THESIS / THÈSE

DOCTOR OF SCIENCES

Building a tested emulating cellular networks

design, implementation, cross-validation and exploitation of a real-time framework to evaluate QoS and QoE in the UTRAN

Van Peteghem, Hugues

Award date:
2008

Awarding institution:
University of Namur

Link to publication

Building a Testbed Emulating Cellular Networks

Design, Implementation, Cross-Validation and Exploitation of a Real-Time Framework to Evaluate QoS and QoE in the UTRAN

by
Hugues Van Peteghem

A dissertation submitted to
the Faculty of Computer Science, The University of Namur
in partial fulfilment of the requirements for the degree of
DOCTOR OF PHILOSOPHY,
March 2008, Namur, Belgium.

Remerciements

A l'issue de ces cinq années de travail, mes premières pensées vont évidemment à mon promoteur, **Laurent Schumacher**, sans qui aucune des pages de cette thèse n'auraient pu s'écrire. Au début de ce projet, il a su m'aiguiller lorsque je n'arrivais pas à cerner le sujet, il m'a ensuite poussé à me poser les bonnes questions afin de mieux orienter ma recherche. De cette interaction continue a jailli de nombreuses idées, souvent fructueuses, tantôt abandonnées, tantôt irréalisables mais toujours très constructives.

Dans cet exercice d'écriture de mes remerciements, la seconde personne qui me vient tout naturellement à l'esprit est bien évidemment **Géraldine**. Elle a eu l'occasion de suivre l'entièreté de l'évolution de ma thèse et a toujours répondu présente lors des moments difficiles, ces périodes de doute si spécifiques aux chercheurs qui se demandent si leur travail a vraiment un sens. Elle m'a continuellement encouragé et soutenu et sans son aide discrète mais si importante, cette dissertation n'aurait sans doute jamais vu le jour non plus.

En dépit des apparences, la recherche n'est certainement pas une discipline individuelle. Au même titre que les échanges entre promoteur et doctorant, les interactions avec d'autres pôles de recherche sont extrêmement importantes. Je tiens donc ici à souligner l'aide que j'ai aimablement reçue de la part de **Marie-Ange Remiche** de l'unité de recherche Méthodes Algorithmiques en Probabilité de l'ULB; de **Christophe De Vleeschouwer** du Laboratoire de Télécommunications et Télédétection de l'UCL; de **Wolfgang Karner** de l'Institut des Communications et des Fré-

quences Radio de l'université de Vienne; ainsi que de **Erwann Thoraval** de l'Institut de Recherche en Informatique et Systèmes Aléatoires de l'université de Rennes 1.

Je tiens également à remercier quelques uns de mes étudiants, en fin de Bachelor ou de Master, qui m'ont aidé à traiter des aspects très particuliers au sein de ma recherche. Je pense ici à **Yuji Sekiguchi**, **Lucia D'Acunto** et **Claudio Greco**.

Le binôme promoteur-doctorant est continuellement chaperonné par un comité d'encadrement. Celui-ci s'assure de l'originalité et de la pertinence du projet de recherche. Il contrôle également le respect des engagements et du planning. Ce travail a été admirablement réalisé par les Professeurs **Troels Bundgaard Sørensen** et **Chris Blondia**. Dans le même ordre d'idée, je tiens également à largement saluer les autres membres de mon jury, **Stefan Engels** et les Professeurs **Jean-Marie Jacquet** et **Vincent Englebert**, qui ont consacré une part importante de leur temps à la relecture de cette thèse. Leurs nombreuses remarques m'ont permis d'améliorer sensiblement la qualité du manuscrit et de véritablement le finaliser.

Un énorme remerciement doit absolument aller à **Sarah Zandona** et **Raphaël Gruber** qui se sont efforcés de corriger à plusieurs reprises les nombreuses fautes que comptait ce document. Sans leur contribution, la qualité linguistique de ce manuscrit n'aurait certainement jamais été ce qu'elle est maintenant.

Mes pensées vont enfin à ma famille et mes amis dont la liste est bien trop longue que pour être totalement énumérée ici. Je tiens à les remercier du fond du cœur pour le soutien qu'ils m'ont tous apporté ainsi que l'énergie qu'ils m'ont procuré durant toutes ces années. Une petite partie de la réussite de ce projet revient à chacun d'entre eux.

Merci à tous!

Hugues Van Peteghem
Février 2008

Contents

List of Figures

List of Tables

List of Acronyms

1G	1st Generation
2G	2nd Generation
3G	3rd Generation
3GPP	3rd Generation Partnership Project
ACK	TCP ACKnowledgement
AM	Acknowledged Mode
AMPS	Advanced Mobile Phone System
AMR	Adaptive MultiRate
ARQ	Automatic Repeat reQuest
B3G	Beyond 3rd Generation
BCH	Broadcast CHannel
BER	Bit Error Rate
BGP	Border Gateway Protocol
BLAS	Basic Linear Algebra Subprograms
BLER	BLock Error Rate
BPSK	Binary Phase Shift Keying
CAC	Call Admission Control
CDMA	Code Division Multiple Access
CN	Core Network
CPICH	Common PIlot CHannel
CPU	Central Processing Unit
D-AMPS	Digital-AMPS
DCH	Dedicated CHannel
DSCH	Downlink Shared CHannel
DSCP	Differentiated Services Code Point
DVB	Digital Video Broadcasting
E2E	End-to-End
EDGE	Enhanced Data rates for GSM Evolution
eNodeB	enhanced NodeB

ESA	European Space Agency
FACH	Forward Access CHannel
FDD	Frequency Division Duplex
GNU	Gnu's Not Unix
GoP	Group of Pictures
GPRS	General Packet Radio Service
GSL	GNU Scientific Library
GSM	Global System for Mobile communications
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
HTB	Hierarchical TB
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
IPv4	IP version 4
IPv6	IP version 6
ISA	Inter-Session Arrival
IST	Information Society Technologies
LAPACK	Linear Algebra PACKage
LTE	Long Term Evolution
MAP	Markovian Arrival Process
MIMO	Multiple Input Multiple Output
MMS	Multimedia Messaging Service
MPEG-4	Motion Picture Experts Group Layer-4
MSE	Mean Square Error
NetEm	Network Emulation
NMT	Nordic Mobile Telephone
NodeB	WCDMA base station
NP	Non-Preemptive
NP-P	NP and Preemptive
NRT	Non Real-Time
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OSPF	Open Shortest Path First
OVSF	Orthogonal Variable Spreading Factor
OWC	Operator's Worst Case
OWC+	OWC + degradation
OWNP2	OWN site Plus 2 other sector interference model
PCH	Paging CHannel
PDC	Personal Digital Communication

PDU	Protocol Data Unit
PEAQ	Perceptual Evaluation of Audio Quality
PER	Packet Error Rate
PESQ	Perceptual Evaluation of Speech Quality
POP	Post Office Protocol
POSIX	Portable Operating System Interface on uniX
PSNR	Peak Signal-to-Noise Ratio
PSTN	Public Switched Telephone Network
QoE	Quality of Experience
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
RAN	Radio Access Network
Rel'4	UMTS Release 4
Rel'5	UMTS Release 5
Rel'6	UMTS Release 6
Rel'99	UMTS Release 99
RIP	Routing Information Protocol
RLC	Radio Link Control
RNC	Radio Network Controller
RRC	Radio Resource Control
RRM	Radio Resource Management
RT	Real-Time
RTP	Real-time Transport Protocol
RTT	Round-Trip Time
SAE	System Architecture Evolution
SDU	Service Data Unit
SF	Spreading Factor
SMS	Short Message Service
SNIR	Signal-to-Noise plus Interference Ratio
SNodeB	Serving NodeB
SSH	Secure SHell
S-UMTS	Satellite-UMTS
TACS	Total Access Communication System
TB	Token Bucket
TC	Traffic Controller
TCP	Transport Control Protocol
TDD	Time Division Duplex
TG	Traffic Generator

TIM	Telecom Italia Mobile
TM	Transparent Mode
TTI	Transmit Time Interval
TV	TeleVision
UDP	User Datagram Protocol
UE	User Equipment
UM	Unacknowledged Mode
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial RAN
UWC	User's Worst Case
VoIP	Voice over IP
WCDMA	Wideband CDMA
WiMAX	Worldwide interoperability for Microwave Access
WLAN	Wireless Local Area Network
WWW	World Wide Web

Part I

Introduction & Motivations

Chapter 1

Introduction

1.1 Telecommunication Evolution

The evolution of mobile communications and the Internet has led to the third generation of cellular networks, of which Universal Mobile Telecommunications System (UMTS) is a part. However, the mobile communication has undergone many evolutionary steps before reaching this state.

Figure 1.1: Simplified telecommunication evolution time line.

As shown in Fig. 1.1, the introduction of cellular systems in the late 1970s and early 1980s was a major step in mobile communication, especially in terms of capacity and mobility. 1st Generation (1G) was the first generation system. 1G systems were analogue and offered only voice services. They were not protected against eavesdropping and offered no roaming possibilities. The most important 1G systems were Advanced Mobile Phone System (AMPS), Nordic Mobile Telephone (NMT) and Total Access Communication System (TACS). With the introduction of 1G, the mobile market had annual growth rates of 30 to 50 percent, reaching nearly 20 million subscribers by 1990.

2nd Generation (2G) heralded then digital voice and messaging services and offered encrypted transmission. 2G cellular systems include Global System for Mobile communications (GSM), Digital-AMPS (D-AMPS), Code Division Multiple Access (CDMA) and Personal Digital Communication (PDC). GSM became the dominant 2G standard and roaming became possible in adhering countries. The development of 2G cellular systems was driven by the need for improving transmission quality, system capacity, and coverage. Speech transmission was dominating. However, the demand for fax, short message, and data transmission grew rapidly. Supplementary services such as fraud prevention and encryption of user data became standard features. Further evolutions of 2G, General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE) appeared. These are often referred to as 2.5G.

After many years during which 2G mobile systems and especially the GSM communication have been very successful yielding important growth rates, the cellular universe is changing. The first commercial release of the 3rd Generation (3G) systems known as Rel'99 or Rel'4 has been in operation since 2003, offering true broadband data. Multimedia messaging, video on demand, videophones and high bandwidth games are available, potentially leading to an explosion in throughput from a few bytes for the Short Message Service (SMS) to a few $kbps$ for the Multimedia Messaging Service (MMS), to several $100kbps$ for video content. 3G systems differ from the 2G in terms of both the bandwidth and data capabilities they provide. With this new generation of mobile communication, users are now able to access the Internet from mobile terminals and use a large variety of added value services. The key feature of 3G mobile systems such as UMTS is the ability to deliver wideband and multimedia services alongside the traditional radio services (mobile and fixed) including voice, messaging and slow rate data. In this context, UMTS is one of the candidates to provide the future mass market with wideband mobile and fixed multimedia services. UMTS offers significant user benefits including high quality multimedia, innovative services and applications.

At the heart of 3G lays the promise to connect users anytime and anywhere, providing the same experience as by fixed broadband at home and at work. However, current 3G deployments are still far from achieving this objective, both in throughput and capacity. 3G is enhanced by improving current technologies in order to deploy the well known Rel'5 (High Speed Downlink Packet Access (HSDPA)) and Rel'6 (High Speed Uplink Packet Access (HSUPA)) UMTS more generically called Beyond 3rd Generation (B3G) systems. Regarding radio access, evolved 3G means higher transmission rates and a much lower cost per bit in order to foster the development of an increasing number of bandwidth-hungry services and applications.

Future steps towards B3G systems will improve coverage in highly populated areas in order to carry more traffic by designing a new air interface. B3G systems will also create a convergence between diverse access technologies in order to deliver the best possible services to subscribers, taking into account both cost and bandwidth efficiency. Rather than targeting multi-megabit/s capability per user, the objective of B3G will be to allow users to connect through a variety of devices anywhere and at a wide range of speed.

A major issue is the timing for the development of evolved 3G and B3G systems. Previous experience with 2G and 3G indicates that standards development for each new generation system technology takes about a decade, and that a further decade is required before the technology reaches commercial maturity. Therefore, as 3G Rel'5 systems are already deployed in many countries, the time has come to start preparing the future. In the short-term (up to about 2010), the future development of 3G builds upon enhancing the capabilities of the initial deployment. It is also believed that the interfaces for B3G systems will exploit the new frequency spectrum that will be identified soon [1]. A speculative time scale for a mature B3G system is expected between 2010 and 2015.

Further evolutions have already been started. UMTS Long Term Evolution/System Architecture Evolution (LTE/SAE) is an evolution of the B3G radio interface based on Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input Multi-

ple Output (MIMO) technologies. This will provide much higher
data rates (over $100Mbps$) while reducing cost-per-bit. A reduc-
tion in latency will enhance the behaviour of time-sensitive ap-
plications, thus improving the user experience. To meet these
requirements, a reduction in the number of network nodes is
needed. One of these new nodes is the enhanced base sta-
tion, the so-called enhanced NodeB (eNodeB) presented in the
3rd Generation Partnership Project (3GPP) standards. It pro-
vides the LTE/SAE air interface and performs Radio Resource
Management (RRM) for the evolved access system. It is a way
of integrating UMTS Radio Network Controller (RNC) and Wide-
band CDMA (WCDMA) base station (NodeB).

At the same time, wireless technologies are also evolving and
compete with the 3GPP standard suites. That is true for the
$802.16e$ standard widely know as Mobile Worldwide interoperabil-
ity for Microwave Access (WiMAX). This new technology comes
from the wireless world whereas 3G, B3G, LTE, etc. are evo-
lutions coming from the cellular world. Since the range of the
wireless access points is increasing these technologies are now
competitors, while users are able to access more services by cel-
lular phone. A kind of war is taking place; future will show
which approach will prevail: Cellular communication networks
initially designed for voice transmission and reshaped to offer
broadband access to digital networks, or wireless networks de-
signed for data transfers with successfully mastered Voice over
IP (VoIP) technologies.
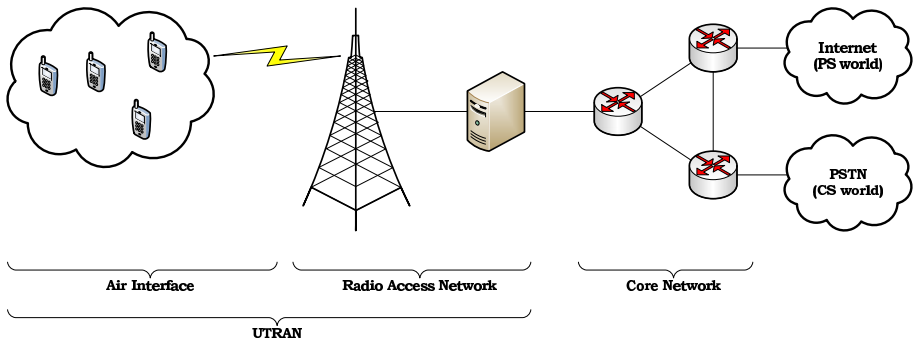
Figure 1.2: Third generation network overview.

The focus of this dissertation is the Rel'99 UMTS and, more precisely, the UMTS Terrestrial RAN (UTRAN) part of the UMTS network.

UMTS systems can be roughly divided into three (network) parts: the air interface, the Radio Access Network (RAN), and the Core Network (CN) as shown in Fig. 1.2. The air interface is the technology of the wireless link located between the User Equipment (UE) and the NodeB. The CN connects the operator switches and routers and extends to a gateway giving access to the wider Internet or Public Switched Telephone Network (PSTN). Finally, the RAN is the 'glue' that links the CN to the NodeBs and deals with most of the consequences of the terminal's mobility.

As will be explained in Chapter 9, the Rel'99 UMTS air interface is only the first step of this research. It was chosen because it was the only flavour of the 3G technology available at the beginning of this project in 2002. Adaptations and evolutions are however possible and could be implemented to support HSDPA, Mobile WiMAX, and even LTE/SAE.

The present dissertation targets readers with an advanced knowledge of UMTS and more precisely UTRAN. References [2], [3] and [4] give a comprehensive overview to readers lacking such prerequisite knowledge.

1.2 Motivation

Different methodologies are available to develop and test new wireless networks. Those methodologies can be classified in four different tracks.

The first track uses analytical developments. With the help of network calculus tools, statistical modelling and finite state machines, researchers aim at modelling the behaviour of computer networks.

The second track is the simulation type. Communication networks have become so complex and intricate that no one can claim to have understood it completely with the sole help of analytical studies. For instance, protocols designed to cope with congestion in wired networks are likely to react in an unex-

pected way when coping with phenomena triggered by wireless communications. Time- and event-driven network simulators implemented in software such as ns-2 [5] or Opnet [6] are valuable tools. They can figure out a number of issues to be solved before even planning the real deployment.

The third track is the measurement campaigns on a testbed. Network simulators also have their limitations. With increasing size and complexity network simulations become more difficult to implement and require even more computing time (scalability issue). Testbeds can be considered as a complementary tool in complex world simulations. They are indeed cheaper than a real deployment. Moreover, they emulate real network behaviour in a closer approximation than what a simulator would do in certain cases such as hardware limitations or delay computations. Besides being able to emulate in detail a larger network, another advantage of testbeds is their capability to reproduce the behaviour of a single user with more accuracy than simulators which are typically more suitable for global systems performance analysis. The testbeds are seen as a good tradeoff between computer simulations and field tests.

The last track combines all the field tests. Even if the testbed enables an approximation of the real network behaviour, at a certain point, it bases on assumptions which could influence and bias the conclusion made. On the other hand, field tests results are of course extremely valuable and directly applicable without any post-processing (results applicability). However, the major disadvantage of this approach is the cost as well as the irreproducibility of the test environment when confronting different results (reproducibility issue).

Table 1.1: Four tracks comparison.

	Analytical	**Simulation**	**Testbed**	**Field**
Scalability	−	−	+	+
Reproducibility	+	+	+	−
Cost	+	+	+	−
Applicability	−	−	+	+

The advantages and inconveniences of each of these tracks are summarised in Table 1.1. However, one should not rely on only one of the above described tracks in order to study a network protocol, architecture or a Quality of Service (QoS) management technique. Instead, parallel progressing and cross-validation such as presented in Sections 5.6, 7.5 and 8.3 is recommended (Fig. 1.3). Each track can serve as a benchmark for the validation of results obtained by the other tracks. This procedure will yield cross-validated results.
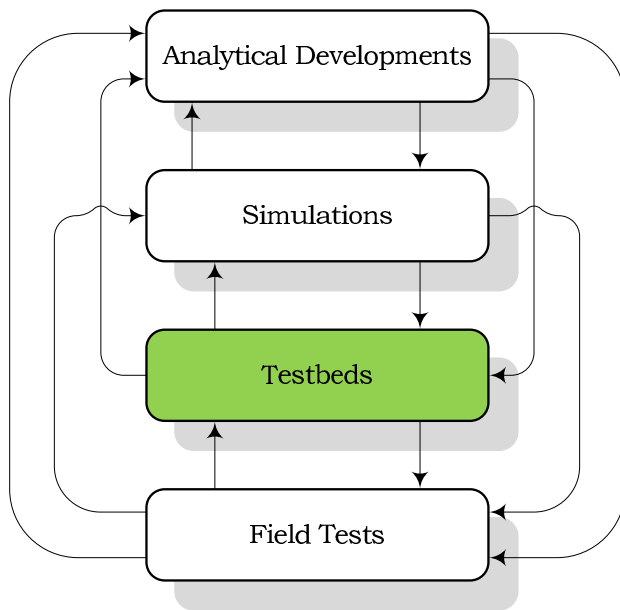
Figure 1.3: Four tracks interaction.

The present dissertation mainly describes a third track approach, i.e. testbeding, while using the other tracks to cross-validate the obtained results. This testbed has been designed so as to emulate a Rel'99 UMTS access network.

The objective is to describe the design, constitution and exploitation of a modular Linux-based testbed of a Rel'99 UTRAN. The platform is able to emulate advanced UMTS capabilities (such as scheduling, Call Admission Control (CAC) or RRM) in a Real-Time (RT) framework whereas simulators are mainly time- or event-driven. This implies the possibility of testing multi-

media applications (video conference, streaming services, web browsing, etc.) over an emulated UMTS network with enhanced RRM features. Moreover, the impact of these RRM algorithms on system performance can also be emulated in different scenarios. Such scenarios may include different traffic load conditions, propagation characteristics, and user mobility patterns.

Besides the common quantitative QoS metrics, such as jitter, delay, packet loss rate, etc., the ultimate measure of a network and its services is how subscribers perceive network performance. Quality of Experience (QoE) is the term coined to describe this perception and how satisfying and user friendly subscribers think the services are. With the growth of mobile services, it has become very important for an operator to accurately measure the QoE perceived over its network and further improve it in order to fulfil customer requirements and maintain a competitive edge. Even if the QoS metrics are above a certain threshold, the resulting QoE can be poor, leading to the dissatisfaction of customers, high churn rates, poor market perception, and ultimately, brand dilution. QoE is therefore seen as an extremely important parameter for the operators.

Due to its structure and its RT conception, the present testbed is able to test End-to-End (E2E) QoS performance and to assess the effects that RRM algorithms have on the QoE perceived by the end user. This is achieved by introducing real life application sessions during the emulation.

This testbed is furthermore able to assess true and representative network-based QoS metrics (packet delay, loss rate, cell throughput) as well as the QoE of a single user in many different scenarios (mobility, traffic load, traffic type, etc.). Due to these advantages, it could be of interest for:

- operators (Vodafone, Orange, Telecom Italia Mobile (TIM), Hutchinson, etc.) who could, on a short-term basis, fine tune their scheduling, RRM and CAC algorithms based on specific scenarios, and

- infrastructure manufacturers (Ericsson, Nokia, Alcatel-Lucent, etc.) who could, on a long term basis, benefit from the testbed's emulations to elaborate new algorithms.

1.3 Thesis Objectives

The main objective of this thesis is to demonstrate through comparison of results from the open literature, that a testbed consisting of a dozen personal computers and open-source software is able to realistically emulate the UTRAN.

Based on this testbed infrastructure, the second objective of this project is to improve RRM, CAC and/or scheduling algorithms in order to fulfil the QoS requirements for upcoming applications and services, and the QoE of users.

1.4 Structure of the Dissertation

The present dissertation is structured as follows:

Part II presents the testbed as a tool and comprises four main chapters. This perfectly matches the modular structure of the testbed. Features from a given UTRAN layer can therefore be found in several chapters. This is particularly true for topics addressing the air interface.

Chapter 2 presents the state of the art of the existing testbed emulating a UMTS network. It also shows the dissertation testbed's perspective, characteristics, assumptions, limitations and differences compared to other testbeds in the same research field.

A generic three-level traffic model created for the purpose of the present dissertation is described in **Chapter 3** in order to generate the four UMTS traffic classes proposed by the 3GPP. Based on this multilevel structure, the chapter shows how the four traffic classes are modelled in terms of stochastic distributions and injected in the testbed.

Chapter 4 depicts the world representation, explains how the testbed emulates user mobility above a wired network infrastructure and describes the testbed handover management.

Chapter 5 shows a methodology of emulating a wireless link over a wired one, based on the Linux Traffic Controller (TC) and some of its components. Based on this implementation, the chapter explains how the testbed emulates the Rel'99 UMTS air

interface, the different transport channels, the higher Bit Error Rate (BER) and its time-correlated errors property.

Part III covers the major achievements of the testbed's operation. It provides results that mostly illustrate the capabilities of the testbed. Many more experiments could have been designed and performed on the testbed for further cross-validation. These chapter presents only three case studies, as a proof of concept of the testbed.

Chapter 6 aims at finding an optimal Spreading Factor (SF) allocation strategy. It confronts the two opposite approaches of this issue: The users' perspective trying to receive the lower SF in order to get the higher possible bandwidth, and the operators' perspective trying to load as much as possible the Orthogonal Variable Spreading Factor (OVSF) tree in order to accept a maximum number of sessions within the same sector.

Chapter 7 evaluates the advantages of a scheme based on an optimal Non Real-Time (NRT) session switching between the shared and the Dedicated CHannels (DCHs). This chapter also analyses the advantage of an improved NRT sessions channel switching strategy based on a division of the downswitch timer.

Chapter 8 measures the impact of design choices at transport, data link and physical layers on the user experience at the reception of RT flows.

In **Part IV**, conclusions and outlook on possible future scenarios are presented.

Finally, **Part V** includes the scientific contributions which have been peer-reviewed and accepted for oral presentation in international conferences.

1.5   Publications

This project has lead to 8 publications among which 5 have been peer-reviewed and accepted for oral presentation in international conferences. The reader can find these major contribution in the Appendix A.

"Description of an IPv6 Linux-Based UTRAN Testbed" emphasises the fact that network modelling and traffic characterisation are essential research activities towards the optimisation of UMTS. It presents the thesis testbed, the statistical traffic distributions, the random walk UE mobility and line-of-sight power-based handover management. It explains also the interconnection of those different modules to obtain the first testbed results about the handover and QoS management.

This article has been presented at TridentCom 2006. This conference brings together all aspects related to experimental telecommunication infrastructures, creating a forum where telecommunication networks researchers, vendors, providers and users can exchange ideas on past experience, requirements, needs, visions for the establishment of such infrastructures [7]. It received the "Best Paper Award" presented in the Appendix B.

"Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS" derives a SF allocation map for the four UMTS traffic classes and for four user profiles with the help of the testbed. The map is based on the E2E delay experienced by the user. Using this allocation map, this article measures the performance of the UTRAN in terms of cell throughput and blocking rate in different scenarios.

This article has been presented at VTC-fall 2006. This conference aims at bringing together individuals from academia, government, and industry, to discuss and exchange ideas in the fields of wireless and vehicular technology. The conference usually features world-class plenary speakers, tutorials, and technical as well as applications sessions [8].

"UMTS Non Real-Time Sessions Channel Switching Emulation" discusses a scheme to improve the Rel'99 UMTS RRM. This article is focused on the two different types of transport channel: shared and dedicated. It evaluates the merits of a scheme based on an optimal NRT session switching between these two kinds of channels. The switching decision may be influenced by several parameters like the downswitch timer's duration, the non-/preemptive split of downswitch timer and the selection of the initial transport channel. Performances of the analysed scheme

are evaluated taking into account different metrics: blocking rate, mean packet delay and cell throughput. The results show that an optimal tuning of these three parameters may significantly decrease the number of blocked calls while keeping the other metrics at a satisfactory level.

This article has been presented at VTC-spring 2007 which is the spring counterpart of the VTC-fall conferences.

"Performance of Downlink Shared Channels in UMTS Scenarios Using Markov Fluid Process" proposes a new model of the Downlink Shared CHannel (DSCH) buffer in Rel'99 UMTS cellular systems. Considering that the available bandwidth dedicated to DSCH traffic is limited by voice connections carried by DCHs on the one hand, and by traffic on the Forward Access CHannel (FACH) on the other hand, it proposes to model the DSCH buffer content as a Markov driven fluid queue with jumps and to derive several performance measures, namely stationary DSCH buffer content, bit loss probability and overflow probability. These measures are obtained through adequate transformation of the original fluid queue.

This article has been presented at ASMTA 2007. This conference is a continuation of the series started as the Conferences in Analytical, Numerical and Stochastic Modelling Techniques held as a part of European Conference for Modelling and Simulation. It features an outstanding technical program ranging from introductory tutorials to software reviews to state-of-the-art research and practice. The conference includes also exhibits, business meetings for professional societies, software user groups [9].

"UMTS Layers Parametrisation for Real-Time Flows" demonstrates the impact of design choices at transport (Transport Control Protocol (TCP) vs. User Datagram Protocol (UDP)), data link - RLC (Acknowledged Mode (AM) vs. Transparent Mode (TM)) and physical (SF 4 vs. 8) layers on the user experience at the reception of RT flows for both synthetic and real life Conversational and Streaming traffics. Over an impaired wireless channel, it appears that TM is not suited for Streaming flows, as it results in significant Packet Error Rate (PER) on UDP or significant delays on TCP, whereas it is acceptable for Conversational flows.

This article has been presented at MobConQoE 2007. This conference address all the technical aspects of implementing a comprehensive process for content provisioning, optimisation and monitoring of the delivered quality, with the scope of increasing mobile subscribers' experience [10].

Bibliography

[1] ITU-R. World radiocommunication conference 2007. Geneva (Switzerland), `http://www.itu.int/ITU-R/`, last visited: March 13, 2008.

[2] Harri Holma and Antti Toskala. *WCDMA for UMTS*. Wiley inter-science, 3rd edition, October 2004.

[3] Harri Holma and Antti Toskala. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. Wiley inter-science, April 2006.

[4] Jaana Laiho, Achim Wacker, and Tomáš Novosad. *Radio Network Planning and Optimisation for UMTS*. Wiley inter-science, second edition, 2006.

[5] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in Network Simulation. *IEEE Computer*, 33(5):59–67, May 2000.

[6] Xinjie Chang. Network simulations with OPNET. In *Proceedings of the 31st Winter Simulation Conference (WSC)*, pages 307–314, Phoenix (AZ -USA), 1999.

[7] TridentCom 2006. 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities. Barcelona (Spain), `http://tridentcom.org/tridentcom06/`, last visited: March 13, 2008.

[8] VTC fall 2006. 64th Vehicular Technology Conference. Montreal (Canada), `http://www.ieeevtc.org/vtc2006fall/`, last visited: March 13, 2008.

[9] ASMTA 2007. 14th International Conference Analytical and Stochastic Modelling Techniques and Application. Prague (Czech Republic), `http://www.comp.glam.ac.uk/ASMTA2007/index.html`, last visited: March 13, 2008.

[10] MobConQoE 2007. 1st International Workshop on Mobile Content Quality of Experience. Vancouver (Canada), `http://www.mobconqoe.org/`, last visited: March 13, 2008.

Part II

Tool

Chapter 2

Testbed Presentation

2.1 Introduction

This chapter first introduces a number of different UMTS testbeds, their specificities and characteristics. It then presents the thesis testbed's perspective, characteristics, differences and new features compared to these testbeds. Finally, assumptions which have been set by the testbed implementation are listed in three different limitations groups.

2.2 Testbed State of the Art

The idea of using a testbed to cross-validate simulation experiments or analytical results as illustrated in Section 1.2 is not new. Several other projects are already using this strategy for UMTS studies and report good results. Here follows a brief presentation of some of the most well-known testbed experiments emulating a 3G cellular network with a special focus on the management of Radio Link Control (RLC), Internet Protocol (IP) and transport layers.

2.2.1 Arrows

The first testbed emulating a UMTS network addressed here is the Information Society Technologies (IST) project Arrows [1].

Arrows is designed to emulate as faithfully as possible a real-istic 3G UMTS system consisting of one UE, the UTRAN, the CN and one application server. This testbed aims at providing advanced RRM and QoS solutions for the support of integrated voice and data services within the UTRAN. Different algorithms related to CAC, congestion control as well as packet schedul-ing procedures are proposed and evaluated for both uplink and downlink paths. This project includes packet access, asymmet-rical traffic and high bit-rate (up to $2Mbit/s$) services for multi-media IP based applications.

A new extension of this project is an integrated UMTS-Wire-less Local Area Network (WLAN) testbed [2] which is a platform including multimedia terminals, UMTS and WLAN elements and IP connectivity. The objective of this testbed is to help manag-ing radio resources cooperatively across multiple radio access technologies in a coordinated manner.

2.2.2   Novembre

The Novembre testbed [3] unites several wireless studies with a special focus on the transport of RT voice traffic over the UTRAN using IP (VoIP). Novembre tried to limit as much as pos-sible the VoIP E2E delay by encapsulating compressed, low bit rate voice traffic arriving at the NodeB into frames of variable size. These frames were then multiplexed into larger, possibly variable in size, IP packets. This aggregation took place in the NodeB and a timer was associated with this component in order to avoid unacceptable large packetization delays. This work is based on the Adaptive MultiRate (AMR) codecs at $12.2kbps$ with silent suppression. Over the radio interface, the transport of radio bearers takes place within DCH.

2.2.3   Net Institute and Fokus

Other projects gather telecommunication partners who grant an access to their wireless infrastructure. In this case, the test-bed is in fact a simple reduction of a real UMTS network. The NET Institute project [4] has such a great opportunity partnered by the operator T-Mobile and the manufacturer Nokia. The main

research focus of this project is on security improvements over the air interface of the UMTS.

The Fokus project [5] has a similar design since this testbed is a part of T-Mobile's technical UMTS infrastructure in Berlin. Testbed users have the opportunity to experiment technology and applications in an integrated network environment ranging from Bluetooth, WLAN 802.11a/802.11b, GSM, GPRS, UMTS Time Division Duplex (TDD) and Frequency Division Duplex (FDD) right up to Satellite. The Fokus test and development environment is mainly for client prospects from industrial and public administration communities.

The flaw of this kind of testbeds is that research activities are highly manufacturer-dependent. They should therefore be considered as commercial showcase networks.

2.2.4   Mobiq

The Mobiq project [6] was focused on multimedia services including VoIP and audio/video streaming. Its key components were compression of Real-time Transport Protocol (RTP)/UDP/IP headers for efficient media packet delivery on radio links, layer2 combined queueing, which integrated wireless layer2 functionality into an IP-layer queueing model, and QoS control utilising feedback information from a network agent called "RTP monitoring agent" in order to counter both, network congestion and radio link errors. These components enhance the quality of multimedia delivery services in 3G and future mobile networks.

2.2.5   Whynet

The Whynet project [7] was an hybrid testbed which gathered many wireless standards and worked on their interoperability. The objective of this project was to develop a scalable testbed in order to evaluate the impact of next-generation wireless technologies on application performance in heterogeneous wireless networking scenarios, while realistically accounting for various cross-layer interactions. Whynet has provided a number of hybrid experimentation modes with seamlessly integration of

physical, simulation and emulation components in order to address issues related to vertical handovers.

2.2.6 Respect

The Respect testbed [8] investigates the performance of packet services in both a standalone UMTS network and a UMTS network coexisting with other networks. The studies deal with the performance of a single link. Respect is a RT emulation platform that uses a combination of simulated and real network protocols enabling both subjective and objective investigations from the user's perspective. It shall give a holistic view of packet service performance by considering the impact of different protocol layers while taking the end user into account.

2.2.7 Virtuous, Future and ATB

Other projects have built a testbed in order to define, validate and demonstrate the adaptations of 3G mobile technologies based upon WCDMA for supporting satellite-based services. Such services are often generically referred to as Satellite-UMTS (S-UMTS). IST Virtuous [9] and Future [10] projects, and the European Space Agency (ESA) Advanced S-UMTS TestBed (ATB) project [11] are examples of such S-UMTS testbed.

2.2.8 Unite

The Unite project [12] encompasses the common features of cellular, wireless and Digital Video Broadcasting (DVB) in order to provide a scalable and extensible testbed investigating cross-layer and cross-system optimisation. The main objective of this project is the investigation, the design and the evaluation of cross-layer and cross-system interactions for current and future generations of wireless systems in terms of both, radio and protocols.

Table 2.1: Testbeds comparison.

| Testbeds | Main Research topics | | | | | | | | |
| | Traffic Classes | | | | | | Others | | |
	Conver-sational	Stream-ing	Inter-active	Back-ground	Vert. Hand.	Horiz. Hand.	Secu-rity	Multi-user	Satel-lite
Arrows		✓	✓			✓			
Novembre	✓								
NET Institute						✓	✓	✓	
Fokus								✓	✓
Mobiq	✓	✓							
Whynet	✓				✓	✓		✓	
Respect		✓	✓		✓				
Virtuous					✓				✓
Future	✓		✓	✓					✓
ATB		✓	✓		✓		✓	✓	✓
Unite		✓					✓	✓	
Thesis testbed	✓	✓	✓	✓		✓		✓	

2.3   Thesis Testbed Architecture

The testbed created for the purpose of the present dissertation [13, 14] differs from the ones previously described both in scope and approach. The objective is to establish a suitable open-source software/hardware RT platform in order to thoroughly analyse and improve QoS management in the UTRAN. As shown in Table 2.1, most of the presented testbeds are focused on only one or two types of the typical UMTS traffic flows defined by the 3GPP (Table 3.1), such as Streaming or Conversational. In contrast, this testbed is not restricted to a single service but is open to all typical UMTS applications. It shall work on an acceptable QoS level for all these simultaneously active services e.g. a population of users within the same cell, some of them having standard conversations (Conversational), others browsing the Internet (Interactive) or retrieving their Emails (Background), while a few more users arrive and initiate a video streaming session (Streaming).

As shown on the upper part of Fig. 2.1, the testbed emulates a Rel'99 UTRAN segment consisting of one RNC managing four NodeBs, each of them serving a population of UEs. The NodeBs are placed on a hexagonal grid being the standard macrocellular set-up [15, 16]. For representation simplicity, the four cells are represented on a rectangle map.

In order to emulate this UTRAN segment, the testbed is composed of nine Linux-operated personal computers interconnected with $100\ Mbit/s$ Ethernet links thus creating a little network isolated from the Internet, as shown on the lower part of Fig. 2.1 and in Fig. 2.2.

The network can be subdivided in three computer groups ($\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$). Group $\mathcal{A}$ comprises second-hand Pentiums III $750MHz$ with $512MB$ of Random Access Memory (RAM). They all have four network interfaces connected to various IP version 6 (IPv6) sub-networks. Group $\mathcal{B}$ is composed of Celerons $300MHz$ with $128MB$ of RAM. Each computer is connected to one of the computers of group $\mathcal{A}$. Group $\mathcal{C}$ is a single computer. It has the same hardware specification as the group $\mathcal{A}$ computers and is directly connected to each of the group $\mathcal{A}$ machines.
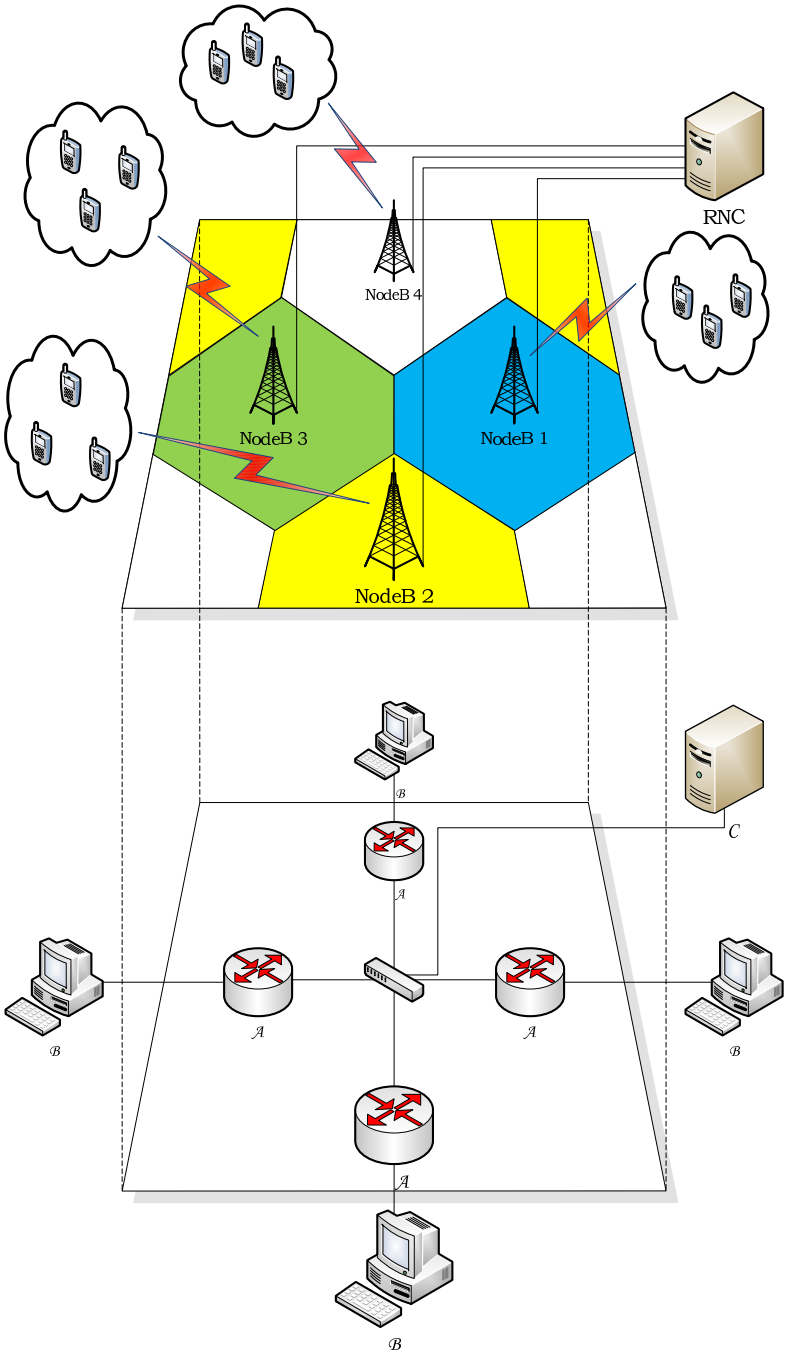
Figure 2.1: UTRAN testbed representation.
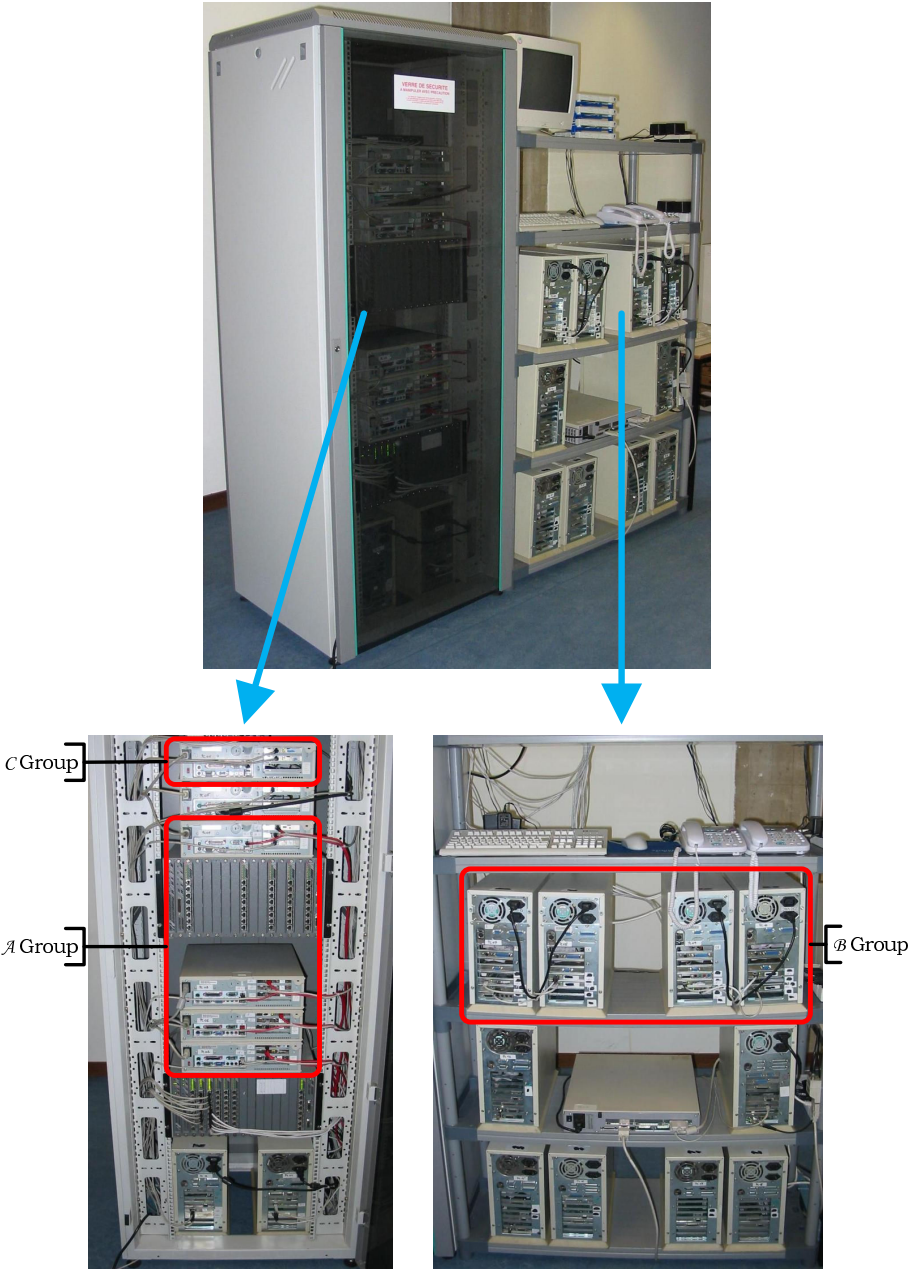
Figure 2.2: The overall testbed.

In order to ensure homogeneity, a Linux Red-Hat-based Fedora Core 6, kernel 2.6.20 has been installed on each of the nine computers. Flexibility, cost vs. performance and the ability to customise the operating system in order to be able to accurately tune its network behaviour and its performance motivated the choice of this operating system.

From the implementation perspective, the present testbed is fully developed in the C language. It can be seen as an assembly of numerous modules. A module is a specialised code part managing a predefined function for all the UEs active during an emulation run. Modules tasks can be as different as traffic generation, mobility evolution, CAC negotiation or handover management. A detailed presentation of each of them can be found in the testbed documentation [17].

Group $\mathcal{A}$ is placed in the centre of the network. These computers run the freeware Quagga [18] and are used as routers. Quagga is a routing software suite providing implementations of multiple routing protocols (Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Border Gateway Protocol (BGP), etc.) for Unix platforms. The computers of groups $\mathcal{B}$ and $\mathcal{C}$ are placed in periphery and are used as traffic sources/sinks. To serve that purpose, the open-source application TG [19] (a traffic generator program able to create one-way UDP or TCP streams between two computers) has been installed. The traffic generation process is detailed in Chapter 3 in terms of inter-arrival times and packet lengths by means of statistical distributions.

Using this network infrastructure, the idea is to emulate a real UTRAN segment:

2.3.1 Group $\mathcal{A}$ - NodeBs

The computers in group $\mathcal{A}$ represent four independent trisectorial macrocell NodeBs. Their role is to take care of the traffic coming in from the UE (resp. RNC) and to forward it to the RNC (resp. UE). Most of the QoS management functions are performed at these spots as they are the bottlenecks of the UTRAN (border between the wireless and the wired networks). Since the network does not contain any real NodeB, these computers also

handle the emulation of the UMTS air interface (BER, Service
Data Unit (SDU) segmentation), etc.).

In the early UMTS standardisation the NodeBs were seen as
simple access points linking the UEs to the UTRAN and the CN.
With the introduction of HSDPA, they gained more responsibili-
ties in terms of QoS management and packet scheduling. There-
fore, in anticipation of a future evolution of the testbed, the com-
puters emulating the NodeBs are designed in a Rel'5 philosophy
while the testbed is still emulating a Rel'99 UTRAN.

2.3.2   Group $\mathcal{B}$ - UEs

Each computer of this group acts as a population of UEs
attached to their Serving NodeB (SNodeB) (emulated on the at-
tached $\mathcal{A}$ group computer). Each of them generates several up-
link flows (from the UE to the RNC) representing the emulated
UEs network activity.

In order to simultaneously emulate several autonomous UEs
on a single computer, these computers use a control panel based
on virtual network interfaces management (Fig. 2.3). A physical
network interface can have multiple IP addresses assigned to it.
This could be done using the Linux virtual IP addressing. In Fe-
dora all information about the network interfaces is kept in the
following directory: `/etc/sysconfig/network-scripts/`. As-
suming that the `eth1` network interface configuration script is:
`/etc/sysconfig/network-scripts/ifcfg-eth1`, it looks like
this:

```
# Interface
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
TYPE=Ethernet
HWADDR=00:04:75:F5:02:6C
USERCTL=no
# IPv4 Configuration
NETWORKING=yes
IPADDR=10.0.3.1
NETMASK=255.255.255.0
NETWORK=10.0.3.0
BROADCAST=10.0.3.255
# IPv6 Configuration
```

```
IPV6INIT=yes
IPV6ADDR=2005:0:0:3::1/64
IPV6_AUTOCONF=no
IPV6_ROUTER=yes
```

In order to allocate another IP address to this interface, one simply has to make a copy in the same directory naming the new file `ifcfg-eth1:1` and adapting the `DEVICE`, `IPADDR` and `IPV6ADDR` fields:

```
# Interface
DEVICE=eth1:1
BOOTPROTO=static
ONBOOT=yes
TYPE=Ethernet
HWADDR=00:04:75:F5:02:6C
USERCTL=no
# IPv4 Configuration
NETWORKING=yes
IPADDR=10.0.3.2
NETMASK=255.255.255.0
NETWORK=10.0.3.0
BROADCAST=10.0.3.255
# IPv6 Configuration
IPV6INIT=yes
IPV6ADDR=2005:0:0:3::2/64
IPV6_AUTOCONF=no
IPV6_ROUTER=yes
```

`eth1:1` becomes an alias of the `eth1` physical network interface. Other aliases could be named `eth1:2`, `eth1:3`, etc. The role of this control panel is to facilitate the UEs management by delegating to them a single IPv6 address and a range of UDP/TCP ports.
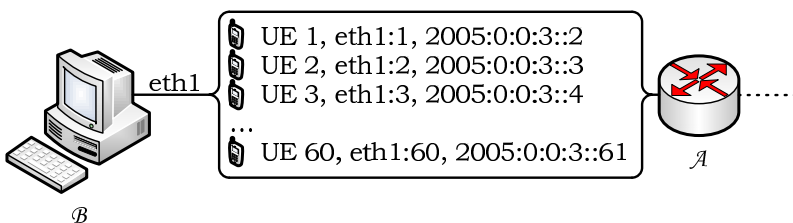
Figure 2.3: UE control panel.

2.3.3   Group $\mathcal{C}$ - RNC

This single computer plays the role of the testbed's unique RNC. It is the sink of the traffic coming from the four NodeBs and generates the downlink traffic flows (from the RNC to the UEs) for each of the emulated UEs. As in a real UTRAN, the complexity is gathered in the RNC since it is in charge of the CAC and RRM. It consequently controls the entire emulation.

2.4   Testbed Limitations

When emulating such a complex network as a UTRAN over a computer network, testbed limitations need to be taken into account. The following section describes three different types of limitations.

2.4.1   Computing Power

A major limitation is performance. Since the testbed was built from second-hand computers with a limited computing power and RAM, the complexity of the emulated UTRAN had to be adapted. This led to several further restrictions:

The NodeBs can serve only a fixed maximal number of 60 UEs due to the low computing power of the group $\mathcal{B}$ computers. However, this number, fixed at the beginning of an emulation run, is important enough to be realistic. The different UEs are switched on during the entire emulation. This means that the testbed does not have any UE birth-death mechanism. However, the UEs activity is controlled by stochastic models which will be presented in Chapter 3. As a result, during an emulation with e.g. 12 users, one may encounter periods of inactivity as well as full load periods when all 12 users are active. This is illustrated in Fig. 2.4 which shows the aggregate downlink traffic of a given NodeB during an emulation of $1,200s$ with 12 users. Up to three users are simultaneously streaming around the 600th second whereas there was no traffic at all a few seconds before.

Due to a lack of computing power, an adaptation to be made is the load balancing of this scarce resource. In a real Rel'99
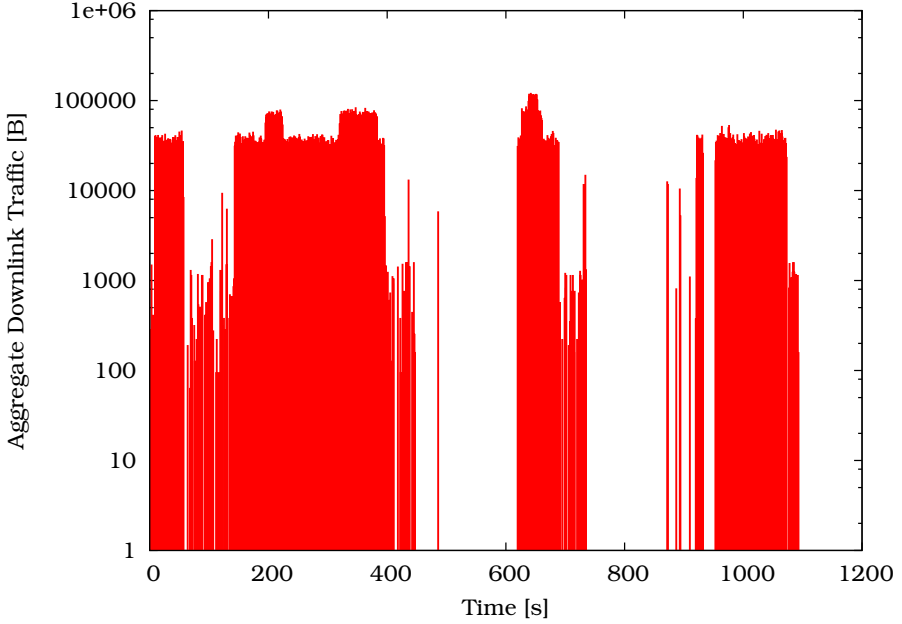
Figure 2.4: NodeB downlink traffic.

