relying on the retransmission timer.

The fast retransmit algorithm is modified in the different versions of TCP. The basic version works the following way that after receiving the threshold number (usually 3 and not 1 since we don't know whether a duplicate acknowledgement is caused by a lost segment or just a reordering of segments) of duplicate acknowledgements (dupacks) for the same TCP segment, the data source infers that a packet has been lost and then performs a retransmission of what appears to be the missing segment, without waiting for a retransmission timer to expire. This leads to higher channel utilization and an increased throughput.

The idea behind fast recovery is that a dupack is an indication of available channel bandwidth since a segment has been successfully delivered. This in turn implies that the congestion window (cwnd) should actually be incremented upon a dupack delivery. In particular, receiving the threshold number of dupacks (usually set to 3) triggers fast recovery. In this phase the sender retransmits one segment, and sets the congestion threshold to half the current congestion window (cwnd). Then, instead of entering slow start like in TCP Tahoe, the sender increases its current congestion window (cwnd) by the dupack threshold number. Thereafter, and for as long as the sender remains in fast recovery, the congestion window (cwnd) is increased by one for each additional dupack received. The fast recovery phase is completed when an acknowledgement for new data is received. The sender then sets the congestion window (cwnd) to the current congestion threshold value and resets the dupack counter.

Fast recovery is particular effective over wireless links where sporadic random packet losses due to the noisy and fading nature of a radio channel and general interference are often (in most TCP versions, not in TCP Westwood) misinterpreted as a symptom of congestion and thus lead to an unnecessary congestion window reduction described above [25], [23].

5.3.4 Delayed ACKs

TCP Reno was the first implementation that introduced delayed ACKs in the receiver. Acknowledgements are sent only for every other packet received, thus reducing traffic in the backward direction. However, TCP receivers shall not delay ACKs more than 500ms to prevent spurious retransmissions of already received segments.

5.3.5 Explicit feedback

Unlike the implicit feedback algorithms used for example in TCP end-to-end solutions (see section 5.5) the explicit feedback used for example in split-connection solutions (see section 5.6) network component (e.g. RNCs) provides congestion indication explicitly to the sources. Unlike local recovery algorithms at the RNC, where timeouts still occur at the source, explicit feedback does not need to make use of those timers in the source. In explicit feedback ICMP source quench packets are sent over the network to inform the source about bit errors occurred on the lossy link and to prevent timeouts during local recovery. This improves goodput and throughput of the connection. In

addition to that no state maintenance is required at any intermediate host. However, modification to the TCP code at the source and modifications to network elements like RNCs are required, which makes the explicit feedback algorithm not very popular [20].

5.4 TCP add ons

The idea behind TCP add ons is to introduce mechanisms that improve TCP performance over wireless links by allowing TCP to distinguish between losses due to congestion and bit errors, or to extend TCP's error recovery mechanisms to be able to repair multiple losses in a single window. However, their potential is limited. Retransmissions caused by losses on the final wireless hop have to travel the whole way from the sender to the receiver, which can take some time. Furthermore, they require that all TCP implementations are modified according to the introduced mechanisms.

In the following subsections the RCP add ons TCP SACK, Explicit Loss Notification and TCP Eifel are lined out.

5.4.1 TCP SACK (Selective ACKnowledgment)

Traditional implementations of TCP use an acknowledgment number field that contains a cumulative acknowledgement, indicating the TCP receiver has received all of the data up to the indicated byte. A selective acknowledgement option allows receivers to additionally report up to three non-sequential blocks of data they have successfully received. So the sender does only need to retransmit the missing TCP segments.

RFC 2018 [10] defines TCP Selective ACKnowledgment (SACK) as an option to standard TCP specification that does not change the basic underlying congestion control algorithms. The main difference between the TCP SACK implementation and the TCP Reno implementation is in the behavior when multiple packets are lost from one window of data. With TCP SACK, the sender is able to identify and retransmit multiple lost packets within the same RTT if there are enough ACKs returning to the sender.

The TCP SACK option field contains a number of SACK blocks, where each SACK block reports a non-contiguous set of data that has been received and queued. The first block in a SACK option is required to report the data receivers's most recently received segment, and the additional SACK blocks repeat the most recently reported SACK blocks. When the SACK option is used with the Timestamp option specified for TCP Extensions for High Performance, then the SACK option has room for only three SACK blocks.

As in TCP Reno, the TCP SACK implementation enters fast recovery when the data source receives more than the threshold number of duplicated acknowledgements (dupacks). The source retransmits a packet and cuts the congestion window (cwnd) in half. During fast recovery, SACK maintains a variable called pipe that represents the estimated number of packets outstanding in the path. This variable is used during fast recovery for limiting the sender's sending rate.The source only sends new or retransmitted data when the estimated number of packets in the path is less than the congestion window (cwnd). Pipe is incremented by one when the source either sends a new packet or transmits an old packet. It is decremented by one when the source receives a duplicated acknowledgements (dupacks) with a SACK option reporting that new data has been received at the receiver. The use of the variable pipe decouples the decision of when to send a packet from the decision of which packet to send [11]. Due to its good performance in case of multiple losses in one data window, the IETF network working group suggest that UMTS should support TCP SACK [41].

However, other researchers claim that particularly for mobile hosts, it is doubtful if the introduced complexity by using TCP SACK, which implies more power consumption and increased memory requirements, is worth implementing [58]. Table 5-2 shows a list of advantages and disadvantages of using TCP SACK in a wireless network.

Table 5-2: Advantages and disadvantages of using TCP Sack over a wireless network

Advantages	Disadvantages
The source has better information of the packets that have been successfully delivered compared to other TCP versions. It can therefore avoid unnecessary delays and retransmissions.	Requires modification to the acknowledgement procedures at both sender and receiver sides.

5.4.2 Explicit Loss Notification (ELN)

Explicit Loss Notification (ELN) is a mechanism that allows to communicate the reason of a packet loss to the source. In case the RNC knows for sure that the loss of a packet was not due to congestion, it sets the ELN bit in the TCP header, caches the information about which packet was lost and sends it to the source of the lost package. The RNC does not perform any retransmission, it just monitors all incoming TCP segments and therefore does not need to cache any data segments. To prevent from wrong marking of segments that have been lost due to congestion, the RNC does only set the ELN bit if the number of packets at the interface queue at the RNC is far apart of the maximum length of the queue [19].

Important to notice is that ELN is very difficult to implement, it does not invoke a congestion control algorithm and there is doubt about an efficient solution for UMTS [58].

5.4.3 TCP Eifel

The TCP Eifel algorithm was propose as an enhancement to TCP's error recovery scheme. It eliminates the retransmission ambiguity, thereby solving the problems caused by spurious timeouts and spurious fast retransmits. Eifel enables TCP to determine what packet an acknowledgement corresponds to, i.e. if it corresponds to an original or a retransmitted packet. This is done by using timestamps and by recording the time when the retransmitted packet is sent. When TCP Eifel detects that the acknowledgement corresponds to the original packet, there is no need to reduce the congestion window. As this also eliminate using the slow start and congestion avoidance mechanism it lead to an increased throughput.

TCP Eifel uses the TCP timestamp option that is not included in older TCP implementations and can be incrementally deployed as it is backwards compatible and does not change TCP's congestion control semantics. In wireless environments where spurious retransmissions occur frequently, the algorithm can improve the end-to-end throughput by several tens of percent. The Eifel algorithm finally makes TCP truly wireless-capable without the need for proxies between the end points. Another key novelty is that the Eifel algorithm provides for the implementation of a more optimistic retransmission timer because it reduces the penalty of a spurious timeout to a single (in the common case) spurious retransmission [43].

Major advantages of TCP Eifel are its simplicity and that it is already implemented and in use in operating systems like Linux.

5.5 End-to-end solutions

These solutions retain a single TCP connection from the sender to the receiver and the newer variants attempt to make TCP aware of wireless losses so that it can deal with them. Many papers, as listed in section 5.2, have been published recently proposing modifications to older TCP implementations. To give an overview of the most important versions considering my simulations they are described briefly in the following subsections.

It is important to notice that enhancements to TCP do not always help on wireless links. Clever design at the link layer (RLC) in combination with older TCP versions may be more effective.

5.5.1 TCP Tahoe

TCP Tahoe as an early version of TCP included slow start (see section 5.3.1), congestion avoidance (see section 5.3.2) and fast retransmit (see section 5.3.3) in the congestion control. The refinements of the original TCP version include a modification to the RTT estimator used to readjust retransmission timeout values. The problem of TCP Tahoe is that slow start is not always efficient, especially if the error was random in nature. In such a case the massive shrinkage of the congestion window is unnecessary. TCP is in such a situation unable to fully utilize the available bandwidth of the radio channel during the phase of window re-expansion.

5.5.2 TCP Reno

The TCP Reno implementation retained the enhancements incorporated into Tahoe, but introduced fast recovery in conjunction with fast retransmit.

Using fast recovery significantly improves performance compared to TCP Tahoe when a single packet is lost from a window of data, but can negatively impact performance for multiple packets drop from a single window of data, which is often the case on a wireless link like in my simulations over UMTS.

Table 5-3: Advantages and disadvantages of using TCP Reno over a wireless network

Advantages	Disadvantages
Performs well over WLAN compared to TCP Tahoe when only a single packet is lost from one window of data [11].	Poor performance over WLAN compared to TCP Tahoe when multiple packets are lost from one window of data [11].
	Cannot distinguish between congestion loss and packet errors.
	Overreacts to packet errors.

Table 5-3 shows a list of advantages and disadvantages by using TCP Reno in a wireless network.

5.5.3 TCP NewReno

TCP NewReno includes a small change to the TCP Reno algorithm at the source side that eliminates the waiting time for a retransmit timer to expire when multiple packets are lost from a window. The change concerns the sender's behavior during fast recovery when a partial acknowledgement is received that acknowledges some, but not all of the packets that were outstanding at the start of that fast recovery procedure. In TCP Reno, partial acknowledgements take TCP out of fast recovery by decreasing the size of the usable window back to the size of the congestion window (cwnd). Thus TCP NewReno has to wait for a timeout in case of multiple losses per window. In TCP NewReno, partial acknowledgements do not take TCP out of fast recovery. Instead, partial acknowledgements received during fast recovery are treated as an indication that the packet immediately following the acknowledged packet in the sequence space has been lost, and should be retransmitted. Thus when multiple packets are lost from a single window of data, TCP NewReno can recover without a retransmission timeout, retransmitting one lost packet per RTT until all of the lost packets from that window have been retransmitted. TCP NewReno remains in fast recovery until all of the data outstanding when fast recovery was initiated has been acknowledged.

Table 5-4 shows a lists of advantages and disadvantages of using TCP NewReno in a wireless network.

Table 5-4: Advantages and disadvantages of using TCP NewReno over a wireless network

Advantages	Disadvantages
Performs better than TCP Reno over UMTS when multiple packets are lost from one window of data [11].	Cannot distinguish between congestion loss and packet errors.
Modifications are only needed in the sender.	
Very popular protocol overall [24].	

5.5.4 TCP Vegas

TCP Vegas approaches the problem of congestion from another perspective than TCP NewReno. The basic idea is that TCP Vegas attempts to estimate the level of congestion before it happens, and consequently avoid sending unnecessary packet drops. Based on sample RTT measurements, and the size of the sending window, the sender calculates the throughput rate every RTT. This rate is compared to an expected rate, which is calculated based on what is up to now known as best RTT. If RTT becomes larger, the source will shrink its congestion window (cwnd), thus reducing its transmission rate. The opposite effect takes place if RTT becomes shorter.

A major disadvantage of TCP Vegas is its path asymmetry. The sender makes decisions based on the RTT measurements, which might not accurately indicate the congestion level of the forward path. Another drawback is that the algorithm is still new and not fully embedded in most TCP implementations [12].

In general TCP Vegas does not seems to be a good choice for a UMTS network as it was shown that it does not achieve high utilization in large bandwidth delay networks like UMTS, because of its slow-start phase [59].

Table 5-5: Advantages and disadvantages of using TCP Vegas over a wireless network

Advantages	Disadvantages
Good performance over WLAN when using Snoop protocol [26].	Cannot distinguish between congestion loss and packet errors.
	Poor performance over WLAN when multiple error bursts (>4 losses) occur without using Snoop protocol [26].
	Asymmetric path.
	Algorithm is currently not embedded in most TCP implementations.

Table 5-5 shows a list of the advantages and disadvantages by using TCP Vegas in a wireless network.

5.5.5 TCP HeAder CheckSum (HACK) option

TCP HACK uses TCP Header checksum to detect bit errors in the payload or in the header of a TCP segment. It assumes that, when packet corruption occurs, it is more likely that the packet corruption occurs in the data than in the packet header because the size of the header is usually much smaller than the size of the payload. The algorithm in TCP HACK is able to recover an uncorrupted header and thus determine that packet corruption and not congestion has occurred in the network. The receiver can then send a “special” acknowledgement back to the source indicating packet corruption. If there is an error in the payload, the congestion window (cwnd) remains the same. TCP

HACK introduces two TCP options. One option is for data packets and contains the checksum of the TCP header (normal TCP carries only the checksum of the entire TCP segment). The other option is for ACKs and contains the sequence number of the TCP segment that is corrupt. However, TCP HACK premise that RLC does not drop the corrupted packet. This is not likely to be the case.

Table 5-6 shows a list of the advantages and disadvantages by using TCP HACK in a wireless network.

Table 5-6: Advantages and disadvantages of using TCP Hack over a wireless network

Advantages	Disadvantages
Performs well over WLAN in cases where burst corruptions are frequent [22].	Assumes that the RLC protocol does not drop corrupted packets.
Can co-exist nicely with TCP SACK[22].	

5.5.6 TCP Westwood (TCPW)

TCP Westwood is a new TCP version that is designed to perform well in wired, wireless and mixed networks [25]. In TCP Westwood the TCP Reno congestion control is used and is modified only on the sender side. TCP Westwood defines that if the congestion window size divided by the minimum RTT is larger than the currently achieved rate the channel is congested and if it is equal to the current rate, the loss is of random nature. The key innovative idea behind TCP Westwood is that it takes advantage of an end-to-end bandwidth estimation mechanism called Bandwidth Share Estimates (BSE) to set the values of slow start threshold (ssthresh) and congestion window (cwnd) after a random (due to the lossy nature of a wireless link) loss (indicated by 3 duplicated acknowledgements that have been received or by a specific timeout).
Instead of setting the slow start threshold after receiving 3 duplicated ACKs or after the timeout expires to half of the size of the congestion window (as in TCP Reno) TCPW sets the ssthresh to the product of the BSE and the minimum RTT. This estimation is not based on measurements performed by lower layers and therefore retains to layer principles like separation and modularity of layers. The bandwidth is estimated by the source that monitors continuously the received TCP acknowledgements. It then estimates the data rate currently achieved by the connection. By doing this, TCP Westwood ensures both faster recovery and more effective congestion avoidance.

In a refinement of TCP Westwood a new Rate Estimate (RE) was introduced that makes Bandwidth Share Estimates (BSE) unnecessary. However, research has been performed showing that in case of using lossy wireless links, BSE performs better than RE [24].

Table 5-7 shows a list of the advantages and disadvantages by using TCP Westwood in a wireless network.

Table 5-7: Advantages and disadvantages of using TCP Westwood over a wireless network

Advantages	Disadvantages
Faster recovery and more effective congestion avoidance lead to better throughput and delay performance over WLAN [25].	Migration problem [50]
Designed for wired-cum-wireless networks.	TCP-W is not sufficiently evaluated.
Makes use of bandwidth estimation without the support of lower layers [25].	
Capability to distinguish and isolate congestion loss from wireless loss [25].	

5.5.7 TCP Westwood Bulk Repeat (TCPW BR)

The TCP Westwood Bulk Repeat proposal [37] introduces a TCP version that runs in two modes: the “error loss” mode and the “congestion loss” mode. The “congestion loss” mode behaves like TCPW. The “error loss” mode however, relies on three modifications: Bulk repeat, fixed retransmission timeout and intelligent window adjustment. Bulk repeat means that all outstanding packets are immediately retransmitted when possible multiple losses are detected in the same data window.

5.5.8 TCP-Real

The TCP-Real proposal [32] introduced a receiver-oriented approach to congestion control. Beside the “blind” increase and decrease window adjustments, TCP-Real implements a measurement based transmission strategy. Goals of this protocol were to enhance the real-time capabilities of the protocol over wired/wireless networks and the decoupling the size of the congestion window from the timeout [34]. In TCP-Real the timeout can be extended but the window size could remain the same or even increase.

5.6 Split-connection solution

Unlike the end-to-end solutions, the split-connection solutions completely shield the wireless link from the sender by terminating the TCP connection at the Node B or the RNC.

These two connections can be highly optimized for the environment present at each part of the connection. However, they impose increased complexity and dependency especially on the RNC, where the two connections come together. Notice that it would be also possible to split up the connections at the Node B, but unfortunately it is not planned to implement the whole protocol stack up to the TCP layer in the Node B and it is therefore not realistic to use this approach.

One of the big advantages of the split-connection solutions is also that the TCP implementation in the sender does not need to be modified to deal with the enhanced functionality required for the wireless hop.

The most prominent examples that fall under this category are Indirect TCP (I-TCP) and Wireless TCP (WTCP).

5.6.1 Indirect TCP (I-TCP)

I-TCP was implemented at Rutgers University as a part of the Dataman Project. The I-TCP proposal [18] splits the transport link at the wired-wireless boundary so as to completely shield the sender from the effect of wireless losses. RNC maintains two connections, one TCP connection (standard TCP is used) over the fixed network, and another connection (does not have to be TCP) over the wireless link. This way, the poor quality of the wireless link is hidden from the fixed network. This provides an elegant method for accommodating the special requirements of a UE in a way that is backward compatible with the existing fixed network. All the specialized functions can be built into the wireless side of the interaction while the fixed side is left unchanged.

However, this approach violates the semantics of end-to-end reliability as I-TCP does not maintain end-to-end TCP semantics. Furthermore acknowledgements can even arrive at the source before the packet actually reaches the intended destination. Another drawback is the overhead that I-TCP incurs since packets are being processed twice. The handover procedure is also a complicated and time-consuming operation [33].

Table 5-8 shows a list of the advantages and disadvantages by using I-TCP in a wireless network.

Table 5-8: Advantages and disadvantages of using I-TCP over a wireless network

Advantages	Disadvantages
Can distinguish between congestion loss and packet error.	A big amount of microflow-states and even data packets need to be stored in large buffers. The base stations have to be complex and scaling problems arise.
Does not require any changes in the TCP protocol as used by the hosts in the fixed network.	Packets go through the TCP protocol stack twice.
Was designed to enhance performance on wireless links with mobile end points [18].	Handover is very time-consuming since the TCP state must be transferred between the Node Bs [33].
	End-to-end TCP semantics is not preserved in this approach.
	Acknowledgement may reach the sender before the data packet is actually received by the receiver.

5.6.2 Wireless TCP (WTCP)

Wireless TCP (WTCP) was proposed to boost the performance of TCP in wireless networks. WTCP splits a TCP connection between a fixed host and a UE into two segments with end-to-end semantics. It is included into the RNC to be able to perform better estimates of the retransmission timeout and to speed up the packet retransmission on the wireless link. WTCP does not require any modification to the TCP code at the fixed or the mobile host. The RNC that monitors the mobile host buffers the incoming data from the CN and tries to hide wireless channel errors by local retransmission over the wireless hop. Unlike other TCP versions like TCP Reno, WTCP interprets the reception of a dupack as an indication that the wireless link itself is in a good state.

Promising about WTCP is that it effectively hides the time spent by the RNC for local recovery so that the TCP RTT estimation at the source is not affected [31].

Table 5-9 shows a list of the advantages and disadvantages by using WTCP in a wireless network.

Table 5-9: Advantages and disadvantages of using WTCP over a wireless network

Advantages	Disadvantages
Performs better than I-TCP when the wireless link quality is bad [34].	WTCP does not scale well.
Designed for wireless environments.	

5.7 Link Layer solution

The main purpose of the link layer solutions is to hide errors due to the wireless link from TCP by using retransmission and forward error correction (FEC) on the link layer. With this, the transmission media heterogeneity remains transparent to TCP and does not force any modification to current TCP implementations. To do the retransmission, the transmitter will have to buffer the outgoing data until it can determine that the data has been received correctly. The receiver on the other side will also need a buffer to store the incoming data and need processing capacity for detecting errors and for ordering retransmissions. This introduces additional complexity to the system and leads to extra costs.

TCP and link layer ARQ are quite similar retransmission mechanisms, the former against loss and the latter against error. Unfortunately, they interaction of RLC with TCP retransmission mechanisms may result in a conflict and cause service degradation mainly in terms of throughput. Therefore, special care needs to be taken when setting the values of the link-level ARQ and the TCP retransmission parameters.

Another major drawback of link layer solution is that TCP has to deal with a significant packet delay variation.

5.7.1 Snoop protocol

Snoop is a link layer protocol that can be applied to all existing TCP variants. Snoop agents, located in the RNC, monitor every packet that passes through the TCP segments sent from the source and that the receiver has not yet acknowledged. The protocol entities store copies of TCP data packets in a cache and monitor the acknowledgements in the reverse direction. If a packet loss is detected (e.g. by receiving duplicated acknowledgments) the cached copy is used for a local retransmission of the packet, and any packets carrying duplicated acknowledgement (dupack) information back to the TCP source are suppressed from the TCP sender until local retransmission success. This is very effective in reducing the number of retransmissions, and more importantly in preventing the associated reduction of the congestion window (cwnd) size.

One of the main objectives of Snoop is to leave existing implementations in the mobile and fixed nodes unmodified. However, snoop has its own limitations. One major disadvantage of using the snoop protocol is that it requires an additional snoop proxy in the RNC that handles the local retransmissions. Another one is that a big amount of microflow-states and even data packets need to be stored in the network, which scales badly. A third is that unlike in I-TCP (see section 5.6.1), Snoop does not completely shield the wireless link losses from the fixed network, and source timeout is still possible. Moreover, Snoop does not cover the cases where the UE gets involved in handovers. In such cases, the Snoop cache in the old RNC remains unused and in the new RNC no information is cached yet. Table 5-10 shows a list of the advantages and disadvantages by using the Snoop protocol in a wireless network.

Table 5-10: Advantages and disadvantages of using Snoop over a wireless network

Advantages	Disadvantages
No changes to the fixed network needs to be made.	Problems with encryption: IP Sec or IP tunnelling does not work with snoop.
	Cannot be used if TCP data and TCP ACKs traverse different paths.
	Designed for systems with small RTT.
	Does not consider packet loss and delay due to handoff.