UTRAN, all the complexity is gathered in the RNC while the NodeB does not have much decision power. As already mentioned in Section 2.3.1, the testbed is designed in a Rel'5 way in spite of the use of Rel'99 transport channels. This means the NodeBs have much more capabilities in terms of CAC, RRM and QoS management which spare the RNC resources.

On the UE side, the low Central Processing Unit (CPU) frequency of the testbed computers induces to a greater action granularity. This means that a UE can move, switch transport channel, start or stop a session, etc. only once per second. Actually, this granularity is as low as $10ms$ in the Rel'99 and even only $2ms$ in Rel'5. This bigger testbed granularity does not cause increased latency or delay while transmitting packet over the emulated air interface. As discussed in Section 5.4.4, the Transmit Time Interval (TTI) length is left unchanged and is set to $10ms$ as the testbed emulates a Rel'99 UMTS network.

The emulation main loop has to manage all UEs one by one as shown in Fig. 2.5. This solution was imperative since a thread parallelism was impossible due to the lack of RAM and a limitation of the operating system which does not accept more than $1,024$ Portable Operating System Interface on uniX (POSIX) threads on the same computer (where a UE needs as many as 12 threads to manage all its complexity).
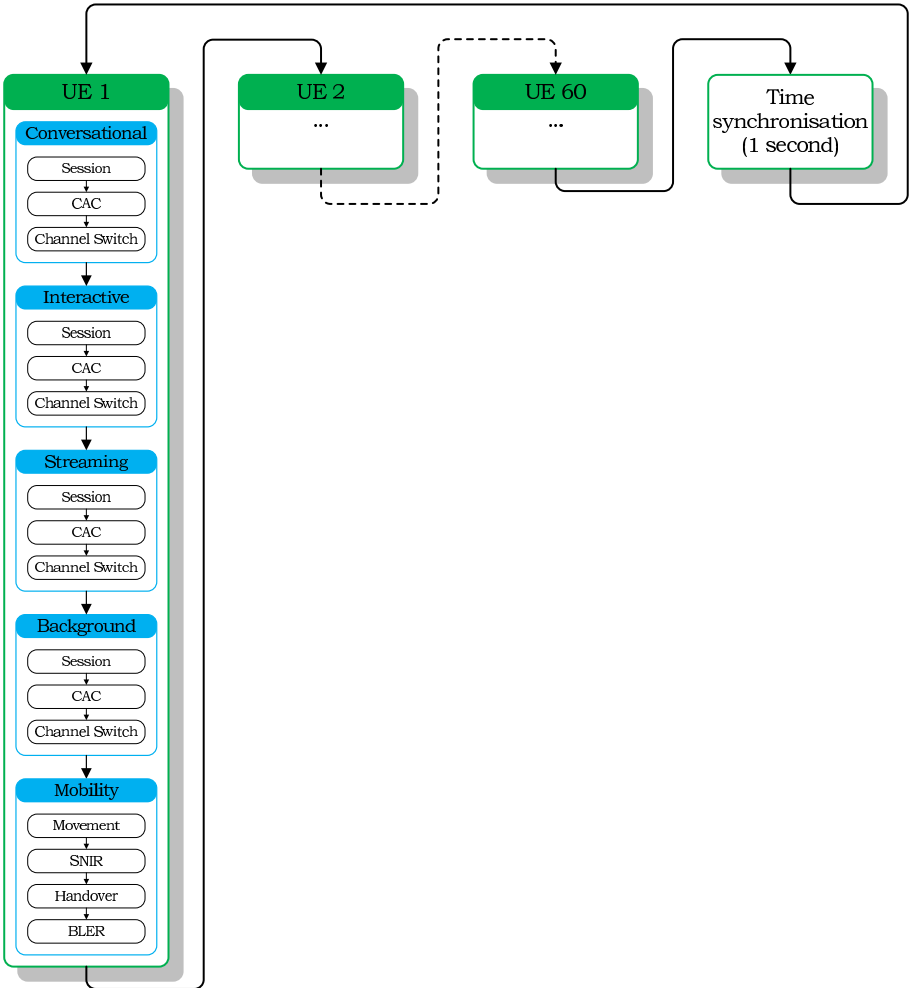
Figure 2.5: Main loop illustration of a $n - user$ emulation run ($n \leq 60$).

In the testbed setting, a single UE treatment is divided into five different phases. The first four phases deal with each of the traffic classes (session starts/ends, channel switching, CAC management, etc.), whereas the last phase manages the UE mobility (handover, Signal-to-Noise plus Interference Ratio (SNIR) evolution, BLock Error Rate (BLER) adaptation, etc.). Once all the five phases are concluded for one UE, the script picks the following UE and so on until all users have been served. The loop then starts again after a synchronisation pause to allow for alignment to the second tick. To ensure time synchronisation between the 9 computers of the testbed, besides emulating the RNC, the computer $\mathcal{C}$ assumes the role of a Network Time Protocol (NTP) server on which all other computers are lined up.

2.4.2 Air Interface

The second major limitation of the testbed is the absence of radio transmissions. To overcome this issue, the lower layers of the UMTS air interface were emulated over a classical Ethernet (wired) link as shown in Fig. 2.6.
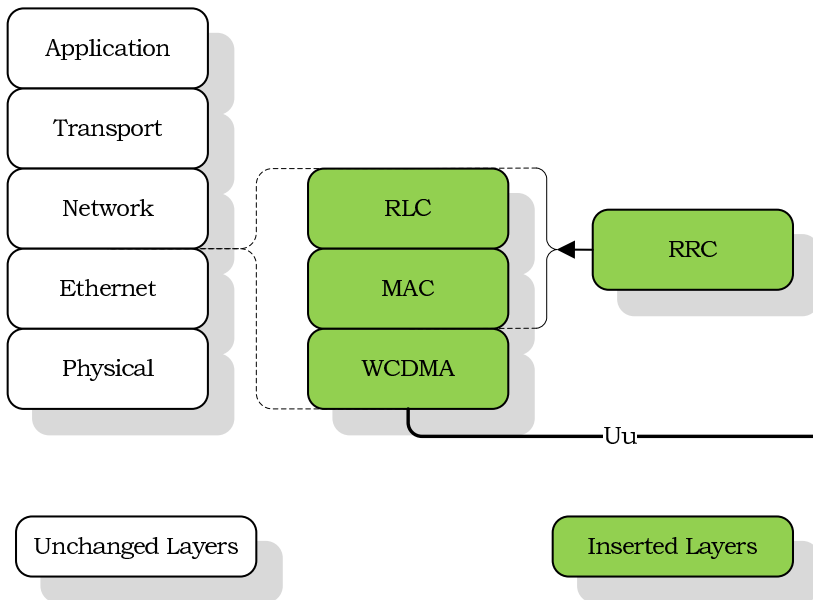
Figure 2.6: UMTS lower layer emulation.

Pondering on out testbed's limitations, it appears that the RAN Round-Trip Time (RTT) (computed between the RNC and the UE) is much more important in a real UTRAN since the packets are evolving on a wired network. In order to be as close to reality as possible, the testbed includes a fixed delay to each packet transiting over the emulated layers. This delay has been fixed to $50.2ms$ following the 3GPP specifications [20].

An other assumption which had been set due to the air interface emulation is the DCH allocation time. This is the delay a UE undergoes between its DCH request and the time point it transmits data through it. This allocation time mainly contains the CAC procedure made by the RNC to allocate a SF to the UE. This delay duration has been fixed to $900ms$ as proposed in [21].

2.4.3 Downlink-Oriented

A typical limitation of the testbed is that it is focused on the downlink. The major part of the data is transported in this direction. Therefore, the air interface emulation influences only on this direction. For example, considering an Interactive session, the testbed affects, with a given BER and bandwidth restriction, the TCP segments used to download a web page, but the TCP ACKnowledgement (ACK) segments returned by the UE do not suffer any additional perturbation or bandwidth restriction.

2.5 Summary

Using testbeds to cross validate simulation results and to investigate new UMTS issues has already been adopted by several researcher teams. In this way of thought, this chapter has exposed the thesis testbed's perspective, characteristics, assumptions, limitations and differences compared to other testbeds. This tool aims at providing advices and hints to correctly tune RRM algorithms over a UTRAN network supporting multiple mobile users generating/receiving flows belonging to the four traffic classes. The underlying objective is of course to optimise the QoS perceived by the final users known as QoE.

Bibliography

[1] Ramon Ferrús, Anna Umbert, Xavier Revés, Ferran Casadevall, Ronan Skehill, and Ian Rice. ARROWS UMTS Test-bed Description. In *Proceedings of the* 11th *IST Mobile & Wireless Telecommunications Summit*, Thessaloniki (Greece), July 2002.

[2] William Kent, Ronan Skehill, Michael Barry, and Sean McGrath. An Integrated UMTS-WLAN Testbed. In *Proceedings of the* 3rd *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Orlando (FL - USA), May 2007.

[3] Abed Ellatif Samhat and Tijani Chahed. Transport in IP-based UMTS Radio Access Network: Analytical Study and Empirical Validation. In *Proceedings of the* 39th *IEEE International Conference on Communications (ICC)*, Paris (France), June 2004.

[4] Carleen Maitland. The Delft UMTS Testbed and End-user Security features. Technical report, NET Institute, Working Paper #03-09, October 2003.

[5] Thomas Magedanz, Dorota Witaszek, and Karsten Knüttel. Service Delivery Platform Options for Next Generation Networks within the national German 3G Beyond Testbed @ Fraunhofer FOKUS. In *Proceedings of South African Telecommunication Networks Architectures Conference (SATNAC)*, Stellenbosch (South Africa), September 2004.

[6] Takeshi Yoshimura, Tomoyuki Ohya, Hosei Matsuoka, and Minoru Etoh. Design and Implementation of Mobile QoS Testbed MOBIQ for Multimedia Delivery Services. In *Proceedings of the* 12th *International Packetvideo Workshop*, Pittsburgh (PA - USA), April 2002.

[7] Junlan Zhou, Zhengrong Ji, Maneesh Varshney, Zhiguo Xu, Yi Yang, Mahesh Marina, and Rajive Bagrodia. WHYNET: A Hybrid Testbed for Large-Scale, Heterogeneous and Adaptive Wireless Networks. Technical Report CSD-TR060002, University of California, Computer Science Department, Los Angeles (CA - USA), January 2006.

[8] Oumer Mohammed Teyeb. *Quality of Packet Services in UMTS and Heterogeneous Networks*. PhD thesis, Aalborg University, October 2006.

[9] Seoung-Hoon Oh, Vladimir Obradovic, Hanspeter Widmer, Massimo Muratore, Massimiliano Torregiani, Leonardo Varasconi,

Alois Huber, and Arnolodo Giralda. Physical Demonstrator Architecture for VIRTUOUS Experiments. In *Proceedings of the 10th IST Mobile Communications Summit*, Stiges (Spain), September 2001.

[10] Daniel Christian Schultz, Mirko Albani, Alois Huber, Lila Dimopoulou, and Arnoldo Giralda. IST FUTURE Project - Extended Service Capabilities in a Multi-Segment Environment. In *Proceedings of the 10th IST Mobile Communications Summit*, Stiges (Spain), September 2001.

[11] André Barbosa, Martin Buck, Giuseppe Chiassarini, Gennaro Gallinaro, Riccardo De Gaudenzi, Joseph Furrer, Andrea Julia Pérez-Carro, Antonio Vernucci, and Hanspeter Widmer. An Advanced Satellite UMTS Testbed for Laboratory and Over-the-air Experiments of Third Generation Mobile Service. In *Proceedings of the 24th AIAA International Communications Satellite Systems Conference (ICSSC) and 4th Annual International Satellite & Communications (ISCe) Conference and Expo*, San Diego (CA - USA), June 2006.

[12] Derek Babb, Luis de Campos, Nicolas Ibrahim, George Kormentzas, Sylvie Mayrargue, Christos Politis, Jorge Rodriguez, Atilio Gameiro, and Rahim Tafazolli. A Virtual Distributed Testbed for Optimisation and Coexistence of Heterogeneous Systems. In *Proceedings of the 15th IST Mobile & Wireless Telecommunications Summit*, Helsinki (Finland), November 2006.

[13] Hugues Van Peteghem and Laurent Schumacher. Description of an IPv6 Linux-Based UTRAN Testbed. In *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Barcelona (Spain), March 2006.

[14] Hugues Van Peteghem and Laurent Schumacher. Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based *Rel'*99 UTRAN Testbed. Technical Report TD(07)011, COST 2100, February 2007.

[15] Technical Specification Group Radio Access Network. Multiple Input Multiple Output in UTRA. Technical Report TR 25.876 V7.0.0, 3rd Generation Partnership Project (3GPP), March 2007.

[16] Technical Specification Group Radio Access Network. Feasibility study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement. Technical Report TR 25.892 V6.0.0, 3rd Generation Partnership Project (3GPP), June 2004.

[17] Hugues Van Peteghem. Description of an IPv6 Linux-Based UTRAN Testbed Reference Manual.

Technical Report 8.1, FUNDP, November 2007. `http://www.info.fundp.ac.be/~hvp/rech/doc/docs/Testbed.pdf`, last visited: March 13, 2008.

[18] Kunihiro Ishiguro. Quagga, A routing software package for TCP/IP networks. Technical report, July 2006. `http://www.quagga.net`, last visited: March 13, 2008.

[19] Paul McKenney, Dan Lee, and Barbara Denny. Traffic Generator Software Release Notes. Technical report, SRI International and USC/ISI Postel Center for Experimental Networking, January 2002. `http://www.postel.org/tg/tg.htm`, last visited: March 13, 2008.

[20] Technical Specification Group Radio Access Network. Delay Budget within the Access Stratum (Release 4). Technical Report TR 25.853 V4.0.0, 3rd Generation Partnership Project (3GPP), 2001.

[21] Harri Holma and Antti Toskala. *WCDMA for UMTS*. Wiley interscience, 3rd edition, October 2004.

Chapter 3

Traffic Generation

3.1   Introduction

3G mobile communications systems offer high-speed mobile access to a great variety of services world-wide. Some of these services have certain E2E QoS requirements to be met by the network in order to function properly (e.g. in multimedia applications). In UMTS, the 3GPP has decided not to standardise a closed group of services. Instead, an open QoS architecture was specified, including the definition of four QoS traffic classes and a group of QoS parameters (tolerance to delays and losses among them). The four different UMTS traffic classes are: Conversational, Streaming, Interactive and Background. They are defined in the 3GPP specifications [1] together with their QoS parameter values.

The 3GPP has defined traffic classes which need to be supported in UMTS. In order to characterise them, this work is focused on four representative applications as shown in Table 3.1.

Table 3.1: Traffic classes and applications.

3GPP Traffic Classes	Representative Applications
Conversational	VoIP
Interactive	Web browsing
Streaming	Video streaming
Background	Email

The above applications and their typical network activity should be supported by the testbed. Complementary to the ability to rely on real life applications to emulate user behaviour, the generation of synthetic traffic using stochastic distributions (detailed in Appendix C) of packets carrying fake/dummy data enables to mimic traffic patterns from real life applications as realistically as possible. This choice was made for the sake of simplicity and scalability. It is much simpler to configure probabilistic traffic generators and launch them at the beginning of an emulation run than to create as many real life user sessions. Introducing one real life application session during the emulation is still possible and would allow the evaluation of the user's QoE. For instance, including a real Streaming session while the emulation is running would give a good hint of what a UMTS user would experience under such circumstances. This will be dealt with in Chapter 8.

The following sections aim at accurately establishing the respective traffic source model of these applications. For this purpose, an extensive literature survey has been performed in both 3GPP standardisation and open sources [2–6] in order to identify models and their respective parametrisation values.

3.2   Generic Traffic Model

Appropriate traffic models are needed as input for the emulator. Traffic models have to take into account the applications' statistical properties. There are a number of studies on such models for use in 3G access networks. Most of the times only two classes of traffic are considered, usually voice (with delay requirements) and data (with reliability requirements). Other studies deal with the modelling of traffic sources in more general mobile environments. Nevertheless, to the best of the author's knowledge, no article in the literature jointly treats all the four traffic classes defined for UMTS.

As the objective of the present dissertation is to imitate the transport layer activity of a single application over a packet-switched, IP-based network, a generic, three-level traffic model

as illustrated in Fig. 3.1 [7] was chosen to characterise the four classes. Here is a quick run-down of the meaning of the levels:

- Session Level: Lasts as long as the application is running. Its statistics are mainly influenced by user behaviour.

- Connection Level: Describes the connection behaviour of a single session. The chosen pattern is based on a High-Low model in which it is possible to generate traffic in both states.

- Packet Level: Represents the packet inter-arrival and size distribution for each state of the Connection Level.
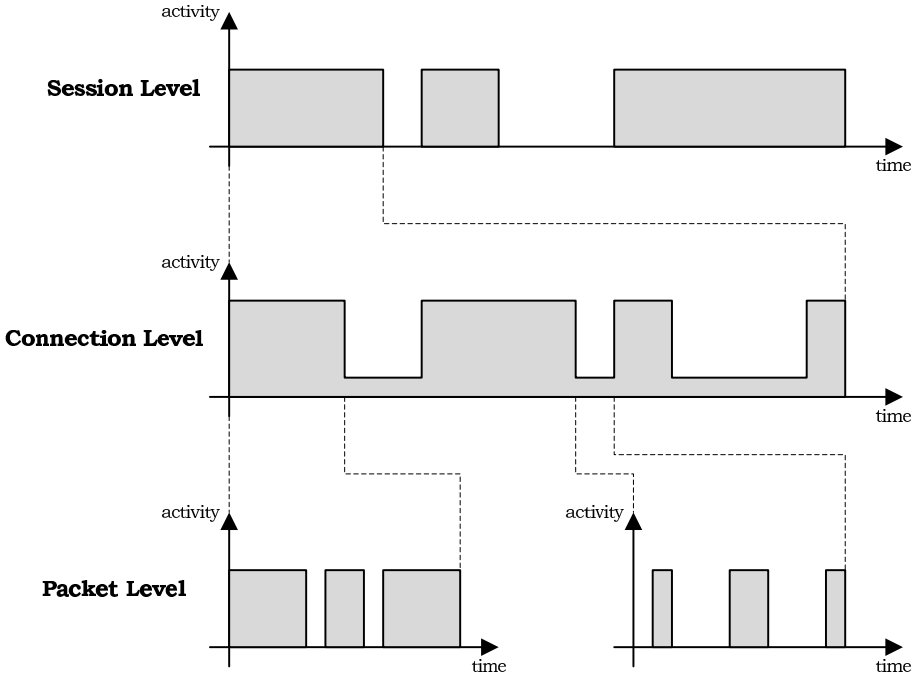
Figure 3.1: Three-level traffic model.

The parametrisation of this model is specific for each traffic class. Each application has been modelled to give physical significance to each of the levels. Let consider the Conversational class. In this case, the session level describes the moments when a user is engaged in a phone call and the moments

when s/he is not. When the user is engaged in a call, the connection level represents the time intervals during which the user is speaking and the time intervals during which s/he is simply listening. Finally, when the user is speaking, the packet level gives the size of each packet sent and the time interval separating two successive packets.

Although an application can be generally considered as bidirectional, many applications are clearly asymmetric. Thus, only the downstream flow is explicitly modelled, since this is the direction in which the majority of the traffic is normally observed. Hence the emulated traffic is simplex for applications emitting a significant amount of data in only one direction, such as Streaming and Interactive classes. The Interactive return traffic, mainly consisting of ACKs, is not explicitly emulated here. As ACKs circulate anyway, the return traffic is only emulated for the Conversational and Background classes.

3.3 Conversational Traffic - VoIP

The Conversational traffic is based on the voice communications. The 3GPP has decided to adopt the AMR vocoder with a rate of $12.2kbps$ to encode voice traffic [8]. Since this kind of traffic is considered as bidirectional, there is a need for synchronisation. In a real conversation, both participants are speaking throughout the session. Therefore, the emulated downlink and uplink traffics have to be synchronised and mixed (Fig. 3.3(a) and 3.3(b)).

In this type of application, a session lasts as long as the user makes a phone call. As soon as the session arrivals are considered to be independent, the customer sessions inter-arrival time follows a Poisson distribution whereas the session duration is exponentially distributed with a mean of $120s$ [9].

Once active, the connection model has to take into account the silence periods in the human voice activity. As the voice activity represents only 32% of the overall call time, a simple but convenient model is the two-state Markovian chain proposed

in [8]. As shown in Fig. 3.2, the probability to remain in voice inactivity states is denoted as $P = 0.99$ and the probability to remain in voice activity state is known as $Q = 0.9875$.

When the user is speaking, the codec produces $32 - Byte$ packets every $20ms$. A single UE connected to a NodeB, generating and receiving Conversational traffic during $1,200s$ has been emulated with a session inter-arrival mean time fixed at $100s$. Fig. 3.3 depicts the resulting activity graphs in the three-level model. The statistical distributions and their parameters are summarised in Table 3.2.
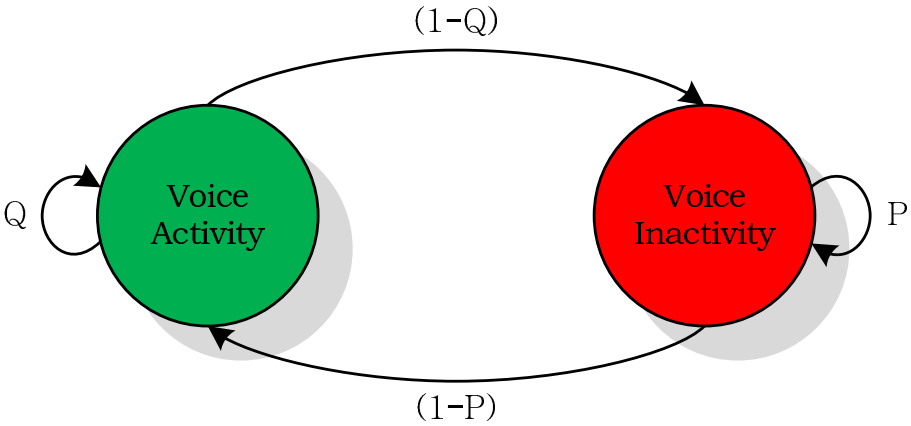
Figure 3.2: Two-state Markovian chain Conversational model.

This model has been chosen to comply with 3GPP requirements. It enables to produce contributions that would be admissible at 3GPP standardisation.

Table 3.2: Conversational statistics summary.

Level	Parameter	Description
Session	Inter-Arrival [s]	$Poisson(100)$
	Length [s]	$exponential(120)$
Connection	Voice Activity	$Markovian(0.99, 0.9875)$
Packet	Length [Bytes]	$constant = 32$
	Rate [kbps]	$constant = 12.2$
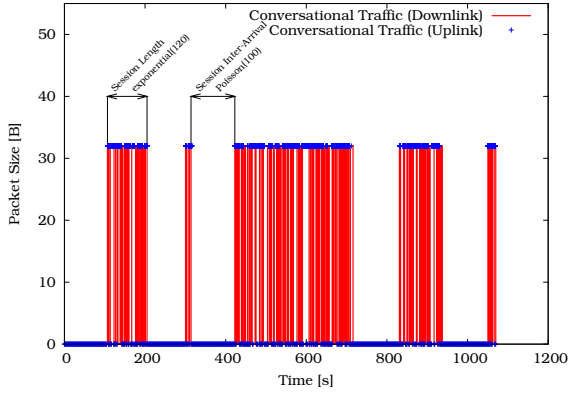
(a) Session level.
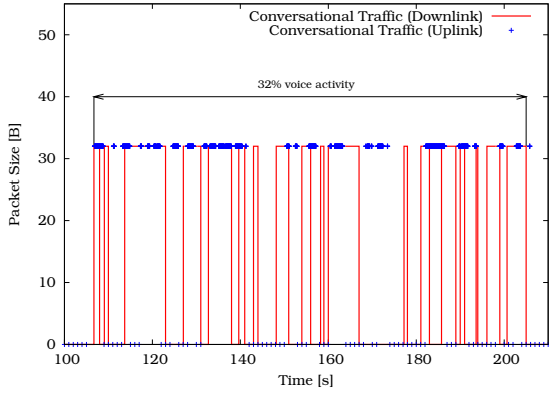
(b) Connection level.
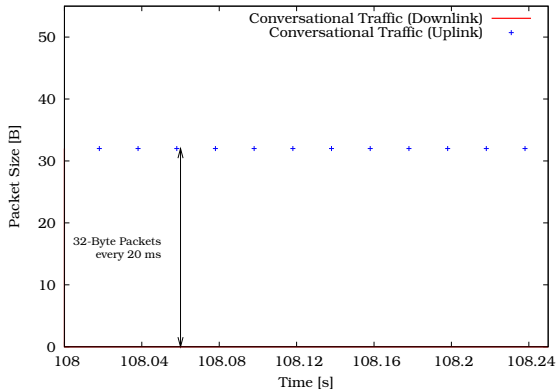
(c) Packet level.

Figure 3.3: Conversational illustrations.

3.4 Interactive Traffic - Web Browsing

Web traffic is nowadays the most important application used by the Internet community. The term web traffic comprises all HyperText Transfer Protocol (HTTP) traffic generated during a session with a typical web browser like Firefox or Internet Explorer.

To fit the above presented multilevel structure to this Interactive traffic, the first step is to study the way in which World Wide Web (WWW) users generate traffic by browsing the web. A user begins a WWW session by running a web browser. A session is considered as the period of time between that starting moment and the exit from the browser. The session arrival process is modelled by a Poisson process with a mean set at $50s$ [10].

During a session, the user visits several web pages and reads them. Therefore, a session is considered as a set of page transfers (connections) separated by reading times. The number of pages a user will browse during a session has been modelled by a geometric distribution around 5 visited pages in [11]. These pages need a certain number of packets in order to be successfully transmitted through the network. This number is considered to be geometrically distributed around 25 packets in [9]. The time the user will spend to read the newly downloaded page is an exponentially distributed random variable centred on $12s$ as proposed in [9].

This model and its parametrisation have been chosen again to comply with 3GPP requirements.

Table 3.3: Interactive statistics summary.

Level	Parameter	Description
Session	Inter-Arrival [s]	$Poisson(50)$
Connection	Packet Call	$geometric(5)$
	Packet Number	$geometric(25)$
	Reading Time [s]	$exponential(12)$
Packet	Length [Bytes]	$Pareto(1.1, 81.5, 66, 666)$
	Inter-Arrival [ms]	$exponential(8.3)$
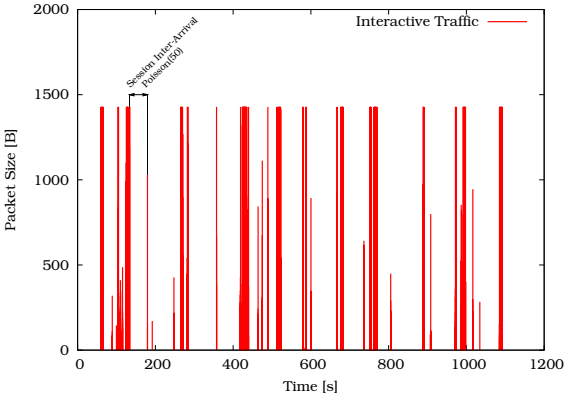
(a) Session level.
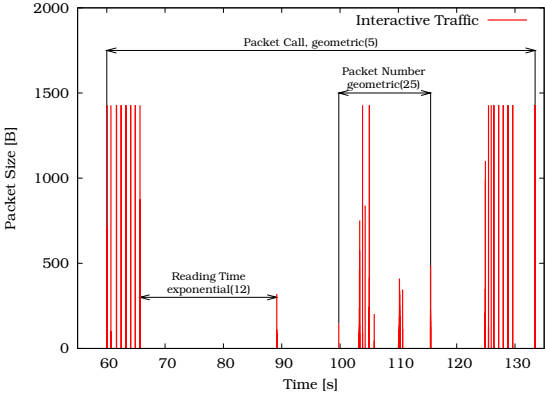
(b) Connection level.
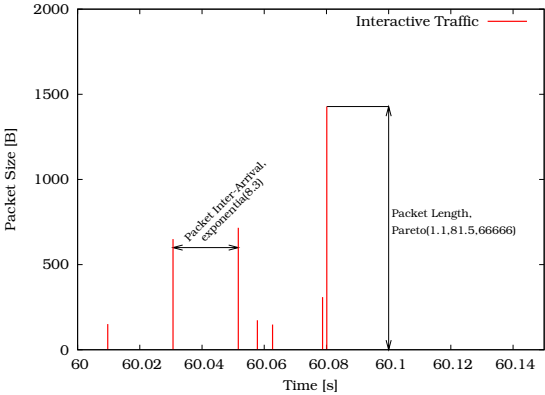
(c) Packet level.

Figure 3.4: Interactive illustrations.

The transfer of a web page is composed of several TCP connections that run in parallel. The resulting packet sizes are distributed following a Pareto distribution and their inter-arrival time is exponentially distributed around $8.3ms$ [9]. The Web browsing traffic then consists in a sequence of file downloads named "Packet Call", each of them modelled as a sequence of packet arrivals. All the parameters are summarised in Table 3.3. Fig. 3.4 represents a typical WWW user activity graph.

3.5 Streaming Traffic - Video Streaming

Multimedia streaming services over the Internet are more and more popular. Movies, news, education and training, video conferences, personal streaming, such as webcams or security surveillance, are only a small part of video streaming applications. UMTS extends these IP-based streaming services to mobile terminals with a video encoding following the Motion Picture Experts Group Layer-4 (MPEG-4) standard [12]. MPEG-4 has been chosen as a standard since it can support very low bit rates down to $5kbps$ and low frame rates of $15frame/s$.

According to the multilevel model, a session is equal to a connection since the session lasts during the whole Streaming connection. The number of new customer arrivals in the system (i.e. video sessions) follows a Poisson distribution with a mean of $300s$, whereas the session length is determined by an exponential distribution around $120s$.

Video content compression in the MPEG-4 standard is done via a Group of Pictures (GoP); either by intra-frame compression removing redundancies within a single frame of video, or by inter-frame compression removing redundancies across a GoP over time. MPEG-4 uses both intra- and inter-frame compression and its frames are designated as I-, P- or B-frames. As illustrated in Fig. 3.5, I-frames (Intra frames) are single frames in which all the content is examined independently by the compressor regardless of the frames preceding or following it. P-frames (Predicted frames) are following an I-frame whose content or colour has changed from the preceding I-frame. Be-

cause a P-frame only contains data that has changed, it cannot
be accessed and displayed on its own. B-frames (Bidirectional
predicted frames) are additional content redundancies that can
be removed during compression by looking at changes in frame
content or colour-based on both previous and subsequent fra-
mes. A B-frame cannot be accessed and displayed on its own, it
requires an I-frame and surrounding P-frames. The unit for the
model is the video frame. Each of these frames is included into
a certain GoP as defined in the MPEG-4 standard: IBBPBBPBB-
PBB.
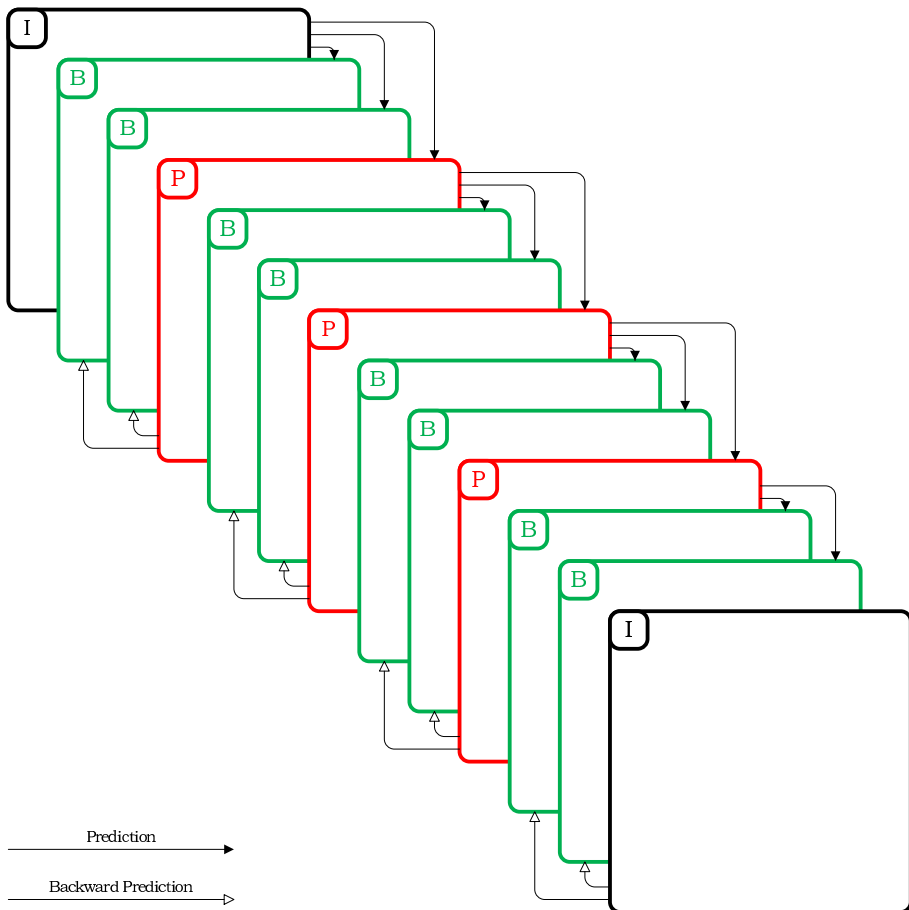
Figure 3.5: GoP illustration.

Considering a constant bit rate video streaming at $25 frame/s$, the size of the frames of each video session is described in [13] by a gamma distribution with different parameters. This depends on whether the user is streaming a high or low quality video. The exact value of these parameters has been determined by extensive statistical analyses made over a publicly available library of frame size traces of long MPEG-4 encoded videos [14, 15]. These frame size traces have been generated from several video sequences of typically $60 - minute$ length each. All those parameters are summarised in Table 3.4 and the resulting activity graphs are depicted in Fig. 3.6.

Table 3.4: Streaming statistics summary.

Level		Parameter	Description
Session	{	Inter-Arrival [s]	$Poisson(300)$
		Length [s]	$exponential(120)$
Connection		*n.a.*	*n.a.*
Packet	{	Length [Bytes]	*gamma*
		Inter-Arrival [ms]	$constant = 40$

3.6 Background Traffic - Email

Background traffic class is based on sending and retrieving e-mails. In this case of bidirectional traffic case, one can think of a user downloading his/her mails and answering some of them one by one after their download. Therefore, emulation then begins with a massive downlink traffic followed by multiple smaller uploads (Figs. 3.7(a) and 3.7(b)).

The session begins when the email application (Thunderbird, Outlook, etc.) is connected to the email server (through Post Office Protocol (POP) for example). According to [16], the session volume is a log normal variable and the session inter-arrival time follows a Poisson process with a mean of $200s$.

Each mail is represented by a single connection. The size of these mails is also defined by a log normal random variable [16].
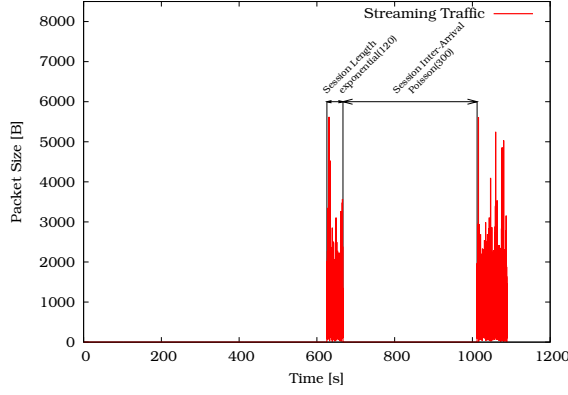
(a) Session level.
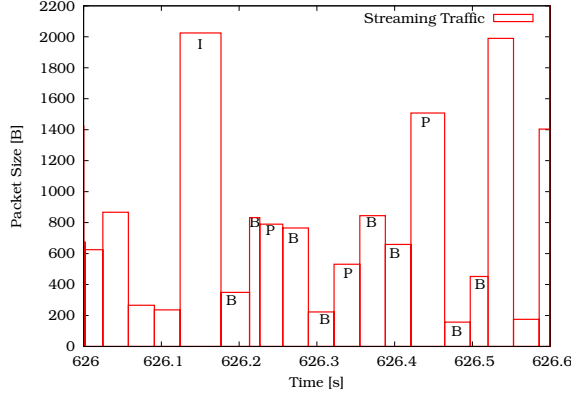
(b) Packet level.

Figure 3.6: Streaming illustrations.

In downstream, all the mails are downloaded one after the other (no mail inter-arrival time), whereas in upstream, each answer is separated by a writing time that follows a Pareto distribution [16].

Finally, the packet inter-arrival is defined once again by a log normal distribution [16]. More details about those distribution parameters may be found in Table 3.5. Fig. 3.7 illustrates the Background traffic.
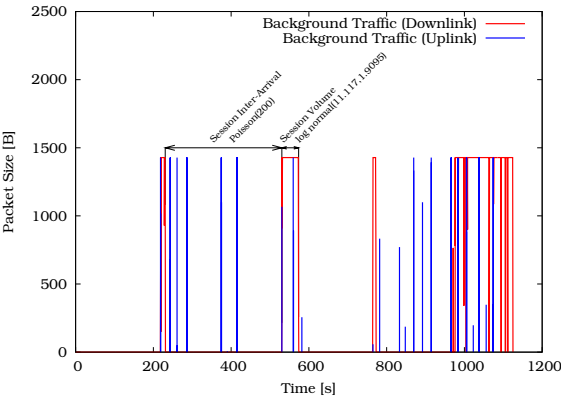
(a) Session level.
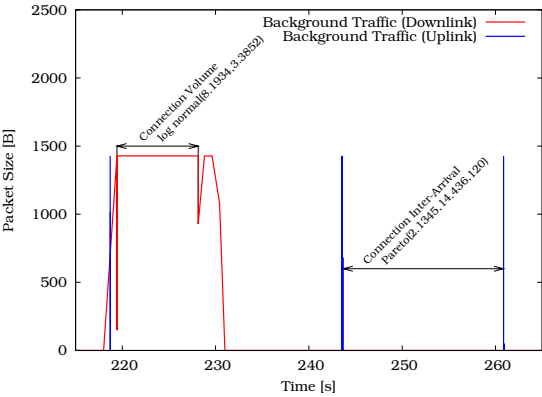
(b) Connection level.
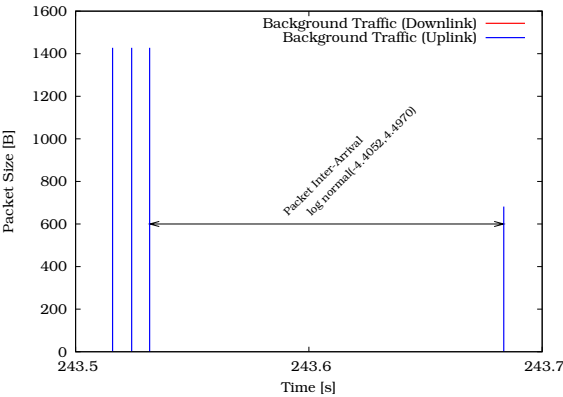
(c) Packet level.

Figure 3.7: Background illustrations.

Table 3.5: Background statistics summary.

Level	Parameter	Description
Session	Inter-Arrival [s]	$Poisson(200)$
	Volume [Bytes]	$log\ normal(11.117, 1.9095)$
Connection	Volume [Bytes]	$log\ normal(8.1934, 3.3852)$
	Inter-Arrival [s]	$Pareto(2.1345, 14.436, 120)$
Packet	Inter-Arrival [ms]	$log\ normal(-4.4052, 4.4970)$

3.7   Traffic Generator

The traffic generator TG [17] was chosen for this work purposes. It is written in the C language, its source code is publicly available and contributions to its further development are encouraged by the community. TG contains most of the statistical distributions needed for the traffic generation except Pareto.

For the purpose of this project, the TG source code was modified to support the Pareto distribution (detailed in Appendix D), to adapt the exponential distribution (detailed in Appendix E), and also to implement the IPv6 addressing as presented in [18, 19]. The code of the enhanced TG was shared with the developer community and published on the Internet [20].

3.8   Traffic Distribution

Probabilistic traffic models are used in the testbed to initiate sessions of any of the four traffic classes at any time according to Poisson processes. An emulated UE may initiate a Conversational session while already active with an ongoing Interactive session. In a real UTRAN, this would be translated by a UMTS user getting called as s/he was browsing the Internet on a 3G terminal.

As in real life, the traffic classes generated over the testbed have to be mixed. Table 3.6 lists a share of calls with the probability of a specific service once a new call/session is generated, as measured on a real life macrocellular outdoor network [21]. To the best knowledge of the author, there is no other open-

Table 3.6: Traffic distribution.

Traffic Classes	Share of Calls
Conversational	27%
Interactive	48%
Streaming	10%
Background	15%

source document stating such figures for the full traffic mix. For the purpose of the dissertation, the above mentioned mix has been chosen to generate traffic in each of the emulated services. This traffic distribution seems realistic enough to claim to mimic the load in a macrocellular network. It should however be adapted for studies focusing on micro-cellular or indoor networks.

3.9 Summary

In this chapter, a generic traffic model has been presented to generate the four UMTS traffic classes proposed by the 3GPP. The model fits each of these classes in a three-level model: session, connection and packet-level. Based on this multilevel structure, the four traffic classes have been modelled in terms of stochastic distributions, in line with open literature or 3GPP standards whenever applicable. It has then been integrated in the testbed through modification of the TG traffic generator source code.

Bibliography

[1] Technical Specification Group Radio Access Network. Quality of Service (QoS) concept and architecture. Technical Report TS 23.107 V7.1.0, 3rd Generation Partnership Project (3GPP), September 2007.

[2] Frank Yong Li and Norvald Stol. QoS Provisioning using Traffic Shaping and Policing in 3rd-generation Wireless Networks. In *Proceedings of the 3rd Wireless Communications and Networking Conference (WCNC)*, Orlando (FL - USA), March 2002.

[3] Jorge Aguiar and Luis Manuel Correia. Traffic Source Models for the Simulation of Next Generation Mobile Networks. In *Proceedings of the 7th Wireless Personal Multimedia Communications (WPMC)*, Abano Terme (Italy), September 2004.

[4] Phuoc Tran-Gia, Dirk Staehle, and Kenji Leibnitz. Source Traffic Modeling of Wireless Applications. *International Journal of Electronics and Communications (AEÜ)*, 55(1):27–36, January 2001.

[5] Paraskevas Kavallaris. Traffic Modelling for Mobile Multimedia Networks. In *Proceedings of the 11th IST Mobile and Wireless Communications Summit*, Thessaloniki (Greece), June 2002.

[6] Ana-Belen Garcá, Manuel Álvarez-Campana, Enrique Vázquez, and Julio Berrocal. Simulation of Quality of Service Mechanisms in the UMTS Terrestrial Radio Access Network. In *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN)*, Stockholm (Sweden), September 2002.

[7] Ana-Belen García, Enrique García, Manuel Álvarez-Campana, Julio Berrocal, and Enrique Vázquez. A Simulation Tool for Dimensioning and Performance Evaluation of the UMTS Terrestrial Radio Access Network. *Lecture Notes In Computer Science*, 2515:49–60, 2005.

[8] Technical Specification Group Radio Access Network. Feasibility Study for Enhanced Uplink for UTRA FDD (Release 6). Technical Report TR 25.896 V6.0.0, 3rd Generation Partnership Project (3GPP), April 2004.

[9] Technical Specification Group Radio Access Network. IP transport in UTRAN (Release 5). Technical Report TR 25.933 V5.4.0, 3rd Generation Partnership Project (3GPP), January 2004.

[10] Arcadio Reyes, Edgar González-Parada, Eduardo Casilari, Julio César Casasola, and Angel Díaz-Estrella. A page-oriented WWW traffic model for wireless system simulations. *Teletraffic Engineering in a Competitive World (ITC16)*, pages 1271–1280, 1999.

[11] Technical Specification Group Radio Access Network. Selection procedures for the choice of radio transmission technologies of the UMTS. Technical Report TS 30.03 V3.2.0, 3rd Generation Partnership Project (3GPP), March 1998.

[12] Atul Puri and Alexandros Eleftheriadis. MPEG-4: An object-based multimedia coding standard supporting mobile applications. *ACM Mobile Networks and Applications*, 3:5–32, 1998.

[13] Michael Frey and Son Nguyen-Quang. A Gamma-Based Framework for Modeling Variable-Rate MPEG Video Sources: the GOP GBAR Model. *IEEE/ACM Transactions on Networking*, 8(6):710–719, December 2000.

[14] Martin Fitzek, Frank Reisslein. MPEG-4 and H.263 Video Traces for Network Performance Evaluation (Extended Version). Technical Report TKN-00-06, TU Berlin, Dept. of Electrical Engineering, Telecommunication Networks Group, October 2000.

[15] Frank Fitzek and Martin Reisslein. MPEG-4 and H.263 Video Traces for Network Performance Evaluation. *IEEE Network*, 15(6):40–54, December 2001.

[16] Alexander Klemm, Christoph Lindemann, and Marco Lohmann. Traffic Modeling and Characterization for UMTS Networks. In *Proceedings of the 44th Globecom conference*, San Antonio (TX - USA), November 2001.

[17] Paul McKenney, Dan Lee, and Barbara Denny. Traffic Generator Software Release Notes. Technical report, SRI International and USC/ISI Postel Center for Experimental Networking, January 2002. `http://www.postel.org/tg/tg.htm`, last visited: March 13, 2008.

[18] Hugues Van Peteghem and Laurent Schumacher. Description of an IPv6 Linux-Based UTRAN Testbed. In *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Barcelona (Spain), March 2006.

[19] Hugues Van Peteghem and Laurent Schumacher. Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based *Rel'*99 UTRAN Testbed. Technical Report TD(07)011, COST 2100, February 2007.

[20] New version of TG. `http://www.info.fundp.ac.be/~hvp/rech/doc/docs/tg2.2.tar.gz`, last visited: March 13, 2008.

[21] David Soldani. *QoS Management in UMTS Terrestrial Radio Access FDD Networks*. PhD thesis, Helsinki University of Technology, October 2005.

Chapter 4

Mobility Management

4.1 Introduction

This chapter depicts several design choices of the testbed which have been set in a way to emulate a real UTRAN segment. First, it presents the emulated world which is composed of four trisectorial macrocell NodeBs. It comprises also the mobility scheme enabling UEs to move across the entire emulated world following a random walk algorithm coupled with the wrap around principle. Finally, the chapter explains the RNC handover management based on the UE's position and a line-of-sight power-based computation.

4.2 Emulated World

The testbed emulates four NodeBs each implemented on the four group $\mathcal{A}$ computers shown in Fig. 2.1. These base stations are situated on a hexagonal grid (standard macrocellular set-up proposed by the 3GPP in [1, 2]). The cell radius equals $1,400m$ therefore, the NodeBs' total covered surface is about $27km^2$ [3,4].

Until now, NodeBs were considered as omnidirectional, i.e. their transmission power is equally distributed all around. In fact, NodeBs are composed of three sectors emitting in different directions. This calls for an adaptation of the world representation. Fig. 4.1 depicts the world evolution from the omnidirectional to the trisectorial NodeBs.

(a) Omnidirectional NodeBs.   (b) Trisectorial NodeBs.
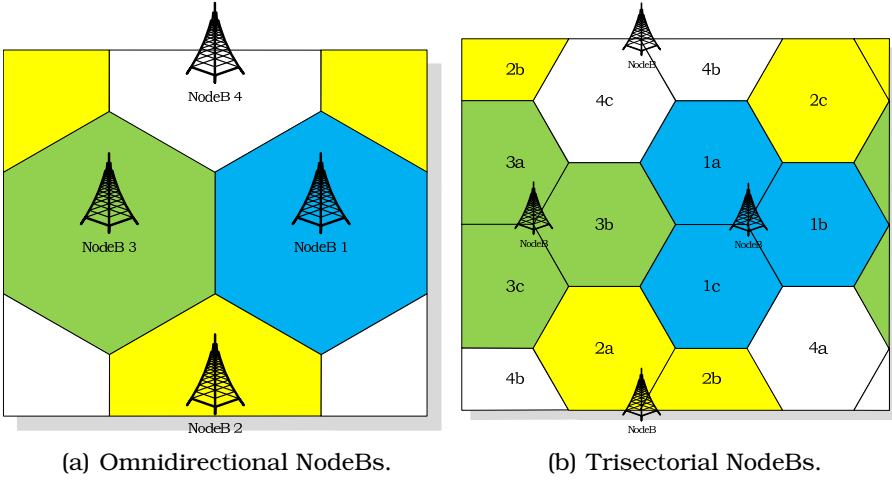
Figure 4.1: Emulated world representations.

In such hexagonal layout network, UEs behaviour at the boundary of the cells can significantly bias the output statistics, especially if the number of hexagonal cells is small in a cellular network simulation, as explained in [5]. For the purpose of this dissertation and to avoid the boundary effect, UEs are not restricted to this $27km^2$ area using a wrap around technique. When a UE is about to leave the emulated world by crossing one edge, it reappears at the opposite edge (still within the same cell if necessary), as if moving on a torus (Fig. 4.2). Moreover, it has been shown in [6] that the introduction of wrap around, when modelling a cellular system, can have a beneficial effect on the results obtained since it appears to be much more realistic.
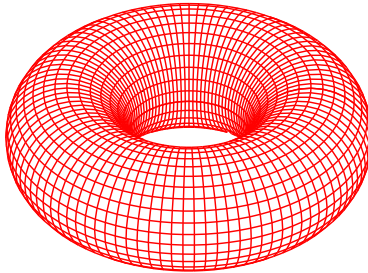
Figure 4.2: Torus world.

4.3 UE Mobility

To allow terminal mobility within the testbed, UEs are able to move with four different fixed speeds defined by the 3GPP in [1]: $3km/h$ representing a pedestrian, 30 and $70km/h$, which represents a UE on board of a vehicle within city limits and $120km/h$, which represents a UE on a train or on a car on the speedway.

To model UE mobility in the testbed design, a random walk scheme based on the Gauss-Markov mobility model presented in [7, 8] was chosen. On every main loop iteration (Fig. 2.5), the distance the UE will travel and its direction during the next iteration are randomly drawn. Keeping in mind that a vehicle moving at $120km/h$ cannot turn as abruptly as a pedestrian, the scope of possible directions is then reduced with increasing UE speed. As illustrated in Fig. 4.3, the angle reduction is inversely proportional to the square root of the UE speed. This is rational to linking the angle a UE is able to turn with its speed. The square root was mandatory since an angle reduction inversely proportional to the UE speed itself was too stringent.
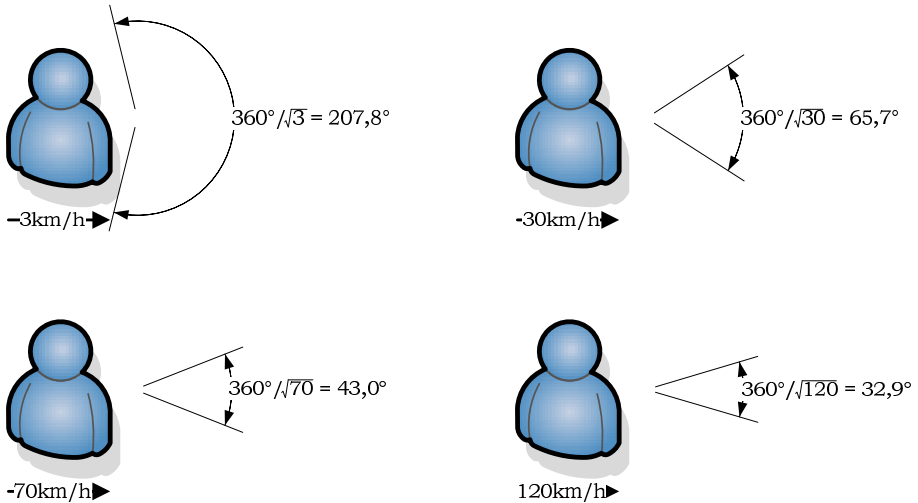
Figure 4.3: Angle amplitude vs. user speed.

As example, Fig. 4.4 presents the course of two different UEs moving at 120 and $30km/h$ during $600s$. The difference of distance travelled appears clearly comparing the two paths. This figure

illustrates also the wrap around principle since the UE moving at $120km/h$ crosses five times the edge of the emulated world.
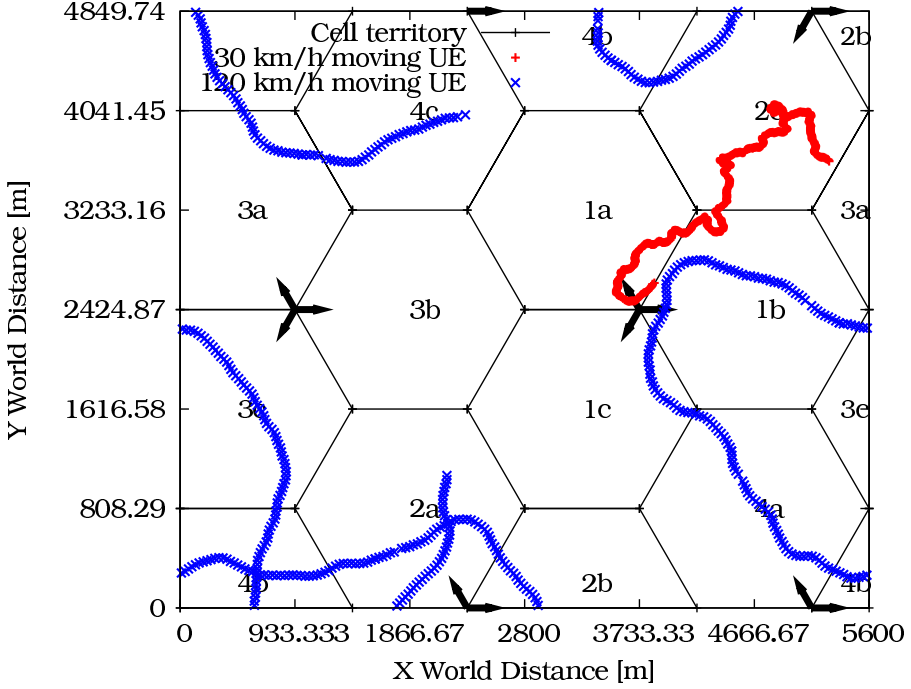
Figure 4.4: Random walk at two different speeds.

This random walk scheme is consistent with the regular deployment of the NodeB (as already presented in Fig. 4.1(b)). Thanks to the modularity of the testbed, a completely different world layout with roads and buildings (Manhattan type map) could be used, triggering a smarter UE mobility scheme.

4.4 Handovers

With moving UEs in the emulated world, the RNC needs to establish a handover procedure which will decide where and when a handover has to be performed. This procedure is based on the SNIR level reached by a UE:

$$SNIR = \frac{Received\ Power}{Interference + Noise} \qquad (4.1)$$

4.4.1 Received Power

The computation of the received power is based on the signal power received on an unmodulated code channel. The main function of this pilot channel is to help with the channel estimation at the terminal for the DCH transmissions. Another important role for this kind of channel is the constant received power measurement for the handover and cell selection. The UMTS specific physical pilot channel is the Common PIlot CHannel (CPICH). The assumptions concerning the CPICH, as well as all the other testbed cellular parameters, are summarised in Table 4.1.

Table 4.1: Testbed cellular parameters.

Parameter	Setting
Transmission power [dBm]	43
Sectorial gain [dBi]	14
UE Noise figure [dB]	9
Thermal noise density [dBm/Hz]	-174
Emitting frequency [GHz]	2.14
NodeBs height [m]	30
UEs height [m]	1.5
CPICH SF length [chips]	256
CPICH power [% of the total power]	10
SNIR handover threshold [dB]	4

The UE received power is then computed by a propagation model mainly determined by the distance separating the UE and its SNodeB and the angle created between the UE position and the SNodeB's sector emitting direction:

- The received power decreases as the distance between the UE and its SNodeB increases. This relation is known as path-loss and it is not a linear one as shown in Fig. 4.5(a). To model this, the UMTS adaptation of the Okumura-Hata wireless propagation equations presented in [9] and approved by the 3GPP in [10] has been chosen.

- As shown in Fig. 4.5(b), if a UE is connected to a trisectorial NodeB, its received power will decrease as it will move away from the sector main lobe. The amplitude of this attenuation has been fixed by the 3GPP in [11].

(a) Received power vs. UE-SNodeB distance.

(b) Antenna gain vs. UE-SNodeB angle.
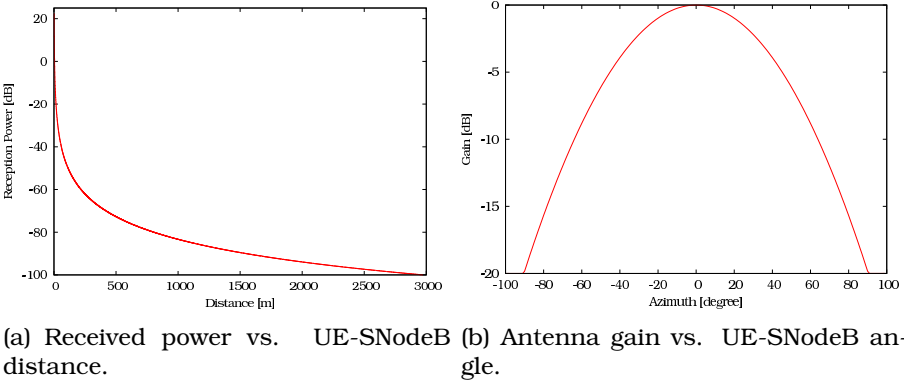
Figure 4.5: UE received power influencing factors.

4.4.2   Interference

Focusing on the interference level a target sector has to support, it has been shown in [2, 12, 13] that taking into account more tiers than the first and second surrounding sectors will not yield significant changes. Therefore, higher level tiers can be neglected in the testbed design. Fig. 4.6 shows the two tiers of sectors surrounding a target sector.

For the testbed design, the interference figure a UE has to support was defined as the sum of the received power from all the testbed sectors except the sector the UE is connected to, multiplied by two. With the testbed emulating 12 sectors (4 NodeBs), the computation of a representative interference figure has to take into account the two tiers surrounding sectors. In the hexagonal grid world representation, this means this computation has to consider 18 neighbouring sectors. Multiplying by two the calculated interference over the 11 neighbouring sectors emulated by the testbed gives a realistic upper bound of what could be the real interference level.
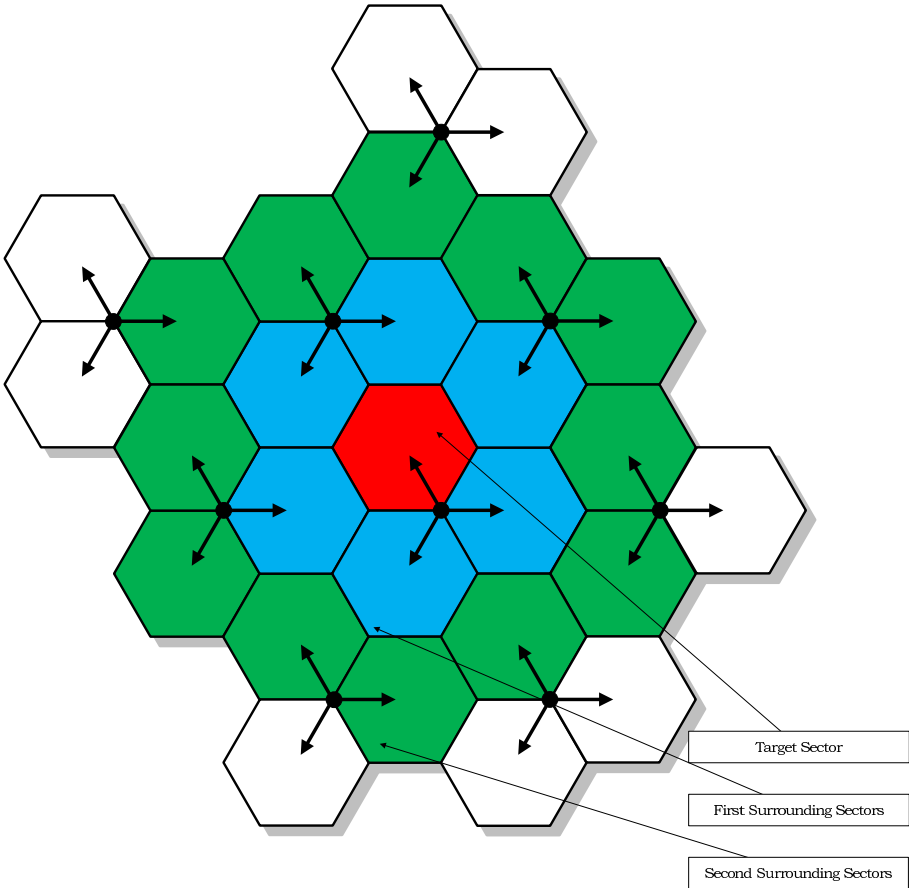
Figure 4.6: Configuration of adjacent tiers of neighbouring cells.

4.4.3 SNIR Level

Another concept embedded in the WCDMA and influencing the UE reception level is signal spreading [14]. It amplifies a signal and attenuates the interferences. The higher the SF, the more amplified the signal will be (resp. the more attenuated the interferences will be). On the other hand, the sending of the data on the wireless interface will be slower. The SF's choice is thus a real challenge the RNC has to overcome. This is a point at issue in the present dissertation and will be thoroughly studied in Chapter 6.

As the noise level is assumed to be constant during an entire emulation, (4.1) finally becomes:

$$SNIR = \frac{Target\ Sector\ Received\ Power\ *\ SF_{CPICH}}{2*(\sum Other\ Sectors\ Received\ Power) + Noise} \qquad (4.2)$$

Derived from (4.2), Fig. 4.7 represents the SNIR level a UE may reach everywhere in the emulated world.
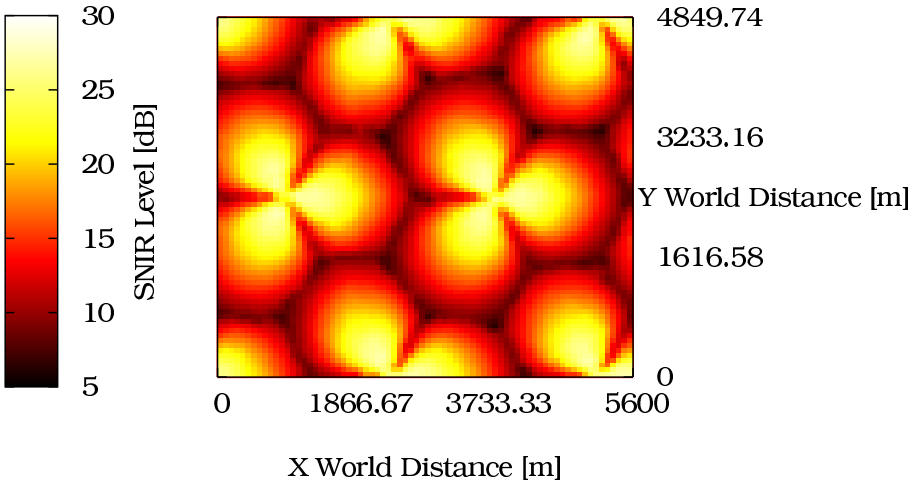
Figure 4.7: SNIR map.

This map represents the SNIR levels under unloaded conditions. The map is not evolving during the emulation runs and, therefore, the testbed (as it is) does not include the cell breathing mechanism. As a result, UEs beyond the geographic cell edge are not able to hook to corresponding NodeB. To implement such a cellular behaviour, the SNIR computation should include the fluctuating CPICH as presented in (5.1).

4.4.4   RNC Decision

Based on the SNIR level constantly evaluated through (4.2), the RNC is able to decide when a handoff is needed. A handover occurs when the difference between the power of the SNodeB's signal and the power of another NodeB's signal is greater or

equal to the handover threshold already introduced in Table 4.1. Based on this continuous SNIR evaluation the RNC knows exactly when to handoff a UE.

Fig. 4.8 illustrates the results from the operation of the testbed. It represents the journey of one UE during 450 seconds generating and receiving Conversational traffic only and moving at $30km/h$. The mobility graph of Fig. 4.8(a) shows that the UE undergoes a handover passing from the NodeB #1 (sector a) to the NodeB #3 (sector b) at about the middle of the emulation run.

The SNIR evolution of the UE (Fig. 4.8(b)) shows that the handover exactly occurs when the power of the signal received by the NodeB #3 (sector b) exceeds by $4dB$ the signal received by the UE's SNodeB power. This behaviour is consistent with the handover SNIR threshold value present in Table 4.1, originally proposed in [15].

Finally, the activity graph of Fig. 4.8(c) shows that the UE was active (sending/receiving data) when the handover occurred. It clearly shows the end of the first connection and the beginning of the second connection, both totally synchronised in uplink and downlink directions. When the handover occurred, the service has been stopped during a $900ms$ representing the transport channel allocation time, as already introduced in Section 2.4.2. The handover impact on UE Radio Resource Control (RRC) state and/or transport channel will be further discussed in Section 7.3.4.

4.4.5   Testbed's Perspective

From the testbed's perspective, this handover triggers a process migration from one computer to another as illustrated in Fig. 4.9. When the RNC decides to handoff the UE, the traffic emitting processes have to be stopped on the source computers (computer $\mathcal{C}$ for the downlink and the eastern computer of the $\mathcal{B}$ group for the uplink) and then new processes corresponding to the traffic generation after the handover can be started on computer $\mathcal{C}$ and the new destination computer (the southern computer of group $\mathcal{B}$). The remote configuration of the TG processes on the $\mathcal{B}$ computer group by the RNC is based on the

(a) Mobility graph.

(b) SNIR graph.

(c) Activity graph.
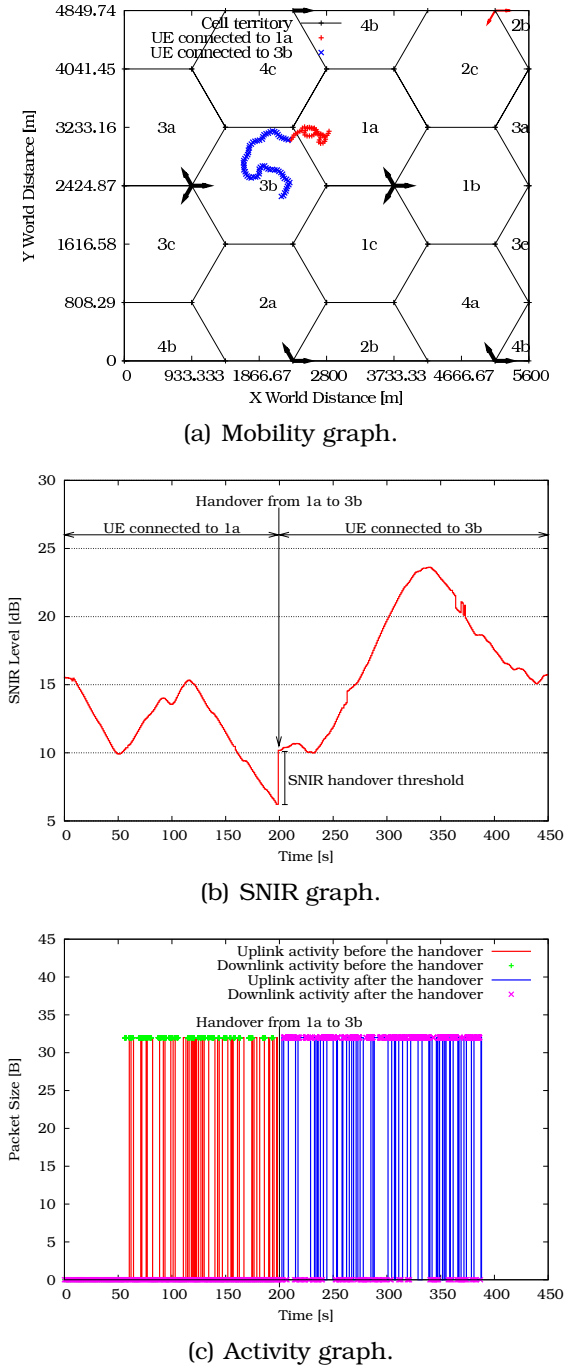
Figure 4.8: Handover impacts.

Secure SHell (SSH). It appeared to be the easiest way to quickly launch command lines on remote computers.
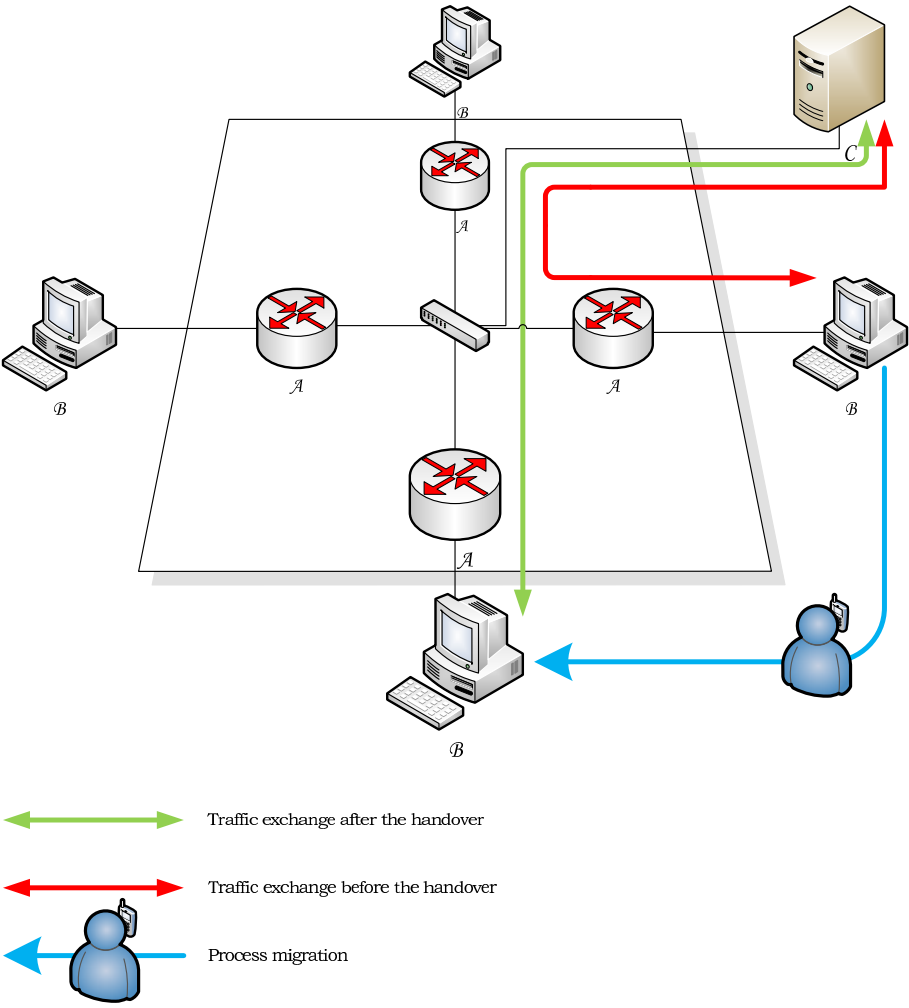
Figure 4.9: Testbed's perspective at handover.

As a result, the packets belonging to the same activity session do not use the same physical route before and after this handover. This illustrates how the testbed implements UE mobility over a wired network.

4.5 Summary

This chapter has presented the testbed main modelisation choices. First of all, it has described the emulated world, the four trisectorial macrocell NodeBs and the integration of wrap around to avoid border effect. The UE mobility scheme based on a Gauss-Markov random walk has also been described and illustrated. I was explained how the RNC manages the handovers through a constant SNIR ratio evaluation and this has illustrated it over the testbed.

Bibliography

[1] Technical Specification Group Radio Access Network. Multiple Input Multiple Output in UTRA. Technical Report TR 25.876 V7.0.0, 3rd Generation Partnership Project (3GPP), March 2007.

[2] Technical Specification Group Radio Access Network. Feasibility study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement. Technical Report TR 25.892 V6.0.0, 3rd Generation Partnership Project (3GPP), June 2004.

[3] Hugues Van Peteghem and Laurent Schumacher. Description of an IPv6 Linux-Based UTRAN Testbed. In *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Barcelona (Spain), March 2006.

[4] Hugues Van Peteghem and Laurent Schumacher. Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based $Rel'99$ UTRAN Testbed. Technical Report TD(07)011, COST 2100, February 2007.

[5] Zheng Qunwei, Hong Xiaoyan, and Ray Sibabrata. Recent Advances in Mobility Modeling for Mobile Ad Hoc Network Research. In *Proceedings of the 42nd Annual ACM Southeast Regional Conference*, pages 70–75, Huntsville (AL - USA), April 2004.

[6] Anna Dinis and John Thompson. The effects of including wraparound when simulating cellular wireless systems with relaying. In *Proceedings of the 65th Vehicular Technology Conference (VTC-spring)*, Dublin (Ireland), April 2007.

[7] Alexandrosz Burulitisz, Sándor Imre, and Sándor Szabó. On the Accuracy of Mobility Modelling in Wireless Networks. In *Proceed-

ings of the 39st *International Conference on Communications (ICC)*, Paris (France), June 2004.

[8] Ben Liang and Zygmunt Haas. Predictive Distance-Based Mobility Management for PCS Networks. In *Proceedings of the* 18th *Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*, pages 1377–1384, New York (NY - USA), March 1999.

[9] COST 231 Final Report. Digital Mobile Radio Towards Future Generation Systems. Technical report, COST 231, November 1999.

[10] Technical Specification Group Radio Access Network. Radio Access Network; Radio network planning aspects (Release 6). Technical Report TR 43.030 V7.0.0, 3rd Generation Partnership Project (3GPP), October 2007.

[11] Spatial Channel Model AdHoc Group. Spatial Channel Model Text Description. Technical report, 3GPP & 3GPP2, April 2003.

[12] Bernd Heideck, Andreas Draegert, and Thomas Kürner. Heuristics for the Reduction of Complexity in UMTS Radio Network Quality Assessment. In *Proceedings of the* 13th *Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Lisbon (Portugal), September 2002.

[13] Lars Torsten Berger. *Performance of Multi-Antenna Enhanced HSDPA*. PhD thesis, Aalborg University, April 2005.

[14] Harri Holma and Antti Toskala. *WCDMA for UMTS*. Wiley interscience, 3rd edition, October 2004.

[15] Igor Tomić and Dukić Miroslav. Soft Handover and Downlink Capacity in UMTS Network. In *Proceedings of the* 12th *Telecommunication Forum (Telfor)*, Belgrade (Serbi), November 2004.

Chapter 5

RAN & Air Interface

5.1 Introduction - Generic Linux Model

Besides the mobility of the users already introduced in the previous chapter, other properties characterise the wireless networks. The most important features are that such networks suffer from a limited bandwidth, encounter typically more errors than wired networks, and that these errors are time-correlated.

In order to mimic a mobile user connected to a wireless access point, two Linux computers connected with a classical Ethernet link were used with TC [1] which is able to perform packet shaping, scheduling, policing and dropping.

The limited bandwidth availability over an air interface can be emulated through the Token Bucket (TB) traffic shaper and rate limiter. The TB is typically used to control the amount of data that is injected into a network, allowing bursts of data to be sent.

The higher and variable BER property is slightly more complicated to emulate, since it has to take into account a larger number of parameters such as user speed, its position, the environment in which it is evolving, etc. For the emulation of this complex behaviour, the TC Network Emulation (NetEm) component [2] is used. It is a waiting queue capable to delay, drop, corrupt, duplicate or even reorder packets. The packet corruption functionality of this component is the most appropriate to mimic what packets really undergo over radio links by introducing a single bit error at a random position. Instead of being

dropped, this corrupted packet is transmitted to the mobile user, and therefore consumes a part of the radio link bandwidth. To cope with the time-correlated error property, the source code of the NetEm component was to be modified, as the original version generates independent errors.

Below follows the explanation of these different steps to fit to the Rel'99 UMTS RAN and air interface.

5.2   Limited Bandwidth

5.2.1   Spreading Factors

Since the UMTS uses WCDMA as air interface, it has to respect the spreading/despreading concept [3]. Transmissions from a single source are separated by channelisation codes. The spreading/channelisation codes of UTRAN are based on the OVSF technique. These codes are picked from a code tree related to one sector and managed at the RNC.

The spreading code allocated to a flow and the bandwidth it will receive are tightly linked: the SF defines how many chips are used to code one user data symbol. Therefore, higher SFs will subsequently reduce the data rate. WCDMA supports a maximal chip rate of $3.84 Mcps$ and the available downlink SFs range from 4 to 512. Table 5.1 (excerpt from [4]) shows the available downlink data rates in Rel'99 UMTS with a Binary Phase Shift Keying (BPSK) modulation[1].

In the testbed design, the SF allocation scheme is based on two main properties: the requirements of the data flow itself (traffic class) and the expectation of the UE receiving the data flow (user profile). The specific QoS needs of the four traffic classes have already been introduced in Chapter 3. Four user profiles have been created. They can easily be mapped on a real cellular network:

---

[1]Data rates in reference [4] are given for Quadrature Phase Shift Keying (QPSK). The present dissertation assumes BPSK. Impact of modulation appears in the available bandwidth by the user. Using QPSK modulation, a user will be entitled to claim twice the bandwidth s/he enjoys with BPSK.

Table 5.1: Available downlink data rates in Rel'99 UMTS.

SF	Maximum User Data Rate (BPSK and ½ rate coding)
512	$3kbps$
256	$7.5kbps$
128	$15kbps$
64	$30kbps$
32	$60kbps$
16	$120kbps$
8	$240kbps$
4	$480kbps$

- Platinum: business contracts. These users will get the best treatment and the larger bandwidth possible according to the network state.

- Gold: postpaid contracts. These are expensive contracts which allow the user to easily access the different applications.

- Silver: postpaid contracts. These are the classical contracts for an ordinary use of a mobile terminal.

- Bronze: prepaid contract. The less expensive contracts designed for a minimal use of a mobile terminal. These users' data will mainly be scheduled as best-effort traffic.

The testbed design assumes a 5% Platinum, 10% Gold, 15% Silver and 70% Bronze distribution inferred from postpaid-prepaid ratios recently published by Belgian cellular network operators [5–7].

5.2.2 Transport Channels

In the UTRAN, data generated at higher layers (e.g. above IP/Datalink) are carried over the air interface using transport channels. The 3GPP has defined all these channels as unidirectional, i.e. uplink or downlink. There are two types of transport

channel: dedicated and common/shared channels. The main difference between them is that a common channel is a resource divided between users in a cell, whereas a DCH is a reserved resource for a single user.

The DCH is the only dedicated transport channel in the UMTS standard. It carries the service data and control information of a single user at variable bit rate. The DCH is characterised by features such as fast power control and fast data rate change on frame-by-frame basis.

The common channels are shared between several users within the same cell. There are six different common transport channel types defined for the Rel'99 UTRAN. The present dissertation focuses on only two of them: The DSCH which carries signalling and service data of several users at variable bit rates, and the FACH which carries also the service data of several users but at a fixed bit rate.

5.2.3   Sector Organisation

Each NodeB sector in the emulated world may only have one FACH, one DSCH and a number of DCHs depending on the RNC's SF management. As presented in Table 5.2, the FACH and DCH carry user data flow with a fixed bit rate depending on their SF. The bandwidth of the DSCH is variable since its SF occupies the free room left in the OVSF tree after the DCHs SF allocation. Fig. 5.10 shows an example of this DSCH bandwidth variation mechanism.

Table 5.2: Comparison of different transport channels.

Channel	DCH	DSCH	FACH
SF	Fixed [512 - 4]	Variable [256 - 4]	Fixed [256 - 4]
Multiusers	Dedicated	Shared	Shared
Multicodes	No	Yes	No

Even if signalling is beyond the scope of this work, the testbed has to consider the radio resource consumption of such traffic. A quarter of the OVSF tree is then devoted to signalisation and data transfer over specified transport channels (Broadcast

CHannel (BCH), Paging CHannel (PCH) and FACH) whereas the rest has to be divided between the DCHs and the DSCH. This is illustrated in Fig. 5.1.
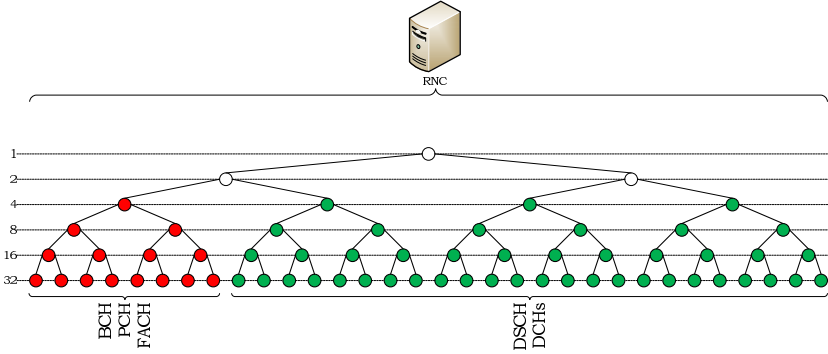
Figure 5.1: OVSF tree division.

An optimal OVSF management is a complex problem and also a major point at issue in the present dissertation. Chapter 6 is therefore dedicated to the study of SF allocation issue.

The RT sessions (Conversational and Streaming) are directly allocated to a DCH, since this traffic cannot tolerate the significant packet delay or delay variation likely to appear when using shared channels. As there is no bandwidth reservation mechanism for shared channels, RT sessions are directed to DCHs only.

On the other hand, NRT sessions (Interactive and Background) may support a much longer delay. These traffics tolerate to be directed to either dedicated or shared channels since the packet scheduling of the shared channels does not affect the NRT sessions as it does to the RT ones.

5.3 BER Specific Properties

5.3.1 Time Correlation

It has been shown that the typical bit errors over a radio link are not totally decorrelated and tend to appear in bursts. One of the analytical models trying to describe this property is the

Gilbert-Elliot model presented in [8,9] and illustrated in Fig. 5.2 at a TTI level. This model uses a two-state (error and error-free) discrete time Markov chain for the description of channel variations. Popular for wireless bit errors, it captures burstiness, while remaining simple enough to be treated analytically.
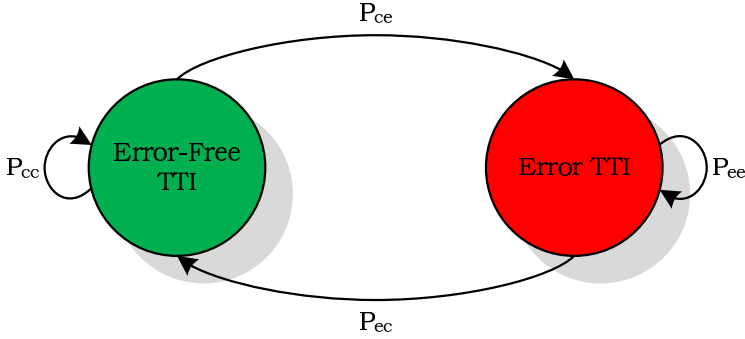
Figure 5.2: Two-state Markovian time-correlated error model.

Recent research has shown that an enhanced version of the Gilbert-Elliot model presented in [10] approaches even better the UMTS radio link. The error state in the enhanced model version is the same as in the simple two-state Markov model. For the purpose of the present dissertation, the error-free state of the new model was modified. The duration of the stay in that state is now derived from a Weibull-distributed random variable which correctly fits the measurements obtained in real networks.
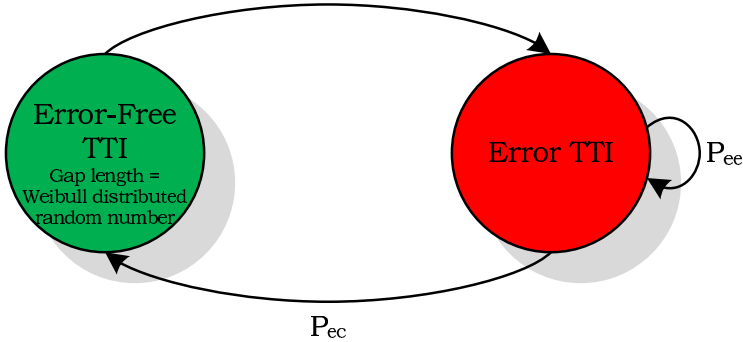
Figure 5.3: Two-state Markovian time-correlated error model with Weibull distribution.

In the testbed design, the NetEm component has been modified to implement this error model considering the two-state of the Markovian chain at the TTI level as shown in Fig. 5.3. If in the error-free (resp. error) state, all the transmitted RLC Protocol Data Units (PDUs) transiting over the air interface during the TTI are considered as well received (resp. corrupted). In line with [10], a burst is now defined as a number of subsequent erroneous TTIs (Fig. 5.4(a)), whereas a gap is considered as a number of subsequent error-free TTIs (Fig. 5.4(b)).

(a) Burst length comparison.

(b) Gap length comparison.
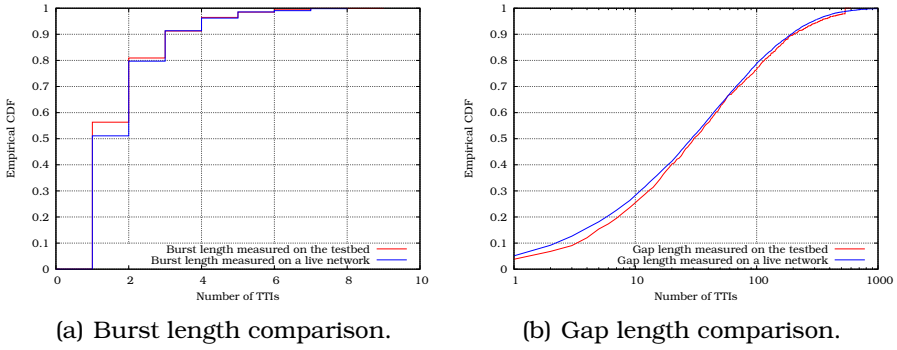
Figure 5.4: Testbed vs. live network measurements.

5.3.2 Variability

As described in Chapter 4, the emulated mobile users are moving around on a torus world. Periodically, the received power of each UE is updated according to [11], in order to gauge the quality of the ongoing transmission. The quality of a transmission between a NodeB and a UE can be evaluated from the received power the UE is getting. This power is a direct function of the NodeB emitted power, and is also influenced by a number of other factors. At the receiver end, the carrier-to-interference-ratio is calculated following the guidelines given in [12] as:

$$\left(\frac{C}{I}\right)_{CH} = \frac{Sh \cdot Pl \cdot (P_t)_{CH}}{\alpha I_{intra} + I_{inter} + P_N} \tag{5.1}$$

where Sh is shadow fading affecting a communication at a precise location in the emulated world. This will be discussed in Section 5.3.4. Pl represents the path loss the communication undergoes at the same position. This has already been addressed in Section 4.4.1. $(P_t)_{CH}$ stands for the transmitted power the NodeB allocates to the transport channel. I_{intra} is the intracell interference which is generated by those users who are connected to the same sector as the observed user. I_{inter} represents the intercell interference which is generated from the other cells. α is the orthogonality factor which takes into account the fact that all the downlink communications are not perfectly orthogonal due to multipath propagation. An orthogonality factor of 0 corresponds to perfectly orthogonal intracell users while with an orthogonality factor of 1 corresponds to the intracell interference having the same effect as intercell interference. The assumed value for the orthogonality factor is 0.4 in the macrocellular case as proposed in [12]. P_N is thermal noise power which has been fixed in [12] to $-99dBm^2$.

From the $(C/I)_{CH}$ evaluated as in (5.1), the normalised energy per information bit is obtained as follows [13, 14]:

$$\left(\frac{E_b}{N_0}\right)_{CH} = \frac{1}{2R_{CH}} \cdot SF_{CH} \cdot \left(\frac{C}{I}\right)_{CH} \qquad (5.4)$$

where R_{CH} is the coding rate of the channel and SF_{CH} its SF. With (5.4) it is possible to calculate the power received in each transport channel:

---

[2]To implement the cell breathing mechanism and considering (5.1), the SNIR computation could include the fluctuating CPICH as follows:

$$SNIR$$
$$= \frac{Sh \cdot Pl \cdot (P_t)_{CPICH}}{\alpha I_{intra} + I_{inter} + P_N} \qquad (5.2)$$
$$= \frac{Sh \cdot Pl \cdot (P_t)_{CPICH}}{\alpha(Sh.Pl.(P_t)_{Tot}.(1-\rho)) + (2*(\sum_{sectors}^{other} Sh.Pl.(P_t)_{Tot})) + P_N} \qquad (5.3)$$

where ρ is the CPICH transmit power. It has been fixed in Table 4.1 to 10% of the maximum cell transmit power, ρ is then equal to 0.1.

FACH

The FACH transmission power is constant and defined as $-5dB$ relatively to the CPICH transmission power as explained in [15]. From Table 4.1, the CPICH transmission power is fixed at $33dBm$ meaning that the FACH is emitted at $28dBm$. The E_b/N_0 of the UE receiving a communication through the FACH is then evolving during the emulation depending on its position.

DSCH

The DSCH is one of the rare common transport channels which support fast power control. Its transmission power is managed using the fast power control of its associated DCH with a power offset. Considering that this offset is a mean to compensate the difference between the SFs of these two channels, it can be fixed to SF_{DCH}/SF_{DSCH} as in [16] giving the following equations:

$$(P_t)_{DSCH} = (P_t)_{DCH}.\frac{SF_{DCH}}{SF_{DSCH}} \tag{5.5}$$

From (5.1) and (5.4), one has:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}}.SF_{DSCH}.\frac{Sh.Pl.(P_t)_{DSCH}}{\alpha I_{intra} + I_{inter} + P_N} \tag{5.6}$$

Using (5.5), (5.6) becomes:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}}.SF_{DSCH}.\frac{Sh.Pl.(P_t)_{DCH}}{\alpha I_{intra}+I_{inter}+P_N}.\frac{SF_{DCH}}{SF_{DSCH}} \tag{5.7}$$

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}}.SF_{DCH}.\frac{Sh.Pl.(P_t)_{DCH}}{\alpha I_{intra}+I_{inter}+P_N} \tag{5.8}$$

Considering that $(\alpha I_{intra} + I_{inter} + P_N)$ has approximately the same value in both cases, one has:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} \cong \frac{1}{2R_{DSCH}}.SF_{DCH}.\left(\frac{C}{I}\right)_{DCH} \tag{5.9}$$

$$\left(\frac{E_b}{N_0}\right)_{DSCH} \cong \frac{R_{DCH}}{R_{DSCH}}.\left(\frac{E_b}{N_0}\right)_{DCH} \tag{5.10}$$

Supposing that the coding rate of these channels is equal, this comes to:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} \cong \left(\frac{E_b}{N_0}\right)_{DCH} \qquad (5.11)$$

In the testbed design, a UE using the DSCH is then considered as enjoying the same E_b/N_0 ratio as by its associated DCH.

Knowing the UE E_b/N_0 ratio in a shared transport channel and considering a coding rate of ½, the curve in Fig. 5.5 (taken from [13][3]) gives the corresponding BER a communication undergoes for each E_b/N_0 value.
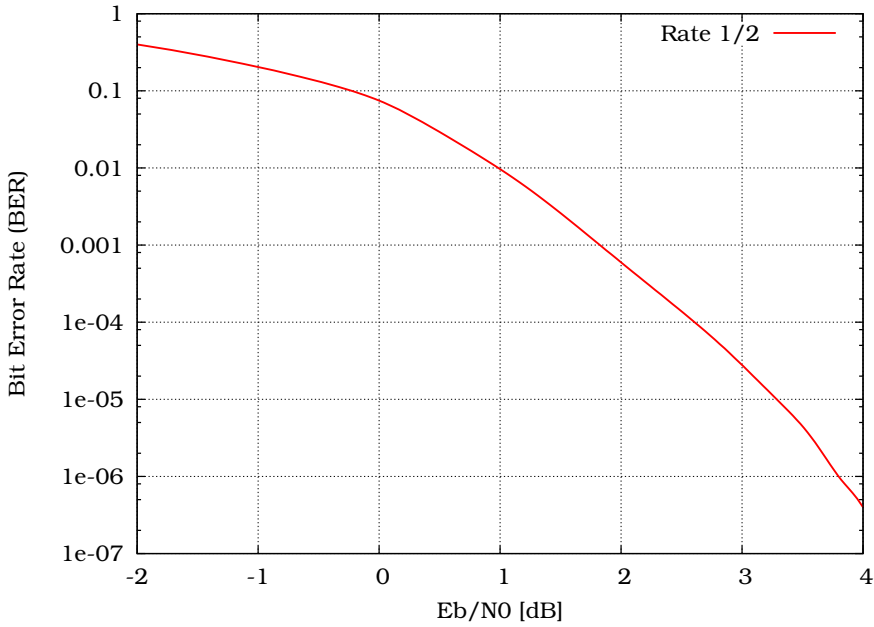
Figure 5.5: BER adopted in the testbed as a function of the bit normalised energy.

___

[3]It has not been possible to clearly identify the simulation assumptions (modulation, channel, speed, etc.) made by the authors to generate this curve. Use of this curve is to be regarded as illustration of the emulation process. Any relevant BER vs. E_b/N_0 curve could be substituted to this one.

DCH

On the DCH, due to its fast power control[4], the target E_b/N_0 set point is permanently adjusted and aims at maintaining a constant quality which is usually defined as a certain target BER, depending on the traffic class. Following the specifications presented in [17], the BER was fixed to 10^{-4} for the RT traffic and to 10^{-5} for NRT traffic.

Given that in DCH and DSCH, the transmit power of the NodeB varies to continually meet the transmission quality required by the traffic class, a mechanism runs on the testbed in order to check whether the total transmit power does not exceed the maximum transmit power of the NodeB. This check must be performed at the end of each iteration of the main loop (Fig. 2.5). The maximum transmit power is set to $43dB$ as shown in Table 4.1. Should the UEs globally require more than $43dB$, the RRM algorithm would drop the DCH requesting the largest part of the transmit power.

5.3.3 Resulting BLER

Assuming independent bit errors despite correlated errors at TTI level, the above computed BER allows the approximation of the resulting BLER as follows:

$$BLER = 1 - (1 - BER)^l \qquad (5.12)$$

where l is the PDU length which has been fixed to 336 bits as in [18]. This BLER is then injected into the modified version of NetEm with two other informations: the allocated SF and the selected RLC mode.

5.3.4 Shadow Fading Emulation

The shadow fading (also known as long-term or slow fading) is a phenomenon occurring especially outdoor. It is caused by the presence of obstacles situated in the propagation path of the radio waves between an emitter (NodeB) and its receiver (UE).

---

[4]Perfect power control is assumed throughout the thesis.

As presented in [19,20] , the shadow fading is a long-term (log normal) fading characterised by a Gaussian distribution with zero mean and standard deviation of $8dB$ as shown in Fig. 5.6(a). Due to the slow fading process versus distance Δx, adjacent fading values are correlated. Its normalised autocorrelation function $R(\Delta x)$ can be described with sufficient accuracy by an exponential function [20]:

$$R(\Delta x) = e^{-\frac{|\Delta x|}{d_{cor}}ln2} \tag{5.13}$$

with the decorrelation length d_{cor}, which is dependent on the environment. This concept can be applied in the vehicular test environment with a decorrelation length of $20m$ as advised by the 3GPP in [21]. Fig. 5.6(b) and 5.6(c) respectively show the shadow fading computed on a part of the emulated world before and after the correlation process.

The calculation of the shadow fading values in the emulated world is not simple due to computing limits. The granularity of the shadow fading map has been increased to $10m^2$ instead of $1m^2$. This gives a 485×560 shadow fading map $\mathcal{SH_W}$:

$$\mathcal{SH_W} = \begin{bmatrix} SH_{1,1} & SH_{1,2} & \dots & SH_{1,560} \\ SH_{2,1} & SH_{2,2} & \dots & SH_{2,560} \\ \vdots & \vdots & \ddots & \vdots \\ SH_{485,1} & SH_{485,2} & \dots & SH_{485,560} \end{bmatrix}$$

Here follows an explanation of the methodology applied to achieve the shadow fading map of the emulated world. Taking as example a $30m^2$ emulated world, $\mathcal{SH_W}$ is set as a 3×3 matrix:

$$\mathcal{SH_W} = \begin{bmatrix} SH_{1,1} & SH_{1,2} & SH_{1,3} \\ SH_{2,1} & SH_{2,2} & SH_{2,3} \\ SH_{3,1} & SH_{3,2} & SH_{3,3} \end{bmatrix}$$

To generate this matrix, the first step is to create another matrix $\mathcal{C}$ containing the correlation coefficient that links the shadow
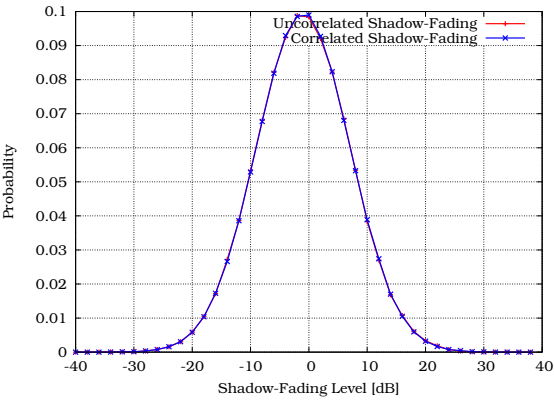
(a) Shadow fading log normal distribution.
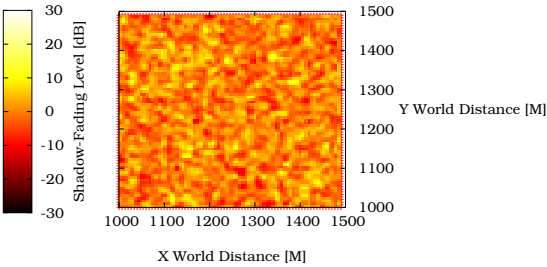
(b) Uncorrelated shadow fading.
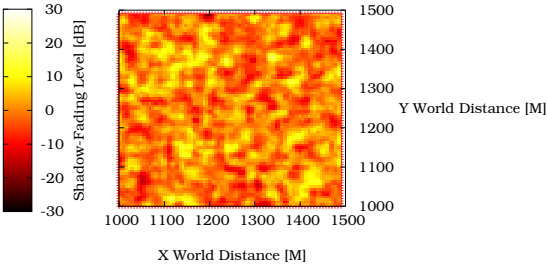
(c) Correlated shadow fading.

Figure 5.6: Emulated shadow fading.

fading value computed in one part of the emulated world to the other parts. Considering a decorrelation length of $20m$, the correlation coefficient linking $SH_{1,1}$ to:

- itself is 1,

- $SH_{1,2}$ and $SH_{2,1}$ is ρ_1,

- SH_{22} is ρ_2,

- $SH_{1,3}$ and $SH_{3,1}$ is ρ_3, and

- other shadow fading values area is 0.

Given the testbed specifications and following (5.13):

$$
\begin{aligned}
\rho_1 &= & e^{-\frac{10}{20}ln2} & \cong 0.71 \\
\rho_2 &= & e^{-\frac{\sqrt{10^2+10^2}}{20}ln2} & \cong 0.61 \\
\rho_3 &= & e^{-\frac{20}{20}ln2} & = 0.50
\end{aligned}
$$

The correlation matrix $\mathcal{C}$ is then:

$$
\mathcal{C} = \begin{bmatrix}
cc_{1,1} & cc_{1,2} & cc_{1,3} & cc_{1,4} & cc_{1,5} & cc_{1,6} & cc_{1,7} & cc_{1,8} & cc_{1,9} \\
cc_{2,1} & cc_{2,2} & cc_{2,3} & cc_{2,4} & cc_{2,5} & cc_{2,6} & cc_{2,7} & cc_{2,8} & cc_{2,9} \\
cc_{3,1} & cc_{3,2} & cc_{3,3} & cc_{3,4} & cc_{3,5} & cc_{3,6} & cc_{3,7} & cc_{3,8} & cc_{3,9} \\
cc_{4,1} & cc_{4,2} & cc_{4,3} & cc_{4,4} & cc_{4,5} & cc_{4,6} & cc_{4,7} & cc_{4,8} & cc_{4,9} \\
cc_{5,1} & cc_{5,2} & cc_{5,3} & cc_{5,4} & cc_{5,5} & cc_{5,6} & cc_{5,7} & cc_{5,8} & cc_{5,9} \\
cc_{6,1} & cc_{6,2} & cc_{6,3} & cc_{6,4} & cc_{6,5} & cc_{6,6} & cc_{6,7} & cc_{6,8} & cc_{6,9} \\
cc_{7,1} & cc_{7,2} & cc_{7,3} & cc_{7,4} & cc_{7,5} & cc_{7,6} & cc_{7,7} & cc_{7,8} & cc_{7,9} \\
cc_{8,1} & cc_{8,2} & cc_{8,3} & cc_{8,4} & cc_{8,5} & cc_{8,6} & cc_{8,7} & cc_{8,8} & cc_{8,9} \\
cc_{9,1} & cc_{9,2} & cc_{9,3} & cc_{9,4} & cc_{9,5} & cc_{9,6} & cc_{9,7} & cc_{9,8} & cc_{9,9}
\end{bmatrix}
$$

where $cc_{i,j}$ represents the correlation coefficient between $SH_{i/3,i\%3}$ and $SH_{j/3,j\%3}$. $\mathcal{C}$ is symmetric and its diagonal is made of 1 only. Expressed with ρ values, $\mathcal{C}$ becomes:

$$\mathcal{C} = \begin{bmatrix} 1 & \rho_1 & \rho_3 & \rho_1 & \rho_2 & 0 & \rho_3 & 0 & 0 \\ \rho_1 & 1 & \rho_1 & \rho_2 & \rho_1 & \rho_2 & 0 & \rho_3 & 0 \\ \rho_3 & \rho_1 & 1 & 0 & \rho_2 & \rho_1 & 0 & 0 & \rho_3 \\ \rho_1 & \rho_2 & 0 & 1 & \rho_1 & \rho_3 & \rho_1 & \rho_2 & 0 \\ \rho_2 & \rho_1 & \rho_2 & \rho_1 & 1 & \rho_1 & \rho_2 & \rho_1 & \rho_2 \\ 0 & \rho_2 & \rho_1 & \rho_3 & \rho_1 & 1 & 0 & \rho_2 & \rho_1 \\ \rho_3 & 0 & 0 & \rho_1 & \rho_2 & 0 & 1 & \rho_1 & \rho_3 \\ 0 & \rho_3 & 0 & \rho_2 & \rho_1 & \rho_2 & \rho_1 & 1 & \rho_1 \\ 0 & 0 & \rho_3 & 0 & \rho_2 & \rho_1 & \rho_3 & \rho_1 & 1 \end{bmatrix}$$

The second step in this methodology is to compute the square root of $\mathcal{C}$. To do so, the easiest way is to apply the Cholesky factorisation since $\mathcal{C}$ is positive semi-definite:

$$\mathcal{C} = \mathcal{Z}^T \times \mathcal{Z} \Rightarrow \sqrt{\mathcal{C}} = \mathcal{Z} \tag{5.14}$$

This kind of operation is typically disc capacity and computing power consuming, especially with large matrices. For this reason, this computation has been coded in Fortran77 using a specialised compiler coupled with specific math libraries [22]. The Intel math kernel compiler offers highly optimised math routines for scientific applications that require maximum computer performance, whereas the Basic Linear Algebra Subprograms (BLAS) and Linear Algebra PACKage (LAPACK) libraries were created to solve a large variety of linear algebra problems.