5.7.2 UMTS RLC AM

Another promising approach to improve the overall performance of TCP over UMTS is to enhance the link layer of UMTS (RLC) by incorporating automatic repeat request (ARQ) and/or forward error correction (FEC). This significantly improves the bit error rate seen by the TCP layer as in case of an error, a RLC frame which is smaller than a TCP segment is affected. However, although the loss rate of packets shown to upper layers is very low due to ARQ and FEC, the recovery of data at the RLC layer can appear as an additional delay and a delay jitter to the upper layers if the recovery of the erroneous packets cannot be done before the TCP retransmission timer expires. In my simulations I try to find out if the overall performance is better with or without using RLC

acknowledged mode. Table 5-11 shows a list of the advantages and disadvantages by using RLC AM in UMTS.

Table 5-11: Advantages and disadvantages of using RLC AM in UMTS

Advantages	Disadvantages
RLC AM hides the lossy wireless link from TCP and improves significantly TCP performance when using standard TCP.	TCP has to deal with large packet delay variation.
	Additional complexity needs to be added to the system.

5.8 Discussion of the different approaches

Automatic Repeat Request (ARQ) in link layer approaches causes a shorter control loop than loss recovery over the end-to-end path. It can operate on individual link frames, which are usually smaller than upper layer PDUs. In addition to that it may use knowledge about the network that is hided from the transport layer. Major drawback of the link layer approaches is that the link transit delay would be increased when frames are retransmitted. Other factors that have a significant influence on the performance of TCP are incompatible timer settings at the link layer and TCP layer that could cause redundant retransmissions or out-of-order delivery of packets at the link level, which invokes unnecessarily TCP's fast retransmit mechanism.

Split-connection solution are an interesting alternative but they don't scale well and they add significant complexity to the system. Every packet that is sent across the TCP connection has to go through the TCP stack four times. It is interesting that most proposals in this area suggest to split up the connections at the base station (Node B in UMTS). However, the TCP layer is not implemented in the Node B to keep the complexity and costs for it at a low level. Another problem with split-connections are that the connection needs to be split up probably at the RNC and not at the Node B. As a consequence, the wireless connection is not fully wireless anymore as the connection between the RNC and Node B is wired. Due to all this drawbacks split connections are not advisable to use.

Solutions that only modify TCP parameters or TCP implementations like TCP Westwood are very attractive, because they improve the performance significantly while no costs in terms of equipment. Another major advantage of TCP Westwood is that it shows significant improvements in wired environment as well as in wireless environments (this shows test over WLAN) without changing the end-to-end semantics of TCP. TCP Westwood is backwards compatible, will work also in a partially modified environment and does not add any overhead to the data transfer. However, if the support for this solution is not widespread, TCP Westwood does not lead to a major improvement in terms of performance. It is important not to underestimate the time it would take for such a migration. Another major drawback with TCP Westwood is that it takes some time in the initial phase of a TCP connection until TCP Westwood can obtain enough information of the link to perform optimal. In addition to that there are uncertainties about how accurate the mechanism that allows distinction of loss due to error or congestion is. The other promising TCP variant for performance evaluation seems to be TCP SACK as it requires less retransmissions compared to TCP variants with cumulative acknowledgement mechanisms.

5.9 Summary and outlook

In this chapter I have presented the most important issues concerning TCP data transmission in wireless networks with focus on the UMTS network. Furthermore, the most promising approaches of how to efficiently send data over an UMTS network using the TCP protocol were defined. In the next chapter I will present my UMTS simulator models along with the simulation scenarios and simulation parameters.

6 Description of the UMTS Simulator and used simulation parameter settings

In this chapter I present my UMTS simulator implementation and simulation models along with the simulation scenarios and simulation parameters.

The goal of my simulations is to study the impact of using different TCP variants and different TCP and RLC settings on the system performance. All simulations were conducted with the network simulator NS-2 (see Appendix C).

6.1 Simulation host

I have used the network simulator NS-2 [70] developed by the Lawrence Berkeley National Laboratory as the host simulator for the developed UMTS radio network simulator module. I chose the network simulator NS-2 because of the range of features it provides and because it is open source and can be modified. Details about the network simulator NS-2 appear in Appendix C.

6.2 General UMTS radio interface simulator model

Due to the complexity of the UMTS radio interface simulator I introduce a general UMTS radio interface simulator model. Based on this model I will explain the basic components of the simulator before presenting details about the implementation and simulation parameter settings.

The general simulator model consists of an environment component (rural, urban, indoor; traffic distribution like hot-spot, uniform or highway), a site component (cell size, type of antennas), a mobility component (mobile speed), a traffic component (speech, web browsing, streaming), a radio network component (Radio Resource Management, radio protocols) and a radio propagation component (distance, attenuation, fading). Figure 6-1 provides an overview of the different components in a general UMTS simulation model.

In the following subsections each component, apart from the core component, the radio network component, is outlined.

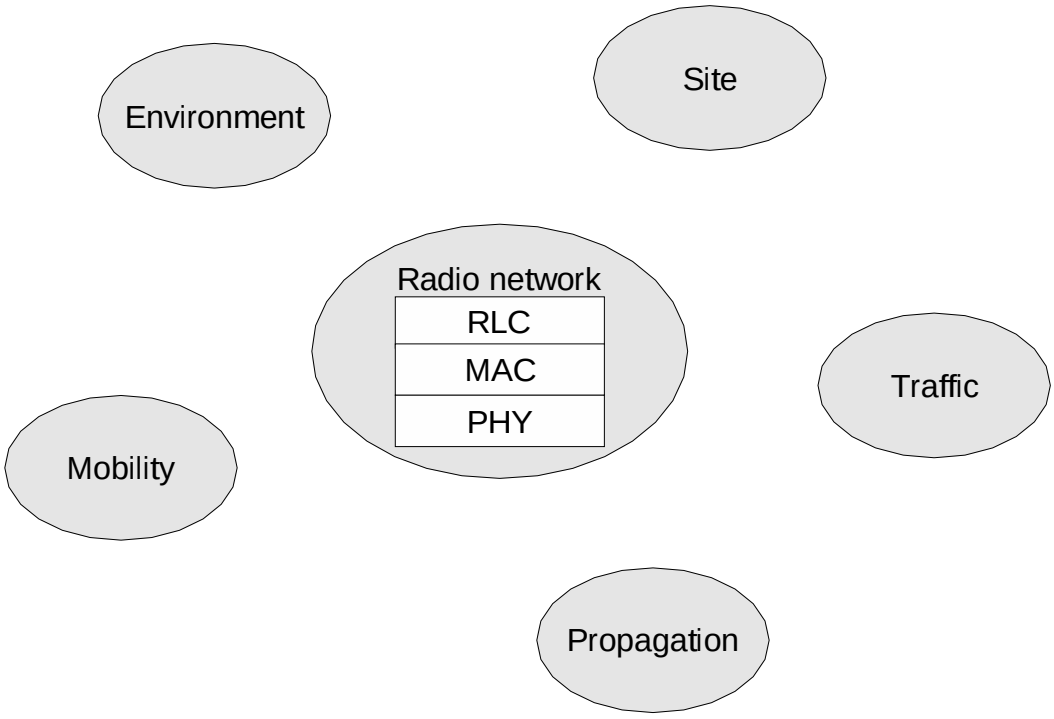

Figure 6-1: Simulator model

6.2.1 Environment model

For the simulations an urban environment with a hot-spot spatial user distribution was used. Details about the environment model settings can be found in Table 6-1.

6.2.2 Propagation model and link-layer simulations

To run simulations within a reasonable time, the details of the physical layer transmission such as modulation, coding etc. have to be ignored. Instead, the physical layer performance has to be phenomenologically described via a mapping function. The idea behind this is that the computationally heavy link-level simulation is performed only once, to create mapping functions. The parts of the simulation model that are analyzed in link-layer simulations are mainly the propagation model and the physical layer of the radio network model. Outputs of link-level simulations are requirements to reach a certain link quality, for example the target Signal-to-Interference Ratio (SIR) to reach a average bit or block error rate. Subsequently, the system simulator uses the mapping functions to convert from Signal-to-Interference Ratio (SIR) into data Bit Error Rate (BER), which allows us to estimate the physical layer performance of each radio channel.

For my simulations I used a mapping function derived from UMTS link-level simulations performed by a research group led by Professor Zorzi at University of Ferrara (see Appendix section D.1).

As already mentioned, the outcome of these link-level simulations is a graph that shows a curve which describes the bit error rate (BER) as a function of the Signal-to-Interference Ratio (SIR). With this the system working point required for each service can be calculated. This is then used by the system-level simulations that consider system-wide parameters, which do not only affect the properties of one link but the performance of the whole system. The system-parameters can be propagation-related parameters, system-dependent parameters, and user-dependent parameters. Examples of propagation-related parameters are path loss and shadowing, noise and interference, cell radius and environment. Examples of system-dependent parameters are power control step size and channel allocation. Examples of user-dependent parameters are spatial user distribution (for example hot-spots), user mobility, traffic symmetry and transmission activity. Table 6-1 shows

the most important system level parameters used in my simulations. In addition to these parameters section 6.7 and section 6.8 provides also protocol level simulation parameters.

6.2.3 Traffic model

To simulate traffic, I use FTP data traffic generators (see Figure 6-2). To simulate different traffic loads I chose three different load scenarios. When indicating high traffic load, I have used thirty, the maximum number of active User Equipments supported by the UMTS simulator, User Equipments for my simulations. When indicating medium traffic load, I have used twenty and with low load only five active User Equipments for my simulations. Each user sets up a FTP connection at the beginning of the simulation and the sender then constantly sends data until the end of the simulation. The number of active User Equipments is therefore equal to the number of FTP flows as each User Equipment produces one FTP flow. The number of active User Equipments in the system is always constant. All data flows are assumed to be in the same traffic class, the background class. Within the traffic class, all user have the same priority level.

6.2.4 Mobility model

Generating node-movement models for large wireless scenarios manually is laborious. Three different mobility scenarios were created with the help of the application sedtest (see Appendix section C.9). All three scenarios have an average pause between movement of 5 s, with a maximum speed of 2 m/s, with a simulation time of 500 s and with a maxx and maxy of 500m. The low load scenario introduces 5 mobile nodes to the system, the mid load scenario 20 and the high load scenario introduces 30 nodes, the maximum number of mobile handsets supported by the simulator.

6.2.5 Site model and simulation topology

The simulation topology consists of a series of FTP traffic generator sending continuously FTP data packets to a corresponding User Equipment. The packets are firstly received by a Radio Network Controller (RNC). The RNC then forwards the data packets to a Node B that finally sends the packets to the corresponding User Equipment, as depicted in Figure 6-2. The wired connection between the FTP traffic generators and the RNC is modeled as a 40 Mb 50 ms Drop Tail link and the wired connection between the RNC and the Node B as a 200 Mb 10 ms Drop Tail link. With this model, it is assured that these two links are not the bottleneck of the system. Notice that although ATM technology is used at the link between the Node B and the RNC in real UMTS systems, no real ATM layer is simulated in the simulation model (see also section 4.2).

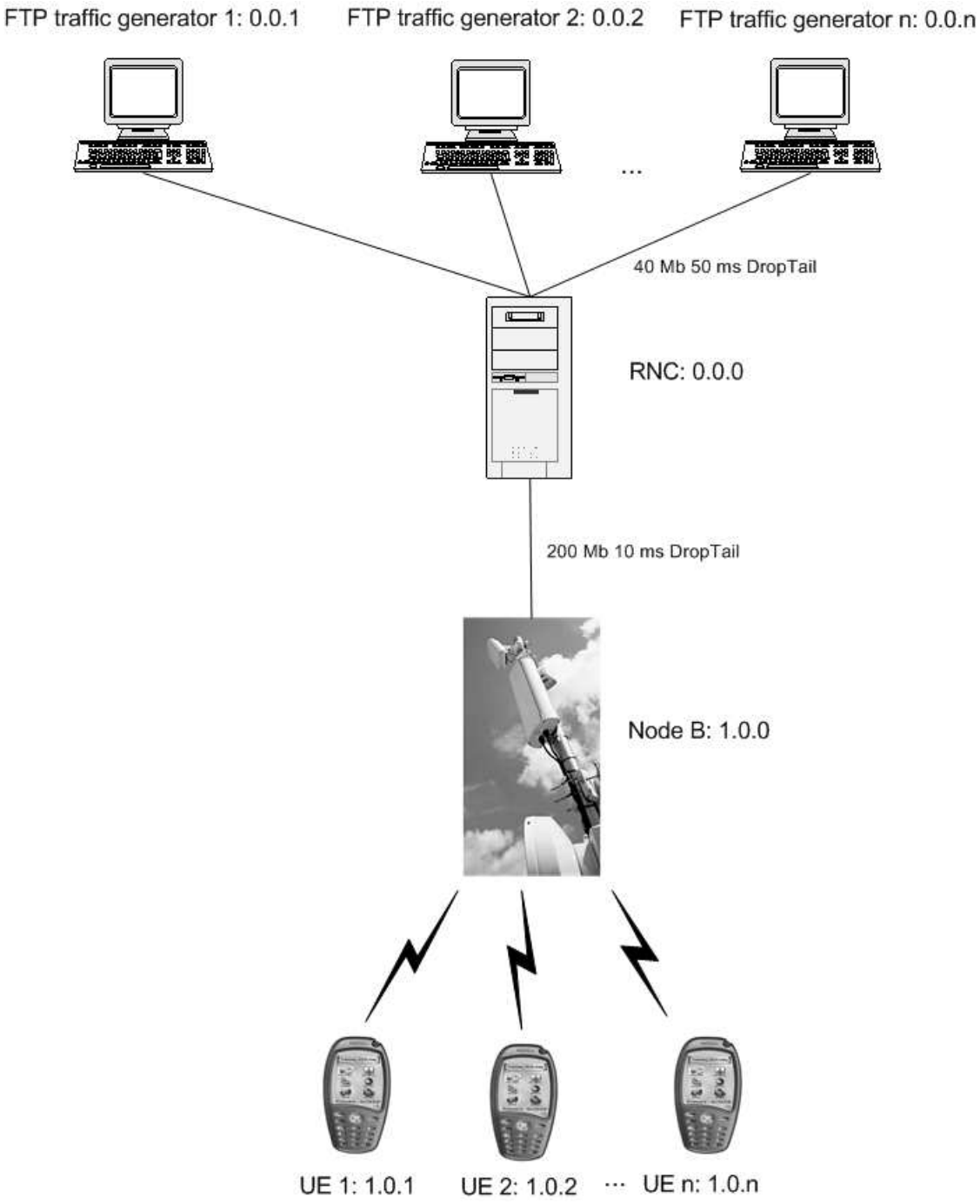

Figure 6-2: Simulation topology

6.3 Performance measures

After each simulation, trace files recording the traffic are generated (see Appendix C.5). These files need to be parsed to extract the information needed to measure the performance metrics like the downlink system goodput, system throughput and packet delay.

In the following I will outline the performance measures used in the simulations.

6.3.1 Downlink system goodput

Goodput, determined as the ratio of useful data received at the destination and the total amount of data transmitted by the source, is an indication for how effective a connection utilizes the network.

Goodput is therefore low when many extra packets where sent over the network due to retransmissions. It is desirable that each connection has a goodput as high as possible.

Most of the figures in chapter 7 use this performance measure for performance analysis.

6.3.2 Downlink user goodput

The average downlink goodput achieved per user is heavily depending on the traffic load.

In Figure 7-14 results from simulations using this performance measure are shown.

6.3.3 Downlink system throughput

The average system throughput, determined as the ratio of the total amount of data received by the end user and the connection time, is an indication for how soon an end user can receive data.

In Figure 7-15 results from simulations using this performance measure are shown.

6.4 Basis of the simulator implementation

In this section I outline some basic information about the implementation of the UMTS module for NS-2 that was provided by a research group of the University of Rome and extended for my needs. With this simulation module both UMTS operation modes FDD and TDD can be simulated for a single cell scenario. However, I have only carried out simulations with the FDD operation mode module. In the following subsections I briefly describe the most important parts of the FDD mode implementation and in the next section I then briefly outline the most important parts that have been subject of amendments [67], [68].

6.4.1 NOAH routing agent [73]

In contrast to the normal routing agent in the NS-2 mobile node that contains a full-fledged ad-hoc routing mechanism, NOAH is a wireless routing agent that only supports direct communication between base stations and mobile nodes. This is more appropriate for the UMTS simulations and it allows simulations of scenarios where multi-hop wireless routing is undesired. Furthermore, NOAH agents do not generate any routing traffic, but learn from the packets they receive which nodes are within reach.

6.4.2 RRC layer

There is no extra RRC layer implemented, however, some of the RRC functions are implemented in other layers (MAC, RLC and PHY).

6.4.3 RLC layer

The RLC layer replaces the LL layer in the wireless protocol architecture of NS-2. The implementation of the UMTS RLC layer is based on the GPRS RLC class module developed at the Indian Institute of Technology [49]. In the UMTS module support for multiple RLC entities in each node were added and extensive modifications to the protocol layer implementation has been performed. The basis simulator implementation provided by the University of Rome allows only simulation with RLC acknowledged mode and RLC unacknowledged mode. RLC transparent mode is not supported for simulations (see section 6.5.2). The RLC acknowledged mode includes a selective repeat Automatic Repeat reQuest (SR-ARQ) scheme to provide link-level retransmissions.

6.4.4 MAC layer

The MAC layer in the UMTS module replaces the Wireless Local Area Network (WLAN) MAC layer in the wireless protocol architecture of NS-2. The implementation of the MAC layer in the UMTS module provides means for packet scheduling with an round-robin time-division scheduler that allocates each frame (10ms) to another user (see also section 9.3). Round Robin is one of the most simple and fair scheduling algorithm. Every user has its own queue. The scheduler assigns the same amount of resources to all users successively in a cyclic manner. Empty queues are skipped, if a user does not use his resources. The Round Robin scheduler needs to know the nominal bit rate of every user and also the users that want to transmit in a certain TTI. The MAC layer implementation also provides power control and RRC signalling procedures like capacity request and channel allocation messages for the shared data channels.

The power control part of the MAC layer implementation provides closed loop power control. In the uplink, the Node B performs estimates of the received Signal-to-Interference Ratio and compares it to a target SIR. On one hand, if the measured SIR is higher than the target SIR, the Node B enforces the User Equipment to reduce the power. On the other hand, if the SIR is lower than the target SIR, the Node B enforces the UE to increase the power. The cycle is executed at a rate of 1500 times per second. The same technique is used on the downlink. The target SIR used in closed loop power control is not fixed; it is dynamically adjusted to maintain a constant quality, defined as a certain target BER or BLER. The target SIR needs to be changed because the required SIR for a certain BLER depends on the mobile speed and the multipath profile.

6.4.5 Physical layer

The most important functions in the implementation of the physical layer are sending packets received from the MAC layer to the wireless channel, calculating received power using a UMTS propagation model and sending packets received from the wireless channel to the MAC layer.

6.5 Features added to the simulator

To carry out my simulations certain parts of the basic simulator implementation needed to be modified and complemented. This required an in-depth, time-consuming analysis of the received basis UMTS simulator implementation and studies of the methods of how to add new modules to the network simulator NS-2.

The most important amendments to the basic UMTS simulator implementation are outlined in the following subsections.

6.5.1 Support for concatenation of RLC data packets

The received UMTS simulator implementation had only means to support segmentation in the RLC layer, but not concatenation. In RLC transparent mode, where the size of a RLC Service Data Unit (SDU) must be a multiple of a RLC Protocol Data Unit (PDU), there is no need for concatenation. However, for simulations with RLC acknowledged mode there needs to be means for concatenation as the size of the RLC SDUs does not have to be a multiple of a RLC PDU and therefore it must be expected that concatenation will take place frequently (see also section 4.8.3).

The RLC acknowledged mode segmentation and concatenation function works as follows: After receiving RLC SDUs from the RLC acknowledged mode service access point, those SDUs are segmented and if possible concatenated to earlier PDUs into RLC payload units (PU) of fixed size. It is doing so by first checking the free space available in the last RLC PDU in the buffer and then, depending on the space available in the RLC PDU, it either inserts the whole RLC SDU, parts of it or if it is already full, it does not insert any data at all. After the segmentation of the remaining data of the original RLC SDU is done, the segmented packets will be stored in the transmission buffer.

6.5.2 Support for RLC transparent mode

In my simulation scenarios I want to compare RLC transparent mode with RLC acknowledged mode to study the tradeoff between additional overhead when using RLC acknowledged mode, and the need for frequent retransmission at TCP level due to the lossy link when using RLC transparent mode.

As the RLC transparent mode was not implemented in the received software package, I have added means to support it to the existing code. The RLC mode used for a simulation run, can be configured in the simulation scenario script. When using the RLC transparent mode, the RLC transmitter segments all received RLC Service Data Units (SDUs) into RLC Protocol Data Units (PDUs) and stores them into the transmission buffer. In contrast to the RLC acknowledged mode, no additional RLC header is added to the packets.

6.5.3 Support for a weighted Round Robin scheduler

I have partly implemented a weighted Round Robin scheduler for the packet scheduling in UMTS. The new scheduler considers the RLC buffer occupancy and the current power situation to compute weights that can be used from the scheduler to decide which users to serve next and how many RLC PDUs to assign. However, before this scheduler can be used for simulations it firstly needs to be completed and tested.

Using other packet scheduler than the Round Robin scheduler currently implemented in the UMTS radio network simulator, will be an important part of future work (see section 9.3.1).

6.5.4 Added mobility model

In the received simulation scripts to run simulations on the UMTS simulator implementation no mobility model was used. At the beginning of each simulation run, a fix number of active User Equipments were introduced to the system with randomly allocated coordinates inside a certain radius. The original model further assumed that the users will not move around with their User Equipments. This approach has two major drawback. Firstly, because of the random allocation of coordinates to the User Equipment each simulation run with the same parameter settings produces significant different performance results. User Equipments that are farther away from the Node B can receive data only with a significantly lower data rate than User Equipments that are close to the Node B. In cases that many mobile handsets are close to the Node B, the system goodput is always high. On the other side, in cases where all User Equipments are far away from the Node B, the system goodput is poor. In order to get useful results it is therefore necessary to conduct many simulations for the same parameter settings, which is enormously time-consuming. Secondly, because it is very important to study the effects on the performance when users are moving around with their mobile terminals as moving cause disturbance to the received radio signal.

When using the new added mobility model, like in the old mobility model, a fixed amount of User Equipments is allocated with random coordinates at the beginning of each simulation run. However, in contrast to the old model, the User Equipment will now move around. Every five seconds the User Equipments will move towards another direction with a speed of around 5 km/h. For each load situation such a mobility pattern has been created (see Appendix section C.9). With this less simulations needs to be performed in order to get useful results. However, the mobility pattern should be chosen carefully because it is then being used in all simulation runs.