The third and final step to create the shadow fading map is to multiply $\mathcal{Z}$ by a vector $\mathcal{V}$ containing log normal values drawn randomly following the specifications given at the beginning of this section:

$$Vec(\mathcal{SH_W}) = \mathcal{Z} \times \mathcal{V} \tag{5.15}$$

where the $Vec()$ operator vectorises a matrix by stacking its columns. For example:

$$\mathcal{M} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \; Vec(\mathcal{M}) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

This will create the desired matrix including correlated log normal values, representing shadow fading levels the UEs will have to undergo in every $10m^2$ area of the emulated world.

5.3.5   Gfactor Validation

As a proof of interference level validation, the Gfactor has been calculated for every possible position in the emulated world. As presented in [23]:

$$Gfactor \;=\; \frac{I_{intra}}{I_{inter}} \tag{5.16}$$

the Gfactor is the ratio of intracell/intercell interference. Following the OWN site Plus 2 other sector interference model (OWNP2) approach presented in [24], the intercell interference is calculated based on the received power from the two other sectors of the SNodeB plus the received power from 2 other sectors generating the most interference.

Fig. 5.7 clearly shows the good match between the testbed's results and those presented in literature [24, 25].

5.4   RLC Modes

The UMTS RLC layer is able to transmit RLC PDUs in three different modes: AM, Unacknowledged Mode (UM) and TM. Below is a brief description of each of those modes.
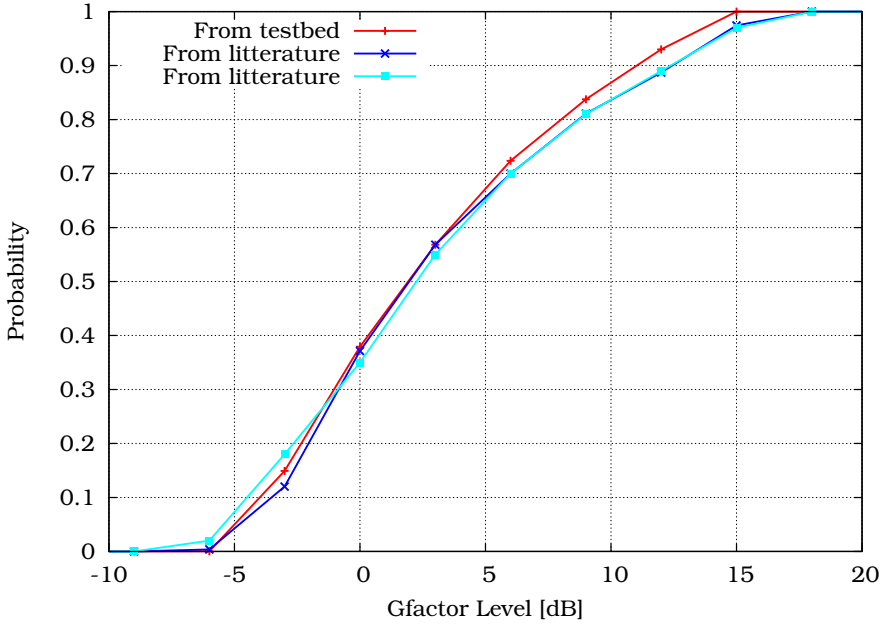
Figure 5.7: Gfactor distribution in the emulated world.

5.4.1   Transparent Mode

The TM is the simplest mode. A TM RLC entity does not add any RLC protocol overhead to SDUs. The entity is transparent to SDUs.

5.4.2   Unacknowledged Mode

A UM RLC entity does not guarantee correct deliveries of RLC PDUs to the peer entity, but it adds RLC overheads to SDUs. This header is made up of a sequence number. Since RLC PDUs are numbered, a UM entity can provide segmentation of SDUs and their reassembling at the receiver side.

If a receiving UM entity detects a missing RLC PDU, the entity automatically discards all RLC PDUs that belong to the same SDU. The UM is mainly utilised for transfers of cell broadcast messages or for transfers of certain RRC messages.

5.4.3   Acknowledged Mode

AM is the most elaborated RLC mode. A RLC entity is config-
ured to operate in AM if a reliable data transfer over the UMTS
radio interface is required. Compared to a UM RLC entity, an
AM RLC entity can be configured to deliver reassembled SDUs
to the upper layer either in sequence or out of sequence.

An AM RLC entity uses the Automatic Repeat reQuest (ARQ)
technique in order to provide reliable data transfers and handle
errors by retransmitting data according to receiver demands.
The ARQ technique requires one bi-directional connection be-
tween the transmitter and the receiver, i.e. the UE and the
NodeB. One direction is reserved for data transmission from the
transmitter to the receiver. The opposite direction is used by
the receiver to transfer status reports. Through status reports,
the receiver informs the transmitter about the RLC PDUs which
have to be retransmitted. The ARQ mechanism can retransmit
an erroneous PDU a fixed number of times before declaring this
PDU as lost. The number of retransmissions is referred as max-
DAT and is fixed at 3 in the emulation following the proposition
made in [26, 27]. The ARQ mechanism can manage either just
one PDU (Stop and Wait) or several PDUs at a time (Go-Back-N,
Selective Repeat). Erroneous PDUs at the receiver side are dis-
carded, as the receiver does not store them in its receiver buffer.
The UMTS RLC protocol uses the ARQ technique of Selective
Repeat type.

5.4.4   Testbed Adaptation

The present dissertation focuses only on two of the RLC mo-
des: TM and AM since cell broadcast and RRC messages trans-
ferring are beyond the scope of this work. The TM is more likely
to transmit the delay sensitive RT traffic. On the other hand,
AM is more suitable to lower the PER for NRT traffic. AM and
TM have been adapted to an Ethernet link and included in the
enhanced version of the TC NetEm component.

In TM (Fig. 5.8(a)), when an IP packet P arrives, NetEm com-
putes the numbers of TTIs needed to transmit P over the air
interface. This number is referred to as m. This division is

(a) TM mode.   (b) AM mode.

Figure 5.8: RLC modes implemented in the TC NetEm component.
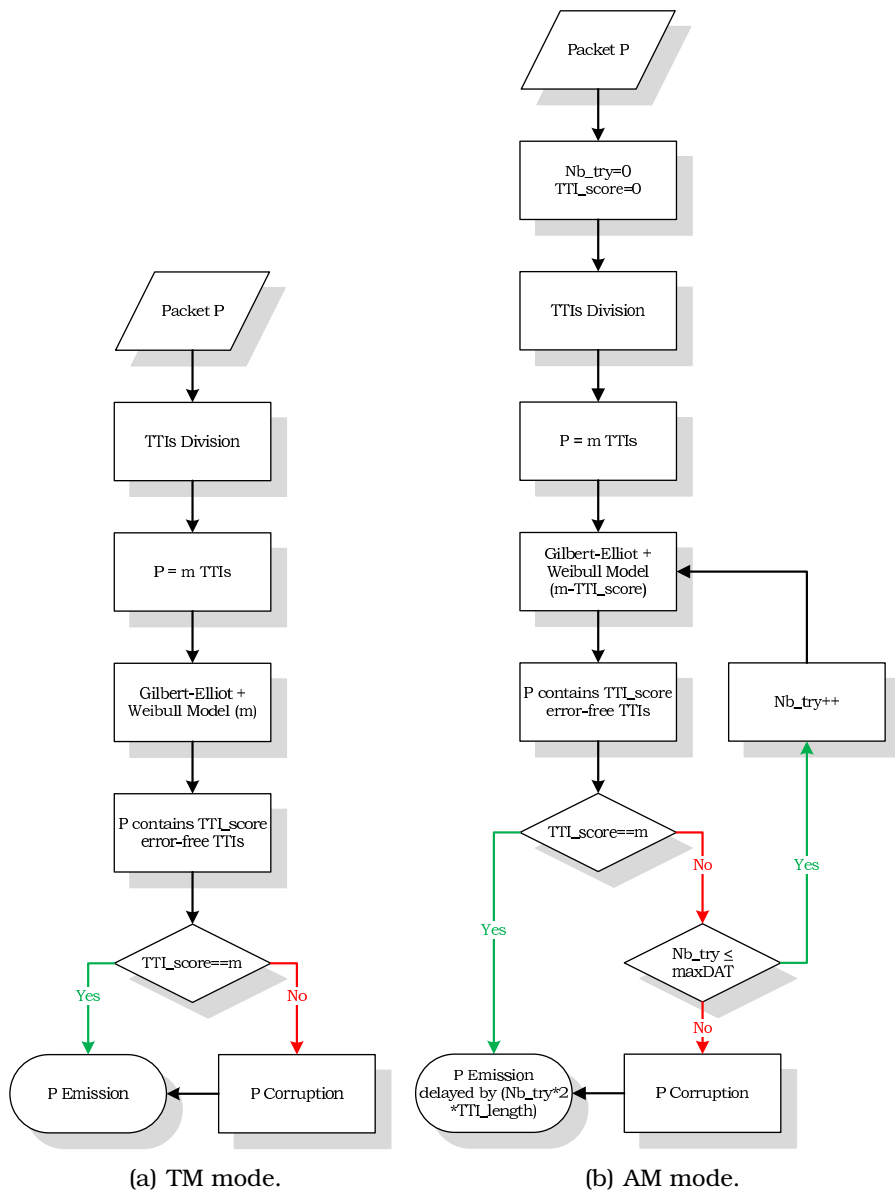
based on P's length and on the SF allocated to the flow P be-
longs to. All these m TTIs are then injected into the two-state
Markovian chain modified by the Weibull distribution presented
in Section 5.3.1 to calculate the number of error-free TTIs re-
ferred to as TTI_score. If P does not have any erroneous TTI
(meaning m is equal to TTI_score), P is correctly transmitted.
However, if P has at least one erroneous TTI, P is corrupted
before being transmitted to its final destination.

In AM (Fig. 5.8(b)), the computation is similar to the compu-
tation in TM, but if P contains erroneous TTIs, $(m - TTI\_score)$
TTIs are reinjected into the Markovian chain until all the TTIs
needed to transmit P are error-free, or until the number of at-
tempts (referred to as Nb_try) exceeds maxDAT. If all TTIs are
error-free, P is transmitted with a $Nb\_try \times 2 \times TTI\_length$ delay
including the ARQ delay due to the retransmissions. If there
is still at least one erroneous TTI at the end of the computa-
tion, P is corrupted and then transmitted with a $(maxDAT +
1) \times 2 \times TTI_length$ delay. It is worth nothing that the delay
formula includes a TTI_length multiplied by 2 in order to rep-
resent the RTT over the air interface. An erroneous RLC PDU
consumes 1 TTI_length to transit from the transmitter to the
receiver and the feed back asking for a retransmission also con-
sumes 1 TTI_length. In a Rel'99 UMTS network, as well as in
the testbed, the TTI_length is fixed to $10ms$.

5.5   Implementation

Using TC and its TB component to mimic the different avail-
able transport channels, and using the enhanced NetEm com-
ponent to emulate the packet loss over the UMTS air interface,
a structure can now be built to manage all the packets going
from a SNodeB to one of its UEs [28]. Fig. 5.9 illustrates the
implementation of one emulated NodeB:

1 First, a new downlink data flow is filtered based on the des-
   tination IPv6 address and the $8 - bit$ "Traffic Class" field of
   the IPv6 header. This field is initialised on the sender side
   by TG. The injected value varies following three parameters:
   the UE's User Profile, the transport Channel Used over the
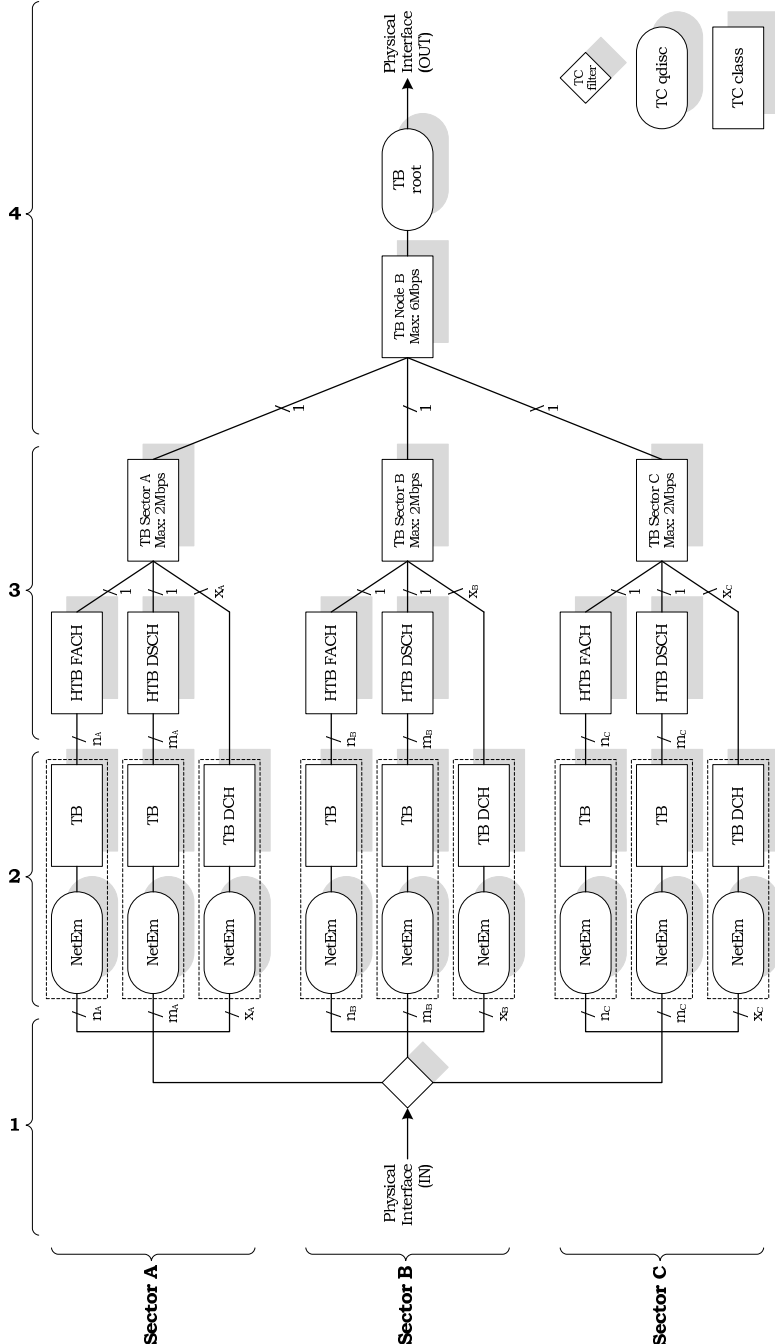
Figure 5.9: Air interface emulation.

Table 5.3: IPv6 header "Traffic Class" field.

User Profile		Channel User			Traffic Class			Comments		
1	1	1	0	0	1	1	1	Platinum	DCH	Conversational
1	1	1	0	0	1	0	0	Platinum	DCH	Interactive
1	1	0	1	0	1	0	0	Platinum	DSCH	Interactive
1	1	0	0	1	1	0	0	Platinum	FACH	Interactive
1	1	1	0	0	1	1	0	Platinum	DCH	Streaming
1	1	1	0	0	0	0	0	Platinum	DCH	Background
1	1	0	1	0	0	0	0	Platinum	DSCH	Background
1	1	0	0	1	0	0	0	Platinum	FACH	Background
1	0	1	0	0	1	1	1	Gold	DCH	Conversational
1	0	1	0	0	1	0	0	Gold	DCH	Interactive
1	0	0	1	0	1	0	0	Gold	DSCH	Interactive
1	0	0	0	1	1	0	0	Gold	FACH	Interactive
1	0	1	0	0	1	1	0	Gold	DCH	Streaming
1	0	1	0	0	0	0	0	Gold	DCH	Background
1	0	0	1	0	0	0	0	Gold	DSCH	Background
1	0	0	0	1	0	0	0	Gold	FACH	Background
0	1	1	0	0	1	1	1	Silver	DCH	Conversational
0	1	1	0	0	1	0	0	Silver	DCH	Interactive
0	1	0	1	0	1	0	0	Silver	DSCH	Interactive
0	1	0	0	1	1	0	0	Silver	FACH	Interactive
0	1	1	0	0	1	1	0	Silver	DCH	Streaming
0	1	1	0	0	0	0	0	Silver	DCH	Background
0	1	0	1	0	0	0	0	Silver	DSCH	Background
0	1	0	0	1	0	0	0	Silver	FACH	Background
0	0	1	0	0	1	1	1	Bronze	DCH	Conversational
0	0	1	0	0	1	0	0	Bronze	DCH	Interactive
0	0	0	1	0	1	0	0	Bronze	DSCH	Interactive
0	0	0	0	1	1	0	0	Bronze	FACH	Interactive
0	0	1	0	0	1	1	0	Bronze	DCH	Streaming
0	0	1	0	0	0	0	0	Bronze	DCH	Background
0	0	0	1	0	0	0	0	Bronze	DSCH	Background
0	0	0	0	1	0	0	0	Bronze	FACH	Background

air interface and the UMTS Traffic Class of the flow the packet belongs to. Table 5.3 regroups all the different values the "Traffic Class" field could have. The filtering could also easily be based on the Differentiated Services Code Point (DSCP) of an arriving packet. From these header fields, the testbed knows exactly in which sector the receiving UE is located and which transport channel this flow will use over the air interface.

2 A new instance of a NetEm queue connected to a TB is then created for each data flow with a default finite queue size of $1,000$ packets. It corrupts some of these packets according to the principles exposed in the previous sections. The TB is only responsible of the bandwidth consumption of the flow ensuring that it never exceeds the maximum bandwidth it has received via the SF allocation. In this architecture, there is a NetEm-TB instance for each single data flow. There are n_i flows sharing the FACH, m_i flows sharing the DSCH and x_i flows each of them occupying a single DCH of the sector i (i=A, B or C).

3 If the flow is using a common channel (FACH or DSCH), it has to share the channel bandwidth with other flows. All NetEm-TB instances belonging to the same common channel are supervised by a Hierarchical TB (HTB) which can borrow some unused tokens from a given TB to transfer them to another TB waiting for tokens. If the flow is using a DCH, it is directly scheduled at the TB responsible for a sector. This TB has to equitably manage (x_i+2) entries: the x_i DCHs + 1 FACH + 1 DSCH of the sector i (i=A, B or C). It also has to guarantee that the total bandwidth consumption of the sector never exceeds $2Mbps$. Each entry of the (H)TB is managed one after the other in a round-robin scheme. This scheduling is fast enough so that the impossibility to simultaneously transmit packets over Ethernet does not impair the E2E performances.

4 The three TBs managing the three sectors are controlled by the last TB representing the NodeB itself. The physical $100Mbps$ Ethernet link is assumed to support a maximal

traffic bandwidth of $6Mbps$. The overdimensioning of the physical link hints that it has no influence on the emulation.

The modularity of the testbed appears in Fig. 5.9.

This implementation requires modifications both in files of Iproute2, which is a collection of utilities for controlling TCP/IP networking and traffic control in Linux, and the NetEm kernel module. These are detailed in the Appendix F.

5.6   Testbed Validation

To validate the testbed implementation, cross-check numerical computations of an analytical model versus emulation results are to be done.

5.6.1   UMTS Scenario

Let consider the downlink of a Rel'99 UMTS cell. The NodeB supervises the delivery of traffic to the users it is serving. This downlink traffic is simply an aggregation of Conversational and Interactive traffic, as previously presented in Chapter 3. These traffic classes are delivered over the air through three different transport channels, as introduced in Section 5.2.2. Under these emulation assumptions, the DCHs will only transport the Conversational traffic. On the other hand, the two other transport channels, the DSCH and the FACH, carry only the Interactive traffic. Whether a given user traffic ends up in a DSCH or a FACH depends on whether the user is simultaneously transmitting through a DCH or not. When a user has already access to a DCH, its Interactive traffic can be transported through a DSCH. With no open DCH connection, his/her traffic will pass through the FACH.

The present validation route studies the fluctuations of the shared DSCH bandwidth, looking to the competing activity of open DCHs and the FACH of the cell. The data is assumed to be spread with a SF 64, and is half-rate coded, which means that each transport channel is eager to achieve a bandwidth multiple

of $30kbps$. At any given time, the full downlink bandwidth could be split into four types of transport channels:

- The FACH of the cell (bandwidth fixed to $30kbps$),

- The $K(t)$ DCHs transporting the $K(t)$ currently active voice connections (each with $30kbps$),

- The DSCH carrying the signalling of the open DCHs (fixed to $30kbps$),

- The DSCH transporting the Interactive traffic of the users already having a DCH.

As shown in Fig. 5.10, the DSCH enjoys residual bandwidth. In a real UMTS system, a single DSCH carries both DCH signalling and Interactive traffic. These traffics have been separated in order to isolate the fluctuating part of the shared DSCH bandwidth. Perfect fast power control is assumed. Every user transmitting in a DCH will enjoy the full prescribed $30kbps$ bandwidth, despite of fading, shadowing, intra- and inter-cell interference. With this model, a NodeB can process up to 62 simultaneous voice connections.
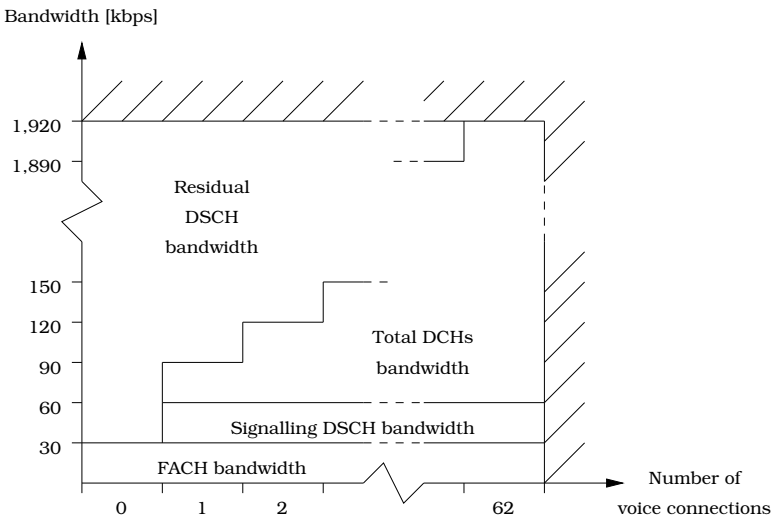
Figure 5.10: Residual DSCH bandwidth vs. number of voice connections.

The DSCH buffer is modelled at layer 2. Interactive traffic is composed of bursts of random size generated by the active users which instantaneously accumulate in the shared DSCH downlink buffer, as delivered by the higher layer. The model of the Interactive traffic captures the fluctuating number of active users. The more active users, the more intense the Interactive traffic.

As perfect fast power control is assumed, the transmit power of a given mobile user is continuously adjusted in order to achieve a fixed BLER at reception. To compute a simple analytic model, Interactive traffic is assumed to be handled by the RLC in TM, there is no reliability mechanism in layer 2, i.e. no link retransmissions. Additionally, to limit complexity, this specific Interactive traffic is supposed to be transported on UDP, with layer 4 retransmissions not taken into account.

Traffic is then transmitted at a speed that depends on $K(t)$. This transmission speed $R(t)$ is determined as follows:

$$R(t) = \begin{cases} 64 - [K(t) + 2] \times 30kbps & \text{when } 1 \leq K(t) \leq 62 \\ 63 \times 30kbps & \text{when } K(t) = 0 \end{cases} \qquad (5.17)$$

5.6.2 Markov-Driven Fluid Model

Markov fluid processes are appealing models for the analysis of complex telecommunication systems. Such mathematical models can be analysed using matrix-analytic techniques and accordingly, efficient algorithms can be developed in order to measure performance of modelled telecommunication systems. A new model of the DSCH buffer in Rel'99 UMTS cellular systems is proposed. This new model is based on Markov fluid processes. The available bandwidth dedicated to DSCH traffic is limited by voice connections carried by DCHs on the one hand, and by traffic on the FACH on the other hand. The resulting model for the DSCH buffer content is shown to be a Markov driven fluid queue with jumps.

For more information about the stochastic processes assumed in the system and the Markov-driven fluid model, the reader should refer to [29].

5.6.3 **Numerical Analysis**

The tests have used the following parameters: The burst arrivals are following a Poisson process in time of rate λ. Their size is in average $18 \times 30kbps$. The Poisson process is a particular case of Markovian Arrival Process (MAP). Voice communication appears at a rate of one every $100s$ in average and lasts for $40s$ on average. Buffer size is about $50 \times 30kbps$.
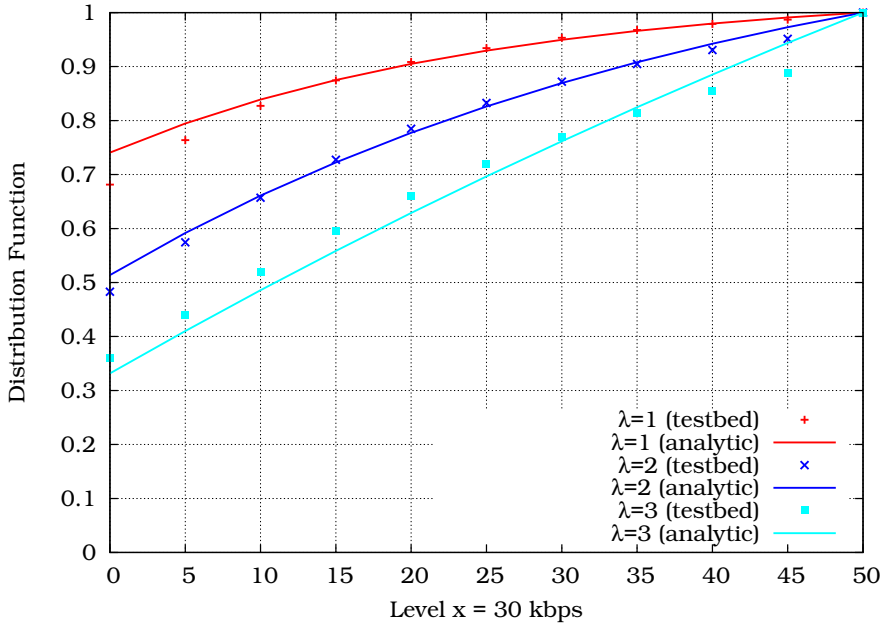
Figure 5.11: Stationary buffer content distribution function for different burst arrival rates.

Fig. 5.11 presents the distribution function of the stationary buffer content when λ ranges from 1 to 3. The curves represent the outcome of the numerical computations of the analytical model, while the markers stand for the emulation results obtained on the testbed. Each emulation run lasted $600s$, with up to 62 simultaneously active users.

There seems to be a good match between the analytical model and the results of the testbed. A goodness-of-fit test, the one-sample Kolmogorov-Smirnov test, has successfully been applied to the samples. Moreover, the previous assumption on the lim-

itation of the loss to the exceeding part of the burst when the buffer is saturated does not significantly affect the validity of the model. As far as the behaviour of the buffer is concerned, it appears that increasing lambda comes with increased mass probability on high buffer content levels.

5.7   Summary

This chapter has presented a methodology for emulating a wireless link above a wired one based on TC and some of its components namely TB, HTB and NetEm. Following this strategy, the testbed emulates the Rel'99 UMTS air interface, the different transport channels and their SF-based limited bandwidth, the higher BER and its time-correlated errors property. Finally, the testbed validation has been detailed by confronting results obtained through an analytical model using the Markov fluid process with measurements obtained on the testbed.

Bibliography

[1] Bert Hubert, Thomas Graf, Gregory Maxwell, Remco van Mook, Martijn van Oosterhout, Paul Schroeder, Jasper Spaans, and Pedro Larroy. Linux Advanced Routing & Traffic Control HOWTO. Technical Report 1.43, October 2003. `http://lartc.org/`, last visited: March 13, 2008.

[2] Stephen Hemminger. Network Emulation with NetEm. In *Proceedings of the 6th Australia's National Linux Conference (LCA)*, Canberra (Australia), April 2005.

[3] Jaana Laiho, Achim Wacker, and Tomáš Novosad. *Radio Network Planning and Optimisation for UMTS*. Wiley inter-science, second edition, 2006.

[4] Harri Holma and Antti Toskala. *WCDMA for UMTS*. Wiley inter-science, 3rd edition, October 2004.

[5] Proximus. `http://www.proximus.be`, last visited: March 13, 2008.

[6] Mobistar. `http://www.mobistar.be`, last visited: March 13, 2008.

[7] Base. `http://www.base.be`, last visited: March 13, 2008.

[8] Edgar Gilbert. Capacity of a Burst-Noise Channel. *Bell Systems Technical Journal*, 39:1253–1265, September 1960.

[9] E. O. Elliot. Estimates of Error Rates for Codes on Burst-Noise Channels. *Bell Systems Technical Journal*, 42:1977–1997, September 1963.

[10] Wolfgang Karner. A UMTS DL DCH BLER Model Based on Measurements in Live Networks. Technical report, COST 273, January 2005.

[11] Hugues Van Peteghem and Laurent Schumacher. Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based $Rel'99$ UTRAN Testbed. Technical Report TD(07)011, COST 2100, February 2007.

[12] Technical Specification Group Radio Access Network. Radio Frequency (RF) system scenarios (Release 6). Technical Report TR 25.942 V7.0.0, 3rd Generation Partnership Project (3GPP), March 2007.

[13] Flaminio Borgonovo, Antonio Capone, Matteo Cesana, and Luigi Fratta. Packet service in UMTS: delay-throughput performance of the downlink shared channel. *Computer Networks*, 38(1):43–59, January 2002.

[14] Jordi Pérez-Romero, Oriol Sallent, and Ramon Agustí. On Dimensioning UTRA-FDD Downlink Shared Channel. In *Proceedings of the 15th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Barcelona (Spain), September 2004.

[15] Harri Holma and Antti Toskala. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. Wiley interscience, April 2006.

[16] Paolo Giacomazzi, Luigi Musumeci, and Giacomo Verticale. Performance of Web-browsing services over the WCDMA-FDD downlink shared channel. In *Proceedings of the 44th Globecom conference*, San Antonio (TX - USA), November 2001.

[17] Technical Specification Group Radio Access Network. Quality of Service (QoS) concept and architecture. Technical Report TS 23.107 V7.1.0, 3rd Generation Partnership Project (3GPP), September 2007.

[18] Special Article on IMT-2000 Services. *NTT DoCoMo Technical Journal*, 3(2), September 2001.

[19] Jay Kumar Sundararajan and Harish Viswanathan. Comparison of Schemes for Streaming Multicast in Cellular Networks with Relays. In *Proceedings of the 7th Wireless Communications and Networking Conference (WCNC)*, Las Vegas (NV - USA), March 2006.

[20] Thomas Klingenbrunn and Preben Elgaard Mogensen. Modelling Cross-Correlated Shadowing in Network Simulations. In *Proceedings of the 50th Vehicular Technology Conference (VTC-fall)*, Amsterdam (The Netherlands), October 1999.

[21] Technical Specification Group Radio Access Network. Selection procedures for the choice of radio transmission technologies of the UMTS. Technical Report TS 30.03 V3.2.0, 3rd Generation Partnership Project (3GPP), March 1998.

[22] Intel Corporation, Hewlett-Packard Development Company, and The Numerical Algorithms Group. Intel® Math Kernel Library, Reference Manual. Technical Report 630813-025US, April 2007. `http://developer.intel.com`, last visited: March 13, 2008.

[23] Javier Rodríguez Fonollosa, Markku Heikkilä, Xavier Mestre, Alba Pagès, Adam Pollard, Laurent Schumacher, Lars Torsten Berger, Ami Wiesel, and Juha Ylitalo. Adaptive Modulation Schemes for MIMO HSDPA. In *Proceedings of the 11th IST Mobile & Wireless Telecommunications Summit*, pages 78–82, Thessaloniki (Greece), June 2002.

[24] Lars Torsten Berger. *Performance of Multi-Antenna Enhanced HSDPA*. PhD thesis, Aalborg University, April 2005.

[25] Howard Huang, Sivarama Venkatesan, Achilles Kogiantis, and Sharma Naresh. Increasing the peak data rate of 3G downlink packet data systems using multiple antennas. In *Proceedings of the 57th Vehicular Technology Conference (VTC-spring)*, Jeju Island (Korea), April 2003.

[26] Oumer Mohammed Teyeb. *Quality of Packet Services in UMTS and Heterogeneous Networks*. PhD thesis, Aalborg University, October 2006.

[27] Olivier De Mey, Laurent Schumacher, and Xavier Dubois. Optimum Number of RLC Retransmissions for Best TCP Performance in UTRAN. In *Proceedings of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Berlin (Germany), September 2005.

[28] Hugues Van Peteghem and Laurent Schumacher. Implementation of an Open-Source UTRAN Testbed. In *Proceedings of the 13th Symposium on Communications and Vehicular Technology (SCVT)*, Liege (Belgium), November 2006.

[29] Marie-Ange Remiche, Laurent Schumacher, Miguel de Vega Rodrigo, and Hugues Van Peteghem. Performance of Downlink Shared Channels in UMTS Scenarios Using Markov Fluid Process. In *Proccedings of the 14th International Conference Analytical and Stochastic Modelling Techniques and Application (ASMTA)*, Prague (Czech Republic), June 2007.

Part III

Results

Chapter 6

Spreading Factor Allocation Map

6.1 Introduction

As introduced in Section 5.2.1 and in [1], the SF allocation strategy located in the RNC is an essential part of the radio link management. Two opposite approaches of the SF allocation issue shall be described: the users' perspective trying to receive the lowest SF to obtain the highest possible bandwidth, and the operators' perspective trying to load as much as possible the OVSF tree to accept a maximum number of sessions within the same sector. Both strategies were injected in the testbed and have been compared with respect to QoS metrics such as the 95th percentile of the packet transfer delay, cell throughput and blocking rate. Based on this comparison, it was possible to identify an optimal SF allocation strategy for the testbed scenario.

6.2 Users' Perspective

6.2.1 Emulation Assumptions

Different scenarios testing SF allocation for the four traffic classes presented in Chapter 3 have been run on the testbed. The idea was to find a range of SFs that would allow meeting the packet transfer delay requirements of each of the traffic classes.

Alongside the implementation assumptions already introduced in Part II, the other specific emulation assumptions for this experimentation were the following.

- The emulation runs were performed with a single static UE, to avoid any code starvation/blocking and handover influences at this stage of the study.

- The different kinds of traffic were generated with an Inter-Session Arrival (ISA) mean of $300s$.

- The emulation length has been fixed to $5,600s$ which is sufficient to capture a statistically significant number of packets.

6.2.2   Allocation Map

Following the definition given by the 3GPP in [2], the packet transfer delay (expressed in ms) indicates the maximum delay for the 95th percentile of the delay distribution for delivered packets during the lifetime of a bearer service. The transfer delay of an arbitrary packet is not meaningful for a bursty source, since the last packets of a burst may suffer from long delays due to queueing, whereas the meaningful response delay perceived by the user is the delay of the first packet of the burst. For this reason, the QoS metric is preferably based on the 95th percentile.

Figs. 6.1-6.2 represent the packet transfer delay during the emulation runs. This delay is computed by post-processing the log files of the synthetic traffic generator program [3]. It corresponds to a user experienced delay since it is calculated at the application layer. The packet delay only refers to delaying in the UTRAN. CN delays were to be added to obtain the full E2E delay of the packets.

Fig. 6.1 depicts the results obtained for the RT flows which have a stringent delay requirement. The access network delay is upper-bounded by $80ms$ in the Conversational case whereas it is fixed to $250ms$ in the Streaming case. These limits were set considering the recommendations made by the 3GPP in [2] to leave some time budget margin for delays in the CN.
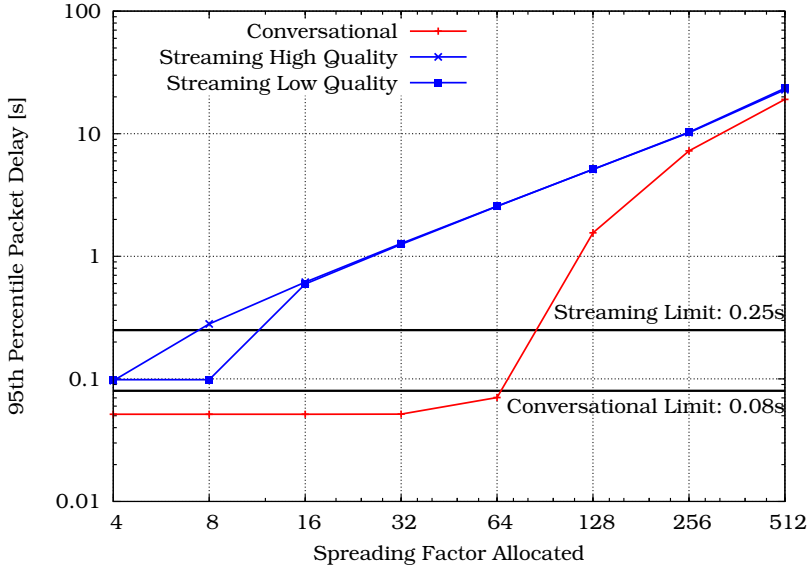
Figure 6.1: RT traffic packet delay vs. allocated SF.

As mentioned in Section 3.3, the Conversational class is based on a $12.2kbps$ AMR vocoder. The network data flow hence never exceeds $20kbps$. Given the above, it would be a waste of network resources to allocate a $SF < 64$, whatever the user profile is. This can be seen in Fig. 6.1 where the transfer delay dramatically increases with $SFs > 64$.

The Streaming flows are divided into two categories: high and low quality. This traffic class is the most bandwidth demanding, so it will require the lowest SF. Fig. 6.1 shows that the high quality undergoes unacceptable packet delay using $SF > 4$. The same conclusion can be drawn for the low quality streaming with $SF > 8$. Therefore, the possibility to use a $SF = 4$ (high quality streaming) is agreed only to Platinum users. This huge radio resources consumption representing a quarter of the OVSF tree of one sector is balanced with the rarity of such a request since only 5% of the emulated population is supposed to have contracted a Platinum subscription as explained in Section 5.2.1. Should the OVSF tree not accept this request, Platinum users would enjoy a lower quality streaming session with $SF = 8$, the only SF available for the other user profiles.
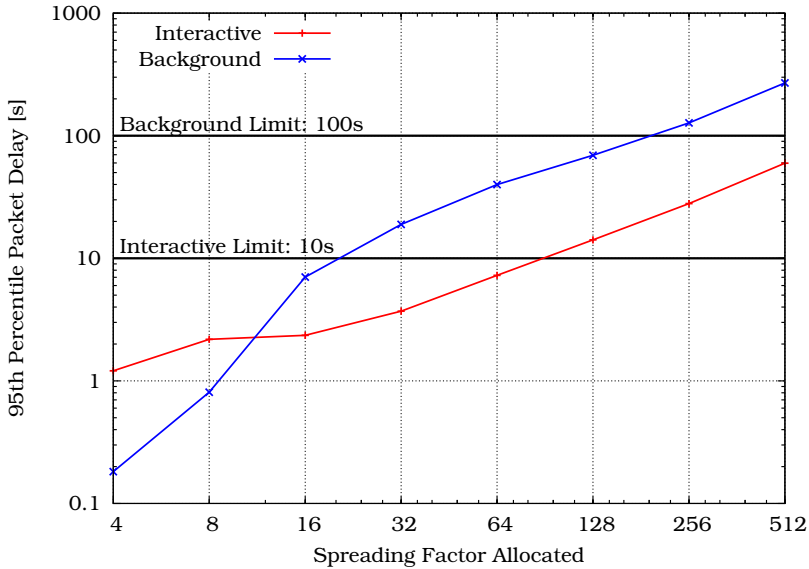
Figure 6.2: NRT traffic packet delay vs. allocated SF.

The NRT traffic packet transfer delays are shown in Fig. 6.2. The NRT traffic sets less stringent delay requirements. However, the Interactive class involves human participation and then requires a delay respecting this responsive behaviour ($< 10,000ms$). The packet delay is not that much important for the Background traffic. Nevertheless, a 95th percentile of the packet delay higher than $100,000ms$ should be avoided.

Since the nature of the Interactive traffic is made up of multiple little downloads (i.e. successive pages) separated with page reading times, the bandwidth required for this traffic class is rather small. The SF allocation logic here is to allocate a $16-chip$ spreading code to the Platinum and Gold users allowing them to downgrade their request to a SF $= 64$ if the OVSF tree has not enough room to accept their original request. In a similar way, the Silver users begin with SF $= 32$ whereas the Bronze users can only access a SF $= 64$.

The Background traffic represents massive downloads and thus, even if it does not require a small mean packet delay, it has to receive some bandwidth. For this reason, the Platinum users receive a SF $= 16$ which can be degraded down to 128. Gold

users have the ability to access a SF as small as 32, a SF $= 64$ for Silver users and a SF $= 128$ for Bronze users.

The allocation strategy does not allow lower SFs to be given to NRT traffic since the resulting very substantial delay gain would be obtained at the risk of code starvation.

Table 6.1 summarises the SF allocation map which has been implemented in the testbed's RNC as a base for the RRM and CAC algorithms.

Table 6.1: SF allocation map.

	Platinum	**Gold**	**Silver**	**Bronze**
Conversational	$[64]$	$[64]$	$[64]$	$[64]$
Interactive	$[16 - 64]$	$[16 - 64]$	$[32 - 64]$	$[64]$
Streaming (H)	$[4]$	*n.a.*	*n.a.*	*n.a.*
Streaming (L)	$[8]$	$[8]$	$[8]$	$[8]$
Background	$[16 - 128]$	$[32 - 128]$	$[64 - 128]$	$[128]$

6.3 Operators' Perspective

In the previous section, the SF allocation scheme has not taken into account the operator's perspective, i.e. the blocking rate and the cell throughput. The SF allocation map from Table 6.1 has been injected into new emulations in order to evaluate its impact in three different scenarios:

- Operator's Worst Case (OWC): All the UEs get the lowest possible SF enabling them to access the maximal bandwidth.

- User's Worst Case (UWC): All the UEs receive their highest SF enabling the operator to load a sector with the maximal number of sessions since the OVSF tree is filling up slowly.

- OWC + degradation (OWC+): All the UEs get the lowest possible SF where downgrading is enabled in order to avoid blocking as much as possible. This is a tradeoff scenario between the OWC and the UWC.

6.3.1   Emulation Assumptions

The mobile UEs are evolving in an urban area. UE speed is fixed at the beginning of the emulation and it remains constant until the end of the emulation. In the urban scenario specified by the 3GPP in [4], the UE speed distribution is the following: 30% at $3km/h$, 40% at $30km/h$, 20% at $70km/h$ and 10% at $120km/h$.

At the beginning of the emulation, each sector is loaded with 8 UEs (total of 96 UEs in the emulated $4 - NodeB$ world) generating the different kinds of traffic which will definitely lead to blocked calls. The emulation length is still fixed at $5,600s$.

When a UE receives a SF at the beginning of a session, be it lowest, highest or degraded, it keeps it until the end of its current session. In other words, code's length renegotiation during a session is not allowed in the testbed. To optimise the code placement and replacement of successive SF requests and in order to avoid the fragmentation of the OVSF tree, the crowded-first algorithm introduced in [5] and explained in the following section was implemented in the testbed.

6.3.2   Code Placement and Replacement Algorithm

When a call arrives requesting a spreading code of a given length, the RNC needs to find a spot in the OVSF tree to accommodate this call. The crowded-first code placement algorithm is simple: Assuming that no code reassignment has to be performed (this will be discussed later in this section) and assuming there is one or more than one free spreading codes in the code tree able to serve the code request, the algorithm picks the code whose ancestor has the least free capacity.

More specifically, assume codes x and x' are both spreading codes of the same length able to serve a call. The algorithm will compare free capacities in their ancestors, e.g. y and y' respectively. The one with less free capacity (i.e., the more crowded) will be picked to accommodate the new call. When there are ties, the algorithm will go one level further up by comparing y and y''s ancestors. This is repeated until the sub-tree with the least free capacity is found. A special case is that y may represent the same sub-tree as y'. If so, the algorithm will pick the code on the left-hand side.
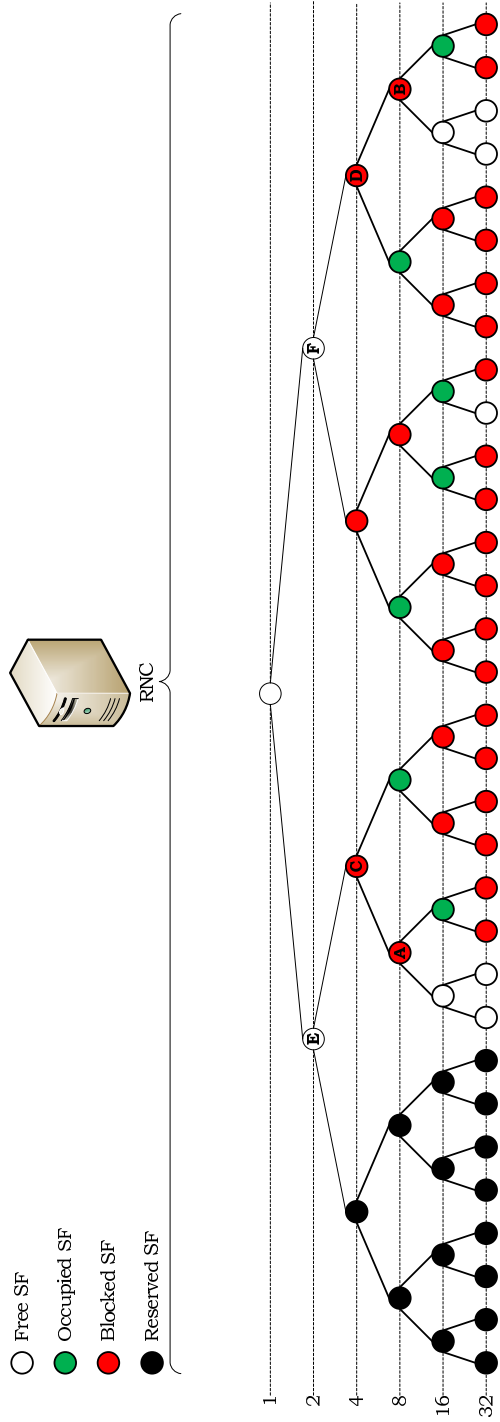
Figure 6.3: Spreading code placement.

For instance, consider the code tree shown in Fig. 6.3. Suppose a new call arrives requesting a SF $= 16$. The crowded-first algorithm will compare all the free spreading codes of the same length's ancestors. Among these ancestors, two of them, **A** and **B**, will be fully occupied if their left free descendant is chosen. The algorithm thus further compares the ancestors of **A** and **B**, respectively **C** and **D**, this is however a tie. After going one level further up (namely **E** and **F**), the algorithm will find that **B** is more crowded than **A**, so the free descendant of **B** will be selected to serve the new call.

When the code tree is used for a long time, it is sometimes inevitable that the tree becomes fragmented. In this case, a new call requesting a spreading code may be rejected even if the total amount of free capacity in the code tree is larger than the requested code's length. This may result in low utilisation of the code tree. To resolve this problem, code reassignment can be conducted to create sufficient capacity for the new call. In case an arriving call requests a spreading code, but there is no free code of that length, the following steps are taken:

- If the total amount of free capacity in the code tree is lower to the requested code's length, the call is simply rejected. Otherwise,

- For each busy code, the algorithm relocates them one-by-one starting with the higher rates busy codes. For each relocated busy code, the crowded-first algorithm is applied. It is possible that there will be no free code to accommodate the relocated call. If so, this will trigger another round of relocation. This is reiterated recursively until all busy codes are relocated. Such a process is guaranteed to succeed as there is sufficient free capacity in the code tree. Despite the low computation power of the computer emulating the RNC, this reallocation algorithm is simple enough to complete within a TTI ($10ms$). Since the control plane is beyond the scope of this work, the signalling overhead (in terms of time and/or throughput) when RNC notifies UEs of their new code(s) is neither computed nor taken into account.

(a) OVSF tree state before a SF = 4 request.

(b) OVSF tree state after a SF = 4 request.
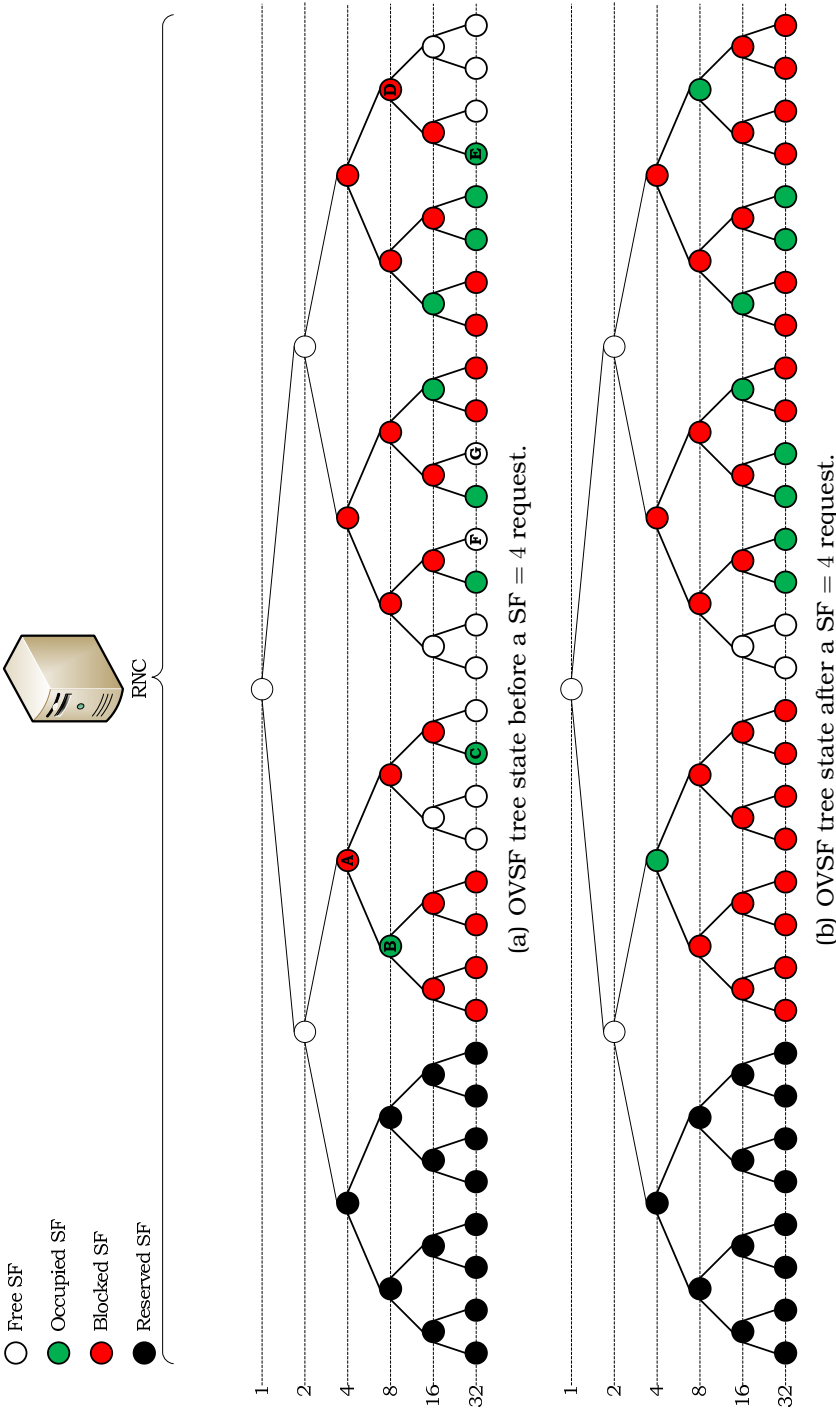
Figure 6.4: Spreading code replacement.

For example, in the code tree presented in Fig. 6.4(a), a new call is arriving requesting a SF $= 4$. The presented OVSF tree can support an additional SF $= 4$ query but it has to reorganise first. Code **A** contains only two busy codes to be reallocated, namely **B** and **C**, whereas the two other codes on **A**'s level count three and four busy codes. Code **A** is then chosen and is referred as the minimum cost branch. When relocating **B**, another reassignment will be triggered. This time, **D** is selected as the minimum cost branch because it contains only one busy code, namely **E**. Based on the crowded-first algorithm, **E** will be relocated at **F**. Then, **B** can be relocated at **D**. Consequently, **C** will be relocated to **G** still following the crowded-first algorithm. Now, **A** is available to accommodate the new SF $= 4$ call. The resulting OVSF tree is shown in Fig. 6.4(b).

6.3.3   Results

The cell throughputs obtained in the different cases are depicted in Fig. 6.5. In the UWC, even if the UEs cannot use as much as bandwidth as they would need, more UEs can transfer data, which leads to a better cell throughput. In the OWC, the UEs can use all the bandwidth they need, but due to the OVSF tree structure, they all waste some resources leading to a lower cell throughput. The OWC+ scenario yields approximately the same cell throughput as the OWC due to the small number of degraded queries (about 1%).

The blocking rates obtained by the different emulation runs are presented in Fig. 6.6. As expected, the number of accepted SF queries in the UWC is greater than in the OWC. In the UWC scenario the OVSF tree fills up more slowly, since all the UEs request their highest SF. That leads to reduced radio resource consumption and less blocked SF queries. The OWC+ yields approximately the same number of accepted queries as in the OWC. However, thanks to the downgrading, the number of blocked queries is lower.

Table 6.2 compares 95th percentile packet delay in the different Cases for the four traffic classes.
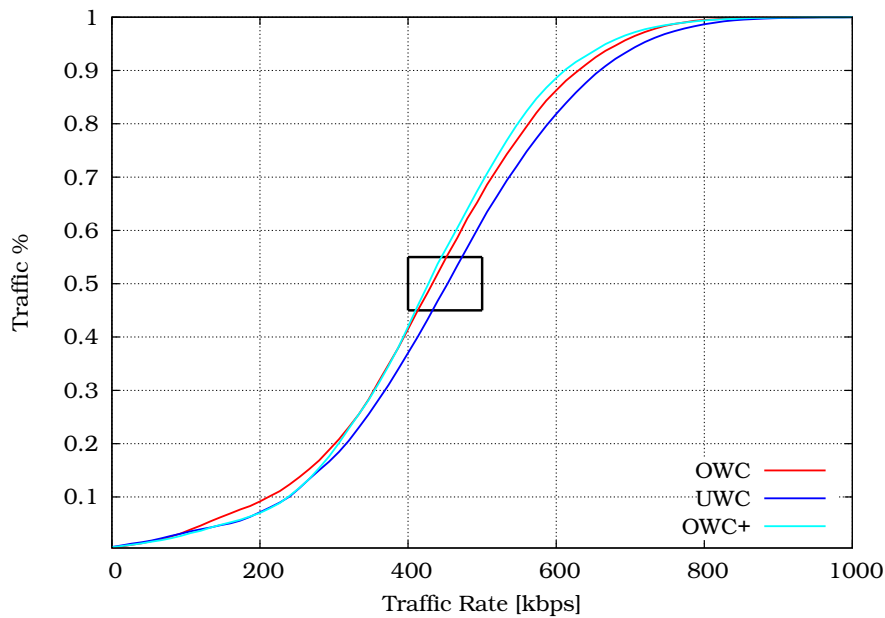
(a) Overall view.
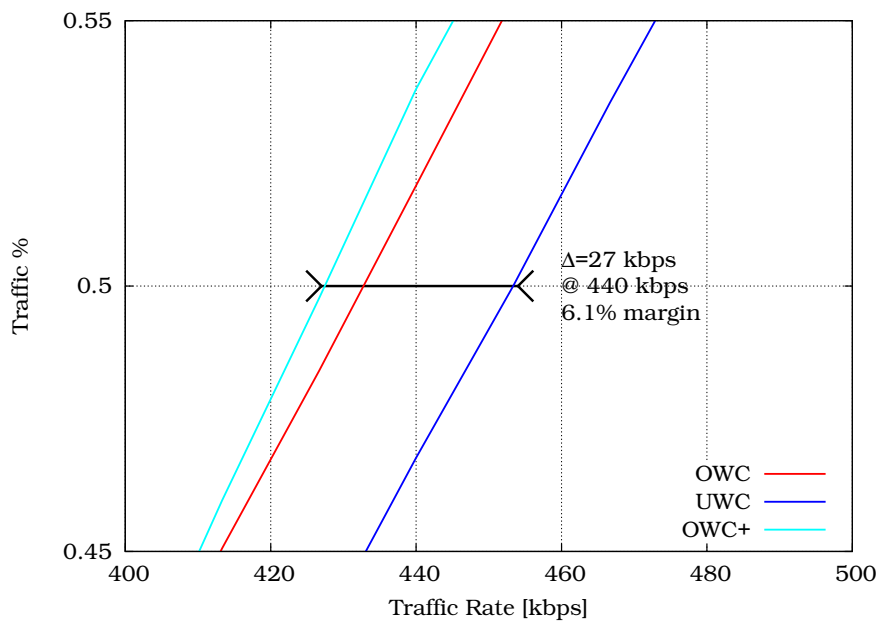
(b) Zoom.

Figure 6.5: Cdf of cell throughputs.
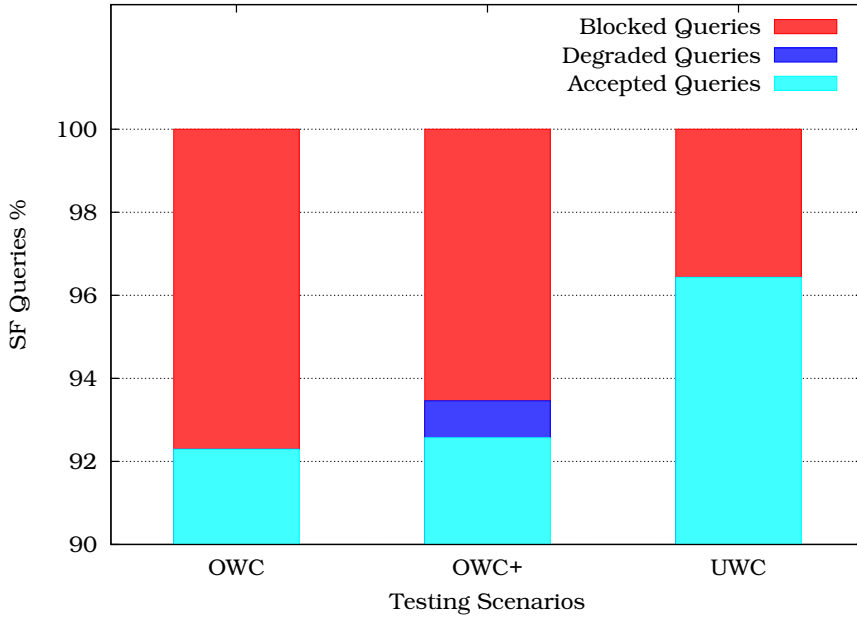
Figure 6.6: Accepted/degraded/blocked SF queries.

Table 6.2: Traffic classes packet delay (95th percentile compared to 3GPP bound).

Traffic Classes	OWC	UWC	OWC+	3GPP Bound
Conversational	$77ms$	$75ms$	$74ms$	$< 80ms$
Interactive	$5,986ms$	$7,702ms$	$7,118ms$	$< 10,000ms$
Streaming	$91ms$	$93ms$	$91ms$	$< 250ms$
Background	$56,657ms$	$72,581ms$	$67,166ms$	$< 100,000ms$

Considering the blocked SF queries percentage (Fig. 6.6), the cell throughput level (Fig. 6.5) and the QoS bounds (Table 6.2), it appears that the OWC+ strategy is the optimal solution. Indeed, it reaches a lower blocked SF queries percentage than the OWC strategy while giving approximately the same cell throughput. It also gives better QoS results than with UWC strategy. It is then chosen as the strategy to implement in the testbed since it seems to be the best tradeoff between the two other extreme strategies (OWC and UWC).

Result sensitivity

The above results appear to be optimal under the assumptions set by the testbed, which tries to be as realistic as possible for the selected scenario of Part II. A confrontation of this allocation map and the OWC+ strategy to other assumptions is needed to evaluate the sensitivity of these results. A neutral scenario has been elaborated for this purpose to check the variability of the obtained results in another traffic distribution. This new scenario can be considered as neutral since it includes a uniform traffic distribution overwriting the one presented in Table 3.6. The new traffic distribution is explained in Table 6.3.

Table 6.3: New traffic distribution.

Traffic	Realistic Scenario		Neutral Scenario		
Classes	Traffic %	ISA	Traffic %	ISA	ISA
Conversational	27%	$100s$	25%	$25s$	$350s$
Interactive	48%	$50s$	25%	$25s$	$350s$
Streaming	10%	$300s$	25%	$25s$	$350s$
Background	15%	$200s$	25%	$25s$	$350s$

In multivariate problems, changing variable influences all others and the overall results. In order to be sure to confront comparable results, a fixed point has to be chosen. Changing the traffic mix while keeping the same fixed point (i.e. the cell throughput) will allow to compare the variation of all the other variables and the output results on the same basis. As presented in Fig. 6.7, two different ISAs ($25s$ and $350s$) yield approximately the same cell throughput as in the realistic scenario. Following the fixed point theory, these three scenarios can therefore be considered as comparable.

Fig. 6.8 shows that the results are highly sensitive to traffic distribution patterns. The percentage of blocked queries computed as 22.82% and 2.17% with an ISA respectively of $25s$ and $350s$ where it has been previously evaluated to 6.54% under the OWC+ assumptions. The degraded queries follow the same trend with a computed percentage of 3.52% and 0.21% while at 0.89 with the OWC+ strategy.
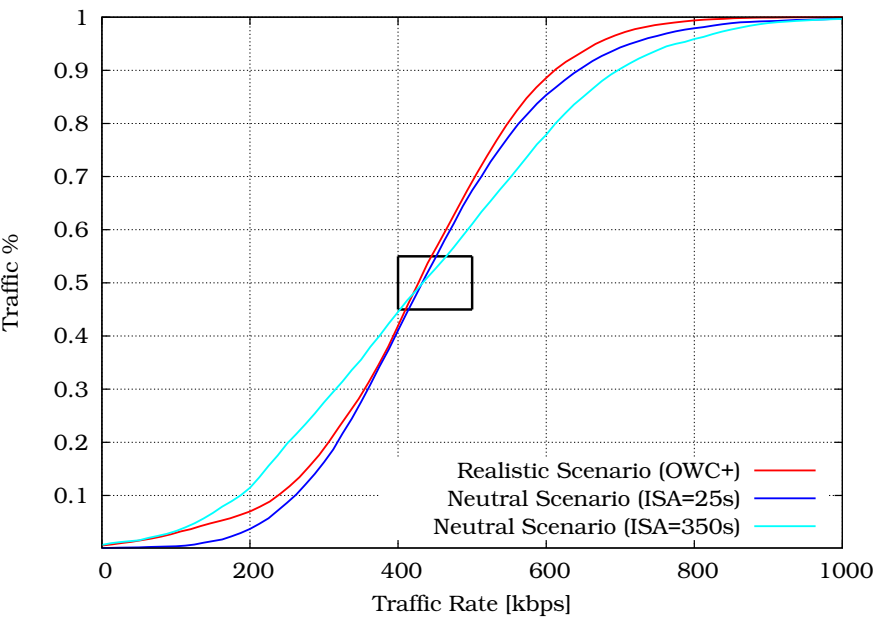
(a) Overall view.
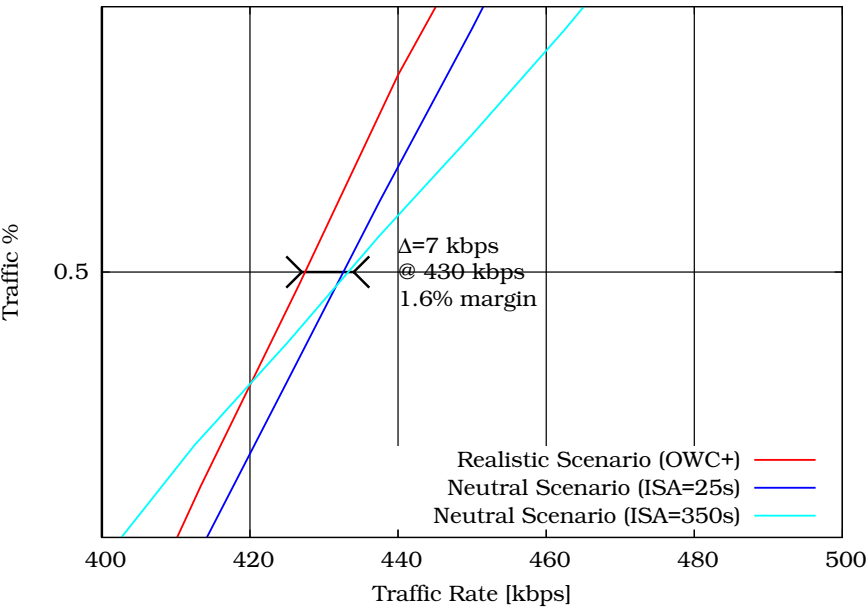
(b) Zoom.

Figure 6.7: Results sensitivity cell throughputs fixed point.
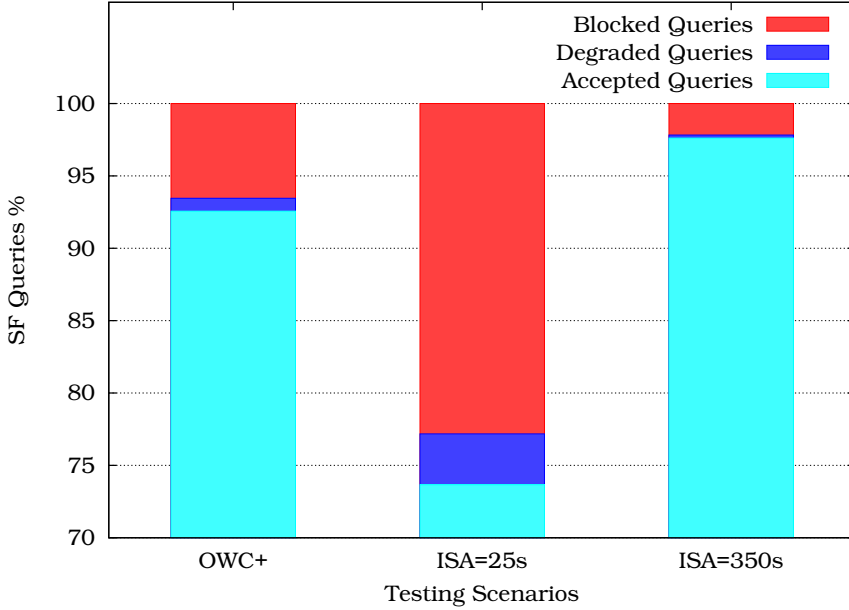
Figure 6.8: Results sensitivity on accepted/degraded/blocked SF queries.

An explanation for this behaviour is that due to their nature, RT sessions generate typically more traffic than NRT sessions. With an equivalent number of sessions, supporting more RT sessions will lead to a higher cell throughput. Considering the neutral scenario, the proportion of Conversational sessions is more or less unchanged whereas the part allocated to the Streaming sessions increases from 10% to 25%. This explains why the testbed can reach the same range of cell throughput as the realistic scenario, with other uniformly distributed traffic scenario. As example, an ISA of $25s$ leads to a very high blocked queries and degradaded sessions number whereas at the opposite side, an ISA of $350s$ gives a few blocked queries and no degradation at all.

6.4 Summary

This chapter has presented the results obtained in terms of QoS management using the computer testbed. First, it has es-

tablished a SF allocation map giving the best possible treatment to a data flow depending on its traffic class and the user profile of the UE receiving/emitting the flow. This allocation map has been established only considering the packet transfer delays, i.e. the users' perspective. The chapter has subsequently evaluated this allocation scheme with respect to the operators' requirements (cell throughput and blocked rate) in different scenarios. Considering both users and operators requirements, it has been possible to find an optimal SF allocation strategy to implement into the testbed. The sensitivity of such a strategy has been evaluated.

To the best knowledge of the author, no similar results can been found in open literature. Hence, this experiment does not only claim to be a case study for the testbed validation, but it is also a completely original result which could be helpful for telecommunication operators.

Bibliography

[1] Hugues Van Peteghem and Laurent Schumacher. Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS. In *Proceedings of the 64th Vehicular Technology Conference (VTC-fall)*, Montreal (Canada), September 2006.

[2] Technical Specification Group Radio Access Network. Quality of Service (QoS) concept and architecture. Technical Report TS 23.107 V7.1.0, 3rd Generation Partnership Project (3GPP), September 2007.

[3] Paul McKenney, Dan Lee, and Barbara Denny. Traffic Generator Software Release Notes. Technical report, SRI International and USC/ISI Postel Center for Experimental Networking, January 2002. http://www.postel.org/tg/tg.htm, last visited: March 13, 2008.

[4] Technical Specification Group Radio Access Network. Multiple Input Multiple Output in UTRA. Technical Report TR 25.876 V7.0.0, 3rd Generation Partnership Project (3GPP), March 2007.

[5] Yu-Chee Tseng, Chih-Min Chao, and Shih-Lin Wu. Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree. *Management IEEE Transactions on Mobile Computing*, December 2002.

Chapter 7

Downswitch Timer

7.1 Introduction

This chapter discusses a scheme to improve the Rel'99 UMTS RRM. It is focused on the two different types of transport channel presented in Section 5.2.2: shared and dedicated. It evaluates the advantages of a scheme based on an optimal NRT session switching between these two kinds of channels. The switching decision may be influenced by several parameters like the downswitch timer's duration, the non-/preemptive split of downswitch timer and the selection of the initial transport channel. Performances of the analysed scheme are evaluated taking into account different metrics such as blocking rate, packet transfer delay and cell throughput. The chapter finally presents results showing that an optimal parameter tuning may significantly decrease the number of blocked calls while keeping the other metrics at a satisfactory level as presented in [1].

7.2 RRC States Simplification

Considering the specifications made by the 3GPP in [2], the two basic operational modes of a UE are the `Idle Mode` and the `Connec-ted Mode`. The `Connected Mode` can be further divided into different service states known as Cell_DCH, Cell_FACH, Cell_PCH, and URA_PCH. The focus of the present dissertation is the UE data flows, it only deals with the user plane of the

UTRAN. The control plane (signalling) is beyond the scope of this work. Therefore, the UE `Connected Mode` RRC states switching can be simplified as shown in Fig. 7.1:

(a) Original version.   (b) Simplified version.
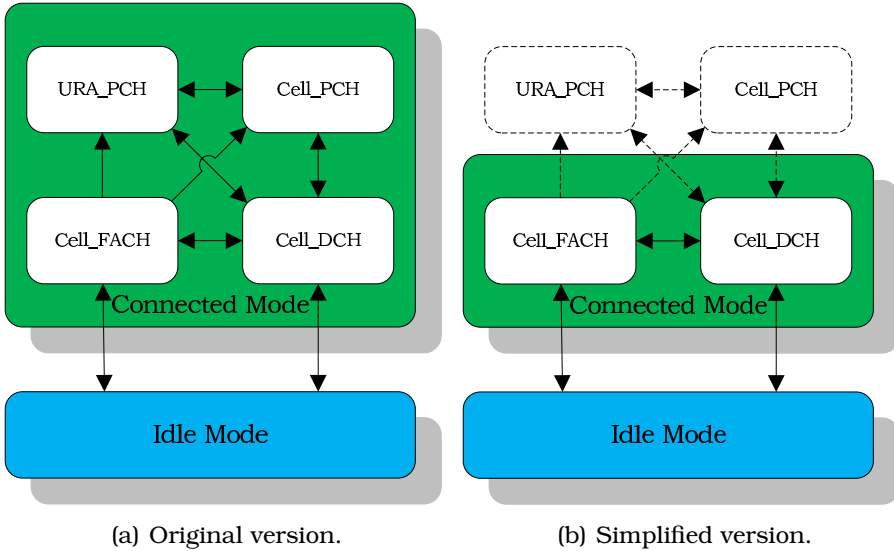
Figure 7.1: UE RRC states switching.

In the Cell_FACH state, no DCH is allocated to the UE but a shared channel. In this case, the FACH will be used. This channel carries small amounts of user data using time multiplexing. Its bandwidth is rather small since it is not expected to support large data transfers.

In the Cell_DCH state, at least one DCH is allocated to the UE. This channel transports a single user data flow with a fixed bit rate depending on the SF allocated by the RNC. A shared channel can also be used in this state to carry different user data flows. Since the testbed is focused on the Rel'99 UMTS, this shared channel is the DSCH. Its bandwidth is variable since its SF occupies the free room left in the OVSF tree after the DCHs SF allocation as shown in Fig. 5.11.

One can now add a line to Table 5.2 to mention in which RRC state the UE has the ability to access a DCH, the FACH and the DSCH (Table 7.1):

Table 7.1: Enhanced comparison of different transport channels.

Channel	DCH	DSCH	FACH
SF	Fixed [512 - 4]	Variable [256 - 4]	Fixed [256 - 4]
Multiusers	Dedicated	Shared	Shared
Multicodes	No	Yes	No
RRC state	Cell_DCH	Cell_DCH	Cell_FACH

A UE in the `Idle Mode` is considered as not active, so the testbed assimilates a UE in this state simply as not connected.

7.3 Channels Switching Policy

As already explained in Section 5.2.3, the RT sessions are directly allocated to a DCH whereas NRT sessions tolerate to be directed to either dedicated or shared channels. As RT sessions use DCHs only, channel switching will influence the NRT sessions only.

7.3.1 NRT Traffic

According to the UMTS standard, the first channel allocated to a NRT session is not the only one For this reason, channel switching is permitted and even encouraged since it can optimise the radio resources use.

If a NRT session uses a shared channel (DSCH or FACH), it will continue using this channel until the RNC receives an indication that the current burst might be long (the packet queue length exceeds a fixed threshold). Then the RNC tries to allocate a DCH to that session. If the OVSF tree has enough room to accept this new DCH, the NRT session moves from the shared channel to its new DCH.

On the other hand, when no packets are sent anymore over the DCH, a timer is initialised and the session remains on the DCH during this period. If there are no new arrivals within this timeout period, the session is switched back to a shared channel.

These two parameters presented in [3] are respectively the upswitch threshold (from shared to DCH, expressed in bytes) and the downswitch timer (from dedicated to shared channel, expressed in seconds). Fig. 7.2 illustrates these mechanisms with an upswitch threshold fixed to $500 Bytes$ and a downswitch timer of $8s$.
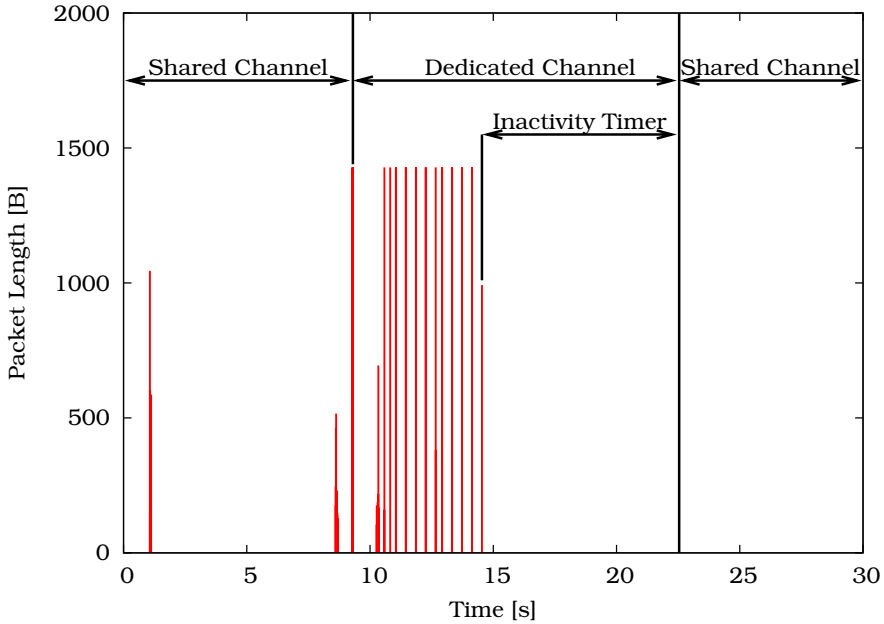
Figure 7.2: Channel switching.

7.3.2   Switching from Cell_FACH to Cell_DCH

Being in Cell_FACH means that the UE is active and does not support any RT session. It can only support at least one low bandwidth demanding NRT session which is satisfied with FACH bandwidth.

Switching from Cell_FACH to Cell_DCH happens when a DCH has been requested. This may be caused by the NRT session exceeding the upswitch threshold or by a beginning RT session. In the first case, the NRT session will leave the FACH and continue into its new DCH. In the second case, the RT session will start

in its own DCH, and the NRT session not transferring enough data to claim a DCH, will not stay in the FACH (since the UE is no longer in Cell_FACH) but will proceed towards the shared channel of the Cell_DCH state: the DSCH.

7.3.3 Switching from Cell_DCH to Cell_FACH

When DCHs are no longer needed by a UE (be it due to a RT session end or/and a NRT session downswitch) and at least one NRT session is still active in the DSCH, the UE falls back to the Cell_FACH state. The NRT session assigned to the DSCH must leave it and proceed in the FACH since the DSCH is not accessible for a UE in the Cell_FACH state.

7.3.4 Handover Impact

When a UE has to undergo a handover, its resources consumption in terms of SF are moving from an OVSF tree to another. There is no SF reservation mechanism for handoff queries implemented in the testbed. This means that if the destination OVSF tree is too crowded to accept the new handoff query, the session requesting the spreading code will be either stopped (RT sessions) or redirected to a shared transport channel (NRT sessions).

Handovers have an impact on the nature of the transport channel a session can use. They also influence the RRC state of a UE. Suppose a UE being in Cell_DCH state and enjoying the use of a single DCH for its Interactive session. When the RNC decides to handoff that UE, suppose the OVSF tree of the destination sector cannot accept the spreading code query: The Interactive session can only continue in a shared channel. The UE will fall back to the Cell_FACH state and its session will continue on into the FACH.

7.4 Optimal Parameter Value Set

The UMTS standard does not give any guidance for setting the value of the upswitch threshold or the downswitch timer. The purpose of the present experiment is to find and evaluate

the best possible values of these parameters in order to achieve optimal resource use under the testbed assumptions.

Focusing on the upswitch threshold, its optimal value has already been identified through extensive simulations in [4]. As a rule of thumb, it has to be some what smaller than the average packet size. As the packet waiting buffer is filled with only one packet (the NodeB can not emit packets as quickly as they arrive), the underlying session responsible of this excess is directly upswitched. This strategy leads to open a DCH for large connections as quickly as possible while preventing small connections from opening one, keeping in mind that reserving a SF and opening a DCH is time costly, i.e. $900ms$ as mentioned in Table 7.2.

Since the optimal upswitch threshold has already been identified, three other parameters still have to be accurately tuned: First, the NRT best suitable value for the downswitch timer has to be identified. If it is high, a NRT session will remain longer on a DCH and, if it has to send more packets after an inactivity period (i.e. a web page reading time), it has to not upswitch again since it has kept its DCH. The connections undergo less upswitches and the DCH allocation time affects the packet delay in a less significant way. The downside in this situation is that other connections may have to stay in the shared channel due to extensive OVSF tree occupation. On the other hand, using a small downswitch threshold will lead to rapidly free the OVSF tree by switching back to the shared channel after using a DCH. In this case, sessions will suffer from multiple upswitches and DCH allocation time will significantly increase the packet transfer delay. A tradeoff between these two options is therefore needed.

The second parameter to accurately tune is based on a strategy initially proposed in [3]. This strategy splits the downswitch timer in two distinct periods. The timer begins in the Non-Preemptive (NP) mode. In this mode, the session admission mechanism is exactly the same as for a normal downswitch timer. A NRT session keeps its SF disregarding all other new sessions requests. After a period of inactivity in NP mode, the downswitch timer moves into the preemptive mode. In this case,

a NRT session keeps its SF until the end of its downswitch timer, unless another session needs this SF. Behind identifying the best possible downswitch timer, is to evaluate the benefit from splitting the downswitch timer in two periods. Fig. 7.3 illustrates this new strategy with an upswitch threshold still fixed to $500Bytes$ but a downswitch timer split with a NP period of $4s$ and a preemptive period of $4s$.
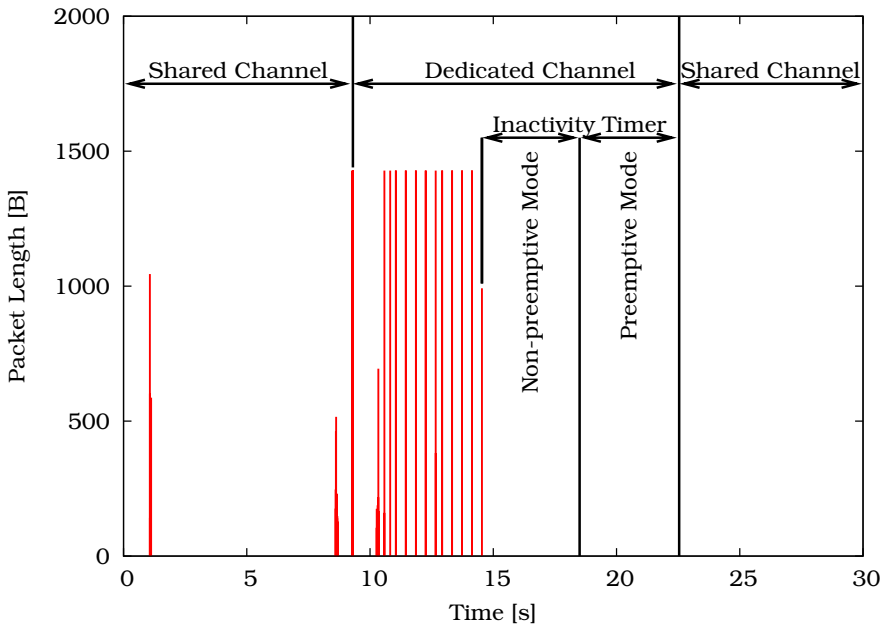
Figure 7.3: Split downswitch timer strategy.

In a scenario with only one remaining DCH for new calls, Fig. 7.4(a) illustrates a classical downswitch strategy:

t_0: User A requests the DCH.

t_1: User A gets the DCH and uses it.

t_2: When transmission from user A stops, its downswitch timer starts.

t_3: User B requests the DCH.

t_4: The user B's request is rejected since the DCH is still allocated to user A even if it does not use it.
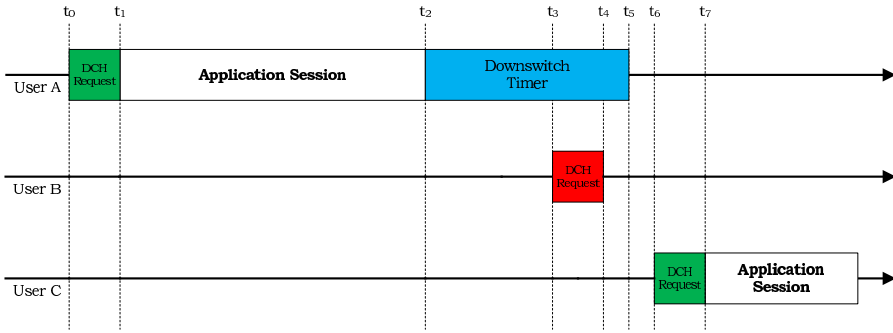
(a) Normal downswitch timer.
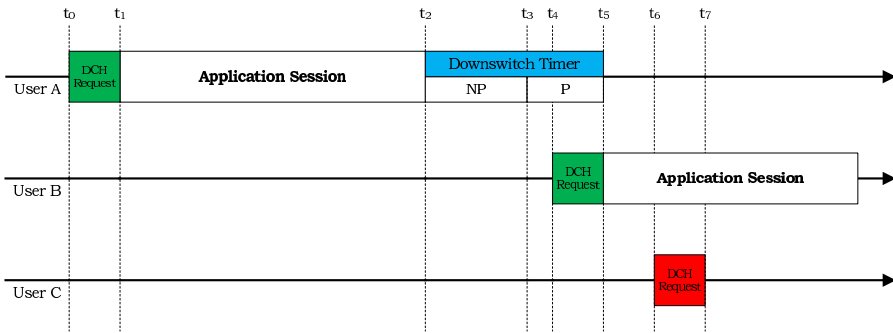
(b) Split downswitch timer.

Figure 7.4: Comparison of downswitch strategies.

t_5: The downswitch timer of user A ends up and user A leaves the DCH.

t_6: User C requests the DCH.

t_7: User C gets the DCH and uses it since user A has released the resource.

whereas Fig. 7.4(b) shows the split downswitch timer strategy:

t_0: User A requests the DCH.

t_1: User A gets the DCH and uses it.

t_2: When transmission from user A stops, its downswitch timer starts in NP mode.

t_3: The downswitch timer of user A passes in preemptive mode.

t_4: User B requests the DCH.

t_5: User B gets the DCH and uses it since the downswitch timer of user A is in preemptive mode; the user A has to leave the DCH.

t_6: User C requests the DCH.

t_7: The user C's request is rejected since the DCH is already allocated to user B.

A new transmission is then able to use the only one remaining DCH earlier with the split downswitch timer strategy (Fig. 7.4(b)) than with the classical downswitch strategy (Fig. 7.4(a)).

Besides the downswitch timer dimensioning, the third option is to determine whether it is beneficial or not to directly open a DCH at the beginning of a NRT session or to allocate it to a shared channel (DSCH or FACH) first. The UMTS standard gives no particular guidance therefore.

The potential merits of the proposed scheme/values is measured by the previously described metrics: the blocking rate, the packet transfer delay and the cell throughput to describe both, the operators and the users perspective.

7.5 Results

7.5.1 Testing Environment

This experiment takes place in the testbed environment implementing the OWC+ scenario as previously presented. All other important emulation parameters are summarised in Table 7.2.

An important characteristic of the emulation runs is the downlink load factor (η_{DL}). It represents the downlink load placed on a sector by a set of channels. η_{DL} approaches 1 when the channels' total allocated bandwidth is being used. In the

Table 7.2: Emulation parameters.

Number of UEs per sector	8
Emulation length [s]	$5,600$
Downswitch threshold [s]	$8; 15; 30; 45; 60$
DCH allocation time [ms]	900
FACH SF	64

testbed, η_{DL} is focusing on the three types of transport channels available (DCHs, FACH and DSCH) and is computed as follows:

$$0 \leq \eta_{DL} = \frac{\sum_{channels} consumed\ bandwidth}{\sum_{channels} allocated\ bandwidth} \leq 1 \qquad (7.1)$$

Under these emulation assumptions, the mean load factor computed during the emulation runs is evolving between 0.35 and 0.4, where it can be as high as 0.75 over real UMTS network.

7.5.2 Observations

The present experiment consists in a single emulation for every downswitch timer value (split or not) allocating either a dedicated or a shared channel first. Furthermore, a set of values, leading to an optimal use of the UMTS wireless interface in terms of blocking rate as well as cell throughput while keeping acceptable packet transfer delays, have been identified. To illustrate these results, Fig. 7.5 shows the blocking rate and Fig. 7.6 depicts the packet transfer delay experienced by the four traffic classes during the emulation runs. These metrics are plotted as a function of the downswitch timer, either NP (lower time scale) or split in NP and Preemptive periods (NP-P, upper time scale). Finally, Figs. 7.7, 7.8, 7.9 and 7.10 present the cell throughput cdfs of all the emulations.

From the downswitch timer perspective, it appears that dividing the downswitch timer in two equally long periods (a preemptive one and a NP one) is beneficial. As shown in Fig. 7.5, the blocking call rate is significantly reduced (almost 2% reduction in the best case), whereas packet transfer delay (Figs. 7.6)

is approximately unchanged and cell throughput (Figs. 7.7-7.10) is lightly increased.

Following the results depicted in the Fig. 7.6-7.10, opening a DCH at the beginning of a new NRT session while ignoring whether it will be a large connection or not, is the better choice. As the DCH allocation time is included into the initial connection establishment time, the average Interactive and Background delays are subsequently reduced.

Focusing on the packet transfer delay (Fig. 7.6(a)) and the cell throughput (Figs. 7.7 and 7.9) when opening a DCH at the beginning of a new session, it appears that a short downswitch timer (between 8 and 15 seconds) yields the best results.

According to these observations, the results of the testbed emulations are inline with those from analytical computations [5] and other simulations [6]. The testbed settings refines their strategy. For the NRT sessions, the optimal strategy appears to be:

- to start the session directly in a DCH (if possible depending on the OVSF tree occupancy),

- to assign a short downswitch timer (as proposed in [6]), and

- to split the downswitch timer in two equivalent periods, the first one NP and the other preemptive.

As mentioned in Section 7.5.1, the load factor of this chapter's experiments is moderate (0.4 compared to a maximum of 0.75). This raises the issue of the sensitivity of the results with respect to the load factor. Depending on the cause of a change of the load factor, results will be impacted or not.

- If the number of sessions, and therefore of transport channels, remains constant, an increase of the load factor due to additional traffic of those ongoing sessions will not translate into higher contention on the resources. Consequently, one can expect that the results, which are focused on the channel switching policy, will not be impacted.
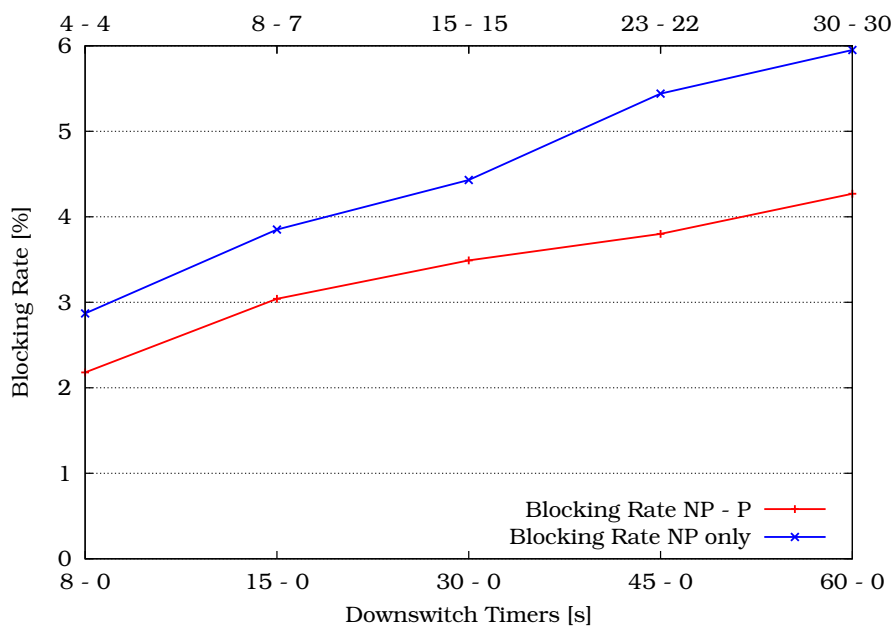
(a) DCH first.
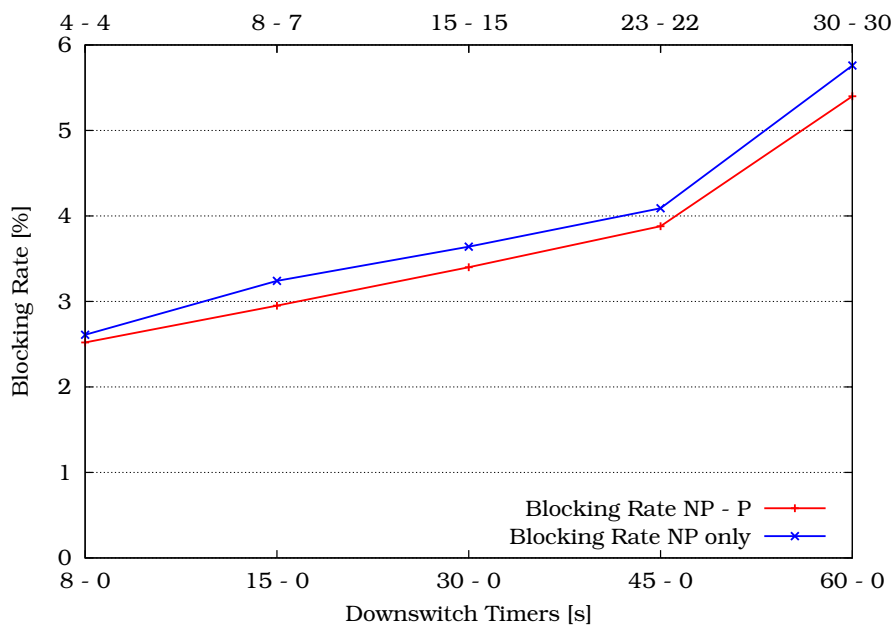
(b) Shared channel first.

Figure 7.5: Blocking rate vs. downswitch timers.
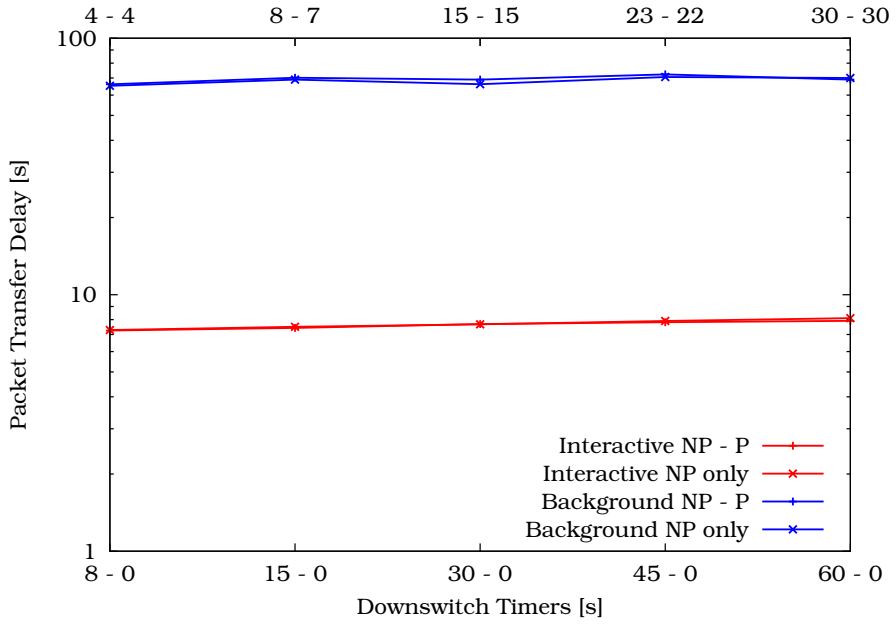
(a) DCH first.
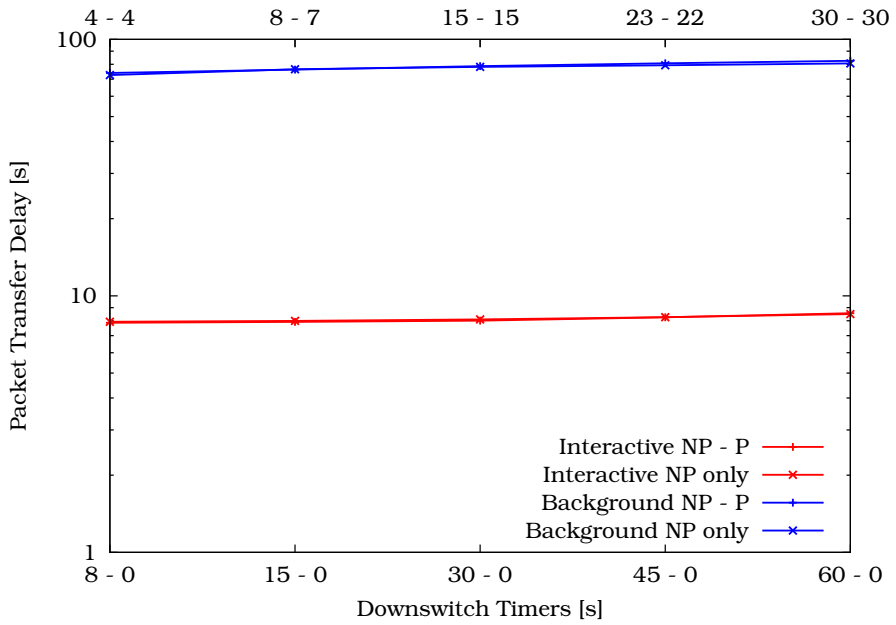
(b) Shared channel first.

Figure 7.6: Packet transfer delay vs. downswitch timers.
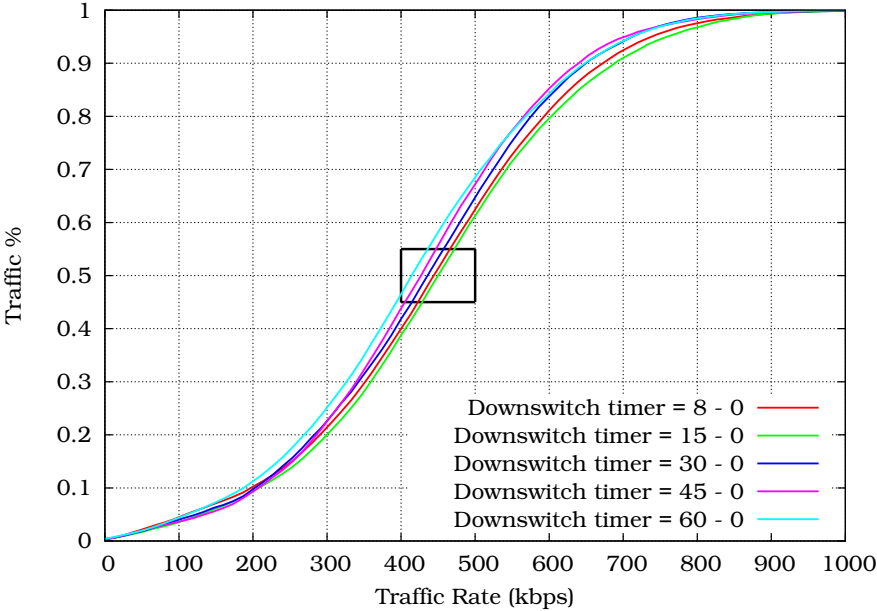
(a) Overall view.
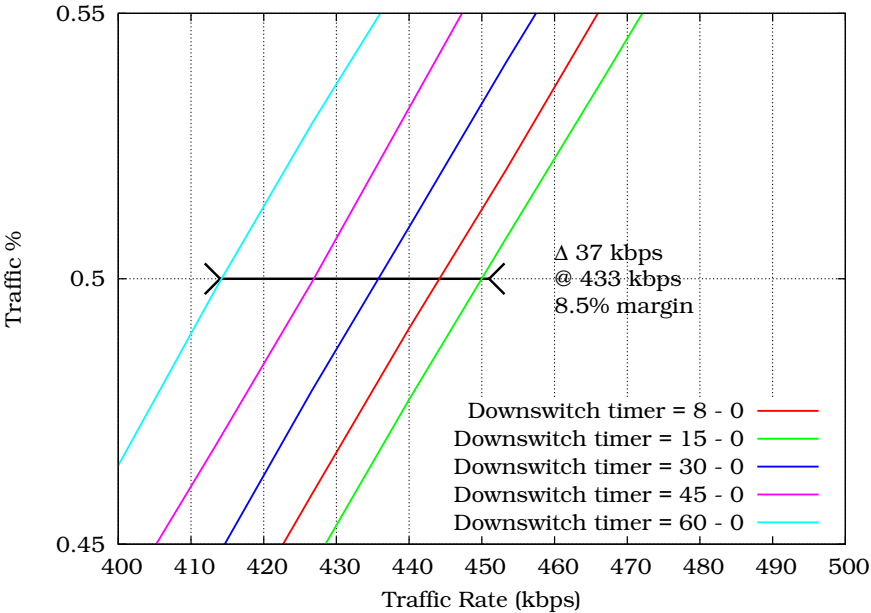
(b) Zoom.

Figure 7.7: Cell throughput vs. downswitch timers (NP only – DCH first).
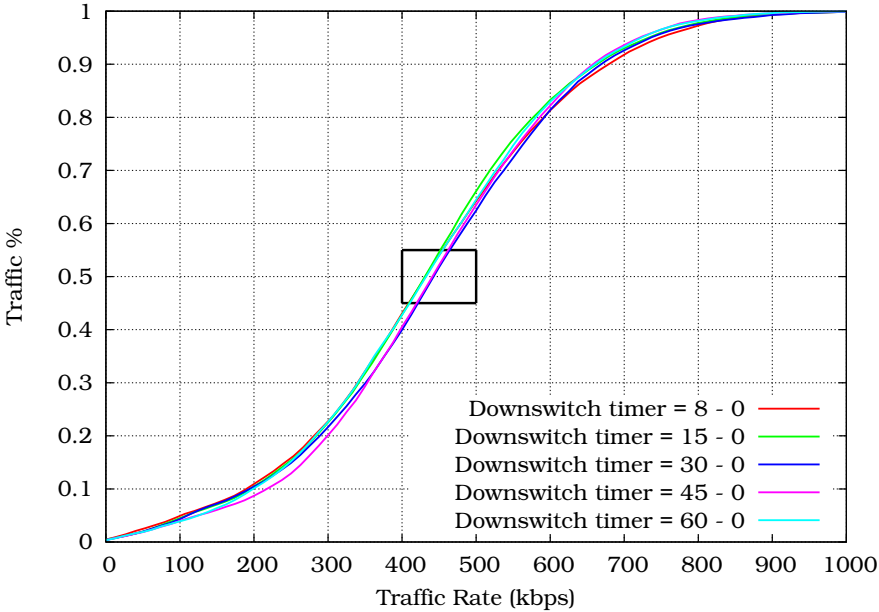
(a) Overall view.
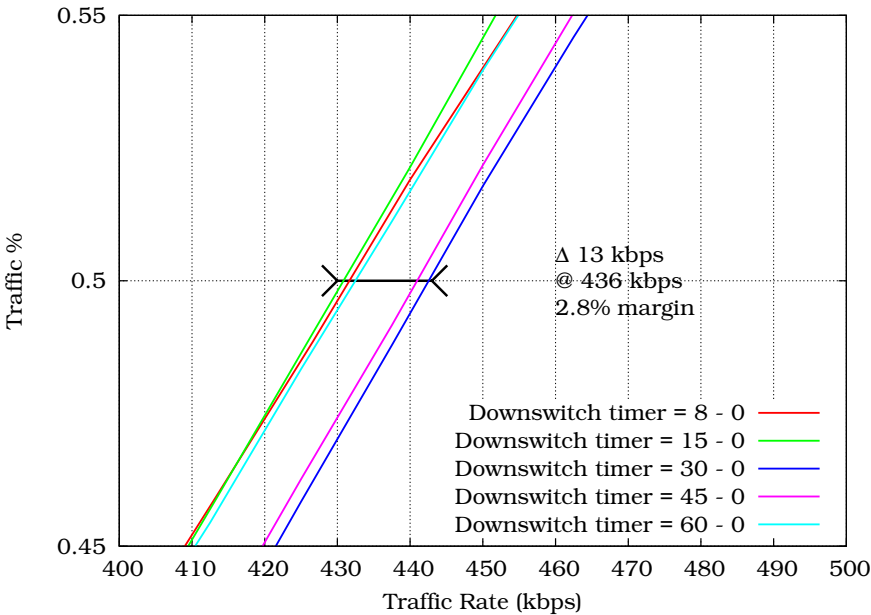
(b) Zoom.