6.6 Simulation steps

When running simulations for UMTS performance optimization I had the choice of using many different parameter settings. As it does not make sense to test all possible combination of the parameters I define three optimization steps to minimize the amount of simulation runs needed to obtain ranges of optimal parameter settings that can be used for profound simulations of certain parameter ranges. I have outlined those steps in the following subsections.

6.6.1 Optimization step one

In the first optimization step I try to make use of scientific results about optimal system, FTP, TCP, RLC, MAC and PHY parameter settings published in papers and Internet Drafts [41]. I have listed those parameter settings in Table 6-1. Note that these parameters are fix during all simulation runs.

6.6.2 Optimization step two

In a second optimization step one tried to fine-tune parameter settings for a maximum system goodput and system throughput by varying the parameter settings for the RLC Mode, the number of link-level retransmissions, the Timer_Poll_Prohibit, the RLC buffer and the transmission window size. This parameter fine-tuning are carried out for three different traffic load scenarios. Details about the different settings are listed in Table 6-2 and information about the UMTS parameters appear in the Appendix A.5. Results of these simulations are presented in chapter 7.

6.6.3 Optimization step three

In the third optimization step I used the optimal values of the parameters obtained from the simulations of optimization step two in order to thoroughly investigate certain parameter settings. The optimal parameter value ranges obtained from optimization step are 0.4 - 0.55 s for the Timer_Poll_Prohibit, 4 - 8 for the maximum number of link-level retransmissions (MaxDAT), disabling for delayed ACKs and 256 - 512 RLC PDUs for the transmission window size and the RLC buffer size. Results of these simulations are presented in chapter 7.

6.7 Set of fixed simulation parameters

There are two types of simulation parameter sets used for simulation in this thesis. In this section I outline the fixed simulation parameter set, used for all subsequent simulations. In the next section I then outline the variable simulation parameter set, containing parameters that may vary from simulation to simulation or within one specific simulation. All descriptions of important UMTS parameters appear in Appendix A. I assume that all User Equipments stay in one cell during the whole duration of a simulation run and therefore do not consider handoffs.

6.8 Set of variable simulation parameters

The configuration of the variable simulation parameters is of great importance as they have direct impact on the TCP performance. However, the parameter fine tuning and optimization is critical to ensure the efficient use of the precious radio resources and to produce high throughput and goodput. In the following, I outline the most important configurations of the variable simulation parameters.

Although a wide variety of TCP implementations are used on the Internet, the current de facto standard is TCP NewReno (see section 5.5.3). I use it as basis in the performance comparison. The two other TCP variants used for simulations are TCP SACK (see section 5.4.1) and TCP Westwood (see section 5.5.6).

Beside the TCP variants, another important variable simulation parameter is the Timer_Poll_Prohibit. Under the circumstances where several poll triggering options are present simultaneously in a system, a potential risk is that the network could be overwhelmed by excessive polling and status reports sent over the air interface. The Timer_Poll_Prohibit is a mechanism that deals with this problem of excessive polling and status report transmission. At the transmitter, the Timer_Poll_Prohibit is started once a PDU with the poll bit set is sent. No polling is allowed until this timer expires. If multiple polls were triggered during the period when this timer was in effect, only one poll is transmitted upon expiry of the timer. More details about polling mechanisms in UMTS are outlined in Appendix A. Information about delayed ACKs appear in section 5.3.4 and information about MaxDAT in A.5.

Layer	Parameter	Value
System	User mobility and speed	Pedestrian (speed of moving user is around 5 km/h)
	Cell radius	350 m
	Environment	Urban
	Spatial user distribution	Hot Spot
	Traffic symmetry	Asymmetric (most traffic is in downlink)
	Transmission activity	Traffic generator send data during the whole duration of a simulation run
	Simulation duration	500 s
FTP	Packet size	1000 bytes
TCP	Initial TCP congestion window size	min(4*MSS, max(2*MSS, 4380 bytes)) [41] => 4000 bytes
	MSS (Maximum Segment Size)	1000 bytes
	ssthresh	20
	Initial slow start threshold (ssthresh)	20
RLC	RLC header size	2 bytes
	Number of RLC PDUs per TTI	12
	Size of a RLC PDU	40 bytes
	Poll_SDU	2 [30]
MAC	TTI	10 ms
	Scheduler	Round Robin
PHY	Intercell Interference	-65 dB
	Intercell Interference variation	0.3333
	Thermal noise	-103 dB
	SIR target	6 dB
	Max. transmission power of a UE	21 dBm
	Max. transmission power of a Node B for one UE	21 dBm

Table 6-1: Fixed simulation parameters

Layer	Parameter	Value
FTP	Number of FTP flows	5, 20, 30
TCP	Variant	TCP NewReno, TCP Sack, TCP Westwood
	delayed ACKs	On, Off
RLC	RLC mode	TM, AM
	RLC Buffer size	64, 128, 256, 512, 768, 1024 PDU
	MaxDAT (Maximal number of link-level retransmissions)	1, 2, 3, 4, 6, 8, 10, 20, 40, 100
	Transmission window size	64,128, 256, 512, 768, 1024 PDU
	Timer_Poll_Prohibit	50, 100, 200, 250, 300, 400, 500, 550, 600, 700, 800 ms

Table 6-2: Variable simulation parameters

6.9 Summary and outlook

In this chapter I introduced a general simulator model for radio interface simulations in UMTS and outlined my simulation scenarios and simulation parameters. In the following chapter I will analyze the obtained simulation results.

7 Simulation results and analysis

In this chapter I discuss the performance impact of different values of the RLC transmission window size, the RLC buffer size, the Timer_Poll_Prohibit and the number of link-level retransmission with changing conditions such as traffic load, TCP variants and TCP options.

Note that only the interesting ranges of the graphs are plotted in the figures in this chapter and care must be applied when interpret the difference between the lines in the graphs as the variations between the worst and best values differ usually only about 10%.

The duration of a simulation run was 500 seconds and the first 10 seconds of the simulation runs were not included into the performance evaluation.

7.1 Comparison of the RLC AM and RLC TM

Then carrying out simulations with RLC transparent mode, there are not many RLC parameters that can be modified. The two most interesting RLC parameter settings are the RLC transmission window size and the RLC buffer size. Simulation results, displayed in Figure 7-1, shows that RLC transparent mode is not well suited for FTP data traffic using TCP as the transport protocol. The average system goodput achieved with RLC transparent mode is around twenty times smaller than the system goodput achieved by using RLC acknowledged mode. Therefore, there will be a focus on the performance analysis of the RLC AM in the following chapters.

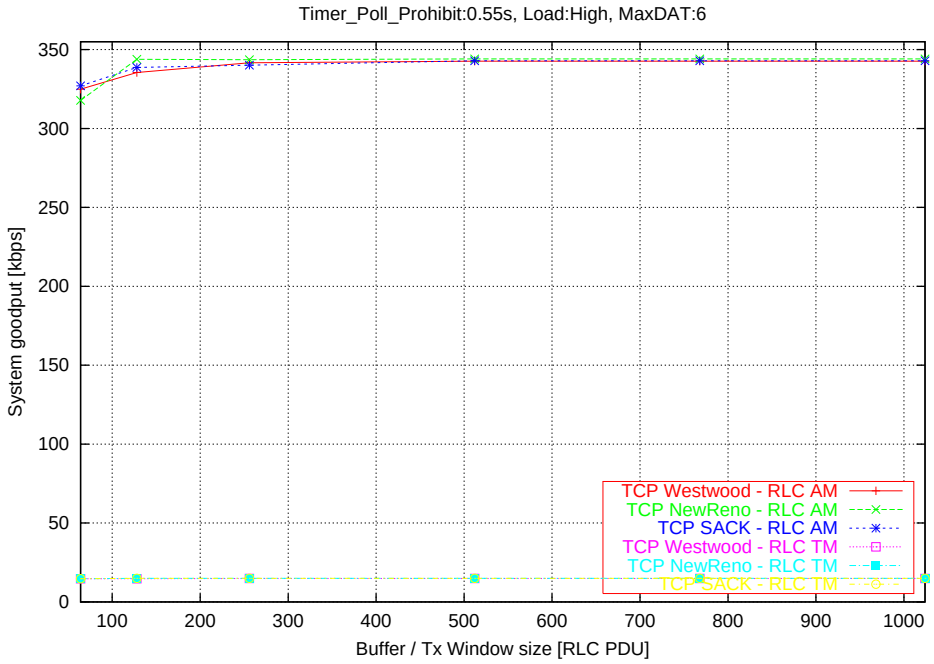

Figure 7-1: Comparison of RLC AM and RLC TM using variable RLC buffer and transmission window size

7.2 Optimization of the parameter MaxDAT

Related papers often suggest a big value or even the infinite value for the number of allowed link-level retransmission in UMTS [63]. Looking at Figure 7-2, Figure 7-3 and Figure 7-4, these suggested values seems to be rather high as the system goodput does not perform significantly better when MaxDAT is set to a higher value than 8, but the delay caused by the link-level retransmission further increases. I therefore suggest to allow only 4 - 8 link-level retransmissions.

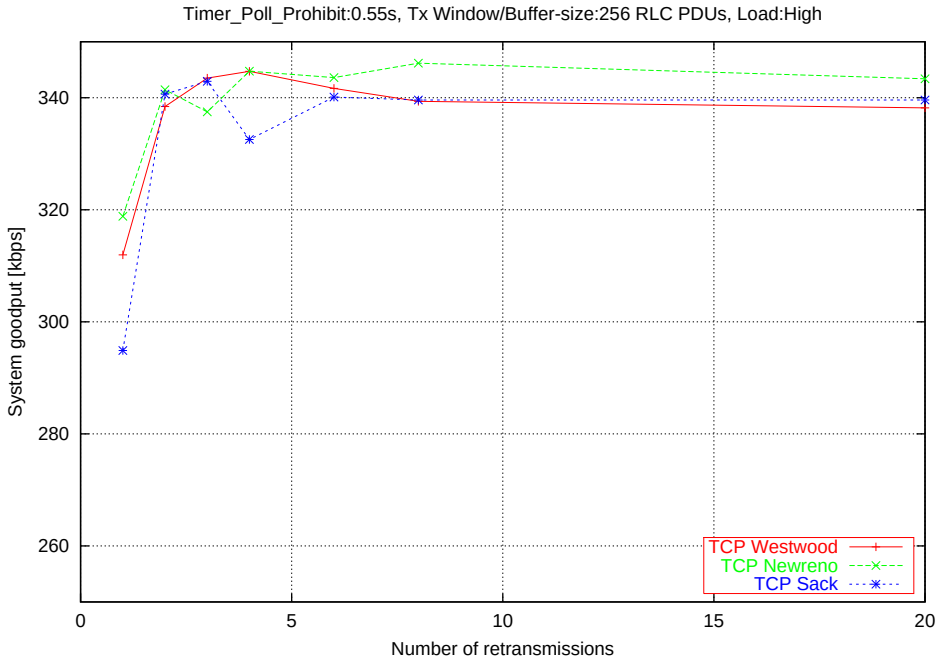

Figure 7-2: System goodput with variable number of link-level retransmission for high load

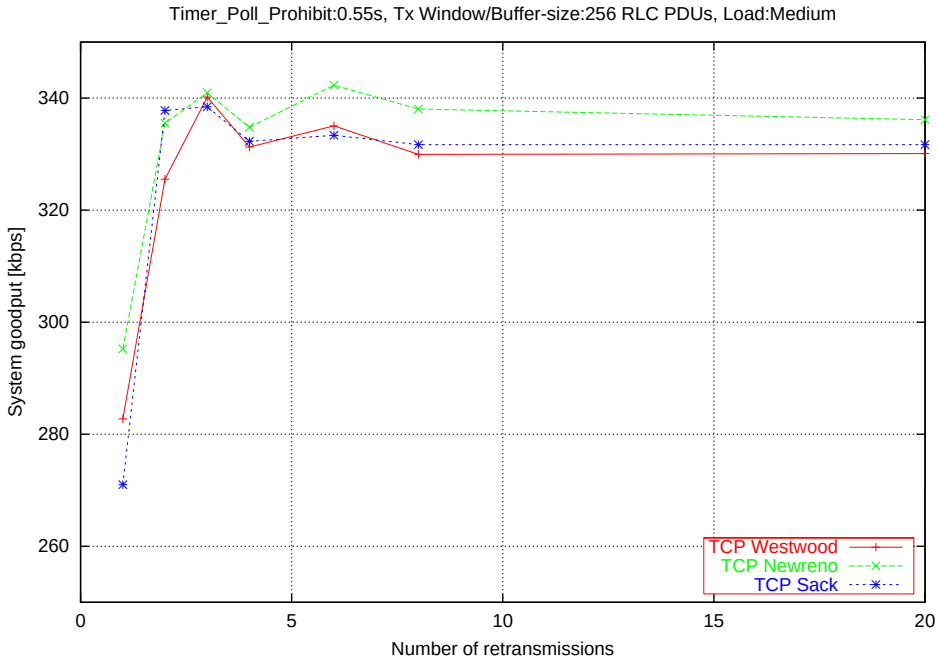

Figure 7-3: System goodput with variable number of link-level retransmission for medium load

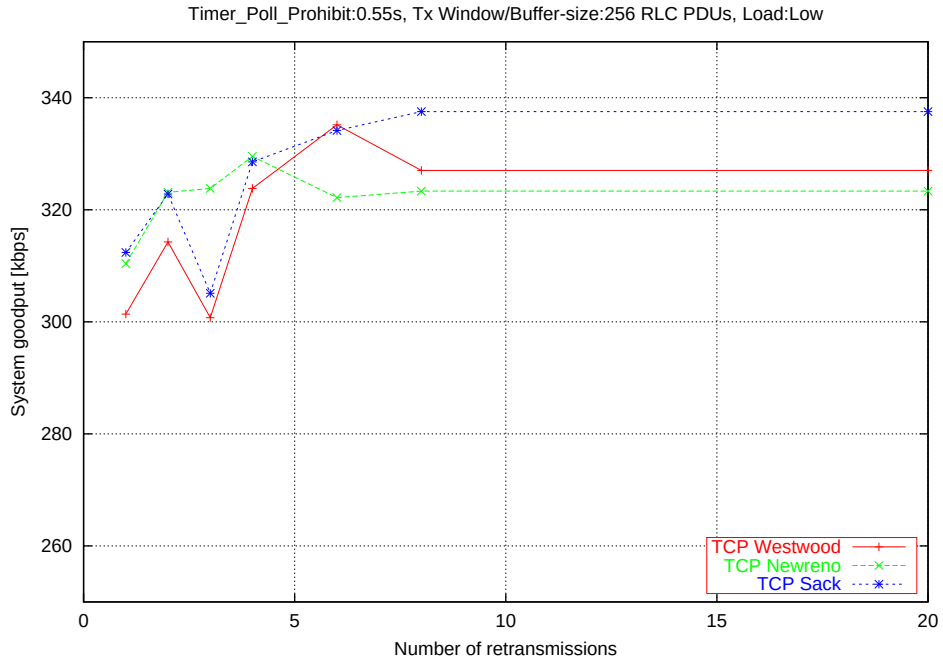

Figure 7-4: System goodput with variable number of link-level retransmission for low load

7.3 Comparison between enabling and disabling of delayed ACKs

Comparing Figure 7-5, where delayed ACKs were enabled with Figure 7-4, where this option was turned off, no big difference in terms of system goodput can be seen. Similar results have been obtained carrying out simulations with other settings. As using delayed ACKs is not resulting in a significant performance improvement it is not recommended to use them.

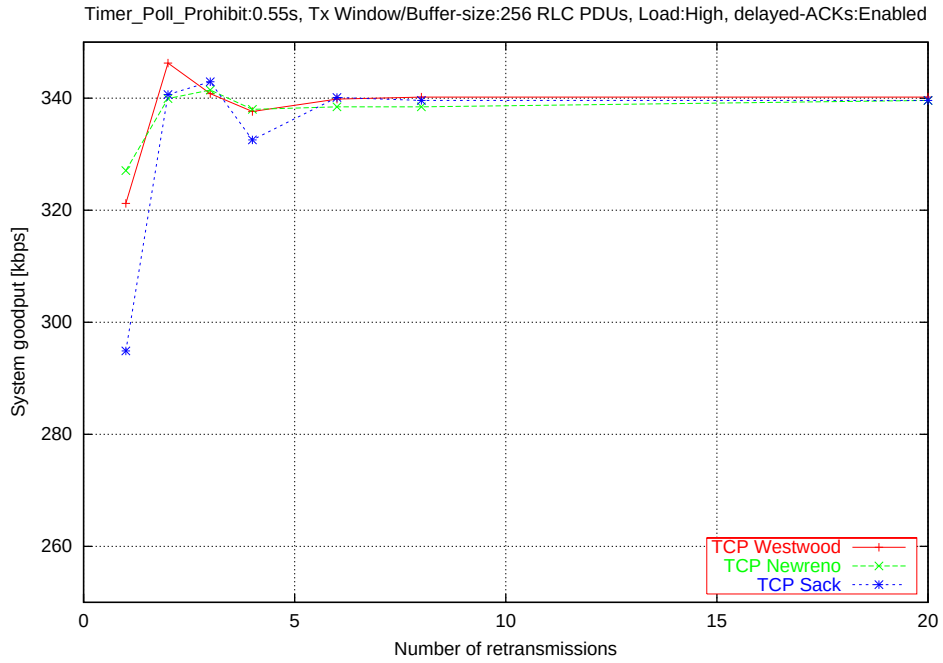

Figure 7-5: System goodput with variable number of link-level retransmission for high load using delayed ACKs

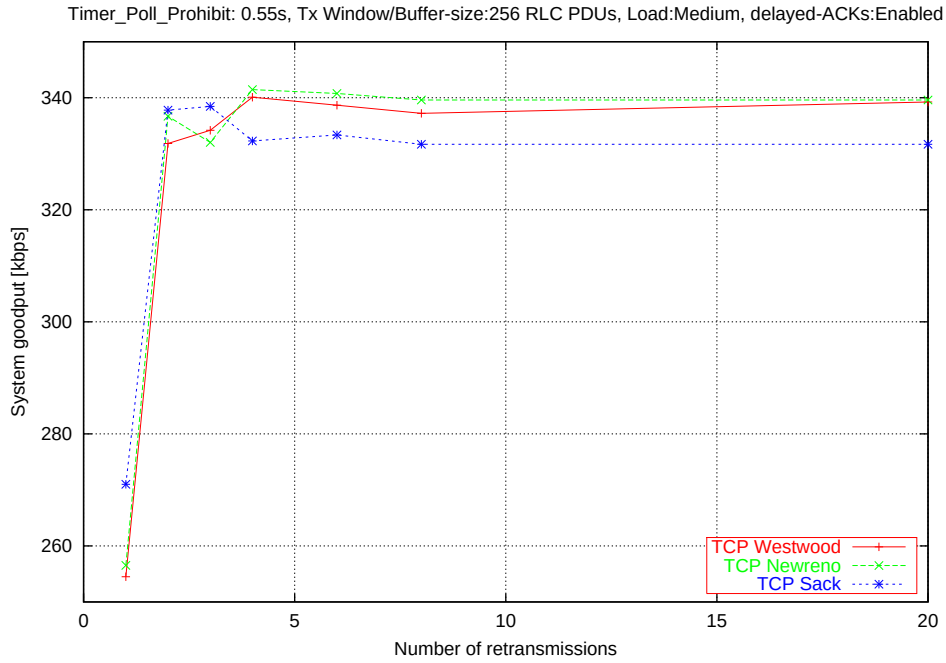

Figure 7-6: System goodput with variable number of link-level retransmission for medium load using delayed ACKs

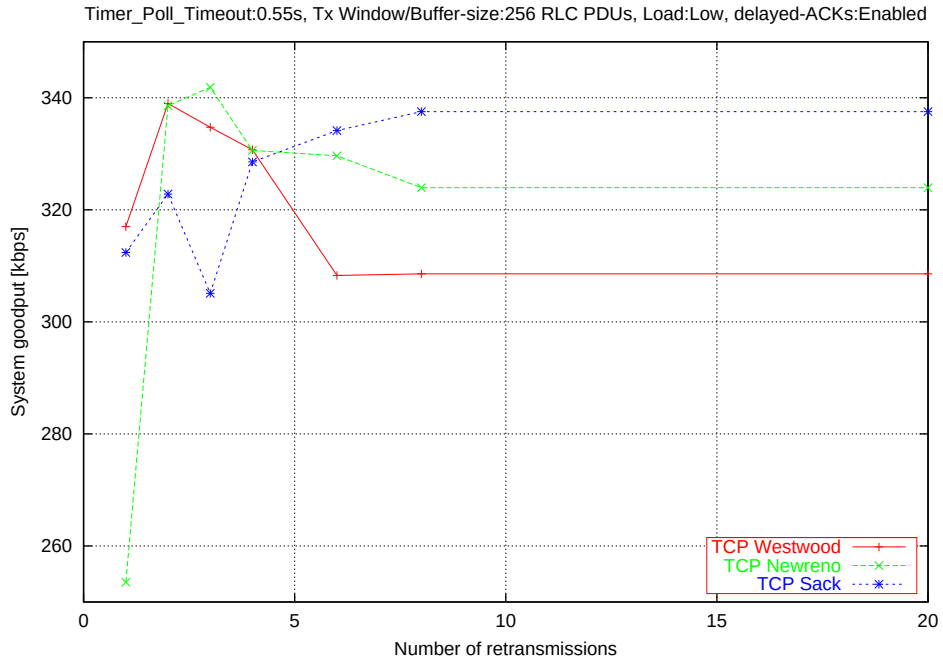

Figure 7-7: System goodput with variable number of link-level retransmission for low load using delayed ACKs

7.4 Optimization of the parameter Timer_Poll_Prohibit

Generally, small values for the Timer_Poll_Prohibit produce low delays but medium up to high throughputs and goodputs, while large values for it still produce high throughputs and goodputs but

long delays. Looking at the system goodput when varying the Timer_Poll_Prohibit value, shown in Figure 7-8, Figure 7-9 and Figure 7-10, an optimal parameter setting interval of 300 to 800 ms for the poll prohibit timer can be defined. Considering the fact that longer values for that timer results in a higher delay I come to the conclusion that 550 s is an optimal value for the Timer_Poll_Prohibit parameter. I therefore use this value of the parameter in the other simulation runs. This result is similar to the results published in [61]. This paper shows also that the throughput will significantly decrease for poll prohibit timer settings bigger than 2000ms. In the paper it is also shown that RLC SDU delays caused by poll prohibit timer settings bigger than 600ms increase rapidly. This assures the proposed setting of 550 ms for the poll prohibit timer.