Figure 7.8: Cell throughput vs. downswitch timers (NP only – shared channel first).
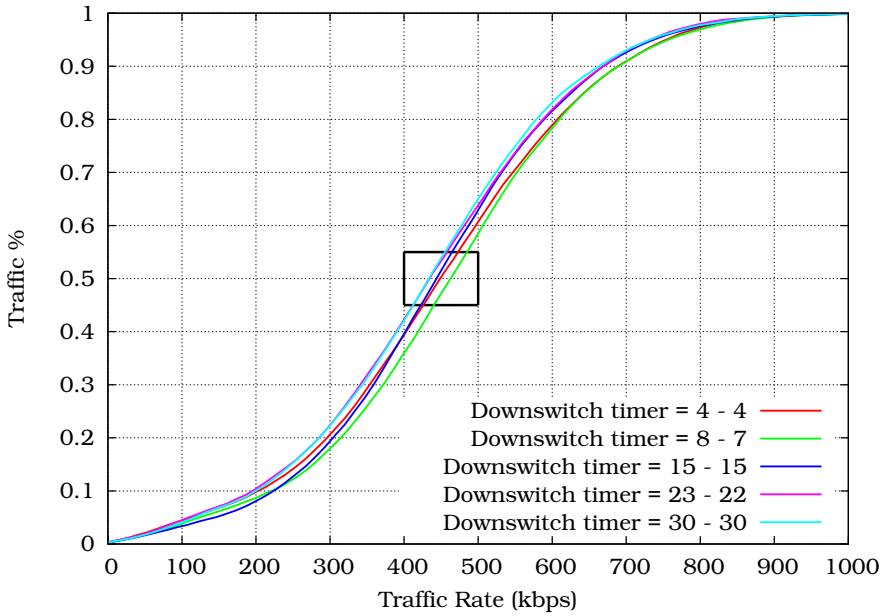
(a) Overall view.
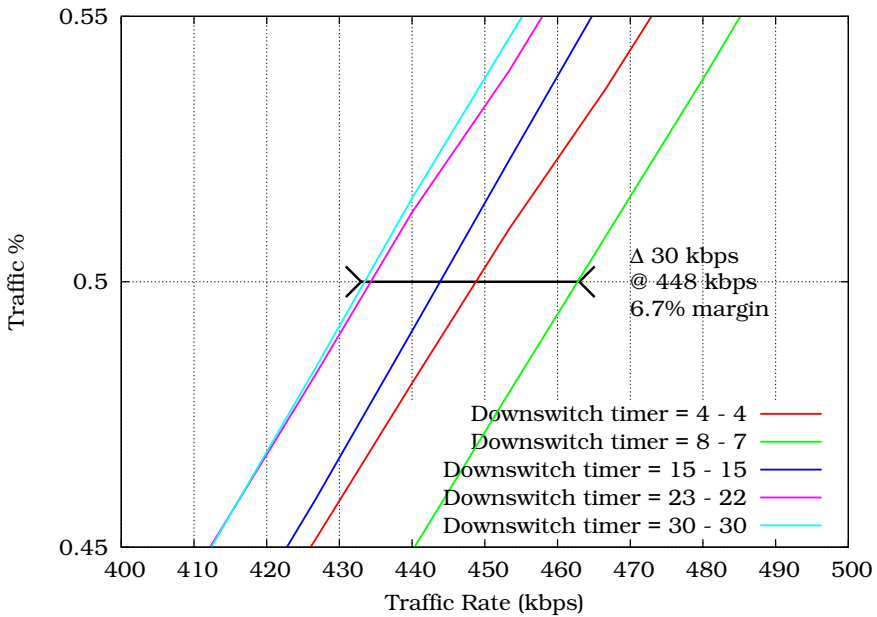
(b) Zoom.

Figure 7.9: Cell throughput vs. downswitch timers (NP-P – DCH first).
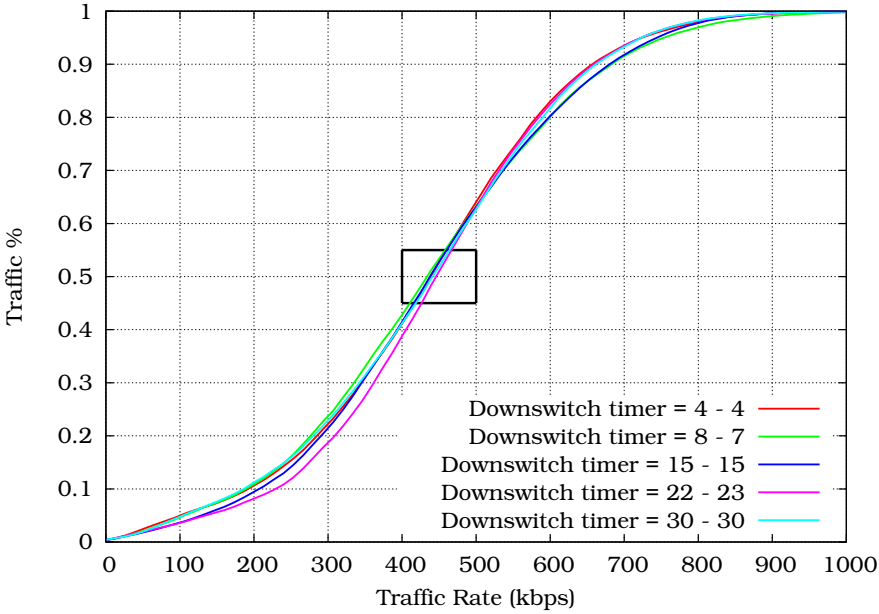
(a) Overall view.
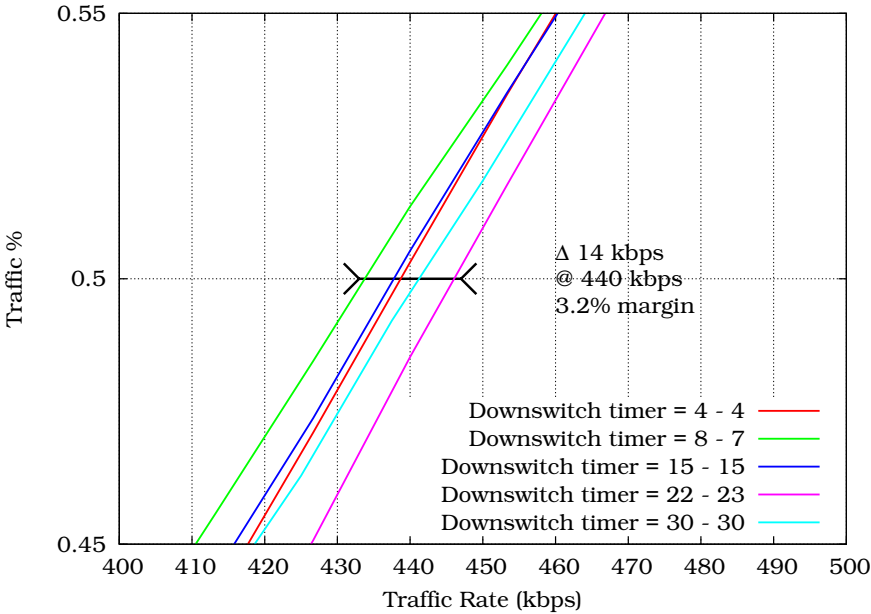
(b) Zoom.

Figure 7.10: Cell throughput vs. downswitch timers (NP-P – shared channel first).

- If the increase of the load factor is due to the opening of new sessions, the contention on the resources increases. In such a scenario, the NP-P downswitch timer split strategy should deliver even better results.

7.6   Summary

This chapter has presented the testbed's results in terms of RRM. This experiment has analysed an improved NRT sessions channel switching strategy based on a division of the downswitch timer. This strategy appears to optimally meet the UMTS network requirements, such as a low blocking rate, a high cell throughput, and a low packet transfer delay. A comparison with simulations and analytical computations results taken from the open literature yields converging sets of results.

Bibliography

[1] Hugues Van Peteghem and Laurent Schumacher. UMTS Non Real-Time Sessions Channel Switching Emulation. In *Proceedings of the 65th Vehicular Technology Conference (VTC-spring)*, Dublin (Ireland), April 2007.

[2] Technical Specification Group Radio Access Network. Radio Resource Control (RRC) protocol specification. Technical Report TS 25.331 V8.0.0, 3rd Generation Partnership Project (3GPP), October 2007.

[3] Balakrishna J. Prabhu, Eitan Altman, Konstantin Avratchenkov, and J. Abadia Dominguez. A simulation study of TCP performance over UMTS Downlink. In *Proceedings of the 58th Vehicular Technology Conference (VTC-fall)*, Orlando (FL - USA), October 2003.

[4] Xavier Dubois. Performance of different TCP versions over UMTS common/dedicated channels. Master's thesis, FUNDP - University of Namur (Belgium), June 2005.

[5] Luk de Bruynseels. Tuning Inactivity Timer Settings in UMTS. Technical report, Commsquare Technical Whitepaper, January 2005.

[6] Mooi-Choo Chuah, Luo Wei, and Xiaowei Zhang. Impacts of Inactivity Timer Values on UMTS System Capacity. In *Proceedings*

of the 3rd *Wireless Communications and Networking Conference (WCNC)*, Orlando (FL - USA), March 2002.

Chapter 8

Real-Time Flows Parametrisation

8.1 Introduction

This chapter describes the results obtained on the testbed based on one of its particularities: the possibility to evaluate the QoE of a given flow a user supports. The goal of this experiment is to measure the impact of design choices at transport, data link and physical layers on the user experience when receiving RT flows. This is illustrated for both synthetic and real life Conversational and Streaming traffics.

This chapter focuses on the influencing parameters which the testbed is able to emulate and modify (from transport down to data link layers). There could be other influencing factors on QoE beyond UTRAN. For instance, E2E application layer mechanisms such as the automatic switch to a lower content resolution could also have a significant impact on the QoE. Therefore, the comparison of the real life traffics is based only on identical application settings (codecs, degradation setup, buffer size, etc.).

8.2 Emulation Assumptions

This experiment focuses on the QoE of a user receiving RT flows. Since the RT flows are using DCHs only as explained in

Section 5.2.3 and since a DCH is fast power controlled by the RNC as exposed in Section 5.2.2, the received power of the user tends to be constant whatever the number of interfering users present in the same sector. For this reason, the experiment does not require more than one active user.

This single user is also set as static. So, the impact of a fluctuating received power or handovers on the user experience is beyond the scope of this experiment. The only objective being the identification of the optimal network parametrisation set for RT flows delivered on a DCH based on the QoE evaluation.

8.3   User Experience

The QoE of the Streaming and Conversational classes is evaluated here as in [1]. In both cases, emulations were first run with synthetic traffic (described in Chapter 3) to measure the impact of the testbed settings. Performance has been evaluated in terms of the classical QoS metrics (packet transfer delay and packet loss rate) computed at the application layer as explained in Section 6.2.2.

Real life experiments were set up with actual content being delivered through the testbed in order to assess the QoE at the end-user when s/he connects through a UTRAN. Conversational and Streaming flows were generated adopting parameter settings based on the options available in different layers:

- Transport layer: TCP vs. UDP.

- Data link layer (RLC): error-free Ethernet (reference), AM with errors and TM with errors.

- Physical layer: selection of the SF.

8.3.1   Streaming Traffic

Synthetic traffic

Synthetic Streaming traffic generation was described in Section 3.5. The results obtained for these synthetic flows are presented in Table 8.1. All possible settings have been taken into

account. Following these QoS figures, it appears that streaming a content over UDP can only be acceptable coupled with the AM in the RLC layer since the TM leads to a too high packet loss rate. Over a TCP connection, the Streaming flow seems to be excellent over a RLC layer in AM; whereas the TM triggers too often the TCP retransmission mechanism leading to an intolerable packet transfer delay.

Table 8.1: Synthetic Streaming flow results.

	Layers		Delay	Packet Lost
Transport	**Data Link**	**Physical**	**[ms]**	**[%]**
	Ethernet		64	0.00
UDP	WCDMA-AM	SF=4	73	0.35
	WCDMA-TM		63	17.73
	Ethernet		67	0.00
TCP	WCDMA-AM	SF=4	108	0.00
	WCDMA-TM		$2,343$	0.00
	Ethernet		51	0.00
UDP	WCDMA-AM	SF=8	71	0.31
	WCDMA-TM		51	16.07
	Ethernet		51	0.00
TCP	WCDMA-AM	SF=8	96	0.00
	WCDMA-TM		$4,294$	0.00

Real traffic

To generate a real streaming flow, the testbed included a Real Network Helix streaming server [2] and a single client able to connect to it through the RealPlayer 10 application. The server is installed on the computer assuming the role of the RNC. An additional delay budget must be added to described results in order to estimate the real E2E delay the user encounters. Nevertheless, the access network delays set by the 3GPP in [3] were considered to identify the optimal solution. The streaming server can serve two different videos representing the high and low quality streaming as presented in Section 3.5. These two videos

are the official Real streaming server testing videos: "White Rain" and "Race Boat". The results obtained for these real Streaming flow tests are presented in Table 8.2. Following these figures, the same conclusion can be drawn as for the synthetic traffic.

Table 8.2: Real Streaming flow results.

| **Transport** | **Layers** | | **Delay** | **Packet Lost** |
	Data Link	**Physical**	**[ms]**	**[%]**
UDP	Ethernet		74	0.00
	WCDMA-AM	SF=4	86	0.13
	WCDMA-TM		69	11.15
TCP	Ethernet		76	0.00
	WCDMA-AM	SF=4	118	0.00
	WCDMA-TM		$2,465$	0.00
UDP	Ethernet		56	0.00
	WCDMA-AM	SF=8	84	0.10
	WCDMA-TM		61	15.62
TCP	Ethernet		51	0.00
	WCDMA-AM	SF=8	109	0.00
	WCDMA-TM		$3,597$	0.00

Besides the QoS metrics, the advantage of this testbed is the capacity to evaluate the QoE of all the possible settings. Methods for video quality assessment are divided into two categories: objective and subjective. Objective methods aim at mathematically estimate the impairment introduced to video during transport whilst the subjective counterparts aid in the compilation and statistical analysis of sample ratings generated by humans. Currently, subjective assessment is the main method used to quantify video quality. However, it is time-consuming and expensive. Therefore, we need an objective video quality assessment method that can estimate subjective video quality from video characteristics effectively.

The most traditional way to objectively evaluate the quality of a streamed video is the calculation of the Peak Signal-to-Noise Ratio (PSNR) [4] between the original video signal and signal

shown on screen. PSNR is the ratio between the maximum possible power of a signal and the power of degradation that affects the fidelity of its representation.

It is most easily defined via the Mean Square Error (MSE). For two $m \times n$ video frames, O and D where O is the original frame and D is the decoded one, MSE is defined as:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||O(i,j) - D(i,j)||^2 \qquad (8.1)$$

Based on (8.1), the PSNR is defined as:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \qquad (8.2)$$

where MAX_I is the maximum pixel value of the image, i.e. if the pixels are represented using 8 bits per sample, this is 255.

Typical values for the PSNR in lossy image and video compression are between 30 and $50dB$, whereas a PSNR under $20dB$ is considered as unacceptable.

The PSNR has been calculated by comparing the high quality video "Race Boat" after decoding at the receiver to the reference video which is the same video but directly read on the server. Table 8.3 presents a summary of the obtained results. The 'Perfect' column refer to the percentage of perfect frames. These frames are exactly identical in both reference and decoded video and played exactly at the same time. The 'Accept.' column presents the percentage of frame being acceptable in the terms of the PSNR ($\geq 20dB$) whereas the 'Unaccept.' column represents the percentage of frame being unacceptable ($< 20dB$). Finally the 'Mean' column shows the mean PSNR calculated on all the frames except the perfect ones.

Fig. 8.1 presents the frame-by-frame evolution of the PSNR of each of the scenario where a perfect PSNR, which is normally equal to $+\infty$, has been arbitrarily set to $100dB$ for representation purposes.

Table 8.3: PSNR evaluation results.

Layers		Perfect	Accept.	Unaccept.	Mean
Transport	**Data Link**	**[%]**	**[%]**	**[%]**	**[dB]**
	Ethernet	57.01	32.23	10.76	22.07
UDP	WCDMA-AM	1.81	40.06	58.13	18.93
	WCDMA-TM	0.62	35.20	64.18	18.82
	Ethernet	59.84	26.07	14.09	22.54
TCP	WCDMA-AM	52.32	23.84	23.84	20.93
	WCDMA-TM	29.77	10.62	59.61	13.29

However, PSNR values do not perfectly correlate with a perceived visual quality due to the non-linear behaviour of human visual system. Hence, there is a need to confront the objective evaluation results to a subjective benchmark. Based on the QoE evaluation the testbed offers, the general conclusion is that TCP over AM is the best choice under the experiment assumptions. As shown in Fig. 8.2, on the 25th second of the "Race Boat" video, using UDP directly leads to packet losses and degradation (Figs. 8.2(e)-8.2(f)), whereas TCP reaches the best resolution due to the ability to buffer content before playing it back at the receiver side. The client application can download the first few seconds of the content before playing it back in order to handle the greater delay implied by the TCP retransmission.

Considering the interest of this experience, a part of the testbed was made publicly available on the Internet. By connecting to `http://www.umtsstreamingexperience.be`, the reader can self assess the user experience of streaming a video file over a DCH.

(a) TCP over Ethernet.

(b) TCP over AM.

(c) TCP over TM.

(d) UDP over Ethernet.

(e) UDP over AM.

(f) UDP over TM.
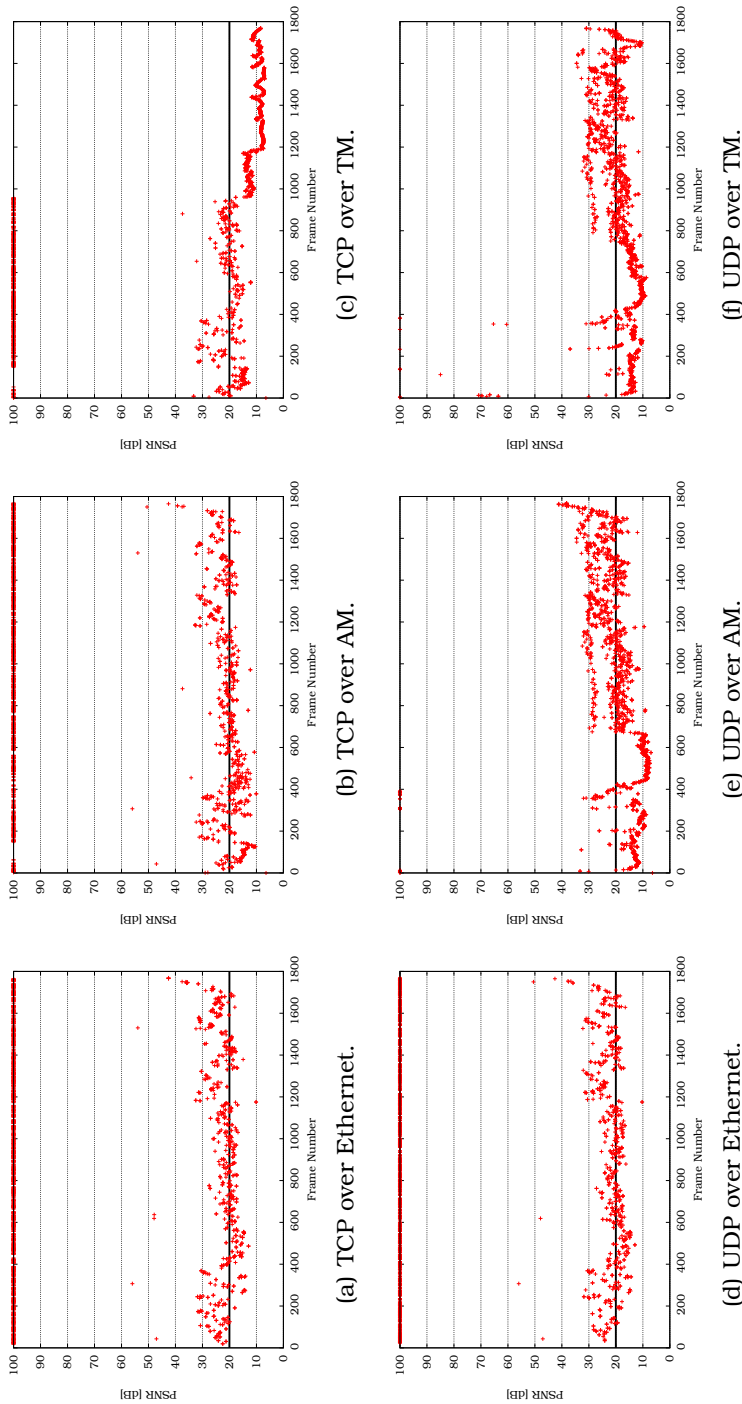
Figure 8.1: "Race Boat" PSNR evolutions.

(a) TCP over Ethernet ($PSNR$ = 100.00dB).

(b) TCP over AM ($PSNR$ = 100.00dB).

(c) TCP over TM ($PSNR$ = 100.00dB).

(d) UDP over Ethernet ($PSNR$ = 100.00dB).

(e) UDP over AM ($PSNR$ = 27.50dB).

(f) UDP over TM ($PSNR$ = 18.61dB).
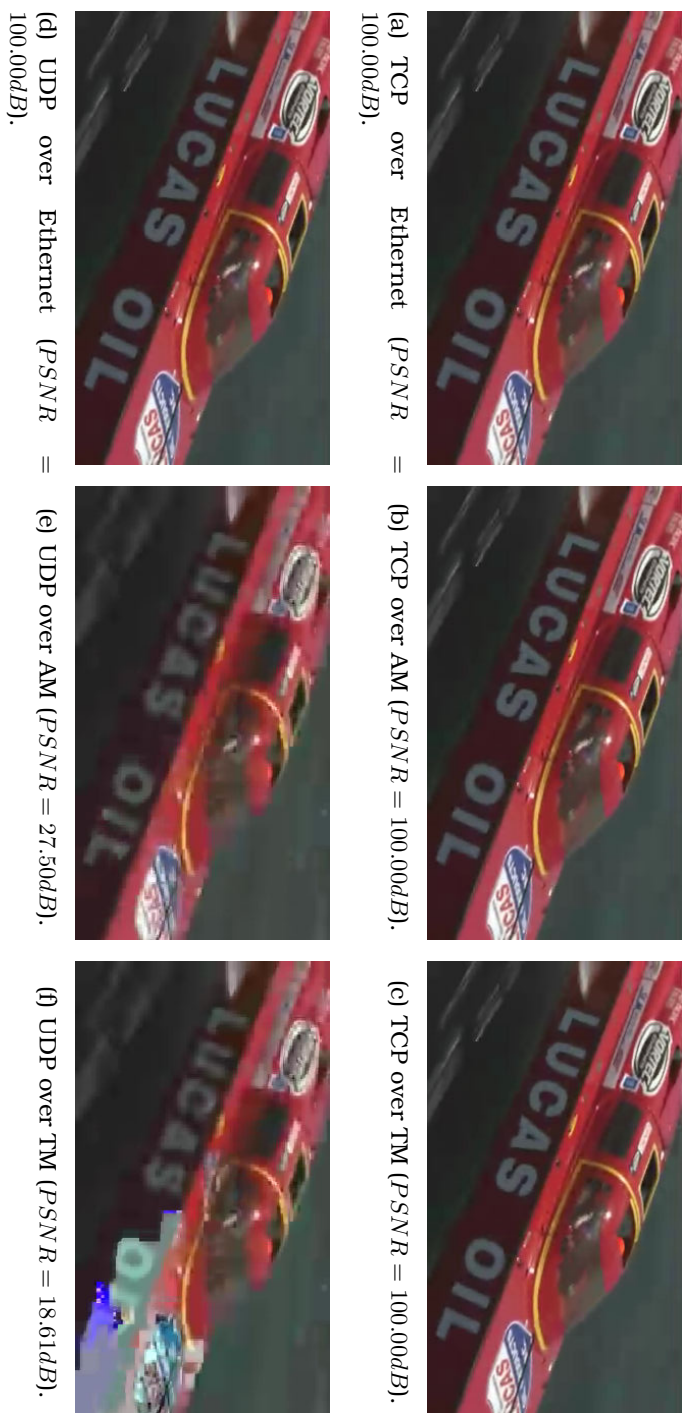
Figure 8.2: "Race Boat" video tests.

8.3.2 Conversational Traffic

Synthetic traffic

For the synthetic Conversational traffic generation (introduced in Section 3.3), there are few setting options. Due to the stringent delay sensitivity, retransmissions at the transport layer are ruled out. Hence, UDP must be used. As exposed in Chapter 6, Conversational flows do not have any SF degradation possibility since they only accept a SF $= 64$. The only option left is the selection of the RLC mode: AM or TM. The results (Table 8.4) show that TM suffers from significant packet losses. AM appears to be a better choice even if it implies a longer packet transfer delay.

Table 8.4: Synthetic Conversational flow results.

	Layers		Delay	Packet Lost
Transport	**Data Link**	**Physical**	**[ms]**	**[%]**
	Ethernet		50	0.00
UDP	WCDMA-AM	SF=64	61	0.39
	WCDMA-TM		50	12.81

Real traffic

The real VoIP tests were run with two Grandstream Networks BudgeTone-100 Series IP phones [5] shown in Fig. 8.3. They used the G.720 vocoder with silent suppression, which is close to the AMR vocoder since it also produces $32 - Byte$ packets every $20ms$. The results of these experiments are summarised in Table 8.5. These figures are inline with the synthetic ones.

From the QoE perspective, it appeared that using TM, even if the packet loss rate is above 10%, the communication quality remains acceptable. In this case, UDP over a RLC layer in TM is recommended.

To be complete, besides this subjective quality evaluation, an objective quality assessment process should also be set up. Objective audio quality assessment also exist similar to the PSNR
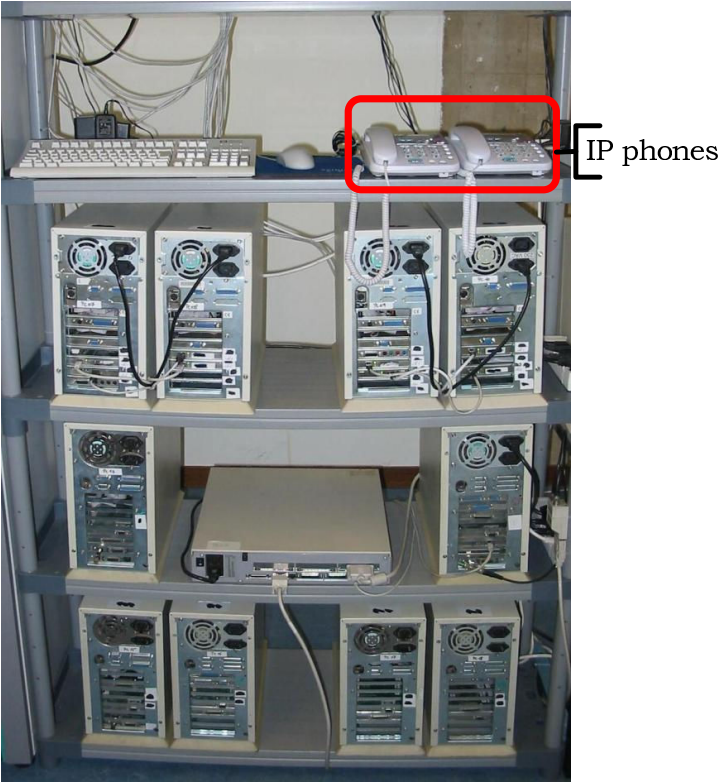
Figure 8.3: Testbed's IP phones.

Table 8.5: Real Conversational flow results.

Layers			Delay	Packet Lost
Transport	Data Link	Physical	[ms]	[%]
	Ethernet		50	0.00
UDP	WCDMA-AM	SF=64	71	0.11
	WCDMA-TM		50	12.24

calculation for streamed video. For example, Perceptual Evaluation of Speech Quality (PESQ) [6] is an enhanced perceptual quality measurement for voice quality in telecommunications and Perceptual Evaluation of Audio Quality (PEAQ) [7] is specifically developed to apply to E2E voice quality testing under real network conditions like in VoIP.

8.4 Summary

This chapter has presented how the testbed enables the evaluation of the QoE of services delivered through a UTRAN. It helped identifying the best parameter setting of the emulated UMTS layers. Over a typically impaired wireless channel, it appears that TM is not suited for Streaming flows, as it results in significant PER on UDP or significant delays on TCP, whereas it is acceptable for Conversational flows.

Bibliography

[1] Hugues Van Peteghem, Laurent Schumacher, and Christophe De Vleeschouwer. UMTS Layers Parametrisation for Real-Time Flows. In *Proceedings of the 1st International Workshop on Mobile Content Quality of Experience (MobConQoE)*, Vancouver (Canada), August 2007.

[2] E. Regan. Helix™ Server Unlimited. Technical report, Real Networks, March 2007. http://www.realnetworks.com/products/media_delivery.html, last visited: March 13, 2008.

[3] Technical Specification Group Radio Access Network. Quality of Service (QoS) concept and architecture. Technical Report TS 23.107 V7.1.0, 3rd Generation Partnership Project (3GPP), September 2007.

[4] Zhou Wang, Hamid R. Sheikh, and Alan C. Bovik. *Handbook of Video Databases: Design and Applications*, chapter 41 - Objective Video Quality Assessment, pages 1041–1078. CRC Press, September 2003.

[5] John Huang. User Manual, BudgeTone-100 Series IP Phone. Technical Report 1.0.8.32, Grandstream Networks, September 2006. http://www.grandstream.com/bt101.html, last visited: March 13, 2008.

[6] ITU-T Recommendation. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-band Telephone Networks and Speech Codecs. Technical Report P.862, International Telecommunication Union, 2001.

[7] Opticom. PEAQ — Perceptual Evaluation of Audio Quality. Technical report, Opticom, 1999.

Part IV

Conclusion & Perspectives

Chapter 9

Conclusion

9.1 Achievements

The present dissertation aimed at designing, building and validating a suitable open-source software/hardware RT platform in order to thoroughly analyse, model, emulate and improve QoS management in the UTRAN. As discussed in Chapter 2, this testbed is not the first attempt in that direction. However, it exhibits a number of innovative achievements which make it unique.

9.1.1 Four Traffic Classes

The testbed simultaneously encompasses the four UMTS traffic classes whereas other testbeds focus on one or two of these classes only. This permits to evaluate each traffic class when competing with the three others in order to gain access to the scarce radio resources, considering its own QoS requirements.

This characteristic of the testbed has been used in Chapters 6 in order to establish a SF allocation map giving the best possible treatment to a data flow in relation to its traffic class, the corresponding QoS requirements, and the user profile of the UE source/sink. Chapter 7 also relied on this property to analyse an improved NRT sessions channel switching strategy based on a split of the downswitch timer.

9.1.2 Air Interface

The emulation of the Rel'99 UMTS air interface over a wired Ethernet link is the second achievement of the present dissertation. Based on a Linux kernel, the testbed is able to mimic the RLC transport channels thanks to several parallel IP traffic queues managed by specific scheduling algorithms. It is also able to emulate the effect of the typical impairments of the wireless links through a modified version of the NetEm component. The emulation includes SF-based limited bandwidth, transmit power ceiling at the NodeB, PDU fragmentation, SNIR-dependent packet loss, and delays due to optional lower layer retransmissions.

Such an emulator would be useless without any validation. This was performed in several cases. The packet loss history was compared to similar statistics collected during field tests in Section 5.3.1. The gap and burst distributions were showing a good match. Another partial validation was performed in Section 5.3.5 where the Gfactor computed on the testbed was compared to Gfactor figures from the open literature. The curves showed a good match, meaning that the interference level computation in the testbed is realistic. Finally, Section 5.6 presented a comparison between results from the testbed and analytical results. There was again a good match between these results. This match validated the modelling of RLC transport channels with the help of parallel IP packet queues.

9.1.3 Multi-User

Where most similar testbeds focus on a single active user, the present testbed emulates several terminals, as shown in Chapters 6 and 7. Nevertheless, a detailed focus on a single active user is still possible, as illustrated in Chapter 8.

The testbed manages each of the UEs independently. Chapter 4 described the UE mobility management based on a Gauss-Markov random walk and SNIR-dependent handover procedure. On the other hand, Chapter 3 exposed the UE traffic generation/reception scheme based on a generic three-level model whose parameters are derived from statistical distributions.

9.1.4   QoE Evaluation

Another innovation of the present testbed is the possibility to truly evaluate the QoE of a flow being delivered over the UTRAN. As presented in Chapter 3, the emulation generates synthetic traffic mainly for the sake of simplicity and scalability. Chapter 8 inserted a real life application session in a full emulation run, enabling the evaluation of the user's QoE in a large variety of scenarios.

9.2   Perspectives

9.2.1   Short-Term Perspectives

The value of the testbed was shown in the different chapters of the present dissertation, especially in terms of QoS and QoE. On a short term basis, the present work is of interest for network operators. Its exploitation helps to provide advice and hints on how to tune RRM, CAC, and scheduling algorithms of a UTRAN network supporting any combination of the four UMTS traffic classes.

9.2.2   Further Testbed Evolution

The modularity of the testbed was designed in such a way as to enable modifications of a single module as straightforwardly as possible. One example would be a modification of the wireless interface modules in order to mimic other access network technologies than just the Rel'99 UMTS like HSDPA, Mobile WiMAX or LTE/SAE.

Such an evolution is happening already since some features of UMTS Rel'5 UMTS, as known as HSDPA, have been implemented in [1] but not yet ported on the testbed. A second step would be to implement another module to emulate the air interface of the last technology to receive the 3G label, namely Mobile WiMAX. This implementation could be based on a simulation project presented in [2] which relies on a channel-aware QoS scheduling architecture similar to the one illustrated in Fig. 5.9.

9.2.3 Exploitation of the Testbed

Besides developing new modules in order to enrich the testbed features, one could still use this testbed to evaluate applications' QoE. This approach has already been introduced in the present dissertation (Chapter 8), but other projects can go sensibly deeper in the evaluation, aiming at consequently adapting the application and lower layers design.

For example, the new WalCoMo project, which has just started at the University of Namur, aims at designing and developing an innovative system which turns regular TV contents into a video flow suited for display on portable terminals, despite the small screen size and the transmission impairments. This requires adapting and protecting the transmitted content. Adaptation implies to reduce the spatial and temporal resolution contents, but also to be focused on the areas of interest for the user [3].

As discussed here above, the outcome of this thesis is potentially the first step on the road of many more projects. It has contributed to design, build and validate a testbed on which other projects can lay down to test their own ideas on a quasi-realistic infrastructure. In that longer term perspective, the testbed could be of interest for infrastructure manufacturers which could benefit from the testbed's emulations in order to elaborate and optimise new RRM, CAC, or scheduling algorithms and protocols.

Bibliography

[1] Lucia D'Acunto and Claudio Greco. Packet scheduling in HSDPA. Technical report, FUNDP, June 2007.

[2] Antonio Iera, Antonella Molinaro, and Sara Pizzi. Channel-Aware Scheduling for QoS and Fairness Provisioning in IEEE 802.16/WiMAX Broadband Wireless Access Systems. *IEEE Network*, 21(5):34–41, 2007.

[3] Christophe De Vleeschouwer, Benoît Macq, Laurent Schumacher, Benoît Quirynen, and Edouard Vanhende. WALCOMO : Worthy visuAL COntent on MObile, submission report. Technical report, Wist2, 2007.

`http://www.fundp.ac.be/recherche/projets/page_view/en/07296002/`, last visited: March 13, 2008.

Part V

Appendices

Appendix A

Major Scientific Contributions

Description of an IPv6 Linux-Based UTRAN Testbed

Presented at the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)
Barcelona (Spain), March 2006

Description of an IPv6 Linux-Based UTRAN Testbed

Van Peteghem H. and Schumacher L.
Pôle "Réseaux et Sécurité"
The University of Namur, FUNDP
Namur, Belgium
`{hvp,lsc}@info.fundp.ac.be`

Abstract—**Network modelling and traffic characterisation are essential research activities towards the optimisation of UMTS. We present here our IPv6 Linux-based UTRAN testbed which aims at emulating a real UTRAN segment. This emulation is relying on statistical traffic distributions, random walk UE mobility and line-of-sight power-based handover management. By interconnecting those different modules, we expose the first testbed results about the handover management and our views regarding the QoS management.**

Index Terms—**UTRAN emulator, testbed, traffic modelling, terminal mobility, handover.**

I. INTRODUCTION

The evolution of mobile communications and the Internet has lead to the third generation of cellular networks (3G), known as Universal Mobile Telecommunications System (UMTS). With the launch of this new generation of mobile communication, users are now truly able to access the Internet from mobile terminals, allowing them to enjoy a large variety of added value services.

In this context, UMTS Terrestrial Radio Access Network (UTRAN) modelling and traffic characterisation are important steps of the research. The goal of our project is to provide a representative testbed which emulates a real UTRAN segment supporting typical traffic samples based on significant statistical properties. This gives us a tool able to emulate real UTRAN behaviours with more realism and less theoretical hypotheses than simulators (ns-2 or opnet), and less costs than a real UTRAN deployment. Such a testbed will be useful to investigate issues like Quality of Service (QoS) or Radio Resource Management (RRM). The objective of this paper is to present the first release of our testbed, its implementation and the first results.

The rest of this paper is organised as follow. Section II presents the testbed architecture. Section III describes the supported traffic classes. Section IV depicts our world representation, explains how we emulate mobility above a wired network infrastructure and exposes the testbed handover management. Section V then presents the first results obtained. Section VI discusses the main future work items. Finally conclusions are drawn in Section VII.

II. TESTBED DESCRIPTION

As shown on the upper part of Fig. 1, we emulate a UTRAN segment consisting of one Radio Network Controller (RNC) managing four NodeBs each of them serving a population of User Equipments (UEs). The NodeBs are placed on a hexagonal grid but, for representation simplicity, the four cells are represented on a rectangle map.
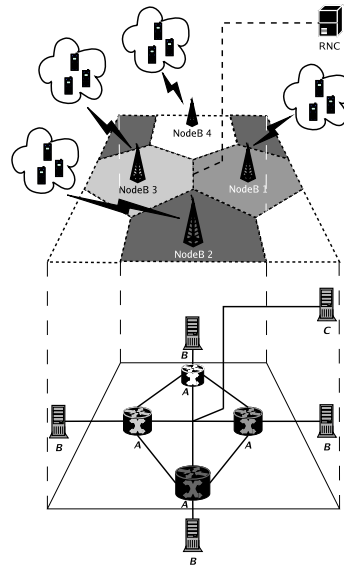

Fig. 1. UTRAN testbed representation.

In order to emulate this UTRAN segment, we have built a testbed which is composed of nine Linux personal computers interconnected with $10\ Mb/s$ Ethernet links, as shown on the lower part of Fig. 1. It is formed by three groups of computers, namely *A*, *B* and *C*.

The first one, the *A* group, is made of Pentiums III 750 MHz with 512 MB of RAM. They all have four network interfaces connected to various sub-networks. The *B* group is composed of Celerons 300 MHz with 64 MB of RAM. Each one is connected to one of the computers of the *A* group. Finally, the *C* group, which is in fact composed of a single computer, has the same hardware specification as the *A* group computers and is directly connected to all the *A* group machines. For the sake of homogeneity, we have installed the same operating system on each machine: a Linux Red-Hat based Fedora Core 1, kernel 2.4.26.

The *A* group is placed in the centre of the network. These computers are used as routers running the routing freeware *Quagga* [1], whereas the computers of the *B* and the *C* groups are placed in periphery and used as traffic sources/sinks. To serve that purpose, we have installed the open-source application *TG* [2] which is a traffic generator program able to create one-way UDP or TCP streams between two computers. This traffic is detailed in Section III in terms of inter-arrival times and packet lengths with the help of statistical distributions.

Using this testbed, the idea is to mimic the working of a real UTRAN segment:

– The *A* group computers act as the NodeBs. Their role is to take care of the traffic coming from the UE (resp. RNC) and to forward it to the RNC (resp. UE). Most of the QoS management functions will be performed at these spots since they are the bottlenecks of the UTRAN (border between the wireless and the wired networks). Actually, in the early UMTS standardisation the NodeBs were seen as simple access points linking the UEs to the UTRAN and the Core Network (CN). But, since the High Speed Downlink Packet Access (HSDPA [3]) introduction, they have much more responsibilities in terms of QoS management and packet scheduling.

– Each computer of the *B* group represents a population of UEs attached to their serving NodeB (emulated on the attached *A* group computer). Each of them will generate several uplink flows (from the UE to the RNC). In order to simultaneously emulate several autonomous UEs, we implemented a control panel based on virtual network interface management. The role of this control panel is to facilitate the UE management by delegating them a single IP address.

– The *C* group computer plays the role of the RNC. It will be the sink of the traffic coming from the four NodeBs and will generate the downlink traffic (from the RNC to the UEs). As in a real UTRAN, the complexity is gathered in the RNC. It consequently controls the entire emulation.

III. TRAFFIC GENERATION

In [4], the 3rd Generation Partnership Project (3GPP) has defined four classes of services that need to be supported in UMTS. In order to characterise them, it might be useful to focus ourselves on four representative applications (Table I):

TABLE I
TRAFFIC CLASSES AND APPLICATIONS.

3GPP Traffic Classes	Representative Applications
Conversational	VoIP
Interactive	Web browsing
Streaming	Video streaming
Background	E-mail

Several studies have already been done about UMTS traffic modelisation [5], but none of them characterised simultaneously all four traffic classes. Most of them consider only two classes, namely conversational and interactive. Although these traffic classes will certainly be the most popular, our aim is to be able to emulate any UMTS traffic mix.

To characterise the composite traffic of a single user, we fit each representative application in a three-level model shown on Fig. 2 [6]. Here is a quick run-down of their meaning:
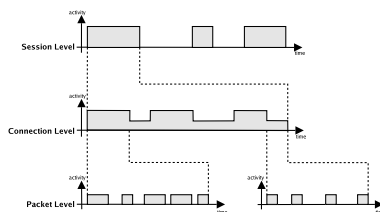
Fig. 2. Three-level traffic model.

– Session Level: it lasts as long as the application is running. Its statistics are mainly influenced by user behaviour.

– Connection Level: it describes the connection behaviour of a single session. The pattern chosen for our sources is based on an On-Off (High-Low) model in which it could be possible to generate traffic on both states.

– Packet Level: it represents the packet inter-arrival and size distribution for each state of the Connection Level.

As an example to illustrate this three-level model, let us consider the conversational class. In this case, the session level describes the moments when a user is engaged in a phone call and the moments when s/he is not. When the user is engaged in a call, the connection level represents the time intervals during which the user is speaking and the time intervals during which s/he is simply listening. Finally, when the user is speaking, the packet level gives the size of each packet sent and the time interval separating two successive packets.

Based on this three-level division, the four traffic classes are described in terms of statistical distributions, in line with literature [7]–[10] and 3GPP standards [11]–[13] when applicable. These descriptions are summarised in Table II.

TABLE II
TRAFFIC CLASSES STATISTICAL DISTRIBUTIONS.

Classes	Session-Level	Connection-Level	Packet-Level
Conversational	Poisson	Two States Markovian	Constant
Interactive	Poisson	Geometric & Exponential	Pareto & Exponential
Streaming	Poisson	–	Pareto
Background	Poisson	Lognormal & Pareto	Lognormal
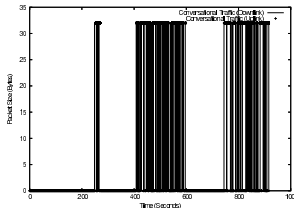
(a) Conversational.

(b) Interactive.
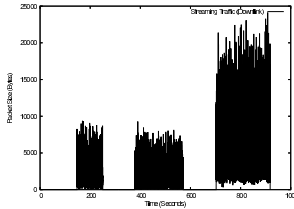
(c) Streaming.
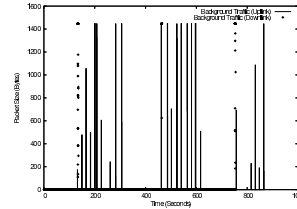
(d) Background.

Fig. 3. UE activity graphs.

This probabilistic model of the traffic is used in the testbed to initiate sessions of any of the four traffic classes at any time in accordance to Poisson processes. It means for example that an emulated UE may very well initiate a conversational session while already having an ongoing interactive sess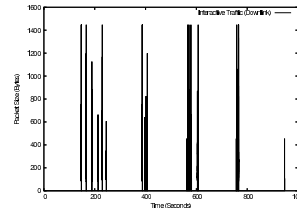ion. This would be translated in a real UTRAN by the fact that an UMTS user gets called as s/he was browsing the Internet on his/her 3G terminal.

Before generating these different traffics, we had to modify the *TG* source code to support the Pareto distribution and the IPv6 addressing[1] . As a matter of validation, we simulated a single UE connected to a NodeB, generating and receiving all kinds of traffic during $1,000$ seconds. Fig. 3 depicts the activity graphs of each of the four classes.

[1]The validation of the Pareto distribution implementation and the new version of *TG* are available on: http://www.info.fundp.ac.be/ ~hvp/rech/doc/index_en.html

Note that we only emulate uplink traffic for the applications emitting a significant amount of data. Consequently, as we can see on Fig. 3(a) and 3(d), the uplink direction of traffic is emulated only for the conversational and background classes. Interactive uplink traffic (mainly constituted of TCP acknowledgements) is not emulated.

There is a need of synchronisation for these two bidirectional traffic classes. Indeed, in a real conversation both of the participants are speaking throughout the session. Therefore, the emulated downlink and uplink traffics have to be synchronised and mixed. In the background case, we can think of a user downloading his/her mails and answering some of them one by one after their download. In this case, the emulation begins with a downlink traffic followed by multiple smaller uploads.

IV. Mobility Management

A. Emulated World

As previously mentioned, we emulate a UTRAN segment consisting of four cells. We decided to place the NodeBs at 2,800 meters away from each other on an hexagonal grid which is the standard macrocellular set-up [14], [15]. This means that the cell radius equals 1,400 meters and the total surface of the four emulated cells is about $27\ km^2$. But in a real world the NodeBs are not omnidirectional, they are composed by three sectors emitting in different directions. This leads us to adapt our world representation. Fig. 4 depicts the world evolution from the omnidirectional to the trisectorial NodeBs. On Fig. 4(b), the arrows represent the sector's main emitting directions.
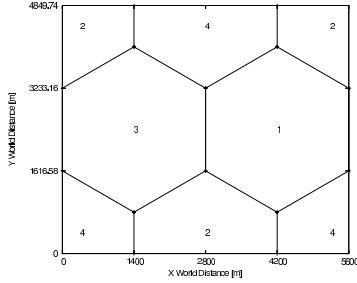
(a) Omnidirectional NodeBs.
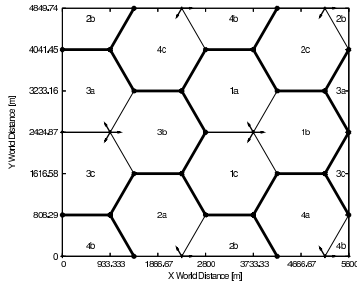
(b) Trisectorial NodeBs.

Fig. 4.   Emulated world representations.

Note that UEs are not constrained within this $27\ km^2$ area thanks to a wrap around technique. When a UE is about to leave the emulated world by crossing one side, it reappears at the opposite side (still within the same cell if necessary), as they were somehow moving on a torus (Fig. 5).
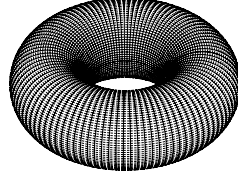
Fig. 5.   Torus world.

The testbed supports up to 30 users by sector (up to 360 users in the entire emulated world). This number is fixed at the beginning of an emulation run and does not vary during it. We assume that the different UEs are switched on during the entire emulation. This means that we do not have any UE birth-death mechanism. However, their activity is controlled by the stochastic models of Section III. As a result, during an emulation with say 12 users, we may encounter periods of inactivity as well as full load periods when all 12 users are active.

This is illustrated on Fig. 6 which shows the aggregate downlink traffic of a given NodeB during an emulation of 1,200 seconds with 12 users. Up to three users are simultaneously streaming around the 600th second whereas there was no traffic at all a few seconds before.
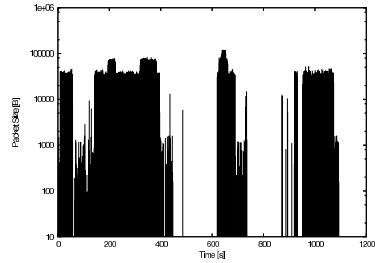
Fig. 6.   NodeB downlink traffic.

B. Random Walk

To enable terminal mobility inside the testbed, we decided to focus on four specific UE speeds: 3 km/h representing a pedestrian walk, 30 and 70 km/h which represents a UE on board of a vehicle inside a city and 120 km/h which finally represents a UE embarked on a train or on a car on the speedway.

To describe the UE mobility, we opted for the Gauss-Markov mobility model [16]: every second, the distance the UE will cover and the direction it will follow during the next second are drawn randomly. Keeping in mind that a vehicle

moving at 120 km/h could not turn as abruptly as a pedestrian, the range of possible directions is then reduced with increasing UE speed. Fig. 7 illustrates the course of two different UEs respectively moving at 120 and 30 km/h during 600 seconds.
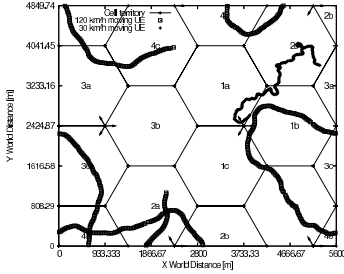
Fig. 7.   Random walk at 2 different speeds.

C. Handovers

Now that our UEs are able to move, we have to establish a handover procedure which will decide where and when a handover has to be performed. This procedure is based on the Signal to Noise plus Interference Ratio (SNIR):

$$SNIR = \frac{Reception\ Power}{Interferences + Noise} \qquad (1)$$

The computation of the reception power is based on the signal power received on a specific UMTS physical channel known as CPICH (Common PIlot CHannel). Our hypotheses concerning the CPICH as well as all the other testbed cellular parameters are summarised in Table III.

TABLE III
TESTBED CELLULAR PARAMETERS.

Parameter	Setting
Emission power [dBm]	43
Sectorial gain [dBi]	14
Noise figure [dB]	9
Thermal noise density [dBm/Hz]	-174
Emitting frequency [GHz]	2.14
NodeBs height [m]	30
UEs height [m]	1.5
CPICH spreading factor length [chips]	256
CPICH power [% of the total power]	10
SNIR handover threshold [dB]	4

The reception level is then emulated by a calculation mainly influenced by the distance separating the UE and its serving NodeB (*SNodeB*, the NodeB the UE is connected to), and by the angle created between the UE position and the *SNodeB*'s main emitting direction:

– The received power decreases as the distance between the UE and its *SNodeB* increases. This relation is known as path-loss and it is not a linear one (Fig. 8(a)). To model this, we use the Okumura-Hata wireless propagation equations and their UMTS adaptation [17].
– As shown on Fig. 8(b), if a UE is connected to a trisectorial NodeB, its received power will decrease as it will move away from the sector main lobe [18].
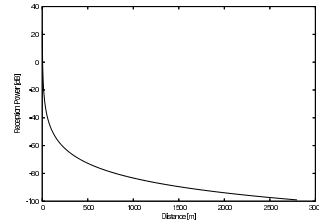
(a) Received power vs. UE – *SNodeB* distance.
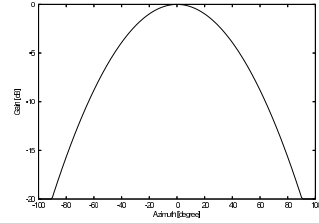
(b) Antenna gain vs. UE – *SNodeB* angle.

Fig. 8.   UE reception influencing factors.

Another concept, embedded in the WCDMA, influencing the UE reception level is the signal spreading [19]. It is able to amplify a signal and to attenuate the interferences. The higher the spreading factor is, the more amplified the signal will be (resp. the more attenuated the interferences will be) but the slower the sending of the data on the wireless interface will be. The spreading factor's choice is then a real challenge the RNC has to overcome.

In the testbed, the interferences are simply considered as the sum of the received power from the NodeBs but the UE *SNodeB*, multiplied by two. Indeed, we only have four NodeBs in our emulated world, but to compute a representative interference figure we have to take into account all the NodeBs lying around the *SNodeB*. In our hexagonal grid world representation, this means we have to consider six instead of three surrounding NodeBs.

As the noise level is assumed to be constant during an entire emulation, (1) finally becomes:

$$SNIR = \frac{SNodeB \; Rec. \; Pow. \; * \; SF_{CPICH}}{2 * (\sum Other \; NodeBs \; Rec. \; Pow.) + Noise} \quad (2)$$

Derived from (2), Fig. 9 represents the SNIR level map of our emulated world.
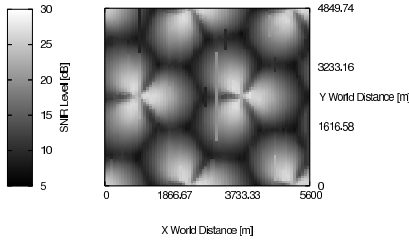
Fig. 9.   SNIR map.

A handover occurs when the difference between the power of the *SNodeB*'s signal and the power of another NodeB's signal is greater or equal to the handover threshold (Table III). Based on this continuous SNIR evaluation the RNC knows exactly when to handoff a UE.

V. RESULTS

Using the previously presented modules (traffic generation, emulated world, random walk and SNIR computation), it is now possible to mimic in a realistic way what happens in a real UTRAN when a UE faces a handover. Fig. 10 illustrates the results from the working of the testbed. It represents the journey of one UE during 450 seconds generating and receiving conversational traffic only.

Thanks to the mobility graph of Fig. 10(a), we figure out that the UE undergoes a handover passing from the NodeB #1 (sector a) to the NodeB #3 (sector b) at about the middle of the emulation.

As we look at the SNIR evolution of the UE (Fig. 10(b)), we observe that the handover exactly occurs when the power of the signal received by the NodeB #3 (sector b) exceeds by $4 \; dB$ the signal received by the UE's *SNodeB* power. It is indeed what we expected regarding the handover SNIR threshold value (Table III).

Finally, if we look at the activity graph of Fig. 10(c), we discover that the UE was active (sending/receiving data) when the handover occurred. We can clearly distinguish the end of the first connection and the beginning of the second connection, both totally synchronised in uplink and downlink directions.
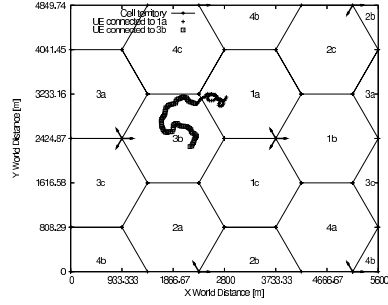
(a) Mobility graph.
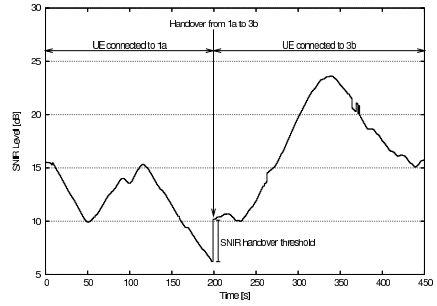
(b) SNIR graph.
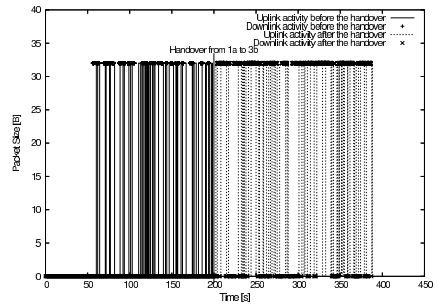
(c) Activity graph.

Fig. 10.   Handover impacts.

From the testbed point of view, this handover is translated by a process migration from a computer to another (Fig. 11): first of all, the traffic emitting processes have to be stopped on the source computers (the *C* computer and NodeB #1 i.e. the eastern computer of the *B* group), then new processes corresponding to the traffic generation after the handover can be started on the *C* computer and new destination computer (NodeB #3 i.e. the southern computer of the *B* group). As a result, the packets belonging to the same activity session do not use the same physical route after this handover. Consequently, we can say that our UMTS testbed supports UE mobility above a wired infrastructure.
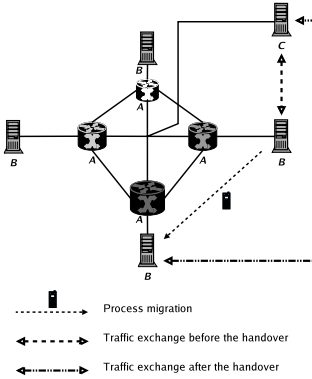
Fig. 11.   Handover testbed's point of view.

VI. FUTURE WORKS

A. OVSF Tree Management

Following the WCDMA principles, before being able to send/receive any data, a UE needs to query a spreading factor. This will help to distinguish the UE's transmission from the other data transiting over the wireless interface. Our next step will thus be to take into account the constraints of the Orthogonal Variable Spreading Factor (OVSF) tree to allocate spreading factors to each user transmitting data over the wireless link. This allocation is established by the RNC and is based on the UE's user profile (platinum, gold, bronze or silver), traffic class (conversational, interactive, streaming or background) and SNIR level.

The RNC has to manage twelve OVSF trees, one for each sector in the emulated world, and tries to satisfy as many users as possible. In this way, a trade-off has to be made between allocating a high spreading factor to each user (which will lead to a spreading code starvation) and allocating a spreading factor to the greatest possible number of users (which will lead to users' general dissatisfaction).

A final OVSF issue the RNC has to overcome is to avoid allocation deadlocks. Because of its tree representation, the OVSF tree may have enough room to accept a new spreading factor but would still have to reject the query due to previous bad spreading factor placements [20].

B. Transport Channels Modelling

To transit over the wireless interface, the data may use different UMTS transport channels: the Dedicated CHannel (DCH), the Dedicated Shared CHannel (DSCH) and the Forward Access CHannel (FACH). Note that the emulation of a fourth transport channel, the High Speed-DSCH (HS-DSCH), is still under study for the time being. Based on the 3GPP standardisation [19], the channel used depends on the traffic class (Table IV) and, as for the spreading factor allocation, the channel assignment is decided by the RNC.

TABLE IV
TRAFFIC AND CHANNELS MAPPING.

3GPP Traffic Classes	Authorised Transport Channel(s)
Conversational	DCH
Interactive	DCH
Streaming	DCH or DSCH
Background	DCH, DSCH or FACH

Our idea is to introduce packet policing/sha-ping/scheduling at the NodeB level (*A* group computers) to emulate those different channels. To do so, we are thinking of using the Linux *TC* tool and the *traffic class* field of the IPv6 header to dynamically create hierarchical token buckets on each of the NodeBs (Fig. 12).
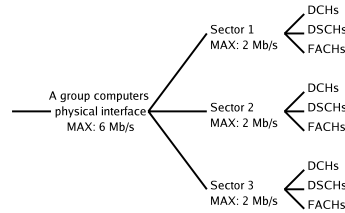
Fig. 12.   Hierarchical token buckets.

The hierarchical property of the token buckets matters since it enables token borrowing between two buckets. We think here to apply this characteristic to the shared transport channels, namely the DSCHs and the FACHs.

C. Wireless Link

Finally, the wireless link emulation is also an important step since the testbed tries to emulate a UTRAN based on a wired network. Our ambition is use the freeware *Click Modular Router* [21] to insert an emulated RLC layer in

the conventional OSI model. It will take place between the MAC and the network layers and will support the three transmission modes (acknowledged mode, unacknowledged mode and transparent mode), the TTI scheduling (firstly $10\,ms$ as in Rel' 99 and later $2\,ms$ needed by the HSDPA) [19] and a random PER-driven packet dropping process based on the UE's SNIR level (since the UMTS wireless interface is much less reliable than a wired link).

VII. CONCLUSION

We have described our IPv6 Linux-based UTRAN testbed. We have exposed how we emulate a real UTRAN by means of different modules such as traffic modelling, random walk algorithm and SNIR level dependent handover procedure. We have also presented the first results obtained by combining all these modules about the handover management.

This project shows to which extend a rather simple testbed based on freewares and an open-source operating system can emulate a much more complex system. It is aiming at going open-source. However, as it is currently still in its infancy, such a move would be premature. Nevertheless, we are open to partnership. Meanwhile the status of the project can be checked on `http://www.info.fundp.ac.be/~hvp/rech/doc/index_en.html`.

REFERENCES

[1] Quagga routing software suite. `http://www.quagga.net`, last visited: 04 January 2006.

[2] Traffic generator tool. `http://www.postel.org/tg/tg.htm`, last visited: 04 January 2006.

[3] Harri Holma and Antti Toskala. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. Wiley inter-science, April 2006.

[4] Technical Specification Group Radio Access Network. TS 23.107 V6.3.0, Quality of Service (QoS) concept and architecture. Technical report, 3rd Generation Partnership Project (3GPP), June 2005.

[5] J.A. Aguiar and L.M. Correia. Traffic Source Models for the Simulation of Next Generation Mobile Networks. In *Proceedings of the 7th Wireless Personal Multimedia Communications (WPMC)*, Abano Terme (Italy), September 2004.

[6] A.B. García, E. García, M. Álvarez-Campana, J. Berrocal, and E. Vázquez. A Simulation Tool for Dimensioning and Performance Evaluation of the UMTS Terrestrial Radio Access Network. *Lecture Notes In Computer Science*, 2515:49–60, 2005.

[7] Alexander Klemm, Christoph Lindemann, and Marco Lohmann. Traffic modeling and characterization for UMTS networks. In *Proceedings of the 44th Globecom conference*, San Antonio (TX - USA), November 2001.

[8] Y. Li Frank and Stol Norvald. QoS Provisioning using Traffic Shaping and Policing in 3rd-generation Wireless Networks. In *Proceedings of the 3rd Wireless Communications and Networking Conference (WCNC)*, Orlando (FL - USA), March 2002.

[9] Soam Acharya and Brian Smith. An Experiment to Characterize Videos Stored on the Web. In *Proceedings of the 5th Multimedia Computing and Networking (MMCN)*, USA, January 1998.

[10] A. Reyes, E. Gonzalez-Parada, E. Casilari, J.C. Casasola, and A. Díaz-Estrella. A page-oriented WWW traffic model for wireless system simulations. *Teletraffic Engineering in a Competitive World (ITC16)*, pages 1271–1280, 1999.

[11] Technical Specification Group Radio Access Network. TS 30.03 V3.2.0, Selection procedures for the choice of radio transmission technologies of the UMTS. Technical report, 3rd Generation Partnership Project (3GPP), March 1998.

[12] Technical Specification Group Radio Access Network. TR 25.896 V6.0.0, Feasibility Study for Enhanced Uplink for UTRA FDD (Release 6). Technical report, 3rd Generation Partnership Project (3GPP), April 2004.

[13] Technical Specification Group Radio Access Network. TR 25.933 V5.4.0, IP transport in UTRAN (Release 5). Technical report, 3rd Generation Partnership Project (3GPP), January 2004.

[14] Technical Specification Group Radio Access Network. TR 25.876 V1.7.1, Multiple Input Multiple Output in UTRA. Technical report, 3rd Generation Partnership Project (3GPP), October 2005.

[15] Technical Specification Group Radio Access Network. TR 25.892 V6.0.0, Feasibility study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement. Technical report, 3rd Generation Partnership Project (3GPP), June 2004.

[16] Alexandrosz Burulitisz, Sándor Imre, and Sándor Szabó. On the Accuracy of Mobility Modelling in Wireless Networks. In *Proceedings of the 39st International Conference on Communications (ICC)*, Paris (France), June 2004.

[17] COST 231 Final Report. Digital Mobile Radio Towards Future Generation Systems. Technical report, COST 231, November 1999.

[18] Spatial Channel Model AdHoc Group. Spatial Channel Model Text Description. Technical report, 3GPP & 3GPP2, April 2003.

[19] Harri Holma and Antti Toskala. *WCDMA for UMTS*. Wiley inter-science, third edition, October 2004.

[20] YeChee Tseng, ChihMin Chao, and ShihLin Wu. Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree. *Management IEEE Transactions on Mobile Computing*, December 2002.

[21] The click modular router project. `http://pdos.csail.mit.edu/click/`, last visited: 04 January 2006.

Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS

Presented at the 64th Vehicular Technology Conference (VTC-fall) Montreal (Canada), September 2006

Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS

Van Peteghem H. and Schumacher L.
Pôle "Réseaux et Sécurité"
FUNDP - The University of Namur
Namur, Belgium
{hvp,lsc}@info.fundp.ac.be

Abstract—Based on the end-to-end delay experienced by the user, this paper derives a spreading factor allocation map for the four UMTS traffic classes and for four user profiles with the help of a computer testbed emulating the UTRAN. Using this allocation map, the performance of the UTRAN is measured in terms of cell throughput and blocking rate in different scenarios.

I. INTRODUCTION

With the huge improvement of bandwidth and strong support of Quality of Service (QoS) in the Universal Mobile Telecommunications System (UMTS) core network, the UMTS Terrestrial Radio Access Network (UTRAN), especially its air interface, is now the bottleneck for the delivery of value added services. Indeed, compared to wired links, wireless links have a smaller bandwidth and are then not fitted to effectively support QoS as we consider it over a wired network. Therefore, an adaptation has to be made to take into account the lower bandwidth of such links. To overcome this problem, the air interface resources have to be carefully managed according to the traffic to be sent over the wireless interface.

The rest of this paper is organised as follows. Section II exposes the link existing between the Spreading Factor (SF) and the data rate available. Section III presents our SF allocation scheme based on the four UMTS traffic classes and four user profiles. Section IV evaluates the resulting SF allocation map to the operator requirements such as cell throughput and blocking rate. Section V explains our next steps in this research. Finally conclusions are drawn in Section VI.

II. SPREADING FACTORS AND DATA RATES

Since the UMTS uses Wideband Code Division Multiple Access (WCDMA) as air interface, it has to respect the spreading/despreading concept [1]. Transmissions from a single source are separated by channelisation codes. The spreading/channelisation codes of UTRAN are based on the Orthogonal Variable Spreading Factor (OVSF) technique. These codes are picked from a code tree related to one sector and managed at the Radio Network Controller (RNC).

The spreading code allocated to a flow and the bandwidth it will receive are tightly linked: the SF defines how many chips are used to code one user data symbol. As a result, the longer the spreading code, the lower the data rate. As WCDMA supports a maximal chip rate of $3.84\ Mcps$ and the available downlink SFs range from 4 to 512, Table I (excerpt from [2]) gives the available downlink data rates in Rel'99 UMTS.

TABLE I
AVAILABLE DOWNLINK DATA RATES IN REL'99 UMTS

Spreading Factor	Maximum User Data Rate (½ rate coding)
512	3 *kbps*
256	7.5 *kbps*
128	15 *kbps*
64	30 *kbps*
32	60 *kbps*
16	120 *kbps*
8	240 *kbps*
4	480 *kbps*

From our point of view, the SF allocation scheme is based on two main properties: the requirements of the data flow itself (traffic class) and the expectation of the user equipment (UE) receiving the data flow (user profile).

A. Traffic Classes

In [3], the 3rd Generation Partnership Project (3GPP) has defined four traffic classes that need to be supported in UMTS. In order to characterise them, it might be useful to focus on four representative applications (Table II).

TABLE II
TRAFFIC CLASSES AND APPLICATIONS

3GPP Traffic Classes	Representative Applications
Conversational	VoIP
Interactive	Web browsing
Streaming	Video streaming
Background	E-mail

According to which traffic class a given flow belongs to, its QoS requirements in terms of delay, packet loss and bandwidth will vary. An important aspect to take into account during the

SF allocation is that the Conversational and Streaming classes are considered as real-time flows, which means their end-to-end delay has to be upper-bounded by a few hundred ms. On the other hand, the non real-time traffic sets less stringent delay requirements. However, the Interactive class involves human participation and then requires a delay respecting this responsive behaviour ($< 1\ s$). Finally, the packet delay is not that much important for the Background traffic. Nevertheless delays higher than $10\ s$ should be avoided.

In accordance with [4], we will assume the following traffic classes distribution: 27% Conversational sessions, 48% Interactive, 10% Streaming and 15% Background.

B. User Profiles

We have define four different user profiles which seems to be easily mapped on a real cellular network:

- Platinum: Business Contracts. These users will get the best treatment and the larger bandwidth possible according to the network state.
- Gold: Postpaid Contracts. These are expensive contracts which allow the user to easily access the different applications.
- Silver: Postpaid Contracts. These are the classical contracts for an ordinary use of a mobile terminal.
- Bronze: Prepaid Contract. The less expensive contracts designed for a minimal use of a mobile terminal. These users' data will mainly be scheduled as best-effort traffic.

In our emulations, we will assume a 5% Platinum, 10% Gold, 15% Silver and 70% Bronze distribution inferred from postpaid-prepaid ratios recently published by Belgian cellular network operators.

III. SPREADING FACTOR ALLOCATION SCHEME

A. Emulation Assumptions

Our work is based on a computer testbed emulating the UTRAN [5]. As shown on Fig. 1, it consists of 9 PCs respectively standing for the RNC, 4 NodeBs and 4 sets of UEs. We emulate 4 macrocell trisectorial NodeBs, each of them managing a population of UEs over a $27\ km^2$ world surface.

Synthetic traffic whose distribution complies with the characteristics of the 4 traffic classes is generated from (to) the RNC to (from) the PC emulating a given UE in the downlink (uplink). These data flows are mapped to Dedicated CHannels (DCHs) which suffer from DCH allocation delay and queueing delay at the NodeB.

Our UEs are mobile and evolve in an urban area. They are mapped on four specific UE speeds: $3\ km/h$ representing a pedestrian walk, 30 and 70 km/h which represents a UE on board of a vehicle inside a city and 120 km/h which represents a UE embarked on a train or on a car on the speedway. The UE speed is fixed at the beginning of the emulation and it remains constant until the end of the emulation. In the urban area scenario, the UE speed distribution is the following: 30% at 3 km/h, 40% at 30 km/h, 20% at 70 km/h and 10% at 120 km/h.

We finally assume a perfect Radio Link Control (RLC) layer such that there is no retransmission delay and we allocate a quarter of the OVSF tree to signalisation and data transfer over common channels.
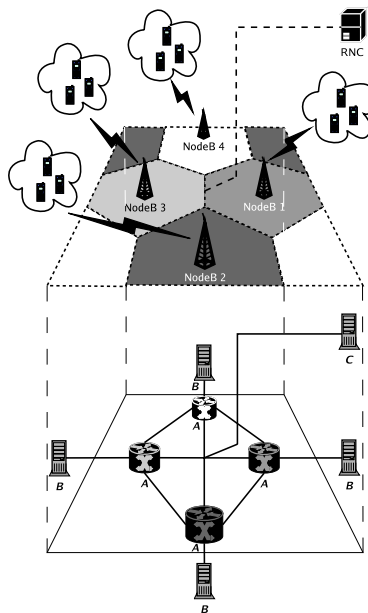
Fig. 1.   UTRAN testbed representation.

With all these hypotheses, we have emulate several scenarios testing different SF allocations for all traffic classes. We were looking for the range of SFs that would enable to meet the mean delay requirements of the 4 traffic classes (Section II-A). Emulation was performed with a single UE, to avoid any code starvation or blocking at this stage of the study. The different kinds of traffic were generated with an inter-session arrival mean of 300 s. The emulation length has been fixed to $5,000\ s$.

B. Results

Figs. 2-3 represent the mean packet delay. This delay is computed by post-processing the log files of the synthetic traffic generator program [6]. It corresponds to a user experienced delay since it is calculated at the application layer. Note that the mean packet delay only refers to delaying in the UTRAN. Core network delays should be added to get the full end-to-end delay.
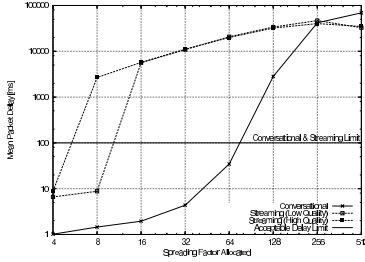
Fig. 2.   Real-time traffic mean packet delay vs. allocated SF.

Fig. 2 depicts the results obtained for the real-time flows which have a stringent delay requirement. The edge delay is upper-bounded by $100\ ms$ so as to leave some time budget margin for delays in the core network.

The Conversational class is based on a $12.2\ kbps$ AMR vocoder. The network data flow hence never exceeds $20\ kbps$. In these circumstances, it would be a waste of network resources to allocate a $SF < 64$, whatever the user profile.

The Streaming flows are divided in two categories: high and low quality. This traffic class is the most bandwidth demanding, so it will require the smallest SF. We can see in Fig. 2 that the high quality undergoes unacceptable packet delay using $SF > 4$. The same conclusion can be drawn for the low quality streaming with $SF > 8$. Therefore we decided to give the possibility to use a $SF = 4$ (high quality streaming) only to the Platinum users. Should the OVSF tree not accept their request, Platinum users would enjoy a lower quality streaming session with $SF = 8$, the only SF available for the other user profiles.

Fig. 3.   Non real-time traffic mean packet delay vs. allocated SF.

The non real-time traffic mean packet delay is shown on Fig. 3. We notice that the Interactive traffic class suffers from a moderate delay regardless of the received SF. This can be explained by the nature of the traffic. Web browsing consists of multiple little downloads (successive pages) separated with page reading times. The bandwidth required for this traffic

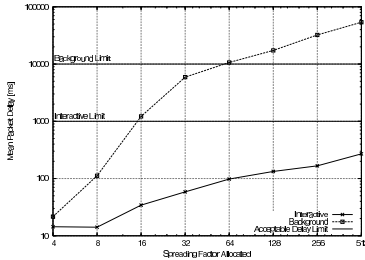class is rather small. The SF allocation logic here is to allocate a 16-chip spreading code to the Platinum users allowing them to downgrade their request to a $SF = 512$ if the OVSF tree has not enough room to accept their previous request. In a similar way, the Gold users begin with a $SF = 32$, the Silver ones with $SF = 64$ and the Bronze users with $SF = 128$. All of them have also the opportunity to downgrade to 512.

Finally the Background traffic represents massive downloads and thus, even if it does not require a small mean packet delay, it has to receive some bandwidth. That is why we choose to allocate a $SF = 8$ to Platinum users which can degrade to 16 or even 32. Gold users have the ability to access a $SF = 16$ or 32 and the only spreading code length available for the other users is 32.

We choose not to allocate smaller SFs to non real-time traffic since the very substantial delay gain that would result would be obtained at the risk of code starvation. Table III summarises our SF allocation map.

TABLE III
SF ALLOCATION MAP

	Platinum	Gold	Silver	Bronze
Conversational	[64]	[64]	[64]	[64]
Interactive	[16 − 512]	[32 − 512]	[64 − 512]	[128 − 512]
Streaming (H)	[4]	n.a.	n.a.	n.a.
Streaming (L)	[8]	[8]	[8]	[8]
Background	[8 − 32]	[16 − 32]	[32]	[32]

IV. BLOCKING RATE AND CELL THROUGHPUT

In the previous Section, the SF allocation scheme has not taken into account the operator's point of view, i.e. the blocking rate and the cell throughput levels. We have then injected this SF allocation map into new emulations in order to evaluate its impact in three different scenarios:

- the operator's worst case (OWC), where all the UEs get the smallest possible SF or are blocked,
- the user's worst case (UWC), where all of them receive their maximal SF, and
- the OWC where downgrading is enabled in order to avoid blocking as much as possible (OWC+).

Note that when a UE receives a SF at the beginning of a session, be it minimal, maximal or degraded, it keeps it until the end of its currently opened session. To optimise the code placement and replacement of successive SF requests and to avoid the fragmentation of the OVSF tree, we have implemented the crowded-first algorithm proposed in [7].

We decided to load each sector with 8 UEs (total of 96 UEs in the emulated 4-cell world) generating the different kinds of traffic which will definitely lead to blocked calls. The emulation length has been fixed at $5,400\ s$.

The different blocking rates are presented on Fig. 4. As we could expect, the number of accepted SF queries in the UWC is greater than in the OWC. Indeed, in the UWC scenario the OVSF tree fills up more slowly since all the UEs requests their
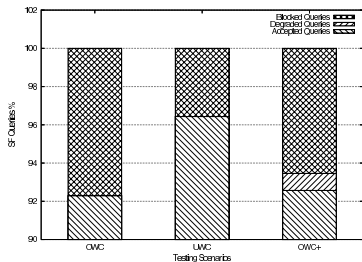
Fig. 4.   Accepted/degraded/blocked SF queries.
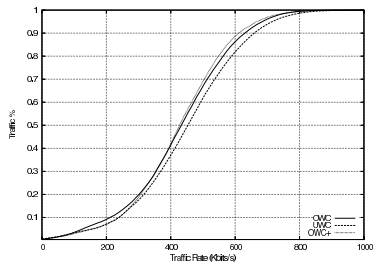
Fig. 5.   Cdf of cell throughputs.

maximal SF. That leads to less resource consumption and less blocked SF queries.

We can also see that the OWC+ gives approximately the same number of accepted queries as in the OWC. But, thanks to the downgrading, the number of blocked queries is lower.

The cell throughputs obtained in the different cases are depicted on Fig. 5. In the UWC, it appears clearly that even if the UEs are not able to use as much as bandwidth as they would need, more UEs are able to transfer data which leads to a better cell throughput. In the OWC, the UEs are able to use all the bandwidth they need but, due to the OVSF tree structure, all of them waste a little bit of resource leading to a smaller cell throughput.

The OWC+ scenario gives approximately the same cell throughput as the OWC due to the small number of degraded queries (about 1%).
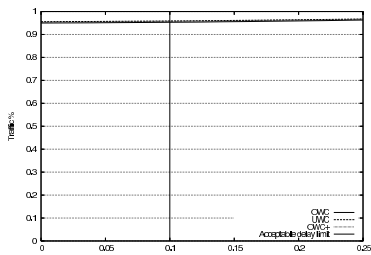
(a) Conversational.
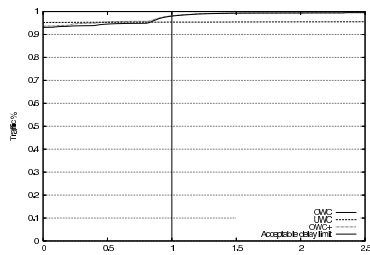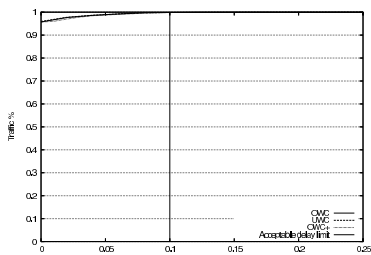
(b) Interactive.
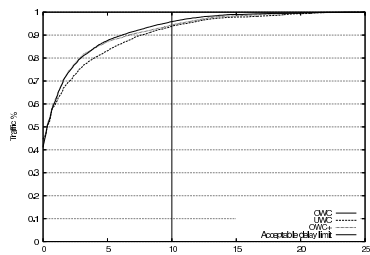
(c) Streaming.

(d) Background.

Fig. 6.   Cdfs of mean packet delay.

TABLE IV
TRAFFIC CLASSES MEAN PACKET DELAY

Traffic Classes	OWC	UWC	OWC+	Limit
Conversational	$23\ ms$	$21\ ms$	$20\ ms$	$< 100\ ms$
Interactive	$97\ ms$	$392\ ms$	$87\ ms$	$< 1,000\ ms$
Streaming	$3\ ms$	$3\ ms$	$3\ ms$	$< 100\ ms$
Background	$2,119\ ms$	$2,375\ ms$	$2,017\ ms$	$< 10,000\ ms$

We can finally compare the packet delay in the different case. Table IV summarises these values for the different traffic classes where Fig. 6 depicts their cdfs, and equivalently the ratio of satisfied users.

We can see that the ratio of satisfied users (whose mean packet delay is lower than traffic class requirement) are identical for Conversational and Streaming traffic classes. This is obvious since a single SF is available regardless of the scenario. Only Interactive and Background traffic could be impacted by downgrading the SF allocated to a session. Indeed, in the UWC this ratio is slightly lower than in the OWC and OWC+ scenarios.

The mean packet delay is also higher in the UWC, although still below its relevant upper bound. This difference is extremely noticeable in the Interactive case where the SF allocated to a session can be as high as 512, where in Background it can not be greater than 32.

It is to be said that the difference between the OWC and the OWC+ scenarios is rather small here due to the small number of degraded sessions. The OWC+ scenario would certainly give better results by allowing downgrading to ongoing sessions.

V. FUTURE WORKS

Disregarding the Dedicated Shared CHannel (DSCH), user's data may use two different Rel'99 UMTS transport channels to transit over wireless interface:

- DCH: it carries a single user data with a fixed bit rate depending on the spreading factor allocated by the RNC, and
- Forward Access CHannel (FACH): it carries small amounts of different user data using time multiplexing. Its bandwidth is rather small since it is not supposed to support large data transfers.

Based on the 3GPP standardisation [1], the transport channel used depends on the traffic class (Table V) according to a decision of the RNC.

In this paper we have directly allocated a DCH to each user session to identify the optimal SF allocation map. The idea now will be to allow the Interactive and Background sessions to use the FACH and to implement the following channel switching policy: a connection uses the common channel (FACH) until we get an indication that the current burst might be long (the queue length exceeds a fixed threshold), then the RNC shall try to allocate a DCH to that connection. In the opposite way, when no more packets are sent using the DCH, a timer is initialised and the connection remains on the DCH during this period. If there are no new arrivals within this timeout period, the connection is switched back to FACH.

TABLE V
TRAFFIC CLASSES AND CHANNELS MAPPING

3GPP Traffic Classes	Authorised Transport Channel(s)
Conversational	DCH
Interactive	FACH or DCH
Streaming	DCH
Background	FACH or DCH

These two parameters are respectively known as upswitch threshold (from DCH to FACH) and downswitch timer (from FACH to DCH) [8] and have to be optimally set up.

VI. CONCLUSION

We have presented the first results obtained in terms of QoS management using our computer testbed. We have established a SF allocation map giving the best possible treatment to a data flow depending on its traffic class and the user profile of the UE receiving the flow. We have then evaluated this allocation scheme with respect to the operators requirements (cell throughput and blocked rate) in different scenarios.

REFERENCES

[1] J. Laiho, A. Wacker, and T. Novosad, *Radio Network Planning and Optimisation for UMTS*, 2nd ed. Wiley inter-science, 2006.
[2] H. Holma and A. Toskala, *WCDMA for UMTS*, 3rd ed. Wiley inter-science, Oct. 2004.
[3] Technical Specification Group Radio Access Network, "TS 23.107 V6.3.0, Quality of Service (QoS) concept and architecture," 3rd Generation Partnership Project (3GPP), Tech. Rep., June 2005.
[4] D. Soldani, "QoS Management in UMTS Terrestrial Radio Access FDD Networks," Ph.D. dissertation, Helsinki University of Technology, Oct. 2005.
[5] H. Van Peteghem and L. Schumacher, "Description of an IPv6 Linux-Based UTRAN Testbed," in *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Barcelona (Spain), Mar. 2006.
[6] "Traffic generator tool," http://www.postel.org/tg/tg.htm, last visited: 16 February 2006.
[7] Y.-C. Tseng, C.-M. Chao, and S.-L. Wu, "Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree," *Management IEEE Transactions on Mobile Computing*, Dec. 2002.
[8] B. J. Prabhu, A. Altman, K. Avratchenkov, and J. Abadia Dominguez, "A simulation study of TCP performance over UMTS Downlink," in *Proceedings of the 58th Vehicular Technology Conference (VTC-fall)*, Orlando (FL - USA), Oct. 2003.

UMTS Non Real-Time Sessions Channel Switching Emulation

Presented at the 65th Vehicular Technology Conference (VTC-spring)
Dublin (Ireland), April 2007

UMTS Non Real-Time Sessions Channel Switching Emulation

Van Peteghem H. and Schumacher L.
Pôle "Réseaux et Sécurité"
FUNDP - The University of Namur
Namur, Belgium
{hvp,lsc}@info.fundp.ac.be

Abstract—We discuss here a scheme to improve the *Rel'*99 UMTS radio resource management. The article focuses on the two different types of transport channel: shared and dedicated. It evaluates the merits of a scheme based on an optimal non real-time session switching between these two kinds of channels. The switching decision may be influenced by several parameters like the downswitch timer's duration, the non/preemptive split of downswitch timer and the selection of the initial transport channel. Performances of the analysed scheme are evaluated taking into account different metrics: blocking rate, mean packet delay and cell throughput. The results show that an optimal tuning of these three parameters may significantly decrease the number of blocked calls while keeping the other metrics at a satisfactory level.

I. INTRODUCTION

Bandwidth consumption has always been an open issue in computer network research since consumers wish to enjoy an ever growing variety of bandwidth demanding services. These last years, the major answer to this problem has often been over-provisioning: to ensure that consumers receive the bandwidth their services require, we deploy significantly more bandwidth than required. This strategy is easily applied in the case of wired networks with the development of gigabits links based on optical fibres.

Rather than provisioning bandwidth, wireless networks have to address this problem in a different way since it may be very costly and in some cases not applicable to increase their bandwidth due to interference, environmental and regulatory constraints. Therefore, the solution for such networks is mainly based on traffic engineering and optimal radio resource allocation strategies within the framework of existing standards.

One of the wireless networks able to support different services is the Universal Mobile Telecommunication System (UMTS). This access network is able to serve a large population of mobile devices, known as User Equipments (UEs), with a single base station. This advantage inevitably leads to a resource allocation issue. Fortunately, traffic can be distributed over two kinds of transport channel, either dedicated or shared, such that an optimal scheduling is able to solve the resource allocation problem. Many parameters may influence this scheduling. The scheme is based on an optimal value setting of a subset of them: the downswitch timer's duration, its split in two periods (preemptive and non-preemptive) and the selection of the initial channel of a session.

The rest of this paper is organised as follows. Section II exposes different *Rel'*99 UMTS characteristics such as the UE's Radio Resource Control (RRC) states, the transport channels and the traffic classes available. Section III explains the channel switching policy and our parameters' value setting to get the optimal radio resource consumption. Section IV then presents the results obtained and discusses some observations. Finally conclusions are drawn in Section V.

II. RRC STATES, TRANSPORT CHANNELS AND TRAFFIC CLASSES

A. RRC States Simplification

The two basic operational modes of a UE are the `Idle Mode` and the `Connected Mode`. The `Connected Mode` can be further divided into different service states known as Cell_DCH, Cell_FACH, Cell_PCH and URA_PCH. Our focus here is on the UE data flows so we only care about the user plane of the UMTS Terrestrial Radio Access Network (UTRAN); the control plane (signalling) is out of the scope of this work. It is then possible to simplify the UE `Connected Mode` RRC states switching initially proposed in [1] into the following figure:
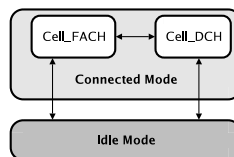
Fig. 1. UE RRC states switching.

A UE in the `Idle Mode` is considered as not active, so we define a UE in this state simply as not connected.

In the Cell_FACH state, no Dedicated CHannel (DCH) is allocated to the UE but a shared channel, the Forward Access CHannel (FACH), can be used instead. This channel carries small amounts of user data using time multiplexing. Its bandwidth is rather small since it is not expected to support large data transfers.

In the Cell_DCH state, at least one DCH is allocated to the UE. This channel transports a single user data flow with a fixed bit rate depending on the Spreading Factor (SF) allocated by the Radio Network Controller (RNC). A shared channel can also be used in this state to carry different user data flows. Since we focus on the *Rel'*99 UMTS, this channel is the Downlink Shared CHannel (DSCH). Its bandwidth is variable since its SF occupies the free room left in the Orthogonal Variable Spreading Factor (OVSF) tree after the DCHs SF allocation.

Table I summarise all these informations.

TABLE I
COMPARISON OF DIFFERENT TRANSPORT CHANNELS

Channel	DCH	DSCH	FACH
SF	Fixed [512 - 4]	Variable [256 - 4]	Fixed [256 - 4]
RRC State	Cell_DCH	Cell_DCH	Cell_FACH
Multiusers	Dedicated	Shared	Shared
Multicodes	No	Yes	No

B. Traffic Classes and Transport Channels

Based on the 3GPP standardisation [2], the four UMTS traffic classes are differently mapped to the three available transport channels (Table II).

TABLE II
TRAFFIC CLASSES AND CHANNELS MAPPING

3GPP Traffic Classes	Authorised Transport Channel(s)
Conversational	DCH
Interactive	FACH, DSCH or DCH
Streaming	DCH
Background	FACH, DSCH or DCH

The Real Time (RT) sessions (Conversational and Streaming) are directly allocated to a DCH. This is natural since this kind of traffic can not tolerate the significant packet delay or the delay variation likely to appear when using shared channels. Since there is no bandwidth reservation mechanism into shared channels, RT sessions are guided to dedicated channels only.

In the opposite way, Non Real Time (NRT) sessions (Interactive and Background) may support a much longer delay. So, these traffics accept to be guided to either dedicated or shared channels since the packet scheduling of the shared channels does not affect the NRT sessions as it does to the RT ones.

As RT sessions use dedicated channels only, channel switching will only influence the NRT sessions.

III. CHANNELS SWITCHING POLICY

A. NRT Traffic

Following the UMTS standard, the first channel allocated to a NRT session is not the only one this session will be able to use until it closes down. Indeed, channel switching is permitted and even encouraged since it can optimise the radio resource consumption.

If a NRT session uses a shared channel (DSCH or FACH), it will continue using this channel until we get an indication that the current burst might be long (the packet queue length exceeds a fixed threshold), then the RNC shall try to allocate a DCH to that session. If the OVSF tree has enough room to accept this new DCH, the NRT session moves from the shared channel to its new DCH.

On the counter part, when no more packets are sent over the DCH, a timer is initialised and the session remains on the DCH during this period. If there are no new arrivals within this timeout period, the session is switched back to a shared channel.
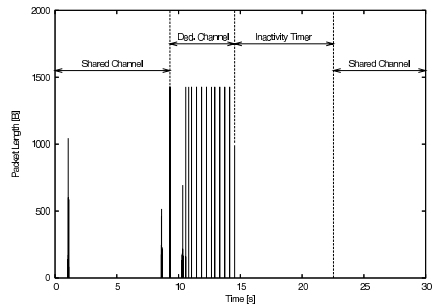
Fig. 2.   Channel switching.

These two parameters are respectively known as upswitch threshold (from shared to dedicated channel, expressed in bytes) and downswitch timer (from dedicated to shared channel, expressed in seconds) [3]. Figure 2 illustrates these mechanisms with upswitch threshold and downswitch timer respectively fixed to $500\ B$ and $8\ s$.

B. Switching from Cell_FACH to Cell_DCH

Being in Cell_FACH means that the UE is active and does not support any RT session. It can only support at least one low bandwidth demanding NRT session which is satisfied with FACH bandwidth. Switching from Cell_FACH to Cell_DCH happens when a DCH has been requested. This may be caused by the NRT session exceeding the upswitch threshold or by a new RT session beginning. In the former case, the NRT session will leave the FACH and continue into its new DCH. In the later case, the RT session will start in its own DCH and the NRT session which was not transferring enough data to claim a DCH, will not stay in the FACH (since the UE is now in CELL_DCH) but will proceed in the shared channel of the Cell_DCH state: the DSCH.

C. Switching from Cell_DCH to Cell_FACH

When a session has to leave its DCH, be it due to a RT session end or a NRT session downswitch, and if no other DCH is allocated but at least one session is still active (in the

DSCH), the UE falls back to the Cell_FACH state. The NRT session assigned to the DSCH leaves it and proceeds in the FACH.

D. Evaluation of the Parameters Optimal Values Set

The UMTS standard does not give any guideline for setting the value of the upswitch threshold or the downswitch timer. Our idea here is to evaluate the best possible values of these parameters leading to an optimal resource consumption.

We will not try to find the optimal upswitch threshold value since it has been identified through extensive simulations [4]. This upswitch threshold has to be smaller than the average packet size. This means that as the packet waiting buffer is filled with only one packet (the NodeB can not emit packets as quickly as they arrive), the session responsible of this excess is directly upswitched. This strategy will lead to open a DCH for large connections as quickly as possible while preventing small connections from opening one, keeping in mind that reserving a SF and opening a DCH is time costly.

So, we have three remaining objectives. First we have to identify the NRT best suitable value for the downswitch timer. If the downswitch timer is high, a NRT session will remain longer on a DCH and, if it has to send more packets after an inactivity period (i.e. a web page reading time), it has not to upswitch again since it has kept its DCH. The connections undergo less upswitches and the DCH allocation time affects the packet delay in a less significant way. But the drawback in this situation is that other connections may have to stay in the shared channel due to expensive OVSF tree occupation. From the opposite point of view, using a small downswitch threshold will lead to rapidly free the OVSF tree by switching back to the shared channel after using a DCH, but in this case, sessions will suffer from multiple upswitches and DCH allocation time will significantly increase the packet mean delay. A tradeoff is therefore to be found between these two options.

A second objective is based on the strategy initially proposed in [3]. This strategy splits the downswitch timer in two distinct periods. The timer begins in the non-preemptive mode. In this mode, the session admission mechanism is exactly the same as with a normal downswitch timer. A NRT session keeps its SF no matter of other new sessions needs. After a period of inactivity into the non-preemptive mode, the downswitch timer moves into the preemptive mode. In this case, the NRT session keeps its SF until the end of its downswitch timer unless another session needs this SF. So, the idea behind identifying the best possible downswitch timer, is to evaluate the benefit of splitting the downswitch timer in two periods.

Besides the downswitch timer dimensioning, the third objective is to determine whether it is beneficial or not to directly open a DCH at the beginning of a NRT session

or to allocate it to a shared channel. When a UE starts a NRT session, does its session get a dedicated channel (DCH) straight ahead or a shared one (DSCH or FACH) first? The UMTS standard gives no particular directive about it and thus, this is another parameter to be identified.

The merits of the proposed scheme/values will be measured with different metrics: the blocking rate, the mean packet delay and the cell throughput.

IV. RESULTS

A. Testing Environment

Our work is based on a computer testbed emulating the UTRAN. It consists of 9 PCs respectively standing for the RNC, 4 NodeBs and 4 sets of UEs. We emulate 4 macrocell trisectorial NodeBs, each of them managing a population of UEs over a $27 \ km^2$ world surface. More details about our emulation platform could be found in [5].

Synthetic traffic whose distribution complies with the characteristics of the 4 UMTS traffic classes [2] is generated from (to) the RNC to (from) the PC emulating a given UE in the downlink (uplink). In order to characterise those traffic classes, it might be useful to focus on four representative applications (Table III) [6].

TABLE III

TRAFFIC CLASSES AND APPLICATIONS

3GPP Traffic Classes	Representative Applications
Conversational	VoIP
Interactive	Web browsing
Streaming	Video streaming
Background	E-mail

Our UEs are mobile and evolve in an urban area. They are mapped on four specific UE speeds. In the urban area scenario, the UE speed distribution is the following: 30% at $3 \ km/h$, 40% at $30 \ km/h$, 20% at $70 \ km/h$ and 10% at $120 \ km/h$.

All the emulation parameters are summarised in Table IV:

TABLE IV

EMULATION PARAMETERS

Number of UEs per sector	7
Emulation length [s]	6,000
Downswitch threshold [s]	8; 15; 30; 45; 60
DCH allocation time [ms]	900
FACH SF	64

B. Observations

We have run a single emulation for every downswitch timer value (split or not) allocating either a dedicated or a shared channel first. We have then been able to identify a set of values leading to an optimal use of the wireless interface
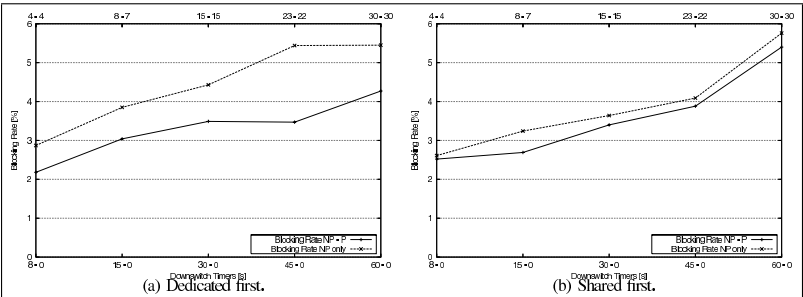
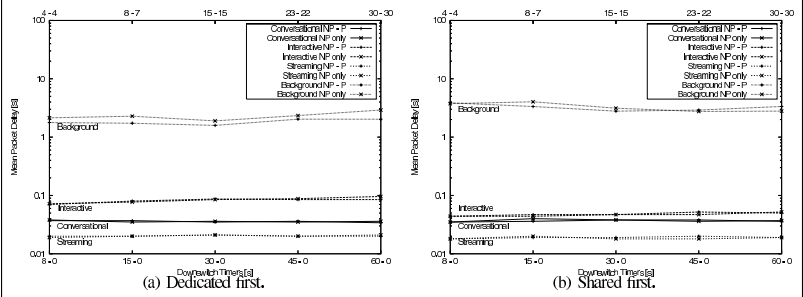Fig. 3.   Blocking rate Vs. Downswitch timers.

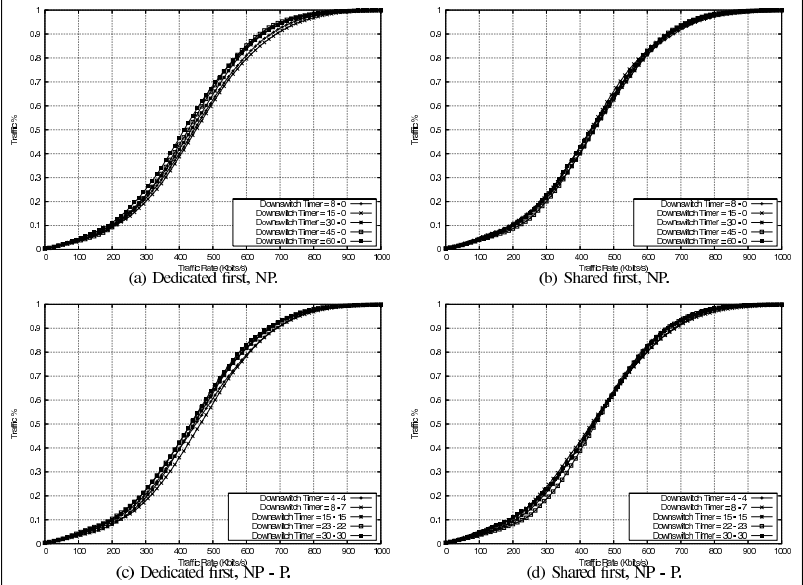Fig. 4.   Packet delay Vs. Downswitch timers.

Fig. 5.   Cell throughput Vs. Downswitch timers.

of the UMTS in terms of blocking rate and cell throughput while keeping acceptable packet delays regarding the traffic class. To illustrate these results, Fig. 3 represents the blocking rate and Fig. 4 shows the mean packet delay experienced by the four traffic classes. These metrics are plotted as a function of the downswitch timer, either non-preemptive only (NP, lower time scale) or split in non-preemptive/preemptive periods (NP-P, upper time scale). Finally, Fig. 5 presents the cell throughput cdfs of all the emulations.

From the downswitch timer point of view, it seems that dividing the downswitch timer in two equivalent periods (a preemptive one and a non-preemptive one) is beneficial. Indeed, as shown on Fig. 3, the blocking call rate is significantly reduced (almost 2% reduction in the best case) whereas mean packet delay (Fig. 4) and cell throughput (Fig. 5) are approximately unchanged.

Following the results depicted on the Fig. 4 and Fig. 5, it appears that opening a DCH at the beginning of a new NRT session, without even knowing whether it will be a large connection or not, is a better choice. Indeed, as the DCH allocation time is included into the initial connection establishment time, the average background delay of the session is then approximately reduced by about 30%. The only negative impact of this scheme resides in the Interactive traffic mean packet delay (Fig. 4(b)) which is smaller when allocating a shared channel first. This could be easily explained by a medium cell load factor (mean between 35% and 40% in the emulations). Since the DSCH bandwidth depends of the DCH SF consumption, if the OVSF tree is lightly occupied, it leaves a large remaining bandwidth to the DSCH which easily emits the Interactive packet bursts without exceeding the upswitch threshold. That is why the Interactive traffic encounters a shorter mean packet delay when allocating a shared channel first.

Focusing on the mean packet delay (Fig. 3(a)) and the cell throughput (Figs. 5(a)-5(c)), it appears that a short downswitch timer (between 8 and 15 seconds) gives the best possible results.

From these observations, the results of our testbed's emulations are inline with those from analytical computations [7]

and from simulations [8]. However we refine their strategy since we focused on different performance metrics. For the NRT sessions, the optimal strategy seems to be:

- to begin the session directly in a DCH (if possible depending on the OVSF tree occupancy),
- to assign a short downswitch timer (as proposed in [8]), and
- to split the downswitch timer in two equivalent periods, the first one non-preemptive and the other preemptive.

V. CONCLUSION

We have presented our results in terms of radio resource management. We have confronted our observations to simulations results from the literature and it appears that these two sets of results are converging. We have then analysed an improved dedicated channel's downswitch timer based scheme which seems to optimally meet the UMTS network requirements such as a low blocking rate, a high cell throughput and a low mean packet delay.

REFERENCES

[1] Technical Specification Group Radio Access Network, "TS 25.331 V7.1.0, Radio Resource Control (RRC) protocol specification," 3rd Generation Partnership Project (3GPP), Tech. Rep., June 2006.

[2] ——, "TS 23.107 V6.4.0, Quality of Service (QoS) concept and architecture," 3rd Generation Partnership Project (3GPP), Tech. Rep., Mar. 2006.

[3] B. J. Prabhu, A. Altman, K. Avratchenkov, and J. Abadia Dominguez, "A simulation study of TCP performance over UMTS Downlink," in *Proceedings of the 58th Vehicular Technology Conference (VTC-fall)*, Orlando (FL - USA), Oct. 2003.

[4] X. Dubois, "Performance of different TCP versions over UMTS common/dedicated channels," Master's thesis, FUNDP - University of Namur (Belgium), June 2005.

[5] H. Van Peteghem and L. Schumacher, "Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS," in *Proceedings of the 64th Vehicular Technology Conference (VTC-fall)*, Montreal (Canada), Sept. 2006.

[6] ——, "Description of an IPv6 Linux-Based UTRAN Testbed," in *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Barcelona (Spain), Mar. 2006.

[7] L. de Bruynseels, "Tuning Inactivity Timer Seetings in UMTS," Commsquare Technical Whitepaper, Tech. Rep., Jan. 2005.

[8] M. Chuah, L. Wei, and X. Zhang, "Impacts of Inactivity Timer Values on UMTS System Capacity," in *Proceedings of the 3rd Wireless Communications and Networking Conference (WCNC)*, Orlando (FL - USA), Mar. 2002.

Performance of Downlink Shared Channels in UMTS Scenarios Using Markov Fluid Process

Presented at the 14th International Conference Analytical and Stochastic Modelling Techniques and Application (ASMTA) Prague (Czech Republic), June 2007

PERFORMANCE OF DOWNLINK SHARED CHANNELS IN UMTS SCENARIOS USING MARKOV FLUID PROCESS

Marie-Ange Remiche
Miguel de Vega Rodrigo
Université Libre de Bruxelles
CoDe - CP 165/15,
Av. F.D. Roosevelt
B-1050 - Bruxelles, Belgium
email: [mremiche,mdevegar]@ulb.ac.be

Laurent Schumacher
Hugues Van Peteghem
FUNDP - The University of Namur
Computer Science Institute,
Rue Grandgagnage 21
B-5000 - Namur, Belgium
email:[lsc,hvp]@info.fundp.ac.be

KEYWORDS

Markov driven fluid flow model, UMTS network.

ABSTRACT

In this work, we propose a new model of the Downlink Shared Channel (DSCH) buffer in UMTS Release '99 cellular systems. Under our assumptions, the available bandwidth dedicated to DSCH traffic is limited by voice connections carried by Dedicated Channels (DCHs) on the one hand, and by traffic on the Forward Access Channel (FACH) on the other hand. We propose to model the DSCH buffer content as a Markov driven fluid queue with jumps and to derive several performance measures, namely stationary DSCH buffer content, bit loss probability and overflow probability. These measures are obtained through adequate transformation of the original fluid queue.

INTRODUCTION

3G/UMTS has been specified as an integrated solution for mobile voice and data with wide area coverage. In its initial phase (Release '99) it offers theoretical bit rates of up to 384 kbps in high mobility situations, rising as high as 2 Mbps in stationary/nomadic user environments. In its latest releases (Releases 5 and 6) the use of High Speed Downlink and Uplink Packet Access technologies (HSDPA and HSUPA, respectively) increase theoretical downlink speeds as high as 14.4 Mbps (and respectively 5.8 Mbps uplink).

In the praxis, a number of factors draw UMTS performance away from such theoretical limits. The users being served by a base station must share common, limited radio resources. Thus, the capacity experienced by each one of them is not only affected by the characteristics of his/her transmission (e.g. user application, connection type, modulation scheme, spreading factor, distance to the base station and scheduling scheme in HSDPA), but also by the number of simultaneously active users and their own transmission characteristics. These cross-dependencies represent a major challenge when trying to describe the behaviour of each shared channel in a certain scenario.

In spite of these difficulties this problem has received much attention by researchers and UMTS developers due to its enormous relevance [4, 5] and references therein. Most of the studies so far, like [1], have focused on the impact of the physical layer on the available bandwidth. However, there is a lack of understanding at system level, where the cross-dependencies among different users play a major role. This stage is extremely important for the performance experienced by the end users since it explains how the bandwidth available at the physical layer is actually distributed among them. Our main focus in this paper is precisely to develop a Markov driven fluid model which includes the effects at system level. To the best knowledge of the authors, such an analytical model has not been proposed for UMTS cellular networks yet.