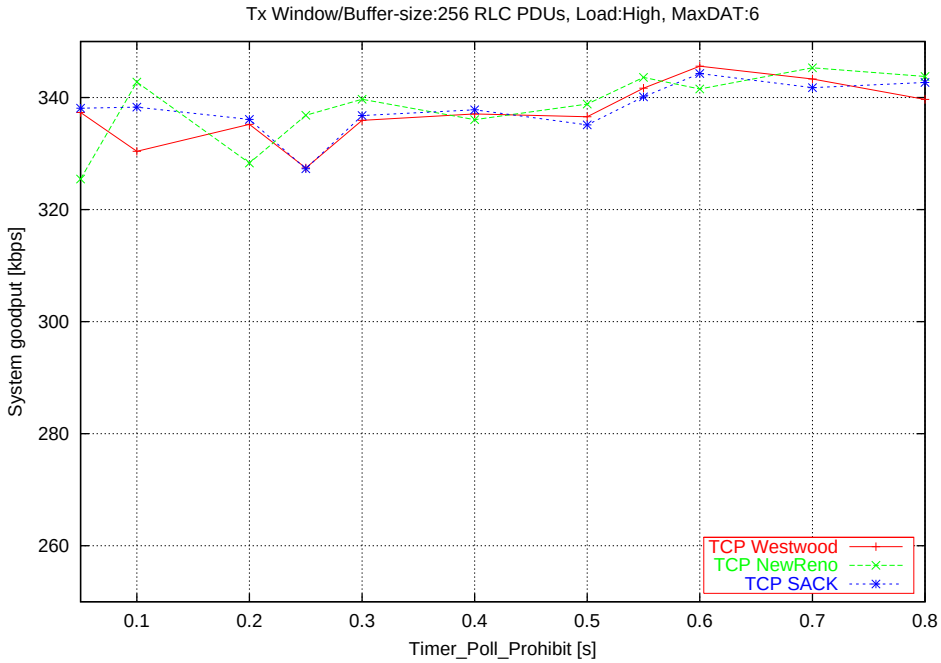

Figure 7-8: System goodput with variable Timer_Poll_Prohibit for high load

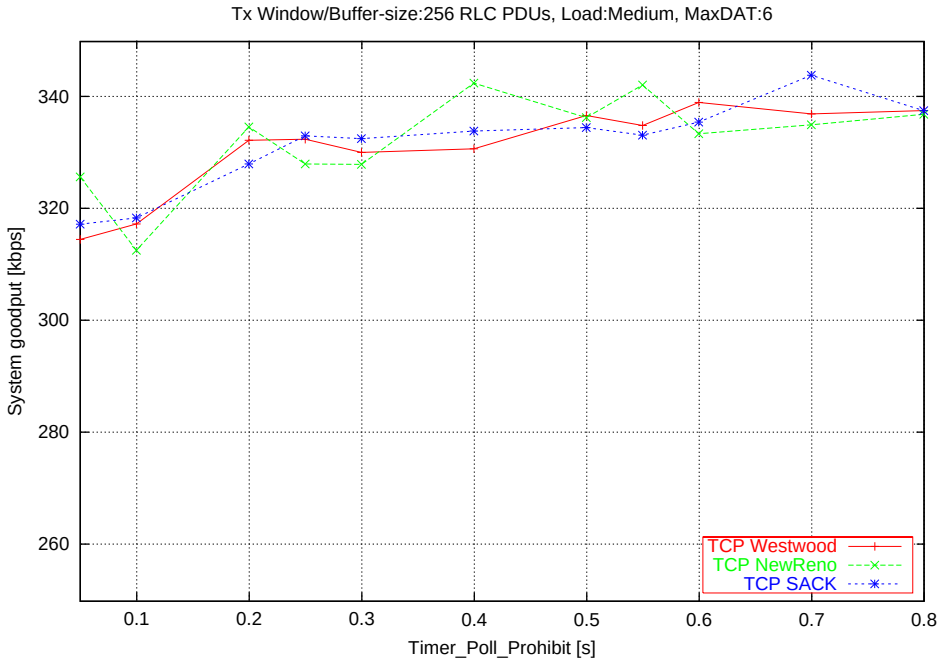

Figure 7-9: System goodput with variable Timer_Poll_Prohibit for medium load

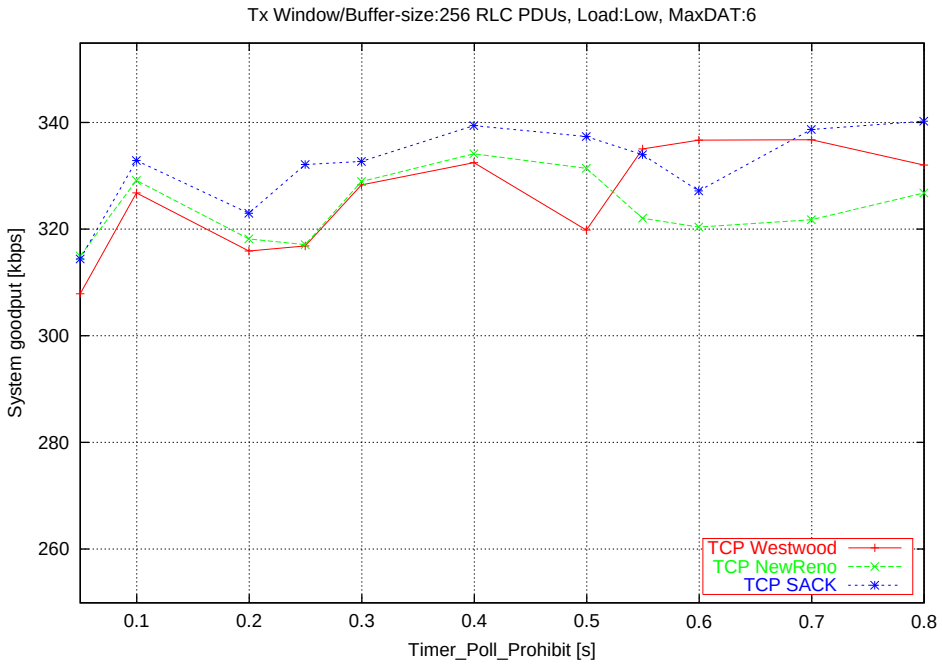

Figure 7-10: System goodput with variable Timer_Poll_Prohibit for low load

7.5 Optimization of the parameters buffer size and RLC transmission window size

In general a large buffer and transmission size can tolerate long RLC RTT and a highly erroneous radio link; and help to simplify the problem as window stall situation don't need to be considered.

However, a large buffer size requires large storage capability at the sender and receiver to store all the RLC PDUs within the current window. This can be expensive and may not be realistic in UMTS system since the User Equipment has limited storage capability due to the drive for low cost and small size. Optimal RLC transmission window size and RLC buffer size, that provides the maximum throughput/goodput, can be found by observing the throughput/goodput changes versus the RLC transmission window size and buffer size shown in Figure 7-11, Figure 7-12 and Figure 7-13. For a small window size, such as 64 RLC PDUs, frequent window stall results in reduced system goodput because the transmitter is constantly waiting for the acknowledgements to free up the window. For window sizes larger than or equal to 256 RLC PDUs, the system goodput maintains nearly at the same level. Considering also the fact that big buffers result in an unwanted delay and storage requirements, the value 256 RLC PDUs seems to be the optimal value for maximum system goodput.

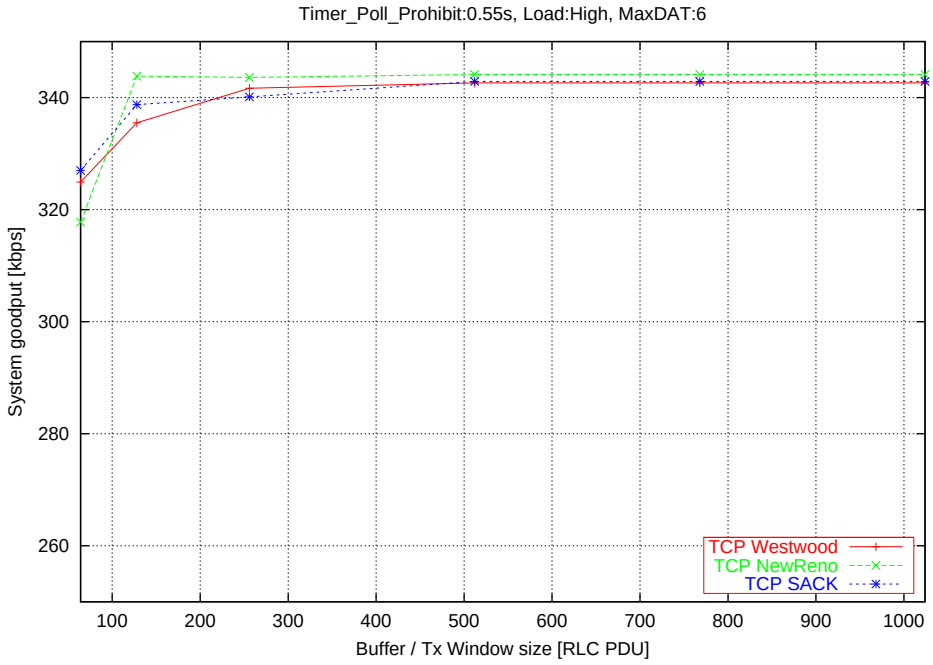

Figure 7-11: System goodput with variable Buffer and transmission window size for high load

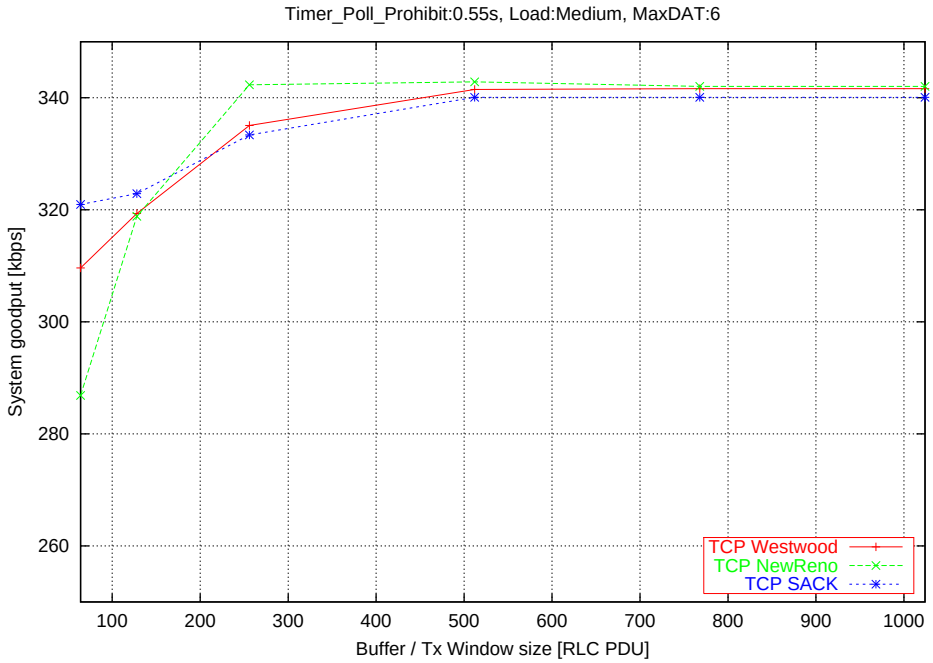

Figure 7-12: System goodput with variable Buffer and transmission window size for medium load

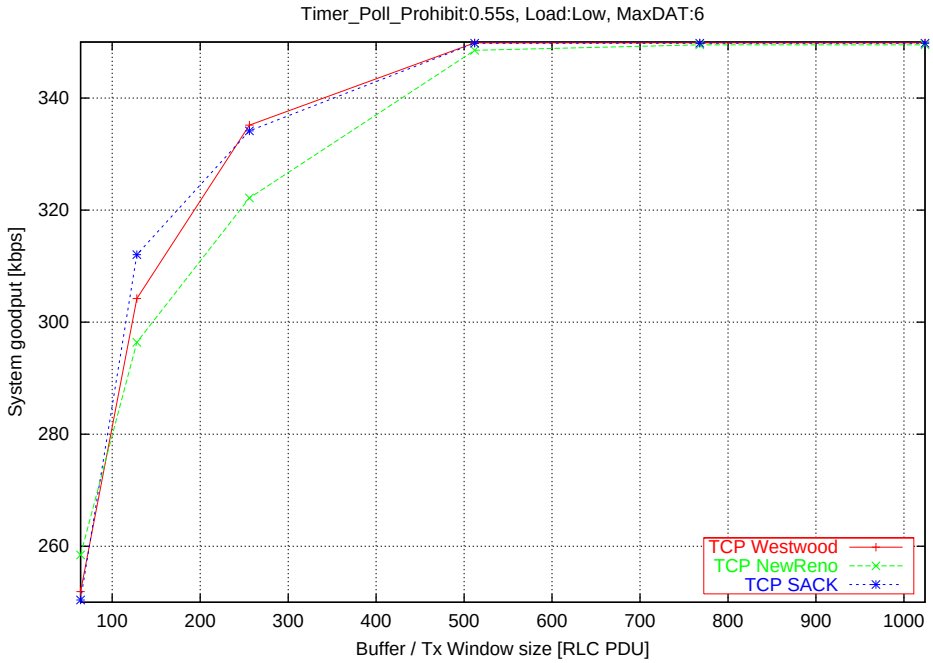

Figure 7-13: System goodput with variable Buffer and transmission window size for low load

7.6 Comparing of average user goodput for the different traffic loads

The only performance metric so far used for the analysis of the simulation results was system goodput. Another interesting performance measure, the average goodput received by each user in different traffic load scenarios is shown in Figure 7-14. The simulation results show that expected result, that the average user goodput is especially high for high traffic load scenarios.

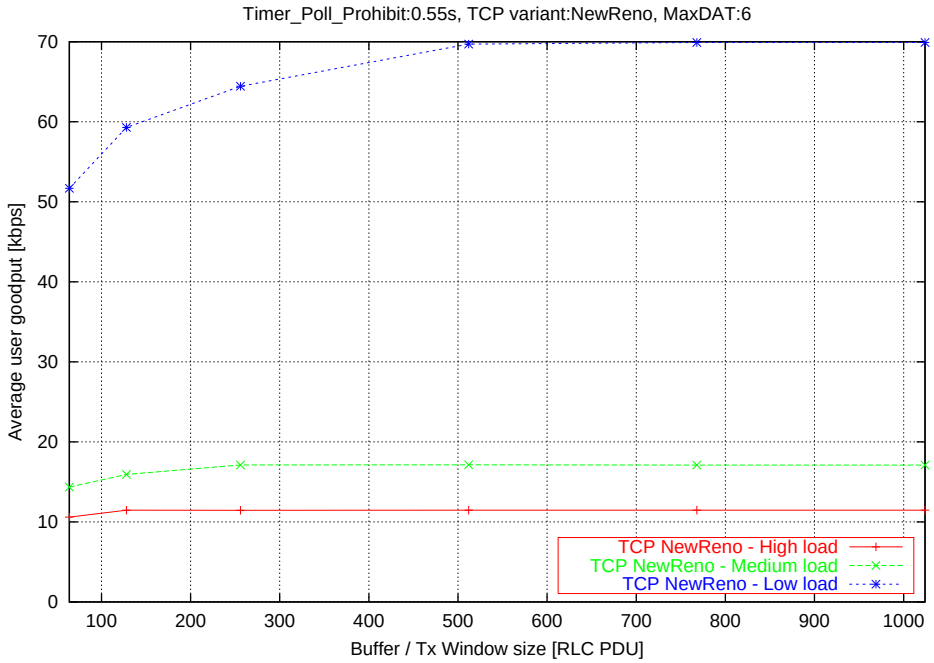

Figure 7-14: Average user goodput with variable Buffer and transmission window size for TCP NewReno

7.7 Comparison of system goodput and system throughput

Goodput is probably the most important performance measure for TCP performance analysis over UMTS. However, it is also interesting to compare the system goodput with the system throughput. I have therefore plotted both performance measures in Figure 7-15 for the same simulation scenario like in Figure 7-2. The difference between the system goodput and the system throughput is around 3%.

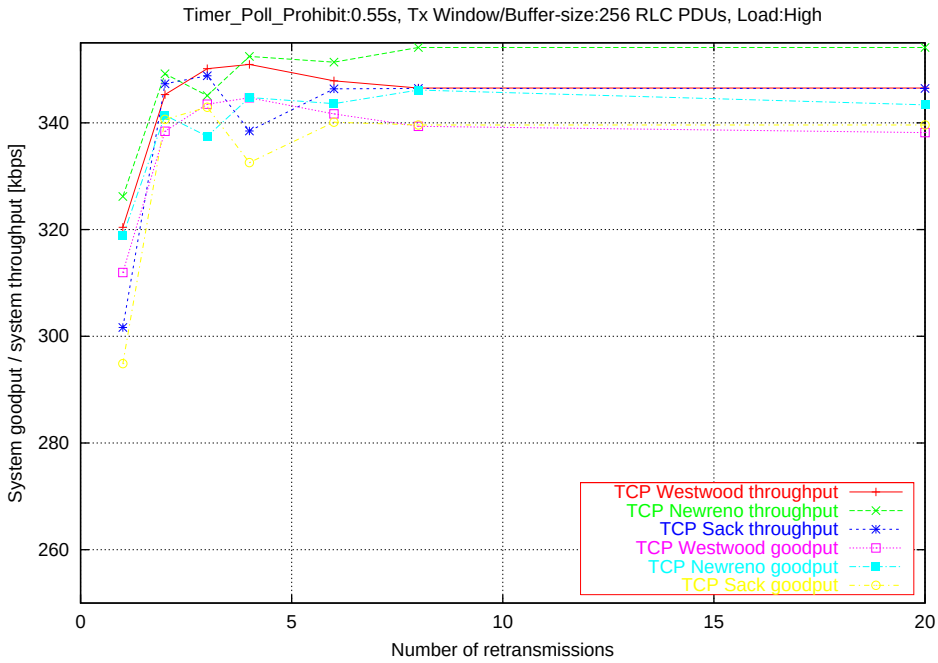

Figure 7-15: System goodput and throughput with variable number of link-level retransmission for high load

7.8 Comparison of TCP NewReno, TCP SACK and TCP Westwood

Looking at the graphs in this chapter it seems that none of the used TCP variant performs always the best. Furthermore, it can be stated, that the system goodputs do not differ much between the different TCP variants. The difference is usually below 10% of the absolute value.

It seems that by using RLC acknowledged mode not many TCP retransmissions are necessary. I therefore suggest to use TCP NewReno as this is the best tested and most wide spread TCP variants of the three evaluated variants.

8 Conclusion

An exciting future in mobile communication is already unfolding and UMTS is going to play an important role in the future mobile communication market. For UMTS network operators, an optimization of the Transmission Control Protocol (TCP) over UMTS for interactive and background traffic is of great importance to reach a better utilization of their scare radio resources, to minimize the number of UMTS radio access points, and finally to offer better service quality to their customers.

The goal of this thesis is to develop a UMTS simulator extension to the existing popular network simulator NS-2 as well as to investigate the influence of using different TCP variants, different link layer solutions and parameter settings on the performance of TCP over UMTS using this simulator.

To this end, extensive literature survey, incorporating related work, UMTS network and TCP over wireless has been carried out. This led to a fundamental design draft for an implementation of a UMTS simulator module for NS-2, before an external implementation of such a UMTS simulator module was provided. This advanced UMTS simulator module was first of all analyzed in-depth and as it was shown to be adequate for most of the planned simulation scenarios, chosen as the basis implementation of the final UMTS simulator used for all simulation runs. To support all planned simulation scenarios, I have partly modified and extended it. Means for concatenation, support for RLC transparent mode were added and a completely new mobility model was introduced to the simulation scenarios.

The analysis of the simulation results carried-out with this UMTS simulator has shown that RLC acknowledged mode generally performs better than RLC transparent mode for FTP traffic. Furthermore, the optimal settings of the RLC transmission window size parameter and the RLC transmission buffer size were shown to be around 256 - 512 RLC PDUs per connection. Furthermore, the maximum number of allowed link-level retransmission was shown to rate between 4 and 6. There is no need to further increase the value of this parameter as the system goodput does not increase significantly with a higher value, but the delay caused by the link-level retransmission further increases. Another result of the performance analysis shows that a value of around 550 ms should be used for the poll prohibit timer in order to prevent from excessive polling. With those optimal parameters an acceptable system goodput and system throughput can be achieved for all simulated scenarios.

It was further shown that, when using RLC acknowledged mode with optimal parameter settings, the system performance does not heavily depend on the used TCP variant.

It was also shown that the average goodput received by each user strongly depends on the traffic load.

Based on the results of this thesis, some suggestions for future work in the research field of TCP performance optimization over UMTS can be provided: open questions and remaining tasks in order to achieve better performance and obtain more realistic results, not covered in the thesis, include the development of an alternative packet scheduler, the use of web traffic in order to simulate interactive services and the use of RLC unacknowledged mode to carry out simulation. More suggestions for future work and some approaches are outlined in chapter 9.

9 Suggested future work

I have not touched upon all aspects of TCP performance optimization over UMTS. This chapter outlines some approaches for further work in this research area based on the work performed in my thesis.

9.1 Support for variable sized Transmission Time Interval (TTI)

The Transmission Time Interval (TTI) defines how often data should be sent from the RLC layer. The amount of data that can be transmitted during one TTI is determined by the bandwidth and the length of the TTI. In my simulations I chose a fixed TTI of 10ms. For future simulations it would be interesting to investigate TCP performance by using a longer TTI (20ms). With this the system needs to conduct less link-layer retransmissions on the other hand this would also result in a bigger delay.

9.2 Modifications on the implementation of the physical layer

I see a big importance of further investigating or even replacing the physical layer model derived from link layer simulations at University of Ferrara (Italy) used in my simulations (see section 6.2.2) by a more adequate solution as these simulations were compute for TDD operation mode and not for FDD operation mode.

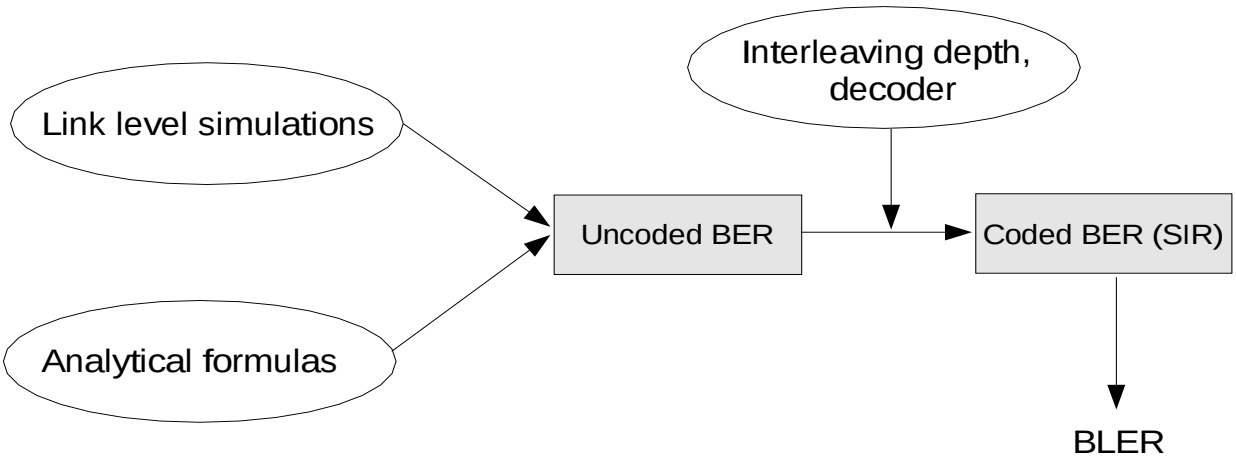

Figure 9-1: Different methods of how to compute the coded BER

Another way of how to obtain the uncoded BER of a UMTS system would be to use the hypothetical factor for analysis described in [54]. However, this approach seems to be complex and elaborate.

Figure 9-1 provides an overview the different approaches.

9.3 Support for other packet scheduler

In my simulations I used a round robin scheduler. To further investigate TCP performance over UMTS other packet scheduler should be also considered. Instead of using the simple Round Robin packet scheduler (see section 6.4.4) that equally serve users (i.e. all users belong to the same traffic class), a weighted Round Robin scheduler would be more adequate as such a scheduler could also serve users that have different requirements like data rate or delay requirements.

In the following, I outline the approaches for a Weighted Round Robin scheduler and for the CHAOS scheduler proposed in [60].

9.3.1 Weighted Round Robin scheduler

Additionally to the standard functions used in Round Robin, the weighted Round Robin scheduler should also consider RLC buffer occupancy, the current power situation and potentially also the mean number of past credits allocated to the users. The weighted Round Robin scheduler needs then to compute the weights from these three parameters and decide which users to serve and how many RLC PDUs to assign.

Werner Perndl proposes such an implementation of a weighted Round Robin scheduler in the environment of UMTS in his diploma thesis [17].

9.3.2 CHAOS (CHannel Adaptive Open Scheduling) scheduler

CHAOS is designed for wireless networks, stands for Channel Adaptive Open Scheduling and has been proposed in [60].

Differently from most other schedulers, CHAOS stresses the importance on an efficient utilization of bandwidth. A user will be given a priority level based on its experienced channel quality; the bandwidth will be preferentially utilized by users with a high-quality channel, thus reducing the global error rate and increasing the efficiency. User who experience a lower quality channel are given lower priority in the allocation, but will be compensated when they are in a more favorable condition.

The scheduling algorithm classifies the User Equipments according to two criterias: a quality-based criterion (e.g. the achievable Signal-to-Interference Ratio and the frame error rate) and a traffic-based criterion (e.g. the amount of traffic and the experienced delay).

The most important input parameter of the CHAOS scheduler are: (i) the capacity request Q; (ii) the previously assigned capacity C and (iii) the radio channel status alpha.

As for (i) the various User Equipments shall signal the capacity requests.

As for (ii) the base station must keep track of the last three values of the assigned capacity for every active User Equipment.

As for (iii) the base station determines the current status of the radio channels by means of physical layer measurements.

The CHAOS performance in terms of throughput-delay trade-off was compared with those resulting from a load adaprive but channel independent scheduler and shows that CHAOS can improve the overall network performance significantly.

9.4 Involving other performance measures

Beside system goodput and system throughput there are a few other performance measures that should be considered like TCP packet delay, or the number of satisfied users.

In the following subsections I outline these two new performance measures.

9.4.1 TCP packet delay

In my simulations I only partly considered the effects of performance optimization on the TCP packet delay as they are not of big importance for FTP background traffic. But especially for interactive services the tradeoff between throughput/goodput and delay should not be ignored. When allowing a high value for the TCP packet delay, a better performance of the system goodput can be expected. I therefore suggest that in future investigations the effects of the parameter settings on the TCP packet delay should also be considered, especially when simulating traffic of interactive services.

9.4.2 ETSI model of user satisfaction in UMTS FDD mode

ETSI proposes a general model of user satisfaction [52]. The user satisfaction criteria for UMTS FDD operation mode includes the following points:

- The user does not get blocked when arriving at the system. If blocking is applied, the proponent must specify used blocking criteria.
- The user doesn't get dropped during transmission. If dropping is applied, the proponent must specify used dropping criteria.
- The active session throughput is equal or greater than the threshold value S=10% of the nominal data rate.

9.5 Support for variable sized RLC PDUs

Currently the size of a RLC PDU is fixed to 40 byte (see Table 6-1). For future investigation it would be interesting to evaluate TCP performance with having the length of a RLC PDU available as a parameter in the simulation script.

9.6 Use more realistic UMTS traffic models

To simulate traffic in my simulations, I used a FTP traffic model. As TCP performance optimization is especially not only important for background traffic where the FTP model is quite adequate, further simulations should be performed with other traffic generators. It would be especially interesting to use a web traffic generator to simulate traffic of interactive services. Furthermore, TCP packet size variation could be interesting as the data volume of mobile services are usually very small.

Another important point considering future traffic modeling is to develop a random user arrival process model. Currently all users for each traffic load class exists for the whole duration of the simulation and receive consequently data from their sending entity.

9.7 More realistic mobility model

To set up a realistic mobility model I have used the application sedtest, available in newer NS-2 releases (see section 6.2.4).

By introducing a random user arrival process model as suggested in the last section, the mobility model needs also to be modified.

Furthermore it would be interesting to carry out simulations with different mobile speeds. Currently the simulator support only simulations with users that move slowly (pedestrian, speed around 4 km/h).

9.8 Performing simulations with RLC UM

In this work I have only performed simulations using RLC AM and RLC TM. As the current simulator also supports RLC UM it would be interesting to compare TCP performance of RLC UM simulations with the performance obtained with simulations using RLC TM.

9.9 Introduce a multi cell scenario with handovers

Handovers between different Node Bs are also an interesting subject for investigation. It is especially important to examine the time it takes for performing a handover. This time might cause TCP to timeout or in some other way decrease the performance.

9.10 Consider the scarce amount of energy available to a UE

A critical factor to be considered when using wireless devices like UMTS User Equipments is the scarce amount of energy available, which leads to issues such as battery life and battery recharge, and which affects the capabilities of the terminal as well as its size, weight and cost. Since dramatic improvements on the front of battery technology do not seem likely, it should be tried to avoid wasting power and to tune protocols and their parameters also in such a way as to optimize the energy use. This way of thinking may lead to completely different design objectives [65], [45].

9.11 Active Queue Management (AQM) for RLC buffer

When using Active Queue Management (AQM) in the Internet, routers can detect congestion before the queue overflows, i.e. congestion feedback is no longer limited to packet drops as an indication of congestion. Random Early Discard was a leading example, demonstrating better performance, and currently reaching implementation in commercial routers. Such a special management of the RLC buffer could lead to better performance and should be further investigated.

9.12 Performing simulations with TDD mode

Performing simulations with TDD operation mode could be another very interesting subject for future work in this area. Especially when pico cell development is more advanced, this would be a very important part of future performance evaluation. Moreover, the current TDD mode implementation of the UMTS module for the NS-2 network simulator is already much more advanced than the FDD mode.

A Technical details in UMTS

In this Appendix chapter I outline some important technical details of UMTS that are relevant for the implementation of a UMTS module for a network simulator. Please notice that this should give only a brief view into the important parts of the UMTS standard and that I have not implemented all of the presented functions or protocol states in the simulator used for my simulations.

A.1 MAC Protocol Data Unit (PDU)

A MAC PDU, shown in Figure A-1, consists of an optional MAC header and a MAC Service Data Unit (MAC SDU). Both the MAC header and the MAC SDU are of variable size.

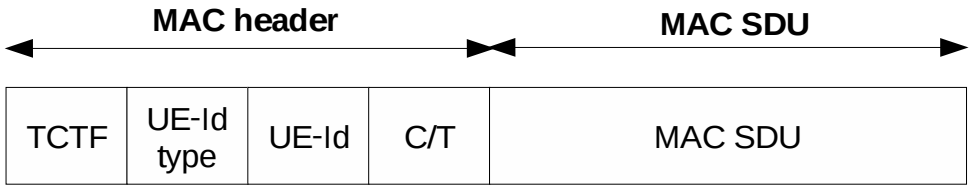

Figure A-1: MAC data PDU

The content and the size of the MAC header depends on the type of the logical channel. The size of the MAC SDU depends on the size of the RLC PDU, which is defined during the setup procedure.

Definitions of the MAC header fields for the DSCH:

- **Target Channel Type Field (TCTF)**
  Is a flag that provides identification of the logical channel class on FACH and RACJ transport channels. It is empty for DSCH
- **UE-Id type**
  Is needed to ensure correct decoding of the UE-Id field. Cell Radio Network Temporary Identity (C-RNTI) is used on DSCH.
- **UE-Id**
  Provides an identifier of the UE on common transport channels.
- **C/T**
  Indicates the logical channel where the data originates

A.2 State model for RLC TM entities

Figure A-2 illustrates the state model of a RLC TM entity.

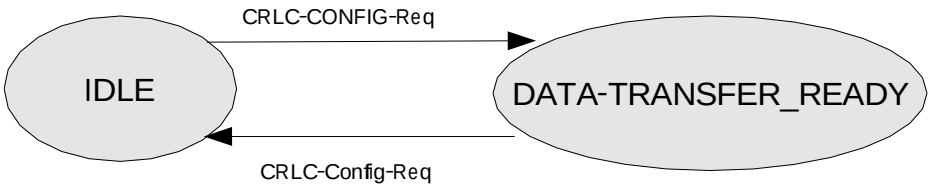

Figure A-2: State model for RLC TM entities [4]

A TM entity can be either in the IDLE state or in the DATA_TRANSFER_READY state.

IDLE state

The RLC entity does not exist and therefore it is not possible to transfer any data through it. Upon reception of a CRLC-CONFIG-Req (see Appendix B.5) from RRC indicating establishment, the RLC entity is created and enters the DATA_TRANSFER_READY state.

DATA_TRANSFER_READY state

Transparent mode data can be exchanged. Upon reception of a CRLC-CONFIG-Req from RRC indicating release, the RLC entity enters the IDLE state and is considered as being terminated.

A.3 State model of RLC AM entities

Figure A-3 illustrates the state model of the AM RLC entity.

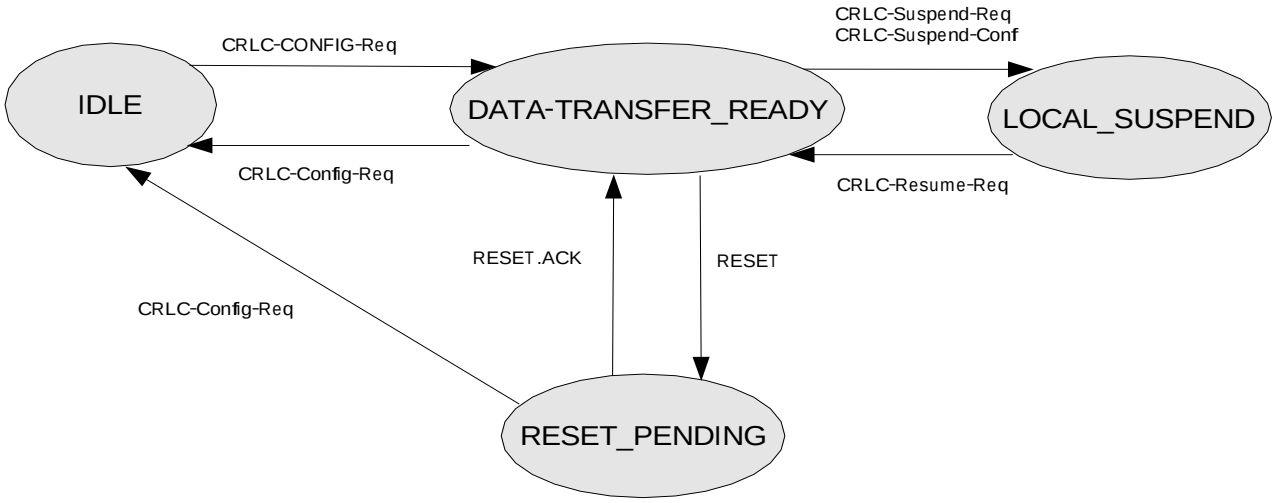

Figure A-3: State model of RLC AM entities [4]

An RLC AM entity can be in one of the following states:

IDLE state

The RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req (see Appendix B.5) from RRC indicating establishment, the RLC entity is created and enters the DATA_TRANSFER_READY state.

DATA_TRANSFER_READY state

Acknowledged mode data can be exchanged between the entities.

Upon reception of a CRLC-CONFIG-Req from RRC indicating release, the RLC entity enters the IDLE state and is considered as being terminated.

Upon detection of an initiating condition for the RLC reset procedure the RLC entity: initiates the RLC reset procedure and enters the RESET_PENDING state.

Upon reception of a RESET ACK PDU, the RLC entity takes no action.

Upon reception of CRLC-SUSPEND-Req from RRC, the RLC entity is suspended and enters the LOCAL_SUSPEND state.

RESET_PENDING state

The entity waits for a response from its peer entity and no data can be exchanged between the entities.

Upon reception of a CRLC-CONFIG-Req from RRC indicating release, the RLC entity enters the NULL state and is considered as being terminated.

Upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the RLC entity enters the DATA_TRANSFER_READY state.

Upon reception of a RESET ACK PDU with a different RSN value as in the corresponding RESET PDU, the RLC entity discards the RESET ACK PDU and stays in the RESET_PENDING state.

Upon reception of a RESET PDU, the RLC entity responds and stays in the RESET_PENDING state.

LOCAL_SUSPEND state

In the LOCAL_SUSPEND state, the RLC entity is suspended, i.e. it does not send AMD PDUs with Sequence Number greater than or equal to certain specified value.

Upon reception of CRLC-RESUME-Req from RRC, the RLC entity resumes the data transmission and enters the DATA_TRANSFER_READY state.

Upon reception of CRLC-CONFIG-Req from RRC indicating release, the RLC entity enters the IDLE state and is considered as being terminated.

In the following subsections state variables and parameters are described. RRC configures and sets these variables. Notice that I only outline those variables that are of importance for my simulations.

A.3.1 State Variables used in RLC AM

The states described in the last section depend on certain state variables that are discussed in this subsection. All state variables used in AM are non-negative integers. AM data PDUS are numbered by modulo integer sequence numbers. All arithmetic operations on most variables like VT(S), VT(A), VT(MS), VR(R), VR(MR) are affected by the AM modulus of 4096. When performing arithmetic comparisons of state variables or Sequence number values a modulus base shall be used. For comparison of transmitter variables VT(A) is assumed to be the modulus base, for comparison f receiver variables VT(R) shall be assumed to be the modulus base.

The following state variables in the sender are maintained by RLC [4]:

- **VT(S) - Send state variable**
  This state variable contains the sequence number of the next AM data PDU to be transmitted for the first time. It shall be updated after the PDU is transmitted.
- **VT(A) - Acknowledge state variable**
  This state variable contains the sequence number of the first PDU that is not acknowledged yet. This forms the lower edge of the transmission window of acceptable acknowledgements. VT(A) shall be updated based on the receipt of a Status PDU including an ACK.
- **VT(DAT)**
  This state variable counts the number of transmissions of a certain PU. There is one VT(DAT) for each PU and it shall be updated each time that it is scheduled to be transmitted.
- **VT(MS) - Maximum Send state variable**
  This state variable denotes the sequence number of the first PDU that can be rejected by the peer receiver due to transmission window size. This value represent the upper edge of the transmission window. VT(MS) = VT(A) + VT(WS).

- **VT(PU)**
  This state variable counts the number of all transmitted PUs, including both new and retransmitted PUs.
- **VT(SDU)**
  This state variable counts the number of transmitted SDUs. It shall be incremented by one for a given SDU when all the PDUs carrying a part of this SDU have been transmitted at least once.
- **VT(WS) - Transmission window size state variable**
  This state variable contains the size that shall be used for the transmission window. The initial value is set to the parameters value Configure_Tx_Window_size and shall be changed upon receiving a Status PDU including a Window SUFI.

The following state variables in the receiver are maintained by RLC [4]:

- **VR(R) - Receive state variable**
  This state variable contains the sequence number of the first PU that hasn't been received yet. It shall be updated upon the receipt of the PU with sequence number equal to VR(R).
- **VR(MR) - Maximum acceptable Receive state variable**
  This state variable contains the sequence number of the first PU that shall be rejected by the receiver. This forms the lower edge of the reception window. VR(MR)=VR(R)+Configured_Rx_Window_Size.

A.4 Format of a RLC AM PDU and a Status PDU

User data and piggybacked status information are transferred by AM data PDUs, shown in Figure A-4. The status PDU, shown in Figure A-5, provides means for control information transfer between transmitter and receiver of peer entities.

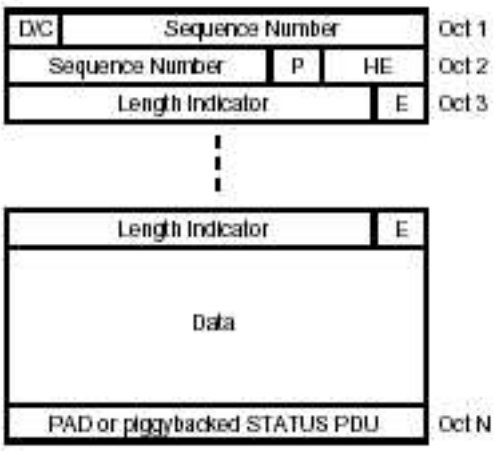

Figure A-4: AM data PDU [4]

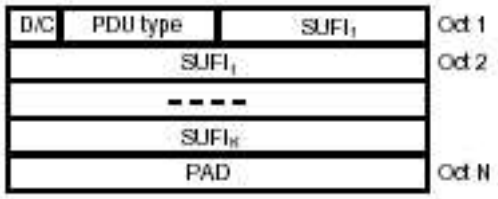

Figure A-5: AM Status PDU [4]

The fields in a RLC AM data PDU have the following meaning:

- **D/C**
  Indicates if the PDU is a data or a control PDU.
- **Polling bit (P)**
  Used to request a status report from the peer entity.
- **HE** and **E** bit
  Indicate the extension of the header within the PDU.
- **Sequence number**
  Indicates the number of the payload unit.
- **Length indicator (LI)**
  Used to define boundaries between RLC SDUs within AMD PDUs. Indicates the ending of a RLC AMD SDU within the payload. It also refers to a padding block or a piggybacked status PDU.
- **Data**
  Represents RLC SDUs or segments of them.
- **Padding**
  To fill up the PDUs to get PDUs of equal length.

The fields in an AM Status PDU have the following meaning:

- **PDU type**
  Indicates the type of control PDU.
- **SUFI**
  To transfer status information to the peer entity. Up to now, eight different SUFIs have been defined.

A.5 Parameters used in AM

Beside the state variables, parameters need to be defined to control the behavior of the RLC entity.

The following parameters are set by RRC:

- **MaxDAT**
  This parameter indicates the maximum number of retransmissions of a PU. It represents the upper limit for the state variable VT(DAT). When VT(DAT) equals the value MaxDAT, SDU discard procedure shall be initiated.

- **Poll_PDU**
  This parameter indicates how often the transmitter shall poll the receiver in the case where “polling every Poll_PDU PDU” is configured by upper layer. It represents the upper limit for the state variable VT(PU). When VT(PU) equals the value Poll_PDU the poll bit is set and is transmitted to the peer entity.
- **Poll_SDU**
  This protocol parameter also triggers the polling function, but only in the case where “polling every Poll_SDU SDU” is configured by upper layers. It represents the upper limit for the state variable VT(SDU). When VT(SDU) equals the value Poll_SDU the poll bit is set and is transmitted to the peer entity.
- **Poll_Window**
  This parameter indicates a relative position within the transmission window beyond that the sender should poll the receiver in the case where “window-based polling” is configured by upper layers. A poll is triggered if J, the transmission window percentage (defined below) is equal to the Poll_Window.

$$J = \frac{(4096 + VT(S) - VT(A)) \bmod 4096}{VT(WS)} \times 100$$

- **Configured_Tx_Window_Size**
  This parameter indicates the maximum allowed transmission window. It represent also the value for the state variable VT(WS). The value can be changed by the peer transmitting entity.
- **Configured_Rx_Window_Size**
  This parameter indicates the size of the reception window size. The value can be changed only by RRC. Normally, the reception window size and the transmission window size are of equal size.

A.6 Timers used in RLC AM

Timers are another factor that is having a noticeable influence on the performance of the RLC layer. The timers are considered active from the time they are started until the time they either expire (signalled by RRC) or are stopped.

The following timers are relevant for the simulation [4]:

- **Timer_Poll**
  This timer should be started when an AM data PDU with a set poll bit is transmitted. If x is the value of the state variable VT(S) after the poll was submitted to lower layer, the timer shall be stopped upon receiving a positive acknowledgements for all the AM data PDUs with sequence number up to and including x - 1 or a negative acknowledgement for the AM data PDU with sequence number = x - 1.
- **Timer_Poll_Prohibit**
  This timer is used to prohibit transmission of consecutive polls within a certain period. The timer is started every time an AM data PDU with a set poll bit is transmitted. The next poll is blocked until the expiry of this timer.
- **Timer_Poll_Periodic**
  This timer ensures a periodically triggered polling. The timer shall be started when the RLC entity is created.
- **Timer_Status_Periodic**
  This timer ensures that status reports are triggered periodically. This timer shall be started when the RLC entity is created.

A.7 Polling functions used in AM

RLC AM is mainly a transmitter driven protocol via the polling mechanism. The Polling function is used by the transmitter to request the peer RLC entity for a status report. The Polling bit in the PDU indicates the poll request. Upon receiving such a request or after an error is detected, the receiver sends a status report to the transmitter. There are a total of eight different polling triggers defined in the RLC specification. RRC controls, which triggers shall be used (polling can also be disabled by using the poll prohibit function).

The throughput and delay performance of each individual entity will be degraded with more frequent status reports.

The following triggers can be used:

- **Last PU in transmission buffer**
  When a PU to be transmitted for the first time is submitted to lower layer, the sender shall trigger a poll when the last PU in the transmission buffer is transmitted. Can avoid deadlock of the RLC entities.
- **Last PU in Retransmission buffer**
  The sender triggers a poll when the last PU in the retransmission buffer submitted to lower layer.
- **Poll timer**
  The sender triggers polls periodically (when the Timer_Poll expires).
- **Every Poll_PU PDU**
  The sender triggers a poll for every Poll_PU PDU.Both retransmitted and new PUs are counted.
- **Every Poll_SDU SDU**
  The Sender triggers the Polling function for every Poll_SDU SDU. The poll is triggered for the first transmission of the last PU that contains segments of the RLC SDU.
- **Window based**
  A poll is triggered if the transmission window percentage (defined in the last subsection) is equal to the Poll_Window.

A.8 Status transmission in RLC AM

The receiver transmits status reports to the sender to inform the sender about received and not received PUs.Each status report consists of one or several Status PDUs. There are several triggers defined for sending a status report. Which triggers should be used obtains the control by RRC.

The following triggers are configurable:

- **Receiving a poll request**
  Every time a poll request (indicated by the poll bit that is set in the PDU header) was received, a status report shall be transmitted.
- **Detection of missing PDU(s)**
  If the receiver detects one or several missing PUs it shall trigger the transmission of a status report to the sender.
- **Timer based status transfer**
  The receiver triggers the transmission of a status report to the sender periodically.

B Interactions between the different protocol layers

In Appendix A I mainly provide information about the different protocols used in UMTS. In this Appendix chapter I give an overview of the sometimes complex interactions between the different protocol layers.

B.1 Interactions between PHY and MAC layer

As the physical layer terminates at the Node B and the MAC layer terminates at the SRNC, interaction between these two layers take place over the connection between Node B and SRNC. The interface between the Node B and its RNC is called Iub, shown in Figure B-1. The radio interface between the UE and the Node B is called Uu. The interface between two adjacent RNCS is called Iur. As I examine the Downlink Shared Channel in my simulations I will first depict the user plane protocol for DSCH over Iub Interface.

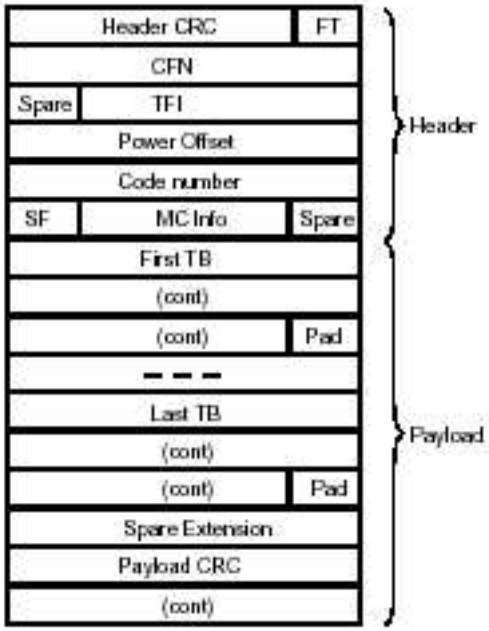

Figure B-1: Iub-Frame [3]

The fields in the Iub Frame have the following meaning:

- **Header CRC**
  Used for error detection. The CRC (Cyclic Redundancy Checksum) is calculated on the header of a data frame.
- **FT (Frame Type)**
  Describes if it is a control frame or a data frame.
- **CFN (Connection Frame Number)**
  Frame counter used for transport channel synchronisation between UE and UTRAN. A CFN value is associated with each TBS.
- **TFI (Transport Format Indicator)**
  Describes length and number of transport blocks transmitted within one TTI.
- **Power Offset**
  Indicates the preferred PDSCH transmission power level.

- **Code number**, **SF (Spreading Factor)**, **MC Info**
  Used to indicate the actual PDSCH code management to the Node B.
- **Spare Extension**
  Indicates the location where new Information Elements can in the future be added in a backward compatible way.
- **Payload CRC**
  CRC calculated on the payload of a data frame.

For my simulations I have not implemented the Iur frame but I have included a delay in the MAC layer that is equivalent to the delay caused by the Iur communication.

The following primitive is of importance for the simulation:

- **Iub-Data-Req**
  Indicates the transmission of a data frame from the DRNC to the Node B in the UTRAN.

The following PHY primitives are used between PHY and MAC and are important for the simulation [2]:

- **PHY-Access-CNF**
  Is used to confirm that physical layer synchronisation has been established and that the physical layer is ready for data transmission using the PHY-Data primitive.
- **PHY-Data-Req**
  Is used to request SDUs used for communications passed to and from the physical layer. One PHY-Data primitive is submitted every Transmission Time Interval for each Transport Channel.
- **PHY-Data-Ind**
  Is used to indicate SDUs used for MAC passed to and from the physical layer. One PHY-Data primitive is submitted every Transmission Time Interval for each Transport Channel.
- **PHY-Status-IND**
  Can be used by the physical layer to notify higher layers of an event that has occurred.

B.2 Interactions between PHY and RRC

The following CPHY primitives are used between PHY and RRC for control of the configuration of the physical layer and are important for the simulation [2]:

- **CPHY-Measurement-Req**
  Is used for RRC to configure PHY measurements.
- **CPHY-Measurement-Ind**
  Is used to report the measurement results.
- **CPHY-Error-Ind**
  Is used to indicate to the management entity that an error has occurred as a result of a PHY layer fault.

Control primitives [2]:

- **CPHY-TrCH-Config-Req**
  Is used for setting up and configure a transport channel, and also to modify an existing transport.
- **CPHY-Trch-Release-Req**
  Is used for releasing a transport channel.

- **CPHY-TrCH-Release-Conf**
  Is used for confirming releasing a transport channel.
- **CPHY-RL-Setup-Req**
  Is sent from RRC to PHY for establishment of a radio link to a certain UE.

B.3 Interactions between MAC and RLC

The following primitives are used between MAC and RLC and are important for the simulation [5]:

- **MAC-DATA-Req**
  Is used to request that a RLC PDU be sent using the procedures for the information transfer service.
- **MAC-DATA-Ind**
  Is used to indicate the arrival of RLC layer PDUs received within one Transmission Time Interval by means of the information transfer service.
- **MAC-STATUS-Ind**
  Is used to indicate RLC for each logical channel the rate at which it may transfer data to MAC.
- **MAC-STATUS-Resp**
  Is used by RLC to acknowledge a MAC-STATUS-Ind primitive in the case that there is nothing to send. This primitive is also used to indicate the current buffer occupancy to MAC.

B.4 Interactions between MAC and RRC

The following primitives are used between MAC and RRC and are of importance for the simulation [5].

- **CMAC-CONFIG-Req**
  Is used to request for setup, release and configuration of a logical channel.
- **CMAC-MEASUREMENT-Req**
  Is used by RRC to request MAC to perform measurements.
- **CMAC-MEASUREMENT-Ind**
  Is used to notify RRC of the measurement result.
- **CMAC-STATUS-Ind**
  Is used to notify RRC of status information.

B.5 Interactions between RLC and RRC

The following primitives are used between RLC and RRC and are of importance for the simulation [4].

- **CRLC-CONFIG-Req**
  Is used by RRC to establish, re-establish, release, stop, continue or modify the RLC.
- **CRLC-STATUS-Ind**
  Is used by and RLC entity to send status information to RRC.

B.6 Interactions between RLC and PDCP (User plane)

The following primitives are used between PDCP and RLC for user data transfer in AM and are important for the simulation [4]:

- **RLC-AM-DATA-Req**
  Is used by upper layers to request transmission of a RLC PDU in AM.

- **RLC-AM-DATA-Ind**
  Is used by the AM RLC entity to deliver to upper layers and RLC SDU that has been transmitted in AM and to indicate to upper layers of the discarded RLC SDU in the peer RLC AM entity.
- **RLC-AM-DATA-conf**
  Is used by RLC to confirm a transmission of a PDCP PDU to PDCP.

The following primitives are used between PDCP and RLC for user data transfer in TM and are important for the simulation:

- **RLC-TM-DATA-Req**
  Is used by upper layers to request transmision of a RLC SDU in TM.
- **RLC-TM-DATA-Ind**
  Is used by the TM RLC entity to deliver to upper layers a RLC SDU that has been transmitted in TM.
- **RLC-TM-DATA-Conf**
  Is used by the TM RLC entity to inform the upper layers of a discarded SDU.

B.7 Interactions between higher layers (>=3) and PDCP (User plane)

The following primitives are used between PDCP and upper layers and are important for the simulation.

- **PDCP-DATA-Req**
  Is used by the upper user plane protocol layer to request transmission of an upper layer PDU.
- **PDCP-DATA-Ind**
  Is used by the PDCP layer to deliver a PDCP SDU that has been correctly received from the peer entity.

C The network simulator NS-2

In this Appendix chapter I outline some basic information about NS-2.

NS-2 is an object oriented, discrete event driven network simulator developed at UC Berkeley written in C++ and OTcl [70]. NS-2 can be used in wired, wireless and satellite environment and implements network protocols like TCP and UDP, traffic source behavior like FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra. The simulator can simulate multicast or unicast and comes with a wide range of functions for statistics, tracing, error models and so on. NS-2 is very popular in academic research areas and a great variety of extensions to NS-2 are available. Many users share information about software bugs and features and the development of the simulator is continuously in progress. However the industry tends to favor OPNET [71]. Major disadvantages of OPNET are that costs for a license are rather high and that not the whole code is publicly available, while NS-2 is open source and freeware. However, one major disadvantage of NS-2 is that it is poorly documented.

C.1 OTcl and C++

The front-end of NS-2 is written in OTcl, which is an interpreted language with support for objects. With this, simulation parameters can be changed in the OTcl scripts for new simulations without having to recompile any code.

Most things, like the main protocols, topology elements and per-packet processing actions are written in separate C++ files. The objects in C++ are then linked to the OTcl objects in such a way that the behavior of the model objects in C++ can be controlled from OTcl objects. This approach has the benefit of both the powerful and fast executing features of C++ and also the flexibility of Tcl. For efficiency reasons new network objects are usually programmed in C++.

There are of course also drawbacks that comes with the shared object approach, for example special procedures are needed to make the C++ objects visible to the Tcl objects.

Figure C-1 shows the shared object design used in NS-2. Every object in OTcl has a corresponding object in C++. Since the objects are shared, their attributes can be accessed from both the OTcl and the C++ interface.

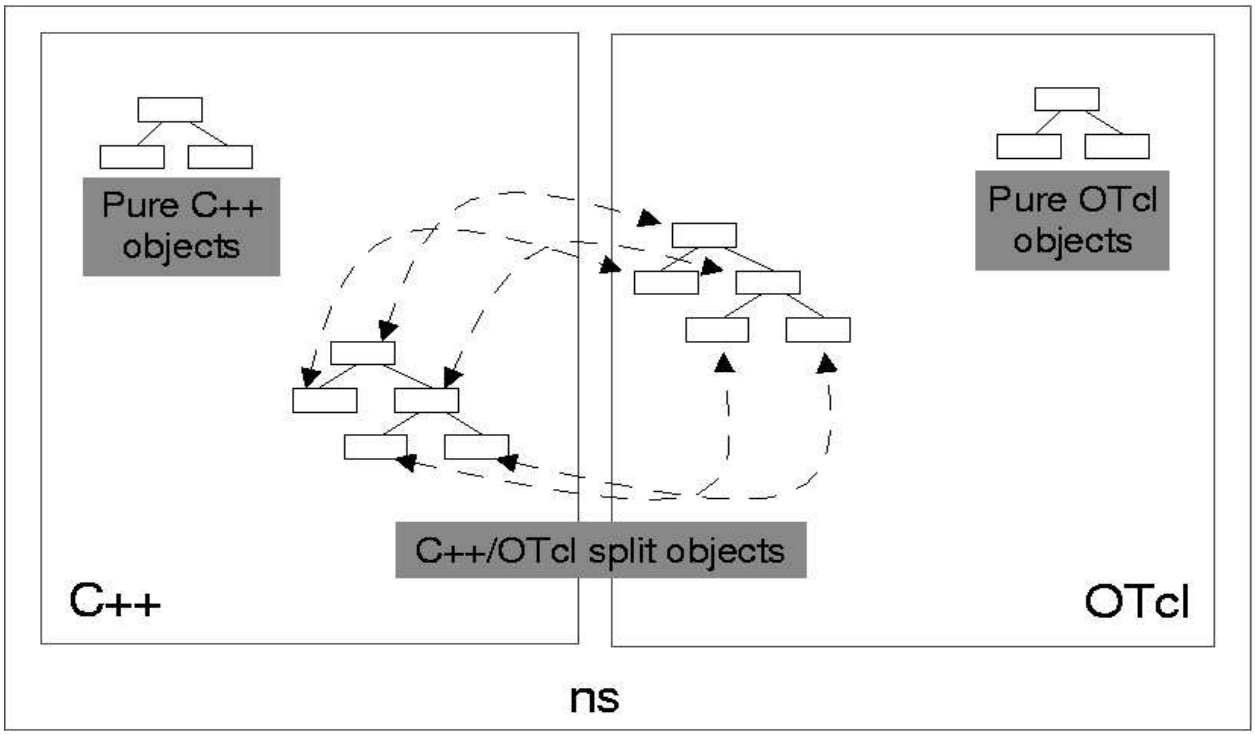

Figure C-1: The duality of OTcl and C++ objects

C.2 Schedulers and Events

The scheduler is the core of the simulation and provides functionality for handling events and making sure that they execute the correct code at the right time. The scheduler class has a great variety of features, but one of the most useful is the one providing a way to schedule your own events. Table C-1 shows how this is done in NS-2.

Table C-1: Scheduling of own events in NS-2

```
schedule(Handler*, Event*, double delay)
```

When an event is ready to dispatch, the handler passed to the scheduler is called. The handler then executes its code and takes appropriate action before passing the event (packet) on. All headers that are available in NS-2 are in fact present in the packet representation. This is in contrast to real packets where only the headers of the current used protocol are present. But this is not a problem, as the size of the packets are set in a special field in the cmn part of the header and the actual size of the packet in the NS-simulation is not of importance. The data field in the packet object are most of the time not used at all. Therefore only headers are passed around in the network simulator without actually carrying any data.

C.3 Nodes and links

To carry out simulations, a network topology needs to be defined. Such a topology consists of nodes and links. The links interconnect the nodes and one node can have multiple incoming and outgoing links. To determine what incoming packet should be sent out to which outgoing link, an internal routing is used. Each object that ever handles a packet has one thing in common: the way that the packet is received. A packet is always received through the method recv(Packet* P, Handle* H). P is

the packet being transferred and H is a reference to a handler. A handler is a pointer to the object that will execute if the receiver chooses to call it.

Beside the nodes, links are the other fundamental element of the simulation topology. Links are comprised of several components like queue, link delay and time to live (TTL).

C.4 Agents

Agents can be used together with traffic generators like constant bit rate (CBR) or file transfer protocol (FTP) to insert traffic into the network. There are many types of agents available like TCP and UDP. Agents are responsible for constructing and destroying packets.

The sender side in my simulations use a TCP agent and the receiver have a TCP Sink agent attached to it. In my simulations I have used different TCP agents for each TCP variant used for simulation. Furthermore, I have used another TCP Sink agent for simulation with delayed ACKs.

C.5 Tracing

NS-2 provides some built-in support for tracing the data flow generated in the simulation. The tracing is done by writing data and describing various events in a text file.

Beside tracing packet related events, the simulator can also be configured to trace other things. This manually configurable tracing utility is one very useful feature when working with TCP since all TCP's internal variables can be traced.

C.5.1 Important trace formats used in my simulations

Table C-2 shows the NS-2 trace format used in my simulations. The leftmost column indicates what type of event that occurred and the next column the time when that event occured.The third entry indicates the current node. The r symbol represents the event that the node received a packet, while + and - symbols are related to queuing. The + means that a packet was inserted in a queue, while - means that a packet was taken from a queue. AGT in the forth column stands for agent. The fifth column shows the id of this event. The sixth column indicates if the packet is either a TCP data packet or an acknowledgement packet. The seventh column indicates the size of the packet. 10:0 in the ninth column is the next important column, it indicates sender ip-address and port number. The tenth column indicates the ip-address and port number of the destination.

Table C-2: NS-2 trace format

```
r 0.264025000 _51_ AGT --- 36 tcp 1000 [0 0 8 0] ------- [10:0 4194314:0
29 4194314] [0 0] 0 0
```

C.6 The wireless domain in NS-2

The wireless node is created differently from a fixed node. It has the mobility to move around in a given topology, and transmit onto and receive from a wireless channel. The mobility features include node movement (only two dimensional), periodic position updates and maintaining topology boundary. These features are implemented in C++ in the files mobilenode.{cc,h}. The stack is linked together by OTcl in ns-mobilenode.tcl (see Appendix C.8.4). By default, each node is configured as "ad-hoc" and can send or listen to any other mobile node within reach.

In my simulations I require each UE to talk to its Node B only, and not to any other UE. To achieve this I can use the Non-Adhoc routing agent NOAH (see Appendix 6.4.1), developed by Joerg Widmer [73].

To do my simulations I introduce a RLC protocol layer and extensions of the MAC protocol layer to the default ns mobile node. This involves changing the nodal structure in ns-mobilenode.tcl and other changes in ns-lib.tcl and ns-default.tcl.

C.7 Wired-cum-wireless scenarios

If a mobile node wants to communicate to a fixed wired node, it can do so by attaching it to a base station (Node B in UMTS). The Node B is implemented as a mobile node in ns but with wired routing switched on and motion disabled. Hierarchical routing has to be used with the wired and wireless parts kept in different domains.

C.8 Where to find what in NS-2?

NS-2 is composed out of a large amount of different files. Figure C-2 gives an overview of the directory structure in NS-2. In the following subsections the most important tcl parameter setting files are outlined.

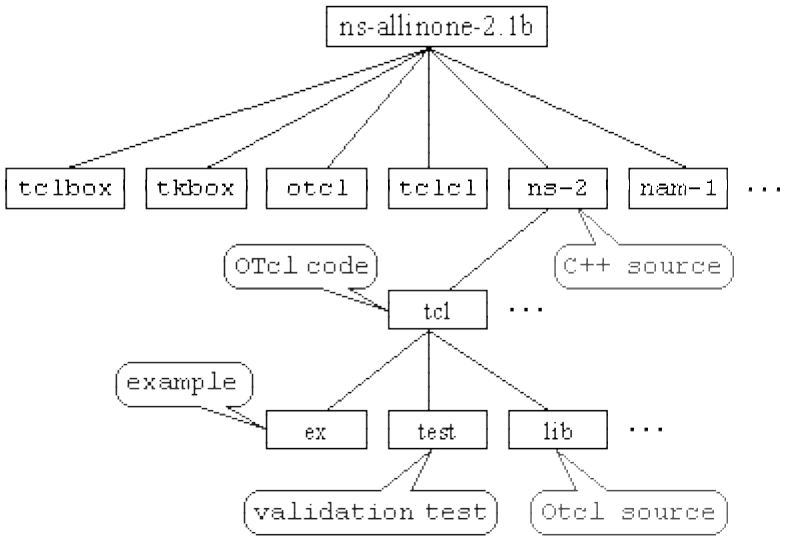

Figure C-2: NS-2 directory structure

C.8.1 ns-default.tcl

The default values for configurable parameters for various network components are located here. Since most of network components are implemented in C++, the configurable parameters are actually C++ variables made available to OTcl via an OTcl linkage function, bind(C++_variable_name, OTcl_variable_name). Parameters of the MAC, RLC and RRC layers are set in this file. To set those parameters in a tcl file has the advantage of not been forced to recompile the code for a simulation with new parameter settings.

C.8.2 ns-lib.tcl

The simulator class and most of its member function definitions are located here.

C.8.3 ns-packet.tcl

The packet header format initialization implementation is located here. To make the UMTS module working information about the RLC and MAC packet header need to be added here.

C.8.4 ns-mobilenode.tcl

Set up link layer, mac layer, network interface and physical layer structures for the mobile node.

For the simulation this file needs to be modified.

C.8.5 packet.h

All the headers that are available in NS-2 are present in the packet representation. This is in contrast to real IP packets where only the headers that are used are present. To compensate for this, the size of the packet is not at all dependent on these headers, instead the size of the packet is set in a special field in the cmn part of the header. The data field in the packet are most of the time not used at all. This results in that only headers are passed around in the network without actually carrying any data. Since NS is only a simulator this is reasonable, and everything that is dependent on the size of the packet, uses the size indicated by the number in the cmn field in the header.

For the UMTS module I needed to add here the RLC packet header and the MAC header.

C.9 Mobility model generation with the help of setdest

Generating manually node-movement models for large wireless scenarios is laborious. Fortunately, a node-movement generator for NS-2 mobile nodes is available under ~ns/indep-utils/cmu-scen-gen/setdest NS-2 directory that consists of setdest{.cc,.h} and a Makefile. To run the generator, go tho the above directory and run setdest with the arguments shown in Table C-3:

Table C-3: Setdest options

./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime] [-x maxx] [-y maxy] > [outdir/movement-file]

The output file begins with the initial position of the nodes and goes on to define the node movements.

C.10 System requirements of UMTS module for NS-2

The minimum required NS-2 version to use the UMTS module is NS 2.1b7a. However, I would like to encourage to use the ns-2.1b9a version with which I performed my simulations or a newer version. All new versions of NS-2 are available for download at the official NS-2 web site [70].

NS-2 is available for most operating systems. I have used Linux Suse version 8.1 to host the network simulator Ns-2.

Furthermore, you need to copy the c++ files (see list bellow) of the UMTS module into the NS-2 directory, change the settings in the library files and run the make file.

The following files have to be added to NS-2: fading.{cc,h}, noah.{cc,h}, umts-{fdd,tdd}-mac.{cc,h}, umts-{fdd,tdd}-phy.{cc,h}, umts-{fdd,tdd}-prop.{c,h}, umts-mq.{cc,h}, umts-rlc-timers.{cc,h}, umts-rlc.{cc,h}, umts-varp.{cc,h} and noah.tcl.

The following files have to be modified: channel.{cc,h}, cmu-trace.cc, god.{cc,h}, packet.h, propagation.{cc,h}, template.h, ns-mobilenode.tcl, ns-default.tcl, ns-packet.tcl and ns-lib.tcl.

D UMTS radio interface simulator details

In this appendix chapter details about the link-level simulations are outlined.

D.1 Link-level simulations

For my simulations I used a mapping function derived from UMTS link-level simulations performed by a research group led by Professor Zorzi at University of Ferrara. This group simulated an outdoor channel. Multipath fading was taken into account by using the Jakes model, shadowing by using the model proposed by Gudmunson, and path loss by using the Hata model. The speed of the user (pedestrian, around 3 km/h) characterizes the maximum Doppler frequency. The contributions of the 5 highest power paths (each path comprises r rays) are then combined according to a power profile typical of an outdoor cellular environment to obtain the effect of a RAKE receiver. These 5 paths correspond to the line of sight path and the 4 primary reflections against obstacles (it has been verified that usually there are 3 or at most 4 relevant reflections). The model is accurate enough. In practice r=8 rays are sufficient to obtain a spectrum similar to that measured in a cellular environment [7], [52].

D.2 Sample simulation script

Bellow a sample simulation script is provided.

```
ns-random 1973272912
# Extract command line arguments
foreach argument $argv {
    scan $argument "duration=%d" duration
    scan $argument "kindoftraffic=%s" kind_of_traffic
    scan $argument "loadfactor=%s" load_factor
    scan $argument "srctype=%s" srctype
    scan $argument "sinktype=%s" sinktype
    scan $argument "rlcparameterset=%d" rlc_parameter_set
    scan $argument "tcpparameterset=%d" tcp_parameter_set
    scan $argument "simulationnumber=%d" simulation_number
}

if {$load_factor == "high"} {
  set num_nodes 30;
}
if {$load_factor == "mid"} {
  set num_nodes 20;
}
if {$load_factor == "low"} {
  set num_nodes 5;
}

# Define options
set opt(chan)    Channel/FddChannel;# channel type
set opt(prop)    Propagation/UmtsFdd;# radio propagation model
set opt(netif)   Phy/UmtsFddPhy;# network interface type
set opt(macbs)   Mac/UmtsFdd/BS;# BS MAC type
set opt(macms)   Mac/UmtsFdd/MS;# MS MAC type
```

```
set opt(ifq)     Queue/MQ;# interface queue type
set opt(ll)      LL/RLC;# link layer type
set opt(ant)     Antenna/OmniAntenna;# antenna model
set opt(ifqlen)  5000;# max packets in ifq
set opt(adhocRouting) NOAH;# routing protocol
set opt(x)       500;     # x coordinate of topology
set opt(y)       500;# y coordinate of topology
set opt(seed)    0.0;# seed for random number generator
set opt(start)   0.1;
set opt(stop)    $duration;# time to stop simulation

if {$load_factor == "high"} {
  set opt(sc)"../scenarios/scen30";
}
if {$load_factor == "mid"} {
  set opt(sc)"../scenarios/scen20";
}
if {$load_factor == "low"} {
  set opt(sc)"../scenarios/scen5";
}

set tot_nodes   [expr $num_nodes + 1];

#********Set name of tracefiles***
set opt(tr)      "../mysimulations_fdd/$kind_of_traffic/$srctype/RLC_AM/
TCP_$tcp_parameter_set/RLC_$rlc_parameter_set/Sim_$simulation_number/
trace_$load_factor.tr"
set opt(trtcp)   "../mysimulations_fdd/$kind_of_traffic/$srctype/RLC_AM/
TCP_$tcp_parameter_set/RLC_$rlc_parameter_set/Sim_$simulation_number/
tcp_$load_factor.tr"
#*********************************

Mac/UmtsFdd set verbose_ 0;# to include MAC verbose output
Mac/UmtsFdd set alloc_trace_ 0;# trace allocated blocks
Mac/UmtsFdd set snr_trace_ 0;# trace SNR
Mac/UmtsFdd set txpower_trace_ 0;# trace tx power
Mac/UmtsFdd set DCH_UL_SF_ 128;# uplink DCH SF
Mac/UmtsFdd set min_DSCH_SF_ 21;
Mac/UmtsFdd set mean_inter_ -65;# is the mean value (in DBm) of intercell
interference, modelled with a gaussian random variable.
Mac/UmtsFdd set var_inter_ 0.3333;# variance of intercell interference
Mac/UmtsFdd set sir_target_ 6; # Eb/No target in dB
Mac/UmtsFdd set rho_inter_ 0.8;# correlation between two successive values

Phy/UmtsFddPhy set error_rate_ -1;# if set to -1, the packet error
probablility is evaluated from the SIR information. if set to 0 an ideal
physical channel is used.if set to n (n>0) errors are generated randomly
with probability 1/n
Phy/UmtsFddPhy set seed_ 13; # is the seed of pseudo random sequence used
to generate the shadowing and the fading.
```

```
Phy/UmtsFddPhy set verbose_ 0;

LL/RLC set rlcfraged_  1;# 1 if segmentation is used, 0 if no segementation
should be applyed
LL/RLC set rlculfragsz_  40;# RLC UL fragment size (in bytes)
LL/RLC set rlcdlfragsz_  40;# RLC DL fragment size (in bytes)
LL/RLC set rlcverbose_ 0;# to include RLC verbose output

LL/RLC set acked_      1;#V 1 if acked, 0 if unacked and 2 if transparent
RLC mode

if {$rlc_parameter_set == "1"} {
LL/RLC set buffer_size_ 64;# Size of the RLC buffer (in RLC PDUs)
LL/RLC set MaxDAT_ 6; # Maximum Number of retransmissions for an RLC PDU
LL/RLC set window_ 64;# Size of the RLC window (in RLC PDUs)
LL/RLC set ptimer_duration_ 0.55;#V Poll timeout
LL/RLC set poll_SDU_ 2;# Polling every x of SDU sent
LL/RLC set mrw_timer_duration_ 0.2;# move receiving window (MRW) timeout
}

Agent/TCP set windowInit_ 4; #initial/reset value of cwnd
if {$tcp_parameter_set == "1" || $tcp_parameter_set == "2"} {
  Agent/TCP set ssthresh_ 20; #V slow-stat threshold (packets);
  Agent/TCP set packetSize_ 1000; # packet size used by sender (bytes);
}
# The following lines make NS 2.1b9a
# behave like NS 2.1b7a
Agent/TCP set syn_ false ;
Agent/TCP set delay_growth_ false ;
Agent/TCP set useHeaders_ false ;

# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
  puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
  puts "Seeding Random number generator with $opt(seed)\n"
  ns-random $opt(seed)
}
#remove extrapkt headers else each pkt takes up too much space.
remove-packet-header LDP MPLS Snoop
remove-packet-header Ping TFRC TFRC_ACK
remove-packet-header Diffusion RAP IMEP
remove-packet-header AODV SR TORA IPinIP
remove-packet-header MIP HttpInval
remove-packet-header MFTP SRMEXT SRM aSRM
remove-packet-header mcastCtrl CtrMcast IVS
remove-packet-header Resv UMP Flags

# create simulator instance
```

```
set ns_   [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical
AddrParams set domain_num_ 2   ;# number of domains
lappend cluster_num 1 1    ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel $tot_nodes $tot_nodes # number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

set tracefd  [open $opt(tr) w]
$ns_ trace-all $tracefd

set tracetcp [open $opt(trtcp) w]

# Create wired nodes
set temp ""
for {set i 0} {$i <= $num_nodes} {incr i} {
  lappend temp 0.0.$i
}
set GW(0) [$ns_ node [lindex $temp 0]]
for {set i 0} {$i < $num_nodes} {incr i} {
    set W($i) [$ns_ node [lindex $temp [expr $i+1]]]
    $ns_ duplex-link $GW(0) $W($i) 40Mb 50ms DropTail
}
# Create topography object
set topo   [new Topography]

# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God
set god_ [create-god [expr $num_nodes+1]]

#set wireless channel before the nodes;they are shared by all nodes
set chan1 [new $opt(chan)]

# configure for base-station node
$ns_ node-config -adhocRouting $opt(adhocRouting) \
                 -llType $opt(ll) \
                  -macType $opt(macbs) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                  -topoInstance $topo \
                 -wiredRouting ON \
                  -agentTrace ON \
                 -routerTrace OFF \
```

```
                -macTrace OFF  \
                 -movementTrace OFF \
                 -channel $chan1

set temp ""
for {set i 0} {$i <= $num_nodes} {incr i} {
  lappend temp 1.0.$i
}

# VARP is needed to resolve IP addresses into MAC addresses using a list
set varp [new UmtsVARPTable]

#create base-station node
set BS(0) [$ns_ node [lindex $temp 0] $varp]
$BS(0) random-motion 0 ;# disable random motion: A base station is fix!
#provide some co-ord (centre of square) to base station node
$BS(0) set X_ [expr $opt(x)/2]
$BS(0) set Y_ [expr $opt(y)/2]
$BS(0) set Z_ 0.0

# create link between BS and GW
$ns_ duplex-link $BS(0) $GW(0) 200Mb 10ms DropTail
#configure for mobilenodes
$ns_ node-config -wiredRouting OFF \
                -macType $opt(macms)

set rng [new RNG]
$rng seed 0

for {set j 0} {$j < $num_nodes} {incr j} {
    set node_($j) [ $ns_ node [lindex $temp [expr $j+1]] $varp ]
    $node_($j) base-station [AddrParams addr2id [$BS(0) node-addr]]
    [$BS(0) set mac_(0)] add_mobile [[$node_($j) set mac_(0)] id]
}
# Define movement model
puts "Loading scenario file..."
source $opt(sc)

# FTP traffic
for {set j 0} {$j < $num_nodes} {incr j} {
  if {$kind_of_traffic == "ftp"} {
        set s($j) [new Agent/TCP/$srctype]
        $ns_ attach-agent $W($j) $s($j)
        $s($j) set packetSize_ 1000
        $s($j) set window_ 10
        $ns_ add-agent-trace $s($j) s($j)
        $ns_ monitor-agent-trace $s($j)
        $s($j) attach $tracetcp
        $s($j) trace cwnd_
        $s($j) trace rtt_
```

```
        $s($j) trace rttvar_

  if {$sinktype =="Newreno" && $tcp_parameter_set == "1"} {
        set null($j) [new Agent/TCPSink]
  }
  if {$sinktype =="Newreno" && $tcp_parameter_set == "2"} {
        set null($j) [new Agent/TCPSink/DelAck]
  }
  if {$tcp_parameter_set == "1" && $sinktype =="WestwoodNR"} {
        set null($j) [new Agent/TCPSink]
  }
  if {$tcp_parameter_set == "2" && $sinktype =="WestwoodNR"} {
        set null($j) [new Agent/TCPSink/DelAck]
  }
  if {$sinktype =="Sack1"} {
        set null($j) [new Agent/TCPSink/$sinktype]
  }
  $ns_ attach-agent $node_($j) $null($j)
  $null($j) set packetSize_ 40
  $ns_ connect $s($j) $null($j)
  set ftp($j) [new Application/FTP]
  $ftp($j) attach-agent $s($j)
  $ns_ at $opt(start) "$node_($j) start"
    $ns_ at $opt(start) "$ftp($j) start"
    $ns_ at $opt(stop) "$ftp($j) stop"
}
    if {$kind_of_traffic == ""} {
        puts "Need to know the kind of traffic"
    }
}

# Tell all nodes when the simulation ends
for {set i 0} {$i < $num_nodes} {incr i} {
    $ns_ at $opt(stop).0 "$node_($i) reset";
    $ns_ at $opt(stop).0 "$W($i) reset";
}
$ns_ at $opt(stop).0 "$GW(0) reset";
$ns_ at $opt(stop).0 "$BS(0) reset";
$ns_ at $opt(stop).0002 "$ns_ halt"
$ns_ at $opt(stop).0001 "stop"
proc stop {} {
global ns_ tracefd tracetcp
    $ns_ flush-trace
    close $tracefd
    close $tracetcp
}
puts "Starting Simulation..."
$ns_ run
```

E Pseudocode for packet processing

In this Appendix chapter I will provide some samples of the many pseudo codes developed to build a UMTS module implementation for NS-2. Note that these samples were only partly implemented in real code as another implementation of an advanced simulator implementation became available that was chosen as basis implementation.

E.1 RLC AM functions of the transmitter in the UTRAN side

The functions discussed in this section are used for transferring data between two RLC AM peer entities (see A.3).

After receiving RLC Service Data Units (SDUs), those SDUs are split up into RLC payload units (PU) of fixed size and are stored in the transmission buffer. To be able to reassemble the upper layer PDUs a header with additional information is added to the PU. Whenever possible the first PU of an upper layer PDU is concatenated to the last RLC PDU stored in the Transmission buffer (see E.1.2). If the RLC PDU is transmitted for the first time, a copy of the data is stored in the retransmission buffer. When RLC PDUs are requested by the MAC layer for transmission, a multiplexer chooses the RLC PDUs to be transferred. One or several PDUs may be transmitted in each transmission time interval (TTI) and the MAC layer decides how many RLC PDUs shall be transmitted in each TTI. After this, Status PDUs are transmitted to the peer entity.

In the following sections some sample pseudo codes are displayed to give a short inside on how they have been developed.

E.1.1 Receive function

Purpose

Filters the received messages and calls the appropriate functions for them.

Called

The function is called each time a message have been received from an other layers.

Pseudocode

```
switch (kind_of_packet)

//A data packet has been received
case RLC_AM_Data_Req:
Concatenation(packet);
Segmentation(packet);
Send_MAC-Status-Resp(); break


//A resource allocation from MAC initiates transmission of PDUs has been received
case MAC_Status_Ind: Send_MAC-Data-Req(packet); break;

//An ACK msg has been received
case ACK: Data_Acknowledged(packet); break;

//A NACK msg has been received
case NACK: Data_Negative-Acknoledgement(packet); Send_MAC-Status-Resp();
break;
```

E.1.2 Concatenation

Purpose

Concatenation of received SDU to earlier received SDU into the same RLC PDU in order to...

Called

Every time a new RLC Service Data Unit (SDU) is received.

Input Parameter

RLC Service Data Unit (SDU)

Pseudocode

```
if (! tx_buf.empty()) {
  if (free_space_in_PDU > 0 AND oldrlc->VT_DAT==1){
        if (rlc_sdu_length > free_space_in_PDU) {
         set_rlc_sdu_length(rlc_sdu_length - free_space_in_PDU);
         set_old_rlc_sdu_lenth(PU_length);
        }
        else {
         set_old_rlc_sdu_lenth(old_rlc_sdu_lenth + rlc_sdu_length);
         add_LI_entity();
         increase_number_of_SDUs();
        }
  }
}
```

Acknowledgements

This masters thesis was performed at the FTW (Forschungszentrum Telekommunikation Wien) telecom research lab in vienna in cooperation with the Computer Engineering and Networks Laboratory Group at the Swiss Federal Institute of Technology (ETH) Zurich in 2003.

I would like to express my gratitude to my tutors at FTW Peter Reichl, Thomas Ziegler and Christoph Mecklenbräuker who guided me through the entire work, for their valuable information and endeavors throughout the time I wrote this thesis. They gave me support and feedback but still let me enough free space to develop and realize my own ideas, which I appreciated very much. They also encouraged me in many different ways and provided valuable hints about developing a UMTS network simulator and on how to carry out good network simulations.

I want also to thank all other members of FTW, especially Eduard Hasenleithner, Hermann Anegg, Ivan Gojmerac, Joachim Wehinger, Erich Plasser, Horst Thiess and Martina Umlauft for their best care, many important advises and numerous constructive technical discussions.

Furthermore, I would like to thank my supervisor Professor Burkhard Stiller for being again my supervisor and for his excellent advises.

Special thanks goes to Pascal Kurtansky for his support and for managing the initial coordination between ETH Zurich and ftw. Vienna.

Furthermore, I would like to direct my acknowledgements to Alfredo Todini, Francesco Vacirca and Werner Perndl for providing me with their UMTS simulator implementations and for their good advises.

Special thanks go to Ed Schofield and Caroline Bécède for proof reading this thesis.

Marcel Lötscher

Vienna, march 2003

References

[1] 3rd Generation Partnership Project. Radio Interface Protocol Architecture (Release 5), 3GPP TS 25.301, V5.2.0, Sep 2002.

[2] 3rd Generation Partnership Project. Services provided by the physical layer (Release 5), 3G TS 25.302, V5.2.0, Sep 2002.

[3] 3rd Generation Partnership Project. UTRAN Iur interface user plane protocols for Common Transport Channel data streams (Release 5), 3GPP TS 25.425, V5.2.0, Sep 2002.

[4] 3rd Generation Partnership Project. Radio Link Control (RLC) protocol specification (Release 5), 3GPP TS 25.322, V5.2.0, Sep 2002.

[5] 3rd Generation Partnership Project. MAC protocol specification (Release 5), 3GPP TS 25.321, V5.2.0, Sep 2002.

[6] 3rd Generation Partnership Project. Physical channels and mapping of transport channels to physical channels (FDD), (Release 5) 3GPP TS 25.211, V5.2.0, Sep 2002.

[7] 3rd Generation Partnership Project. Technical Specification Group Radio Access Networks RF System Scenarios, 3GPP TS 25.942 (Release 5), V5.1.0, Jun 2002.

[8] S. Keshav: An Engineering Approach to Computer Networking; ATM Networks, the Internet and the Telephone Network, 1997.

[9] V. Jacobson: Congestion Avoidance and Control, Proceedings of ACM SIGCOMM Conference, Aug 1988.

[10] M. Mathis, J. Madhavi, S. Floyd, A. Romenow: TCP Selective Acknowledgment Options, RFC 2018, IETF, Oct 1996.

[11] K. Fall, S. Floyd: Simulation-based Comparisons of Tahoe, Reno and SACK TCP, Berkeley, Mar 1996.

[12] O. A. Hellal, E. Altman: Analysis of TCP Vegas and TCP Reno, Sophia Antipolis, 2000.

[13] R. Prasad, W. Mohr, W. Konhäuser: Third Generation Mobile Communication Systems, Artech, 2000.

[14] J. Zander, S. Kim: Radio Resource Management for Wireless Networks, Artech, 2001.

[15] J. Laiho, A. Wacker, T. Novosad: Radio Network Planning and Optimization for UMTS, Wiley, 2002.

[16] J. Schiller: Mobile Communications, Addison-Wesley, 2000.

[17] W. Perndl: Scheduling Algorithms for UMTS FDD Downlink, diploma thesis, TU Wien, 2001.

[18] A. Bakre, B. R. Badrinath: I-TCP: Indirect TCP for Mobile Hosts, ICDCS 1995, Oct 1994.

[19] H. Balakrishnan, R. Katz: Explicit Loss Notification and Wireless Web Performance, Proc. IEEE GLOBECOM Global Internet Conf., Sydney, Nov. 1998.

[20] B. Bakshi, P. Krishna, N. H. Vaidya, D. K. Pradhan: Improving Performance over Wireless Networks, 17th Int. Conf. Distributed Computing Systems, Baltimore, May 1997.

[21] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. H. Katz: A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, IEEE/ACM Transactions on Networking, Dec 1997.

[22] R. K. Balan: TCP Hack: TCP Header Checksum Option to Improve Performance over Lossy Links, IEEE Infocom, 2001.

[23] S. Floyd, T. Henderson: The NewReno Modification to TCP's Fast Recovery Algorithm, RFC 2582, Apr 1999.

[24] R. Wang, M. Valla, M.Y. Snadidi, B. Ng: Efficiency Friendliness Tradeoffs in TCP Westwood, Proc. Of IEEE ISCC'02, Toaromina, Italy, 2001.

[25] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sandadidi, R. Wang: TCP Westwood: End-to-End Bandwidth Estimation for Enhanced Transport over Wireless Links, ACM Wireless Networks, 2002.

[26] S. Vangala, M. A. Labrador: Performance of TCP over Wireless Networks with the Snoop Protocol, University of South Florida, 2001.

[27] W.R. Stevens: TCP/IP Illustrated, Volume 1. The Protocols, Wesley, 1994.

[28] Transmission Control Protocol, RFC 793, Sep 1981.

[29] J. Peisa, M. Meyer: Analytical model for TCP file transfer over UMTS. Proceedings of the 2001 International Conference on Third Generation Wireless and Beyond, San Francisco, USA, Jun 2001.

[30] Q. Zhang, H. Su: Performance of UMTS Radio Link Control, Bell Laboratories, Lucent Technologies, 2002.

[31] K. Ratnam, I. Matta: WTCP: An Efficient Transmission Control Protocol for Networks with Wireless Links, IEEE Symp. Computer and Communications (ISCC), Jun 1998.

[32] V. Tsaoussidis, C. Zhang: TCP-Real: receiver-oriented congestion control, COMNET 2002, Apr 2002.

[33] S. Papayiannis, S. Hadjiefthymiades, L. Merakos: Implications of proactive datagram caching on TCP performance in wireless/mobile communications, University of Athens, Greece, Mar 2002.

[34] V. Tsaoussidis, I. Matta: Open Issues on TCP for Mobile Computing, the Journal of Wireless Communications and Mobile Computing, John Wiley & Sons, Issue 1, Vol. 2, Feb 2002.

[35] IP Header Compression, RFC 2507, Feb 1999.

[36] J. Kang, D. Kim, Y. Shin, H. Park, J. Lee: Performance Evaluation of TCP over WCDMA RLC, ICOIN 2002.

[37] G. Yang, R. Wang, M. Y. Sanadidi, M. Gerla: Performance of TCPW BR in Next Generation Wireless and Satellite Networks,UCLA CSD Technical Report Nr. 020025, 2002.

[38] M. Thoppian, A. Veduru: TCP for Wireless Networks, University of Texas in Dallas, 2001.

[39] A. Natani, J. Jakilinki, M. Mohsin, V. Sharma: TCP for Wireless Networks, The University of Texas at Dallas, Nov 2001.

[40] A. Mate, C. Caldera, M. Rinne: Performance of the Packet Traffic on the Downlink Shared Channel in a WCDMA cell, IEEE ICT2001, Jun 2001.

[41] TCP over 2.5G and 3G Wireless Networks, IETF Network Working Group Internet-Draft, Version 11, Nov 2002.

[42] G. Smaragdakis: TCP Performance over UMTS Network, master's thesis, Technical University of Crete, 2002.

[43] R. Ludwig, R. H. Katz: The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions, ACM Computer Communications Review, Vol. 30, No. 1, Jan 2000.

[44] A. Floberg: Channel type switching in WCDMA, masters thesis, Ericsson Erisoft AB and Lulea University of Technology, Apr 1999.

[45] M. Zorzi,M. Rossi,G. Mazzini: Throughput and energy performance of TCP on a Wide-band CDMA air Interface, WCDMA for UMTS: radio access for third generation mobile communications, John Wiley, 2001.

[46] E. Cianca, M. Ruggieri, R. Prasad: Improving TCP/IP performance over CDMA wireless links: a Physical Layer Approach, PIMRC'01, San Diego, Sep 2001.

[47] M. Chatterjee, G. D. Mandyam, S. K. Das: Fast ARQ in High Speed Downlink Packet Access for WCDMA Systems, University of Texas, 2001.

[48] S. A. Malik, D. Zeghlache: Improving Throughput and Fairness on the Downlink Shared Channel in UMTS WCDMA Networks, Institut National des Telecommunications France, 2002.

[49] R. Jain: GPRS Simulations using ns-Network Simulator, masters thesis, Indian Institute of Technology Bombay, Jun 2001.

[50] E. Lundsten: Improving 3G Performance for the Mobile Internet, KTH and Telia Research, Dec 2002.

[51] L. Nuaymi, X. Lagrange, P. Godlewski: A Power Control Algorithm for 3G WCDMA System, EW2002, 2002.

[52] ETSI: Universal Mobile Telecommunications System(UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS, TR 101 112 V3.2.0, Apr 1998.

[53] M. Hunukumbure, M. Beach, B. Allen: Downlink orthogonality factor in UTRA FDD systems, IEE Electronics Letters, Vol. 38, No. 4, Feb 2002.

[54] M. Simon, M. Alouini: A Unified Approach to the Performance Analysis of Digital Communication over Generalized Fading Channels, Proceeding of the IEEE, Nov 1998.

[55] S. Marikar: Resource Management in 3G Systems employing Smart Antennas, master's thesis, Virginia Polytechnic Institute, Jan 2002.

[56] M. Loetscher, P. Kurtansky, B. Stiller: Business-models for wireless networks, student thesis, Swiss Federal Institute of Technology Zurich, Jul 2002.

[57] G. Loeffelmann, E. Bonek: TCP/IP over UMTS, diploma thesis, TU Wien, Sep 2000.

[58] K. Pentikousis: TCP in wired-cum-wireless environments, State University of New York, May 2000.

[59] S. Lee, B. Kim, Y. Choi: TCP Vegas Slow Start Performance in Large Bandwidth Delay Networks, Seoul National University Korea, ICOIN, Sep 2002.

[60] A. Baiocchi, F. Cuomo, C. Martello: Optimizing the radio resource utilization of multi-access systems with a traffic-transmission quality adaptive packet scheduling, University of Rome La Sapenzia Italy, Aug 2001.

[61] X. Xu, Y.C. Chen, H. Xu, E. Gonen: Simulation analysis of RLC timers in UMTS systems, WSC 2002, 2002.

[62] H. Xu, Y.C. Chen, X. Xu, E. Gonen: Performance analysis on the radio link control protocol of UMTS system, VTC 2002, 2002.

[63] F. Lefevre, G. Vivier: Optimizing UMTS Link Layer Parameters for a TCP Connection, VTC 2001, 2001.

[64] H. Holma, A. Toskala: WCDMA for UMTS, Wiley, Oct 2000.

[65] M. Zorzi: Energy Management in Personal Communications and Mobile Computing, IEEE Personal Communications Magazine, vol.5, Jun 1998.

[66] R. Kwan and M. Rinne: Performance Analysis of the Downlink Shared Channel in a WCDMA Network, Nokia Research Center Finland.

[67] A. Todini, F. Vacirca: UMTS-FDD modules for NS, manual of the FDD operation mode implementation of the UMTS simulator for NS-2 developed at University of Rome, Italy, Jan 2003.

[68] A. Todini, F.Vacirca: UMTS-TDD manual, manual of the TDD operation mode implementation of the UMTS simulator for NS-2 developed at University of Rome, Italy, Jan 2003.

[69] M. Hunukurnbure, M. A. Beach, B. H. Allen: Quantifying code orthogonality factor for UTRA-FDD downlinks, University of Bristol, U.K.

[70] Network Simulator (NS-2) web site: http://www-mash.cs.berkeley.edu/ns

[71] Opnet Technologies Inc. web site: http://www.optnet.com

[72] Cooperative Association for Internet Data Analysis (CAIDA), Traffic Workload Overview: http://www.caida.org/outreach/resources/learn/trafficworkload/tcpudp.xml

[73] None-Adhoc (NOAH) routing for NS-2n: http://www.informatik.uni-mannheim.de/informatik/pi4/projects/MobileIP/ns-extension/

Institut für
Technische Informatik und
Kommunikationsnetze

Prof. Dr. Burkhard Stiller, ETZ G 61.4, CH-8092 Zürich, Tel. +41 1 632 7016, Fax +41 1 632 1035, E-Mail stiller@tik.ee.ethz.ch
Dr. Thomas Ziegler, FTW, A-1220 Wien, Tel. +43 1 5052830-30, Fax +43 1 5052830-99, ziegler@ftw.at
Dr. Christoph Mecklenbräuker, FTW, A-1220 Wien, Tel. +43 1 5052830-23, Fax +43 1 5052830-99, mecklenbraeuker@ftw.at
Dr. Peter Reichl, FTW, A-1220 Wien, Tel. +43 1 5052830-31, Fax +43 1 5052830-99, reichl@ftw.at

Diplomarbeit

für

Herrn Marcel Lötscher

Aufgabenstellung:	*Dr. Thomas Ziegler* *Dr. Christoph Mecklenbräuker* *Dr. Peter Reichl*
Thema:	***Simulative Performance Optimization for TCP over UMTS***
Beginn der Arbeit:	*08.11.2002*
Abgabetermin:	*07.03.2003*
Betreuung:	*Dr. Thomas Ziegler* *Dr. Christoph Mecklenbräuker* *Dr. Peter Reichl* *Dipl. Inf.-Ing. ETH Pascal Kurtansky*
Arbeitsplatz:	*ftw. Wien*
Hilfsmittel:	*ns-2 Simulationsumgebung am ftw.*

1. Einleitung

Mehr als 90% aller Verkehrsflüsse im Internet werden durch das TCP (Transmission Control Protocol) [12] gesteuert. Beispielapplikationen, die TCP verwenden, umfassen WWW, email, FTP, etc. Die Optimierung und Untersuchung dieses Protokolls erfährt daher in der Research Community entsprechend hohe Aufmerksamkeit.

Mit der Erweiterung des Internets über drahtlose Netze hat sich die Untersuchung und Optimierung der Leistung von TCP über spezifischen drahtlosen Subnet-Layern zu einem weiteren wichtigen Forschungsgebiet entwickelt [13]. Neben Technologien wie Wireless LAN wird in den nächsten Jahren vor allem das Universal Mobile Telephon System UMTS an Bedeutung gewinnen [1-5]. Deshalb ist die Untersuchung und Optimierung von TCP über UMTS, die als Thema dieser Diplomarbeit gewählt wurde, von essentieller Wichtigkeit.

TCP Congestion Control hat in den letzten Jahren einige Verbesserungsstufen durchlaufen. Die erste bedeutende Verbesserung gegenüber Van Jacobsons Originalpaper von 1988, in dem Slowstart und ein neues Verfahren zur Berechnung des Retransmission Timeout vorgeschlagen wurden, war der sog. Fast Retransmit/Recovery-Algorithmus, der es TCP erlaubt, mit dem Verlust eines einzelnen Paketes pro RTT zurechtzukommen, ohne das Congestion Window auf 1 reduzieren zu müssen. Diese TCP-Version ist als "TCP Reno" bekannt. TCP NewReno erlaubt

die Ausführung von mehreren Fast Retransmits hintereinander und ist daher besser geeignet, falls mehrfache Verluste pro RTT auftreten. TCP Vegas führt sensitive Kontrollmechanismen ein, um die TCP-Dynamik zu glätten und weniger "greedy" zu machen. Diese letztere Eigenschaft hat allerdings dazu geführt, dass es niemals eingeführt wurde, da TCP Vegas den Wettbewerb um knappe Bottleneck-Bandbreite gegen die weitverbreitete Reno-Version verlieren würde. TCP SACK führt sog. "selective acknowledgements" in TCP ein (d.h. bis zu drei "Löcher" im Sequenznummernraum können überbrückt werden) und ist bereits weit verbreitet (z.B. Linux OS).

All diese Versionen der TCP Error/Congestion Control wurden vor einem Festnetzhintergrund mit vernachlässigbarer Bit Error Rate auf dem Übertragungskanal entwickelt. Daher unterscheiden Standard-TCP-Versionen nicht zwischen Verlusten aufgrund Congestion und Verlusten aufgrund Bitfehlern im Kanal und verkleinern folglich das Congestion Window für beide Fälle von Paketverlusten gleichermassen.

In den vergangenen Jahren fand TCP/IP auch mehr und mehr in drahtlosen Netzen Verbreitung. In einer drahtlosen Umgebung, in der Bitfehler auf dem Kanal u.U. mit hoher Wahrscheinlichkeit vorkommen können, führt die Verkleinerung des Congestion Windows ohne Unterscheidung zwischen beiden geschilderten Verlusttypen zu signifikanter Performance-Verminderung und macht de facto die TCP Congestion Control ineffizient. Wenn etwa in einem bei weitem nicht ausgelastetem Netz ein einziger drahtloser Link mit hoher Bitfehlerrate vorliegt, kann die Verringerung des Congestion Windows aufgrund von Verlusten auf diesem einen Link dazu führen, dass der Durchsatz der TCP Flows gegen Null konvergiert, obwohl das Netzwerk keinerlei Congestion aufweist.

Es gibt mehrere Vorschläge, dieses Problem zu lösen. "Indirect TCP" [6] splittet die TCP-Verbindung zwischen einem fixen und einem mobilen Host in zwei getrennte Verbindungen und verbirgt TCP vor dem Verlust-Link durch Verwendung eines Protokolls, das auf verlustreiche Links hin getrimmt ist. Das Berkeley SNOOP Protokoll [7] speichert Pakete an der Basisstation zwischen und führt lokale Retransmissions über das Verlust-Link durch. Bei beiden Ansätzen ist zu beachten, dass in grossem Umfang Microflow-States und sogar Datenpakete in den Basisstationen gespeichert werden müssen, was auf die Skalierungsfähigkeit entsprechende Auswirkungen hat.

Explicit Loss Notification (ELN) [8] ist ein Mechanismus, der es erlaubt, den Grund für einen Paketverlust an den TCP-Sender zu kommunizieren. Wenn die Basisstation sicher weiss, dass der Verlust eines Segments nicht auf Congestion zurückzuführen ist, setzt sie das ELN-Bit im TCP-Header und schickt es zur Quelle. Die Basisstation überwacht alle über das drahtlose Link ankommenden TCP-Segmente, führt allerdings keine Retransmissions durch und muss daher auch keine Segmente zwischenspeichern. Stattdessen führt sie Buch über die verlorenen Segmente und setzt das ELN-Bit in den entsprechenden ACK-Nachrichten. Um eine falsche Markierung von Segmenten, die aufgrund von Congestion verlorengingen, zu vermeiden, setzt die Basisstation das ELN-Bit nur, wenn die Paketanzahl an der Interface-Queue der Basisstation noch weit entfernt von der maximalen Länge dieser Warteschlange ist. ELN erfordert das Führen einer Liste der verlorenen Sequenznummern pro Flow, was hinsichtlich Skalierbarkeit akzeptabel ist. Allerdings kann die Markierung der ACKs aufgrund von Routensplitting unmöglich werden, wenn sich das drahtlose Link nicht am Edge des Netzes befindet.

[9] untersucht die Performance von TCP over wireless bei variabler MTU-Grösse und schlägt vor, ICMP-Pakete zu senden, um die TCP-Quelle vom Auftreten von Bitfehlern auf einem drahtlosen Link in Kenntnis zu setzen. Schliesslich schlägt [10] vor, die TCP Header Checksum-Option zu verwenden, um festzustellen, ob ein Bitfehler im Body des TCP-Segments oder im

Header geschehen ist. Im ersteren Falle wird das Congestion Window nicht verkleinert. Dieses Paper setzt allerdings voraus, dass drahtlose Layer 2-Protokoll fehlgeleitete Pakete nicht vernichten, was ziemlich unrealistisch ist.

Das IST-Projekt MobyDick (www.ist-mobydick.org) beschäftigt sich unter anderem mit dem Thema "TCP Optimization over Wireless". Ziel dieser Arbeiten ist es, (rückwärtskompatible) TCP-Verbesserungen zu untersuchen, die eine Unterscheidung von Paketverlusten aufgrund Congestion von solchen aufgrund Bitfehlern ermöglichen. Für nähere Details zu den hierzu untersuchten Ansätzen sei auf [11] verwiesen.

2. Aufgabenstellung

Die generelle Aufgabe dieser Diplomarbeit besteht darin, eine Untersuchung der Performance von TCP über UMTS ohne Modifikationen an Algorithmen vorzunehmen. Neben der Untersuchung der Auswirkung von verschiedenen Parametereinstellungen bestehen also Freiheitsgrade in der optimalen Einstellung von Parametern sowohl in TCP als auch in UMTS. In Anbetracht der Vielzahl von vorhanden Parametern (z.B. UMTS Buffersize, Einstellung Scheduling Mechanismus, TCP Window Size, delayed ACKs on/off, verschiedene Lasten durch Benutzer, ...) wird dies hinreichend komplex betrachtet.

Die Untersuchung von TCP über UMTS soll durch Simulationen erfolgen. In der Internet Community gilt der ns-2 Simulator [14] als das State-of-the-Art Tool. In ns-2 wurden sämtliche relevanten TCP Derivate, Traffic Control Mechanismen wie Packet Scheduling und Queue Management, Routing Mechansimen, Support for Tracing und Monitoring, sowie eine Vielzahl von Verkehrsgeneratoren implementiert; durch jahrelange Verwendung können essentielle Fehler nahezu ausgeschlossen werden. Daher soll ns-2 auch im Rahmen dieser Diplomarbeit verwendet und erweitert werden. Während ns-2 über eine Wireless-LAN Implementation verfügt, fehlt ein entsprechendes Gegenstück für UMTS. Daher wird es eine wichtige Teilaufgabe dieser Diplomarbeit sein, die für diese Untersuchung relevanten Teile von UMTS in ns-2 zu implementieren.

Da es allerdings den Rahmen einer Diplomarbeit sprengen würde, sämtliche Teilaspekte von UMTS zu implementieren, erfolgt eine Einschränkung wie folgt:

- Wir beschränken uns auf den kommerziell relevanten FDD-Mode, TDD wird nicht implementiert.

- Am Physical Layer wird nur der UMTS Shared Channel implementiert, da dieser relevant ist um die Interaktion von verschieden TCP-Flüssen zu untersuchen. Dedicated Channel wird nicht implementiert.

- Ein oder zwei Scheduling Mechanismen für den Shared Channel werden implementiert (z.B. Shortest Queue First und Round Robin).

- Am RLC Layer wird der Transparent Mode und der Acknowleged Mode implementiert. Unacknowledged Transfer Mode wird nicht implementiert.

- Wir schränken uns auf Single Cell-Szenarien ein, D.h. obwohl Handover-Untersuchungen zwischen UMTS und Wireless LAN als wünschenswert erachtet werden, wird sowohl Intra- als auch Inter-Technology-Handover nicht gefordert.

Untersuchte Szenarien beinhalten:

- mehrere Lastsituationen mit FTP und Web-ähnlichem TCP-Verkehr
- mehrere Versionen von TCP: SACK [6], NewReno [7], TCP Westwood [8] (eine neue TCP Variante, die für drahtlose Netze optimiert ist)
- Verändern von TCP-Parametern: Delayed Acks, send window size, initial congestion window size, RTO Granularität, und andere die im Laufe der Arbeit zu spezifizieren sind
- Verändern von UMTS Parametern: Buffer sizes für shared Channel, Parmeter am Scheduling Mechanismus, Parameter für den Ackowledged Mode (z.B. Anzahl Pakete für die ein ACK gesendet wird). Die detaillierte Spezifikation der zu verändernden UMTS-Parameter soll ebenfalls im Laufe der Arbeit erfolgen.

3. Vorgehensweise

Folgende Schritte beschreiben einen sinnvollen Vorschlag für den Ablauf der Arbeit:

- Machen sie sich ausgehend von den angegebenen Literaturhinweisen mit den Themen TCP, UMTS und Simulation unter ns-2 hinreichend vertraut und arbeiten Sie sich in die praktischen Fragen der Implementierung von ns-2-Modulen ein.
- Suchen sie selbständig nach weiteren Veröffentlichungen im Bereich der Diplomarbeit. Verwenden sie dazu Bibliotheken und vor allem auch das Internet. Dabei sind vor allem Veröffentlichungen im Rahmen von Konferenzen und Workshops von Interesse. Diese sollten nur in Ausnahmefällen aus dem Zeitraum älter als zwei Jahre sein. Machen sie sich auf diese Art und Weise mit existierenden Ansätzen vertraut und dokumentieren sie diese in ihrem Bericht.
- Ausgehend von den damit ermittelten Grundlagen erstellen sie ein eigenes Konzept im Sinne der Aufgabenstellung. Erstellen Sie insbesondere eine genaue Spezifikation der zu untersuchenden Szenarien (z.B. welche UMTS Parameter).
- Implementieren Sie die in Kapitel 2 spezifizierten UMTS-Module in ns-2. Hierbei kann auf eine Diplomarbeit an der TU Wien zurückgegriffen werden [18], in deren Rahmen ein UMTS-Simulator entwickelt wurde.
- Führen Sie die Simulationen entsprechend der Spezifikation der Szenarien durch.
- Abschliessend muss die Arbeit beschrieben und zusammengefasst werden. Dabei ist es besonders wichtig, noch offene Punkte aufzuzeigen und mögliche Lösungen zu skizzieren.

4. Bemerkungen

- Programmiersprachen: C++, OTCL unter Linux
- Der Student kann auf des ftw. Simulationsenvironment zugreifen: 9 PCs, auf denen ns-2 ausgeführt werden kann.

- Ein Arbeitsplatz-PC wird zur Verfügung gestellt.

- Nach Ablauf der ersten Woche muss ein Zeitplan für den Ablauf der Arbeit eingereicht und mit den Betreuern diskutiert werden.

- Am Ende der Arbeit muss ein schriftlicher Bericht eingereicht werden, der die geleistete Arbeit dokumentiert. Dieser Bericht sollte so geschrieben sein, dass er für einen Nicht-Spezialisten verständlich ist. Weiterhin muss er alle wichtigen Vorbedingungen, Design-Entscheidungen und Architektur-Details enthalten.

- Es muss ein regelmässiger Kontakt (möglichst mindestens einmal pro Woche) zwischen dem Studenten und seinen Betreuern erfolgen, per Telefon, E-Mail, Treffen oder auf anderem Wege. Diese Kontakte sollen dazu verwendet werden, den Fortgang der Arbeit darzustellen und auftretende Probleme zu diskutieren.

- Besonders wichtig ist das regelmässige (möglichst tägliche) Lesen von E-Mails.

5. Ergebnisse der Arbeit

Die Arbeit muss innerhalb eines 20minütigen Vortrags am TIK vorgestellt werden. Der genaue Termin dieses Vortrags wird vorausichtlich der 14. März 2003 sein. Abgesehen vom Vortrag müssen folgende Dokumente eingereicht werden:

- Der in englisch abgefasste Bericht. Dieser Bericht muss folgende Punkte beinhalten: Eine Beschreibung des untersuchten Forschungsgebiets, eine Beschreibung der untersuchten Design-Alternativen, eine Begründung für die Wahl der ausgewählten Design-Variante, eine Liste von gelösten und ungelösten Problemen, Inhalts-, Tabellen- und Bildverzeichnisse, Literaturangaben und eventuell Anhänge (z.B. Programmcode, Glossar und ähnliches). Der Bericht muss mit einer Beurteilung abschliessen, in wie weit die ursprüngliche Aufgabenstellung erreicht und er aufgestellte Zeitplan eingehalten werden konnten. Es müssen insgesamt fünf gebundene und doppelseitig gedruckte Kopien des finalen Berichts eingereicht werden. Der Bericht sollte in FrameMaker erstellt werden.
- Die Zusammenfassung sollte wie folgt gegliedert sein:

  - Einleitung - Ziele - Ergebnisse - Offene Punkte -
- Ein deutscher Abstract von ein bis zwei Seiten Länge. Dieses sollte einen schnellen Überblick über die gesamte Arbeit ermöglichen. Es muss hinter der ersten weissen Seite in den Bericht eingefügt werden.
- Eine elektronische Version des Berichts und alle sonstigen erstellten Dokumente (z.B. Modelle). Bilder sollten dabei zusätzlich als gesonderte Dateien vorliegen, in einem gut verarbeitbaren Format (z.B. EPS). Die Abgabe sollte entweder auf einer CD oder durch ein seperates Verzeichnis im für die Arbeit angelegten Account erfolgen.
- **Für Referenzen auf Dokumente aus dem Internet gelten folgende Grundsätze**:

  Alle referenzierten PDF Dokumente sind als Kopie auf die CD zu legen, wobei die Dateinamen die Referenzierungsnummer enthalten müssen. In der Referenzliste im Text muss zusätzlich der verwendete Dateinamen, wie er auf der CD enthalten ist, angegeben sein.
  Referenzen auf Websites mit allgemeinem Charakter wie beispielsweise "ww.xyz.com" können als Link in die Referenzliste aufgenommen werden. Alle referenzierten Webseiten (HTML, ASP etc.) aus bestimmten Websites sind als PDF Dokumente auf die CD

zu legen, wobei die Dateinamen die Referenzierungsnummer enthalten müssen. In der Referenzliste müssen einerseits der gesamte Link vorhanden sein, andererseits der verwendete Dateinamen, wie er auf der CD enthalten ist.

- Ein Handbuch zum erstellten System. Darin müssen beschrieben werden: Systemvoraussetzungen, Installation, Handhabung.
- Der gesamte erstellte Quellcode und das ausführbare Programm.

6. Literaturangaben

[1] Antti Toskala et al.: Wideband-CDMA for UMTS. Wiley

[2] Laiho, Wacker, Novosad: Radio Network Planning and Optimization for UMTS. Wiley

[3] Schiller: Mobile Communications. Addison-Wesley.

[4] Prasad, Mohr, Konhäuser: Third Generation Mobile Communication Systems. Artech.

[5] Zander, Kim: Radio Resource Management for Wireless Networks.

[6] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In Proc. 15th International Conf. on Distributed Computing Systems (ICDCS), May 1995.

[7] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R.H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, IEEE/ACM Trans. on Networking, 5(6), December 1997.

[8] Hari Balakrishnan and Randy Katz, Explicit Loss Notification and Wireless Web Performance, Proc. IEEE GLOBECOM Global Internet Conf., Sydney, Australia, Nov. 1998

[9] B. Bakshi, P. Krishna, N. H. Vaidya, D. K. Pradhan, Improving Performance of TCP over Wireless Networks,17th Int. Conf. Distributed Computing Systems, Baltimore, May 1997.

[10]R.K. Balan et. al , TCP Hack: TCP Header Checksum Option to Improve Performance over Lossy Links, IEEE Infocom 2001

[11]MobyDick Deliverable D201.

[12]V. Jacobson, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM Conference, August 1988

[13]H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R.H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, IEEE/ACM Trans. on Networking, 5(6), December 1997.

[14]ns network simulator homepage, http://www-mash.cs.berkeley.edu/ns

[15]S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 2582, April 1999

[16]M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, October 1996

[17]R. Wang et al, "Efficiency Friendliness Tradeoff in TCP Westwood". Proc. Of IEEE ISCC'02, Toaromina, Italy

[18]Werner Perndl, "Scheduling Algorithms for UMTS FDD Downlink", DA TU Wien, 2001