In this work, we extend some of the most recent results in the field of Markov driven fluid process in order to analyse the performance of a model of UMTS DSCH channel. Numerical computations of the derived analytical model are checked against emulation results obtained with an IPv6, Linux-based testbed mimicking the behaviour of a UMTS Terrestrial Radio Access Network segment [10].

This paper is organized as follows. In the first section, we completely describe the system under analysis and highlight input stochastic parameters of the model. Next, we derive the resulting Markov driven fluid model that exhibits jumps. Performance measures of interest, that is overflow and loss probability and stationary buffer content distribution, are derived in the following section. In the Numerical Analysis Section we validate our model by comparing numerical computations to emulation results on the testbed. We finally highlights the future work items we are currently working on.

UMTS Scenario

Let consider the downlink of a Release'99 Universal Mobile Telecommunication System (UMTS) cell. The base station, called NodeB, supervises the delivery of traffic to the users it is serving. This downlink traffic is split into four classes, depending on the nature of data to be transmitted. Real-Time (RT) traffic falls into Conversational and Streaming classes, whereas Non Real-Time (NRT) traffic is identified as either Interactive or Background

traffic. These four classes of traffic are delivered over the air through three different transport channels. The most exclusive of them, the Dedicated Channel (DCH), only carries traffic related to a given user. The user enjoys thus the full benefit of the resources allocated to that transport channel. On the other hand, the two other transport channels, the Downlink Shared Channel (DSCH) and the Forward Access Channel (FACH), are shared by the users, meaning that their data are multiplexed within these channels. The fact that a given user traffic ends up in a DSCH or a FACH depends whether the user is simultaneously transmitting through a DCH or not. When a user has already access to a DCH, its NRT traffic can be transported through a DSCH. With no open DCH connection, his/her traffic will go through the FACH.

In the present paper, we will study the fluctuations of the shared DSCH bandwidth, considering the competing activity of open DCHs and the FACH of the cell. For the sake of simplicity, we will assume that the data is spread with a Spreading Factor (SF) 64, and is half-rate coded, which means that each transport channel is eager to achieve a bandwidth multiple of 30 kbps. At any given time, we regard the full downlink bandwidth to be split into four types of transport channels:

1. The FACH of the cell (bandwidth fixed to 30 kbps),

2. The $K(t)$ DCHs transporting the $K(t)$ currently active voice connections (each with 30 kbps),

3. The DSCH carrying the signaling of the open DCHs (fixed to 30 kbps),

4. The DSCH transporting the NRT traffic of the users already having a DCH.

We do not constrain the mapping of transport channels to the Orthogonal Variable Spreading Factor (OVSF) tree as in [5]. As a result, the DSCH enjoys the residual bandwidth to its full extend, as shown in Figure 1. Actually, in a real UMTS system, a single DSCH carries both DCH signaling and NRT traffic. We separate these traffics in order to isolate the fluctuating part of the shared DSCH bandwidth. Finally, we will assume perfect Fast Power Control. Consequently, every user transmitting in a DCH will enjoy the full prescribed 30 kbps bandwidth, despite of fading, shadowing, intra- and inter-cell interference. With this model, a NodeB can process up to 62 simultaneous voice connections.

We model the DSCH buffer at layer 2. NRT traffic is composed of bursts of random size generated by the active users, that instantaneously accumulate in the shared DSCH downlink buffer as delivered by the higher layer. As a result, the model of the NRT traffic captures the fluctuating number of active users. The more active users, the more intense the NRT traffic.

Since we assume perfect Fast Power Control, the transmit power of a given mobile user is continuously adjusted in order to achieve a fixed Block Error Rate
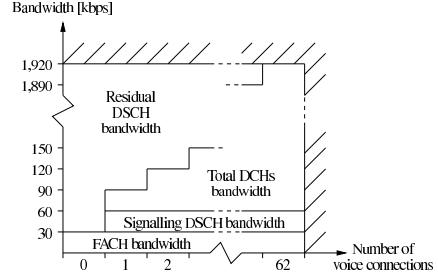
Figure 1: Residual DSCH bandwidth vs. number of voice connections

(BLER) at reception. Assuming also that the NRT traffic is handled by the Radio Link Control (RLC) in Transparent Mode, there is no reliability mechanism in layer 2, i.e. no link retransmissions. Additionally, we regard this specific NRT traffic to be transported on User Datagram Protocol (UDP), such that we do not have to take layer 4 retransmissions into account in the traffic model.

Traffic is then transmitted at a speed that depends on $K(t)$. This transmission speed $R(t)$ is determined as follows:

$$R(t) = \begin{cases} \{64 - [K(t) + 2]\} \cdot 30\,\text{kbps} \\ \qquad \text{when } 1 \le K(t) \le 62 \\ 63 \cdot 30\,\text{kbps} \\ \qquad \text{when } K(t) = 0 \end{cases} \qquad (1)$$

The cellular network industry is actually rather focusing on HSxPA and Long-Term evolutions. In that perspective, this Rel'99 scenario is used to calibrate the comparison of the numerical computations of the analytical model against the emulation results of the testbed. A next step will be to address HSxPA scenarios, where the variation of the bandwidth will no longer depend on the fluctuating number of active voice sessions, but rather on the selected Modulation and Coding Scheme (MCS) based on the Channel Quality Indicator (CQI) fed back by the receiver to the transmitter.

MARKOV DRIVEN FLUID MODEL

This section is composed of three parts. First, we briefly recall some background definitions. Next, we specify the stochastic processes assumed in the system we model. Finally, we put into evidence the resulting Markov driven fluid model, that will be further analysed.

Background definitions

Let $\{L(t), t \in I\!\!R^+\}$ be an absorbing Markov process with generator

$$\begin{pmatrix} \mathbf{0} & 0 \\ G & \boldsymbol{g} \end{pmatrix},$$

and initial phase distribution $(0\ \boldsymbol{\alpha})$, defined on the state-space $\{0, 1, 2, \dots, n\}$, with n finite and 0 being the absorbing state. The time to absorption X is a continuous Phase-type random variable of order n with parameter $(\boldsymbol{\alpha}, G)$; and

$$P[X \leq x] = 1 - \boldsymbol{\alpha} \exp\{Gx\}\mathbf{1},$$

with $x \in \mathbb{R}^+$.

We say that $\{(N(t), J(t)), t \in \mathbb{R}^+\}$ is a stationary Markovian Arrival Process (MAP) with representation (D_0, D_1) and initial stationary distribution $\boldsymbol{\xi}$ if

- $\{(N(t), J(t)), t \in \mathbb{R}^+\}$ is a bidimensional Markov process defined on $\mathbb{N} \times \{1, \dots, m\}$ with m finite,

- whose the corresponding generator is

$$\begin{pmatrix} D_0 & D_1 & 0 & \dots \\ 0 & D_0 & D_1 & \dots \\ 0 & 0 & D_0 & \dots \\ & & & \ddots \end{pmatrix},$$

- and $\boldsymbol{\xi}$ is the minimal non-negative solution of

$$\begin{aligned} \boldsymbol{\xi}(D_0 + D_1) &= \mathbf{0} \\ \boldsymbol{\xi}\mathbf{1} &= 1 \end{aligned}$$

Mathematical model assumptions

We make the following system assumptions. First, voice communications are initiated randomly in time as a Poisson process with parameter λ and have an exponential random duration, with parameter μ. According to the system's description, the number $K(t)$ of active voice communications at time t is determined by a $M/M/62/62$ queue, with generator:

$$A = \begin{pmatrix} -\lambda & \lambda & 0 & \dots & 0 \\ \mu & -(\lambda + \mu) & \lambda & \dots & 0 \\ 0 & 2\mu & -(\lambda + 2\mu) & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & -(62\mu) \end{pmatrix}.$$

Second, we assume that successive burst sizes are independent and identically distributed PH-distributed random variables with parameters $(\boldsymbol{\alpha}, G)$, while random time epochs $\{T_n; n \in \mathbb{N}_0\}$ are given by a MAP with representation (D_0, D_1) and stationary distribution δ. Such an assumption is not restrictive due to the ability of the MAP to mimic self-similar and bursty traffic.

Resulting Model

Our objective is to model the DSCH buffer content. Because a bit can be considered as an infinitesimal quantity with respect to the maximum buffer capacity B, the DSCH buffer content $\{X^B(t)\}$ can be modelled as a fluid process.

Let us assume that the buffer is initially empty. It will remain so as long as the system records no burst arrival. When entering the buffer, a burst is instantaneously available for transmission. Between two consecutive burst arrivals, the buffer content decreases at a speed determined by $R(t)$, as defined in (1). When $K(t) = 62$, the buffer content remains at the same level. When the burst size is larger than the available buffer capacity, only the exceeding burst part is lost, while in real systems, all bits of the packet that is responsible for the overtaking would be lost. However, in our analysis, such an assumption will reveal acceptable in the light of results presented in the Numerical Analysis Section. Upon touching level B, the buffer content immediately decreases or stays at that level in case $K(t) = 62$.

The fluid queue $\{X^B(t)\}$ evolution is completely driven by a particular Markov process $\{\phi(t)\}$ defined on a state-space $\mathcal{S}$. We also define the processes $\{\tilde{X}^B(t)\}$ and $\{\bar{X}^B(t)\}$ as follows. The first is obtained by replacing each jump occurring in $\{X^B(t)\}$ by an interval of time where the process continuously increases at a rate of 1. The latter $\{\bar{X}^B(t)\}$ is obtained through an adequate transformation of the process $\{\tilde{X}^B(t)\}$: level decreases now occur instantaneously and correspond to downwards jumps. Each jump's amplitude is equal to the total level decrease observed over time periods where $\{\tilde{X}^B(t)\}$ was continuously decreasing. According to the definitions above, the evolution of $\{X^B(t)\}$, $\{\tilde{X}^B(t)\}$ and $\{\bar{X}^B(t)\}$ are driven by the same Markov process $\{\phi(t)\}$, the only difference being the behaviour of each fluid queue as a function of the phase visited by $\{\phi(t)\}$.

In the appendix, we completely detail the construction of the generator of the Markov process $\{\phi(t)\}$ and explain how it drives the content of the DSCH buffer. It results that the state-space $\mathcal{S}$ might be decomposed into three disjoint subsets $\mathcal{S}_0 \cup \mathcal{S}_- \cup \mathcal{S}_u$ regrouping phases that correspond to null, negative rates of evolution and finally phases that make the DSCH buffer content instantaneously increases.

PERFORMANCE MEASURES

Our objective is threefold. We aim at characterizing first the stationary buffer content, next the bit loss probability $P_L(B)$ and finally the overflow probability $P_O(B)$.

The bit loss probability, as defined in Lin et al. [7] is the fraction of average lost work divided by the average total workload in the same period of time. The overflow probability may be approximated with the well-known tail probability method as suggested in Sakasegawa et al. [9] for example.

Stationary buffer content distribution

Our approach can be summarized as extending the results of Dzial et al [3] to the finite buffer case analysis. The stationary distribution of the process $\{\tilde{X}^B(t)\}$ is first characterized. With an appropriate transformation, the sta-

tionary distribution of the process $\{X^B(t)\}$ is next completely obtained. We put ourselves in the specific context of the process described earlier but it is a simple matter to extend our analysis to any kind of jump processes.

As mentioned in the appendix, the state-space of $\{\phi(t)\}$ might be decomposed into three disjoint subspaces, those are $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_- \cup \mathcal{S}_+$, following a same decomposition for the generator T, we get:

$$T = \begin{pmatrix} T_{00} & T_{0-} & T_{0+} \\ T_{-0} & T_{--} & T_{-+} \\ T_{+0} & T_{+-} & T_{++} \end{pmatrix}.$$

It is a simple matter to determine the elements of this matrix according to their exact content that is provided in the appendix.

We assume T is irreducible and let $\boldsymbol{\xi}$ be its stationary probability vector, that is $\boldsymbol{\xi} = (\boldsymbol{\xi}_0, \boldsymbol{\xi}_-, \boldsymbol{\xi}_+)$. In our case study, both $\mathcal{S}_+$ and $\mathcal{S}_-$ are not empty, in which case the fluid queue is positive recurrent for any value of the drift $\mu = \boldsymbol{\xi}_+ - \boldsymbol{\xi}_- \boldsymbol{r}_-$, where $\boldsymbol{r}_-$ is a vector composed of different values of $R(t)$ determined for $K(t) = i$ with $(i, j, 0) \in \mathcal{S}_-$.

This finite buffer case was studied in da Silva and Latouche, see [2] and we propose here to recall some of their results which are of interest in the sequel.

Let us first define the matrix $\dot{T}$ as

$$\dot{T} = \begin{pmatrix} \dot{T}_{++} & \dot{T}_{+-} \\ \dot{T}_{-+} & \dot{T}_{--} \end{pmatrix}$$
$$= \begin{pmatrix} C_+^{-1} & 0 \\ 0 & C_-^{-1} \end{pmatrix} \left\{ \begin{pmatrix} T_{++} & T_{+-} \\ T_{-+} & T_{--} \end{pmatrix} + \begin{pmatrix} T_{+0} \\ T_{-0} \end{pmatrix} (-T_{00})^{-1} \begin{pmatrix} T_{0+} & T_{0-} \end{pmatrix} \right\}$$

with $C_+ = I$ and C_- being a diagonal matrix obtained from the vector $\boldsymbol{r}_-$. We define C as a diagonal matrix composed of the different rates of the system. A key matrix is the matrix Ψ, that is the minimal non-negative solution of the following algebraic Riccati equation:

$$\dot{T}_{+-} + \dot{T}_{++}\Psi + \Psi\dot{T}_{--} + \Psi\dot{T}_{-+}\Psi = 0,$$

such that $\Psi\mathbf{1} = \mathbf{1}$. Based on Ψ, we define

$$\begin{aligned} U &= \dot{T}_{--} + \dot{T}_{-+}\Psi, \\ K &= \dot{T}_{++} + \Psi\dot{T}_{-+}. \end{aligned}$$

It has to be noted that efficient matrix analytic algorithms have been developed to obtain Ψ, see Ramaswami [8].

Another key matrix is $\hat{\Psi}$, defined as the minimal non-negative solution of the next algebraic Riccati equation:

$$\dot{T}_{-+} + \dot{T}_{--}\hat{\Psi} + \hat{\Psi}\dot{T}_{++} + \hat{\Psi}\dot{T}_{+-}\hat{\Psi} = 0.$$

We ultimately build up the next matrices:

$$\begin{aligned} \hat{U} &= \dot{T}_{++} + \dot{T}_{+-}\hat{\Psi}, \\ \hat{K} &= \dot{T}_{--} + \hat{\Psi}\dot{T}_{+-}. \end{aligned}$$

We define the vector $\boldsymbol{p}^0 = (p_0^0, p_-^0, \mathbf{0})$ (respectively $\boldsymbol{p}^B = (p_0^B, \mathbf{0}, p_+^B))$) as the probability mass vector of the fluid process $\{X^B(t)\}$ at level 0 (respectively level B). The density probability vector $\boldsymbol{\pi}^B(x)$ is composed of $(\boldsymbol{\pi}_0^B(x), \boldsymbol{\pi}_-^B(x), \boldsymbol{\pi}_+^B(x))$, for $0 < x < B$.

In da Silva and Latouche [2], the following results are established. Let us first define the matrices $\Lambda_{++}^{(B)}, \Psi_{+-}^{(B)}, \hat{\Psi}_{-+}^{(B)}$ and $\hat{\Lambda}_{--}^{(B)}$ as the solutions of the following system:

$$\begin{pmatrix} \Lambda_{++}^{(B)} & \Psi_{+-}^{(B)} \\ \hat{\Psi}_{-+}^{(B)} & \hat{\Lambda}_{--}^{(B)} \end{pmatrix} \begin{pmatrix} I & e^{KB}\Psi \\ e^{\hat{K}B}\hat{\Psi} & I \end{pmatrix}$$
$$= \begin{pmatrix} e^{\hat{U}B} & \Psi \\ \hat{\Psi} & e^{UB} \end{pmatrix}$$

whose coefficient matrix is nonsingular if $\mu \neq 0$.

The next theorem establishes equations for the stationary probability distribution of the fluid process without any jumps.

Theorem 1 *If $\mu \neq 0$, the stationary density of the finite buffer fluid queue is given by*

$$(\boldsymbol{\pi}_+^B(x), \boldsymbol{\pi}_-^B(x)) =$$
$$\gamma(\boldsymbol{\nu}_+^B, \boldsymbol{\nu}_-^0)|\dot{C}|^{-1} \begin{pmatrix} e^{Kx} & e^{Kx}\Psi \\ e^{\hat{K}(B-x)}\hat{\Psi} & e^{\hat{K}(B-x)} \end{pmatrix}$$

where

$$(\boldsymbol{\nu}_+^B, \boldsymbol{\nu}_-^0) =$$
$$(\boldsymbol{\eta}_+^B, \boldsymbol{\eta}_-^0) \begin{pmatrix} 0 & \dot{T}_{+-} \\ \dot{T}_{-+} & 0 \end{pmatrix} \begin{pmatrix} I & e^{KB}\Psi \\ e^{\hat{K}B}\hat{\Psi} & I \end{pmatrix}^{-1},$$

with

$$\dot{C} = \begin{pmatrix} C_+ & 0 \\ 0 & C_- \end{pmatrix}.$$

The parameter γ is $\boldsymbol{\xi}|C|\mathbf{1}$ and the vector $(\boldsymbol{\eta}_+^B, \boldsymbol{\eta}_-^0)$ is

$$(\boldsymbol{\eta}_+^B, \boldsymbol{\eta}_-^0) = (\dot{\boldsymbol{\xi}}_+\Lambda_{++}^{(B)}, \dot{\boldsymbol{\xi}}_-\hat{\Lambda}_{--}^{(B)})$$

where $\dot{\boldsymbol{\xi}} = (\dot{\boldsymbol{\xi}}_+, \dot{\boldsymbol{\xi}}_-)$ respects $\dot{\boldsymbol{\xi}}\dot{T} = \mathbf{0}$ and $\dot{\boldsymbol{\xi}}\mathbf{1} = \mathbf{1}$. Finally,

$$\boldsymbol{\pi}_0^B(x) = (\boldsymbol{\pi}_+^B(x), \boldsymbol{\pi}_-^B(x)) \begin{pmatrix} T_{+0} \\ T_{-0} \end{pmatrix} (-T_{00})^{-1}.$$

Furthermore, the mass probability vector is given by the following theorem.

Theorem 2 *The vector $\boldsymbol{p}^0$ is such that*

$$\begin{aligned} \boldsymbol{p}_-^0 &= \gamma\boldsymbol{\eta}_-^0(C_-)^{-1} \\ \boldsymbol{p}_0^0 &= \boldsymbol{p}_-^0 T_{-0}(-T_{00})^{-1} \end{aligned}$$

and $\boldsymbol{p}^B$

$$\begin{aligned} \boldsymbol{p}_+^B &= \gamma\boldsymbol{\eta}_+^B(C_+)^{-1} \\ \boldsymbol{p}_0^B &= \boldsymbol{p}_+^B T_{+0}(-T_{00})^{-1} \end{aligned}$$

As it is stated in Latouche and Takine [6], the stationary distribution of the original fluid queue can be obtained by an appropriate transformation of the stationary distribution of the fluid queue $\{\tilde{X}^B(t)\}$. In our case, the probability mass in increasing phase has not to be taken into account in the fluid queue with upwards jumps. This is stated in the next theorem.

Theorem 3 *For* $0 < x < B$, *the stationary density vector* $\zeta^B(x)$ *of the fluid queue with upwards jumps* $\{X^B(t)\}$ *is given by*

$$\zeta^B(x) = (\zeta_0^B(x), \zeta_-^B(x), \zeta_+^B(x)) = \delta(\pi_0^B(x), \pi_-^B(x), \mathbf{0}).$$

The stationary probability mass vector ρ^0 *of level 0 of the process* $\{X^B(t)\}$ *is given by*

$$\rho^0 = (\rho_0^0, \rho_-^0, \rho_+^0) = \delta(p_0^0, p_-^0, \mathbf{0}),$$

similarly, we have for the stationary probability mass vector ρ^B *of level B*

$$\rho^B = (\rho_0^B, \rho_-^B, \rho_+^B) = \delta(p_0^B, \mathbf{0}, \mathbf{0}),$$

where δ *is a normalization factor, that is:*

$$\delta = \left(\rho^0 \mathbf{1} + \int_0^B \zeta(x)\mathbf{1}dx + \rho^B \mathbf{1}\right)^{-1}.$$

The bit loss probability

We propose to obtain the bit loss probability by an adequate transformation of the model $\{\tilde{X}^B(t)\}$ into $\{\bar{X}^B(t)\}$. Those two processes were introduced in Resulting Model Subsection.

Let the vector $\bar{p}^0 = (\bar{p}_0^0, \bar{p}_-^0, \mathbf{0})$ (respectively $\bar{p}^B = (\bar{p}_0^B, \mathbf{0}, \bar{p}_+^B)$) be defined as the probability mass vector of the fluid process $\{\bar{X}^B(t)\}$ at level 0 (respectively level B). The density probability vector $\bar{\pi}^B(x)$ is composed of $(\bar{\pi}_0^B(x), \bar{\pi}_-^B(x), \bar{\pi}_+^B(x))$, for $0 < x < B$.

Following again an approach similar to that of Dzial et al. [3], we obtain the next result that concerns the stationary probability of the fluid process $\{\bar{X}^B(t)\}$, where only downward jumps occur.

Theorem 4 *For* $0 < x < B$, *the stationary density vector* $\bar{\pi}^B(x)$ *of the fluid queue with downwards jumps only is given by*

$$\bar{\pi}^B(x) = (\bar{\pi}_0^B(x), \bar{\pi}_-^B(x), \bar{\pi}_+^B(x)) = \delta(\mathbf{0}, \mathbf{0}, \pi_+^B(x)).$$

The stationary probability mass vector $\bar{p}^0$ *of level 0 of the process with jumps is given by*

$$\bar{p}^0 = (\bar{p}_0^0, \bar{p}_-^0, \bar{p}_+^0) = (\mathbf{0}, \mathbf{0}, \mathbf{0}),$$

similarly, we have for the stationary probability mass vector $\bar{p}^B$ *of level B*

$$\bar{p}^B = (\bar{p}_0^B, \bar{p}_-^B, \bar{p}_+^B) = \delta(\mathbf{0}, \mathbf{0}, p_+^B),$$

where δ *is a normalization factor, that is:*

$$\delta = \left(\int_0^B \bar{\pi}(x)\mathbf{1}dx + \bar{p}^B \mathbf{1}\right)^{-1}.$$

In the framework of $\{\bar{X}^B(t)\}$, the loss probability is readily obtained by noting that $\bar{p}_+^B$ bits are in average lost over a total workload of $\int_0^B \bar{X}_+^B(x)\mathbf{1}dx + \bar{p}_+^B$ bits; that is $P_L(B) = \bar{p}_+^B$.

Indeed, we immediately observe that over a unit period, the time spent by $\bar{X}^B(t)$ in levels $(0, B]$ increasing can be also interpreted as the total workload the system has to transmit; while the time spent in level B with an increasing phase is exactly the lost part of this workload.

The overflow probability

This performance measure is obtained by considering the same fluid process with no buffer limit, that is $\{X(t)\}$. This fluid queue is still driven by the evolution of the phase process $\{\phi(t)\}$. As mentioned earlier, we approximate the overflow probability $P_O(B)$ with the tail distribution method, that is

$$\begin{aligned}P_O(B) &= \lim_{t\to\infty} P[X(t) \geq B] = \int_B^\infty \pi(x)\mathbf{1}dx \\ &= 1 - \left(\int_0^B \pi(x)\mathbf{1}dx + p_0\right)\end{aligned}$$

Again, we transform this fluid queue with jumps into a fluid queue such that, each time there is an upwards instantaneous jump, we replace it by an interval of time, of length equal to the size of the jump, and during which the fluid decreases at a rate 1. This exactly what was studied in Dzial et al. [3].

NUMERICAL ANALYSIS

We present a short numerical analysis of our model. Results presented here are of two types. First, we measure the impact of burst arrival rate on the stationary DSCH buffer content. Next, we illustrate the effect of burst arrival burstiness on the stationary DSCH buffer content.

We use the following parameters. Burst arrivals form a Poisson process in time of rate λ, their size is in average 18×30 kbits. Indeed, the Poisson process is a particular case of MAP. Voice communication appears at a rate of one every 100 seconds in average and lasts for 40 seconds in average. Buffer size is about 50×30 kbits.

We present in Figure 2 the distribution function of the stationary buffer content when λ ranges from 1 to 3. The curves represent the outcome of the numerical computations of the analytical model, while the markers stand for the emulation results obtained on the testbed [10]. Each emulation run lasted 600s, with up to 62 simultaneously active users.

Clearly, there is a good match between the analytical model and the results of the testbed. A goodness-of-fit test, the one-sample Kolmogorov-Smirnov test, has successfully been applied to the samples. Moreover, the previously made assumption, that is the limitation of the loss to the exceeding part of the burst when the buffer is saturated, does not significantly affect the validity of the
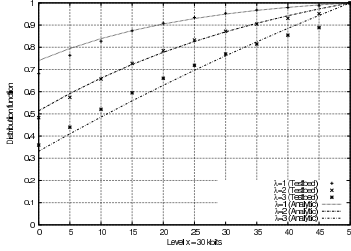
Figure 2: Stationary buffer content distribution function for different burst arrival rates

model. As far as the behaviour of the buffer is concerned, we notice that the greater the lambda, the greater the mass probability on high buffer content levels.

Let us now assume hyper-exponential distributions for the burst inter-arrival times, where coefficient of variation ranges from 4 to 100 and average arrival rate is 3 bursts per second. Figure 3 pictures the distribution function for different coefficients of variation.
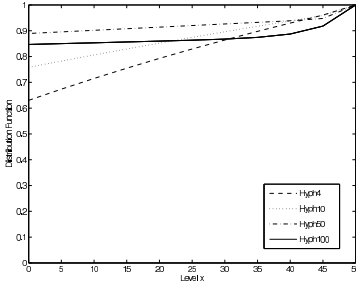

Figure 3: Stationary buffer content distribution function for different burst arrival coefficient of variation

We may clearly identify the effect of the burstiness of the processes: for very bursty arrivals, the buffer is either empty or very nearly full and spends very little time in the intermediate states. Impacts of such arrival patterns and of other more realistic ones (like on-off models) on DSCH buffer content have to be further studied.

CONCLUSION AND FUTURE WORK

In this paper, we have extended results from Dzial et al. [3] in order to model the buffer content of the DSCH of UMTS Rel'99 as a Markov driven fluid model. Performance measures, i.e. buffer content, bit loss and overflow probabilities, have been derived analytically. Assuming Poisson distributed voice and data traffics, preliminary numerical results hint at a higher buffer occupation when the traffic is heavier.

While Poisson-like traffic patterns permitted us to validate our model, these other patterns'analysis will now allow one to rapidly zoom on meaningful and interesting sets of parameter values to feed into our UMTS testbed [10]. This is the subject of future works.

Appendix

For sake of clarity, we explain the evolution of the process $\{\tilde{X}^B(t)\}$, that is driven by the phase process $\{\phi(t)\}$. In our model, the corresponding state-space $\mathcal{S}$ might be decomposed into the following three disjoint subspaces:

$$
\begin{aligned}
\mathcal{S}_0 &= \{(62, j, 0); 1 \leq j \leq m\} \\
\mathcal{S}_- &= \{(i, j, 0); 0 \leq i < 62, \ 1 \leq j \leq m\} \\
\mathcal{S}_+ &= \{(i, j, k); 0 \leq i \leq 62, \ 1 \leq j \leq m, \ 1 \leq k \leq n\}.
\end{aligned}
$$

We have that $\mathcal{S}_+ = \mathcal{S}_u$, since in the framework of $\{\tilde{X}^B(t)\}$, these phases when visited make the fluid linearly increases.

The evolution of $\{\tilde{X}^B(t)\}$ is different according to observed level $\tilde{X}^B(t)$. One needs indeed to distinguish between three different types of evolution, i.e. when the buffer content is empty, partly filled or full. We explain each one in a row.

First, we assume $0 < \tilde{X}(t) < B$. Between two increasing periods, the buffer content may decrease or may remain at the same level. When the observed phase at time t is $(i, j, 0)$ then the decrease rate is $R(t)$ with $K(t) = i$, see Equation (1). When the content is decreasing, we may only observe the following change of phase:

$$
(i_1, j_1, 0) \rightarrow (i_2, j_2, 0) : I_{i_1 i_2}(D_0)_{j_1 j_2} + A_{i_1 i_2} I_{j_1 j_2},
$$

where I_{ij} is the ijth element of the identity matrix I. Indeed only two types of events are possible: a change of MAP phase or a change in the current number of ongoing voice communications. When the number of ongoing voice communication is 62, the buffer content remains at the same level, that is

$$
(62, j_1, 0) \rightarrow (62, j_2, 0) : -62\mu I_{j_1 j_2} + (D_0)_{j_1 j_2}.
$$

The system switches from decreasing to increasing phase when the MAP records an arrival. The first visited phase is thus determined according to α, it gives:

$$
(i_1, j_1, 0) \rightarrow (i_2, j_2, k) : I_{i_1 i_2}(D_1)_{j_1 j_2} \alpha_k,
$$

with $1 \leq i_1, i_2 \leq 62$.

Upon an arrival of a burst, only the phase of the absorbing Markov process (that controls the size of a burst) may change, until reaching phase 0. Other phases evolution are frozen. This corresponds to rates of transition in the process $\{\phi(t)\}$

$$
\begin{aligned}
(i_1, j_1, k_1) \rightarrow (i_2, j_2, k_2) &: \ I_{i_1 i_2} I_{j_1 j_2} G_{k_1 k_2}, \\
(i_1, j_1, k_1) \rightarrow (i_2, j_2, 0) &: \ I_{i_1 i_2} I_{j_1 j_2} g_{k_1},
\end{aligned}
$$

the latter corresponding to the system switching back to its preceding kind of evolution.

Table 1: Fluid process generators according to the fluid level

$$T = \begin{pmatrix} -62\mu I_m + D_0 & e_{62}^t 62\mu \otimes I_m & D_1 \otimes \boldsymbol{\alpha} & O_{m \times 62mn} \\ e_{62}\lambda \otimes I_m & I_{62} \otimes D_0 + A \otimes I_m & O_{62m \times mn} & I_{62} \otimes D_1 \otimes \boldsymbol{\alpha} \\ I_m \otimes \boldsymbol{g} & O_{mn \times 62m} & I_m \otimes G & O_{mn \times 62mn} \\ O_{62mn \times m} & I_{62} \otimes I_m \otimes \boldsymbol{g} & O_{62mn \times mn} & I_{62} \otimes I_m \otimes G \end{pmatrix}, \tag{2}$$

$$T^0 = \begin{pmatrix} -62\mu I_m + D_0 & e_{62}^t 62\mu \otimes I_m & D_1 \otimes \boldsymbol{\alpha} & O_{m \times 62mn} \\ e_{62}\lambda \otimes I_m & I_{62} \otimes D_0 + A \otimes I_m & O_{62m \times mn} & I_{62} \otimes D_1 \otimes \boldsymbol{\alpha} \end{pmatrix}. \tag{3}$$

$$T^B = \begin{pmatrix} -62\mu I_m + D_0 & D_1 \otimes \boldsymbol{\alpha} & O_{m \times 62mn} & e_{62}^t 62\mu I_m \\ I_m \otimes \boldsymbol{g} & I_m \otimes G & O_{mn \times 62mn} & O_{mn \times 62m} \\ O_{62mn \times m} & O_{62mn \times mn} & I_{62} \otimes I_m \otimes G & I_{62} \otimes I_m \otimes \boldsymbol{g} \end{pmatrix}. \tag{4}$$

Only when the maximum number of DCH channels is reached, the DSCH buffer content will remain at the same level. This happens with a rate of probability:

$$(i_1, j_1, 0) \to (62, j_2, 0) : (e_{62})_{i_1}\lambda I_{j_1, j_2},$$

where e_{62} is a column vector of size 62 full of 0's except for a 1 at entry 62. When a voice communication ends up, then the content process rate of evolution decreases back. This happens when:

$$(62, j_1, 0) \to (i_2, j_2, k) : 62(\mu e_{62}^t)_{i_2} I_{j_1, j_2}.$$

If a burst arrives in such a period, then we observe

$$(62, j_1, 0) \to (62, j_2, k) : (D_1)_{j_1, j_2}\alpha_k.$$

Let us consider the previously proposed partition of the $\mathcal{S}$, that is $\mathcal{S}_0 \cup \mathcal{S}_- \cup \mathcal{S}_+$. Let us partition the generator T of the phase process $\{\phi(t)\}$ in a manner conformant to that of $\mathcal{S}$, we thus obtain Generator (2) explicitly given in Table 1. where $O_{m \times n}$ is a matrix full of zeros of size $m \times n$, and I_m is the identity matrix of size m.

In the same manner, one can establish that the evolution of the phase process when the fluid content is at level 0; is the generator T^0 as described in (3). When the fluid touches level B and so only phases belonging to $\mathcal{S}_0$ and $\mathcal{S}_+$ make the fluid staying at that level, this gives Generator (4). It is worth noting that when the fluid stays at level B with a phase belonging to $\mathcal{S}_0$, we still record the next burst arrival. One may object that this does not reflect reality. Fortunately, to obtain the stationary density of $X^B(t)$, a time transform is used so that those added virtual times have no effect at all as we prove in the analytical study. Furthermore, we observe that in this particular case study, matrices T, T^0 and T^B have the same content; for corresponding indexes.

References

[1] R. Bestak, P. Goldlewski, and P. Martins. RLC Buffer Occupancy When Using a TCP Connection Over UMTS. In *Proceedings of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2002*, pages 161–165, Lisbon, Portugal, September 2002.

[2] A. da Silva Soares and G. Latouche. Matrix-analytic methods for fluid queues with finite buffers. *Perform. Eval.*, 63:295–314, 2006.

[3] T. Dzial, L. Breuer, A. da Silva Soares, G. Latouche, and M.-A. Remiche. Fluid queues to solve jump processes. *Perform. Eval.*, 62(1-4):132–146, 2005.

[4] H. Holma and A. Toskala. *WCDMA for UMTS - Radio Access for Third Generation Mobile Communications - Third Edition*. John Wiley & Sons, Ltd., 2004.

[5] J. Perez-Romero, O. Sallent and R. Agusti. *On Dimensioning UTRA-FDD Downlink Shared Channel*. In *Proceedings of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2004*, pages 1777–1781, Barcelona, Spain, September 2004.

[6] G. Latouche and T. Takine. Markov renewal fluid queues. *J. Appl. Probab.*, 41:746–757, 2004.

[7] G. Lin, T. Suda, and F. Ishizaki. Loss Probability for a Finite Buffer Multiplexer with the M/G/∞ Input Process. *Telecommunication Systems*, 29(3):181–197, 2005.

[8] V. Ramaswami. Matrix analytic methods for stochastic fluid flows. In D. Smith and P. Hey, editors, *Teletraffic Engineering in a Competitive World (Proceedings of the 16th International Teletraffic Congress)*, pages 1019–1030, Edinburgh, UK, 1999. Elsevier Science B.V.

[9] H. Sakasegawa, M. Miazawa, and G. Yamazaki. Evaluating the Overflow Probability Using the Infinite Queue. *Management Science*, 39(10):1238–1245, 1993.

[10] H. Van Peteghem and L. Schumacher. Implementation of an Open-Source UTRAN Testbed. In *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (Trident-Com)*, Barcelone, Spain, 2006.

UMTS Layers Parametrisation for Real-Time Flows

Presented at the 1st International Workshop on Mobile Content Quality of Experience (MobConQoE)
Vancouver (Canada), August 2007

UMTS Layers Parametrisation for Real-Time Flows

H. Van Peteghem
FUNDP - The University of Namur
21, Rue Grandgagnage
5000 Namur, Belgium
hvp@info.fundp.ac.be

L. Schumacher
FUNDP - The University of Namur
21, Rue Grandgagnage
5000 Namur, Belgium
lsc@info.fundp.ac.be

C. De Vleeschouwer
UCL - The University of Louvain
2, place du Levant
1348 Louvain-la-Neuve, Belgium
devlees@tele.ucl.ac.be

ABSTRACT
This paper describes the emulation of a UMTS *Rel'*99 radio access network on a Linux-based testbed. The impact of design choices at transport (TCP vs. UDP), data link - RLC (Acknowledged vs. Transparent Mode) and physical (spreading factor 4 vs. 8) layers on the user experience at the reception of Real-Time flows is demonstrated for both synthetic and real life Conversational and Streaming traffics. Over a typically impaired wireless channel, it appears that Transparent Mode is not suited for Streaming flows, as it results in significant PER on UDP or significant delays on TCP, whereas it is acceptable for Conversational flows.

1. INTRODUCTION

The evolution of mobile communications and the Internet has lead to the third generation of cellular networks (3G), known as Universal Mobile Telecommunications System (UMTS). With the launch of this new generation of mobile communication, users are now truly able to access the Internet from mobile terminals, allowing them to enjoy a large variety of added value services.

In this context, the Real-Time (RT) flows are no longer limited to Conversational sessions. Streaming requests are capturing a growing share of the traffic, forcing operators to a clever planning and a more efficient operation of their radio network.

This paper addresses the Quality of Experience (QoE) of Voice over IP (VoIP) and streaming RT flows in UMTS *Rel'*99. Section 2 describes the motivations of this project. Section 3 introduces UMTS traffic classes. The next section presents the testbed designed to emulate the UMTS Radio Access Network (UTRAN). With the help of that testbed, results on the QoE of RT flows under various assumptions are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. MOTIVATIONS

Researchers investigating new architecture designs and protocols in cellular networks can hardly rely on real life networks. Running experiments on a real life network is very expensive, and the experiment is hardly reproducible, due to the varying nature of the aggregated traffic.

They then turn to time- and event-driven computer simulators. Simulators can really help to figure out a number of issues to be dealt with before even thinking of a real deployment. But simulators have their limitations too. As the system to simulate becomes very large and complex, scalability issues can appear. On the other hand, simulators are usually based on simplifying assumptions that do not match real life conditions.

In this perspective, a testbed can be a complementary tool for researchers. Indeed, testbeds are cheaper than a real deployment while more faithfully mimicking the behaviour of the system under study than simulators would do, especially with respect to hardware limitations. Contrary to experiments on real life systems, experiments on testbeds are reproducible, since the whole system is under control. The testbeds can then be seen as a good trade-off between computer simulations and experiments on a real life network.

The first objective of this project has therefore been to design a modular Linux-based testbed mimicking a *Rel'*99 UTRAN. With the help of this emulator, it has been possible to stream multimedia content so as to assess the QoE of a mobile user connecting to the Internet through such a wireless access network.

3. TRAFFIC CLASSES

In [1], the 3rd Generation Partnership Project (3GPP) has defined four classes of services that need to be supported in UMTS. In order to characterise them, it might be useful to focus ourselves on four representative applications (Table 1).

Table 1: Traffic classes and applications.

Traffic Classes	Representative Applications
Conversational	VoIP
Interactive	Web browsing
Streaming	Video streaming
Background	E-mail

As we are only focused on RT flows, we will only consider the Conversational and the Streaming flows in the rest of this paper.

4.   EMULATION OF A UTRAN SEGMENT

The objective of the testbed is to emulate the UTRAN over an Ethernet local network. Details on the implementation are given in [2]. The following paragraphs highlight some design choices, namely the handling of UMTS transport channels, the emulation of time-correlated errors at Layer 3, the emulation of two of the three RLC modes and finally, the derivation of the BER of a mobile user.

4.1   Transport Channels

Each of the data flows presented in the Section 3 is mapped to one of the three available UMTS transport channels following the guidelines given in [3]:

- The Dedicated CHannel (DCH): this is the only dedicated transport channel in the UMTS standard, it carries the service data of a single user at variable bit rate,

- The Dedicated Shared CHannel (DSCH): this transport channel carries signalling and service data of several users at variable bit rate, or

- The Forward Access CHannel (FACH): this last transport channel carries also the service data of several users but at fixed bit rate.

The RT sessions are directly allocated to a DCH. This is natural since this kind of traffic can not tolerate the significant packet delay or the delay variation likely to appear when using shared channels. Since there is no bandwidth reservation mechanism into shared channels, RT sessions are guided to dedicated channels only.

In the opposite way, Non Real Time (NRT) sessions (Interactive and Background) may support a much longer delay. So, these traffics accept to be guided to either dedicated or shared channels since the packet scheduling of the shared channels does not affect the NRT sessions as it does to the RT ones.

4.2   Time-Correlated Errors

The emulation of the Radio Link Control (RLC) and Medium Access Control (MAC) layers is achieved using the Linux Traffic Controller (TC) which copes with packet shaping, scheduling, policing and dropping. More precisely, we use the TC NETwork EMulation component (NETEM) [4] which is a waiting queue able to delay, drop, corrupt, duplicate or even reorder packets.

The NETEM packet corruption functionality is the most appropriate to approach what packets really undergo over the UMTS radio link. Indeed, it introduces a single bit error at a random position in the packet. Instead of being dropped, this corrupted packet is transmitted to the UE, hence consuming a part of the bandwidth allocated to the flow it belongs to. Eventually, this corrupted packet is dropped, because of the CRC error detected by Ethernet.

It has been proven that the typical bit errors over a radio link are not totally decorrelated and tend to appear in bursts. One of the analytical models trying to describe this property is the Gilbert-Elliot model [5, 6]. It uses a two-state (error and error-free) time-homogeneous discrete time Markov chain to describe the channel variations. This model is a popular one for wireless bit errors since it is complex enough to capture burstiness and simple enough to be treated analytically.

Moreover, recent results have shown that an enhanced version of the Gilbert-Elliot model approaches even more the UMTS radio channel [7]. The error state in this new model is the same than the one in the simple two-state Markov model. However, the error-free state of the new model has been modified. The duration of the stay at that state is now taken from a Weibull distributed random variable which correctly fits the measurements over real networks.

We have then modified the NETEM module to respect to these specifications. We will consider the two states of the Markovian chain at the Transmit Time Interval (TTI) level: if we are in the error-free (resp. error) state, all the transmit RLC Packet Data Units (PDUs) transiting over the air interface during the TTI are considered as well received (resp. corrupted). We may then define a *burst* as a number of subsequent erroneous TTIs (Fig. 1) whereas a *gap* is a number of subsequent error-free TTIs (Fig. 2).
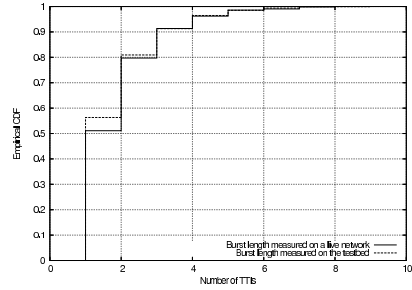

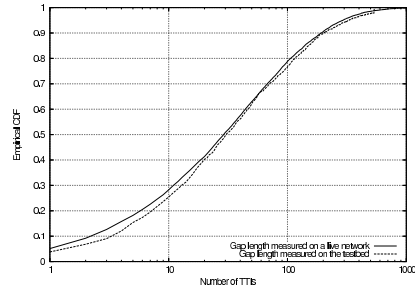Figure 1: Bursts length comparison.

Figure 2: Gaps length comparison.

4.3   RLC modes

The RLC layer is able to transmit RLC PDUs in three different modes: Acknowledged Mode (AM), Unacknowledged Mode (UM) and Transparent Mode (TM). Note that the Automatic Repeat Request (ARQ) mechanism is only available in the AM.

We decided to focus on two of them, namely the TM likely to transmit the delay-sensitive RT traffic. On the other hand, we use the AM for NRT traffic to lower the packet error rate. The ARQ mechanism is able to retransmit an erroneous PDU a fixed number of times before declaring this PDU as lost. The number of retransmissions is known as $maxDAT$ and is fixed at 3 in the emulation following the proposition made in [8, 9].

Those two modes have been adapted to an Ethernet link and included in the enhanced version of the NETEM module as described on Fig. 3-4. Focusing on the Fig. 3, when a packet P arrives, NETEM computes the numbers of TTIs needed to transmit P over the air interface. This number is referred to as m. This division is mainly based on P's length and on the SF allocated to the flow P belongs to. All these m TTIs are then injected into the two-states Markovian chain modified with the Weibull distribution presented in Section 4.2 to calculate the number of error-free TTIs referred to as TTI_score. If P does not have any erroneous TTI (meaning m is equal to TTI_score), P is correctly transmitted. However, if P has at least one erroneous TTI, P is corrupted before being transmitted to its final destination.
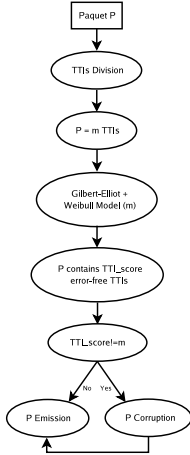

Figure 3: RLC TM using NETEM.

From the AM point of view (Fig. 4), the computation is just like in the TM case but if P contains erroneous TTIs, we reinject these $(m - TTI\_score)$ TTIs into the Markovian chain until all the TTIs needed to transmit P are error-free or until the number of attempts (referred to as Nb_try) exceeds $maxDAT$. If all TTIs are error-free, P is transmitted with a $Nb\_try \times 2 \times TTI\_length$ delay including the ARQ delay due to the retransmissions. On the other hand, if there is still at least one erroneous TTI at the end of the computation, P is corrupted, then transmitted with a $(maxDAT+1) \times 2 \times TTI\_length$ delay. Note that in a $Rel'99$ UMTS network, the TTI_length is fixed to $10ms$.
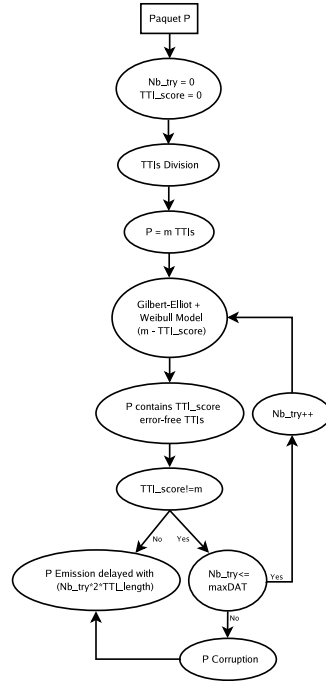

Figure 4: RLC AM using NETEM.

4.4 Received Power

As described in [2], the emulated mobile users are moving around in a wrap-around world. Periodically, the received power of each User Equipment (UE) is computed, so as to derive the quality of the on-going transmission. The quality of a transmission between a NodeB and a UE can be evaluated by the received power the UE is getting. This power is a direct function of the NodeB emitted power and is also influenced by a number of factors. At the receiver side, the carrier to interference ratio is calculated as followed [10]:

$$\left(\frac{C}{I}\right)_{CH} = \frac{Sh \cdot Pl \cdot (P_t)_{CH}}{\alpha I_{intra} + I_{inter} + P_N} \qquad (1)$$

where Sh is shadow fading affecting a communication at a precise location in the emulated world. Pl is the path loss the communication undergoes at the same position. $(P_t)_{CH}$ is the transmitted power the NodeB allocates to the transport channel. I_{intra} is the intracell interference which is generated by those users that are connected to the same base station that the observed user. I_{inter} is the intercell

interference which is generated from the other cells. α is the orthogonality factor which takes into account the fact that all the downlink communications are not perfectly orthogonal due to multipath propagation. An orthogonality factor of 0 corresponds to perfectly orthogonal intracell users while with the value of 1 the intracell interference has the same effect as intercell interference. Assumed value for the orthogonality factor is 0.4 in the macrocellular case. P_N is thermal noise power which is assumed to be equal to -99 dBm.

At the receiver side, from the $(C/I)_{CH}$ evaluated as in (1), the normalised energy per information bit is obtained as followed [11]:

$$\left(\frac{E_b}{N_0}\right)_{CH} = \frac{1}{2R_{CH}} \cdot SF_{CH} \cdot \left(\frac{C}{I}\right)_{CH} \qquad (2)$$

where R_{CH} is the coding rate of the channel and SF_{CH} its Spreading Factor (SF).

On the DCH, the target E_b/N_0 setpoint is permanently adjusted and aims at a constant quality, usually defined as a certain target BER, depending on the traffic class. Following specifications presented in [1], we have fixed the target BER for the RT traffic at 10^{-4}. Assuming independent bit errors despite correlated errors at TTI level for simplification purposes, this BER enables to approximate the BLock Error Rate (BLER) as follows:

$$BLER = 1 - (1 - BER)^l \qquad (3)$$

where l is the PDU length which has been fixed to 336 bits [12]. This BLER is then injected into our modified version of NETEM with two other informations: the allocated SF and the RLC mode used.

Conversely, the associated DSCH benefits from the same transmit power. However, as a different spreading factor can be allocated to the DSCH, the resulting BLER can vary. Finally, the FACH is transmitted at constant power, such that its BLER degrades when the mobile user moves away from the NodeB.

5.   USER EXPERIENCE

With the help of the emulator described in the previous sections, we have investigated the QoE of Real-Time services, in UMTS Streaming and Conversational classes. In both cases, we have first run some emulations with synthetic traffic, to measure the impact of the testbed settings on the service under study. Performance have been evaluated with respect to the following Quality of Service (QoS) metrics: mean packet delay, jitter and packet loss rate. We have then set up a real life experiment, with actual content being delivered through the testbed so as to figure out the QoE at the end-user when s/he connects through a UTRAN. It should be mentioned here that these quality experiments did not involve a significant number of end-users, and hence do not claim to be consistent with ETSI recommendation on usability evaluation [13].

In the case of synthetic traffic, its characteristics are described in [2] in terms of inter-arrival times and packets lengths with the help of statistical distributions. To gen-

erate the packets we have installed the open-source application TG [14] which is a traffic generator program able to create one-way UDP or TCP streams between two computers. We have then generated Conversational and Streaming flows adopting parameter settings based on the options available in the emulated layers and on earlier results [15]:

- Transport layer: TCP vs. UDP
- Data link layer (RLC): error-free Ethernet, Acknowledged Mode with errors and Transparent Mode with errors
- Physical layer: SF = 4 vs. SF = 8

5.1   Streaming

5.1.1   Synthetic traffic

The results obtained for the synthetic streaming flows are presented in Table 2. All possible settings have been taken into account. It seems that streaming in TCP connections over a RLC layer in AM is the best option, despite the longer end-to-ed delay experienced by the TCP segments.

5.1.2   Real traffic

We have installed a Real Network Helix streaming server, with a client connecting to it through a RealPlayer 10 application. We have tested the QoE of all possible settings, and reached the same conclusion: TCP over AM is the best choice, thanks to the ability to buffer content before playing it back at the receiver. The reader can self assess the user experience of streaming a video file over a DCH by connecting to http://www.info.fundp.ac.be/~hvp/rech/doc/streaming/.

The quality of the streamed flow may be degraded due to a high packet loss rate. This can be the results of two different principles. First, the bandwidth available between the client and the server can't meet the requirement of the fully detailed flow. The server has to lower the video and/or audio quality to decrease the bandwidth consumption. On the other hand, streaming a video over the UMTS air interface involves a higher BER which inevitably involves a higher packet loss rate.

5.2   Conversational

5.2.1   Synthetic traffic

For the synthetic conversational traffic, the setting choices are reduced. Due to the delay sensitivity, retransmissions at transport layer are ruled out. Hence, UDP is to be used. Moreover, we limited ourselves to SF = 8, since the typical VoIP bandwidth requirement do not exceed 70 kbps, which is far below the 240 kbps ideally achieved with SF = 8. The results (Table 3) show that TM suffers from significant packet losses, but the communication remains acceptable. In this case, we would recommend to use UDP over a RLC layer in TM

5.2.2   Real traffic

We have tested VoIP over UTRAN using two BudgeTone-100 IP phones configured to use the G.720 vocoder with silent suppression. It appeared that, even with a PER of 10%, the communication was excellent.

Table 2: Synthetic streaming flow results.

Transport Layer	Data Link Layer	Physical Layer	Delay[s]	Jitter[s]	Packet Lost[%]
	Ethernet		0.003	0.011	0.00
UDP	WCDMA-AM	SF=4	0.005	0.013	0.35
	WCDMA-TM		0.003	0.011	17.73
	Ethernet		0.003	0.011	0.00
TCP	WCDMA-AM	SF=4	0.011	0.019	0.00
	WCDMA-TM		0.682	1.120	0.00
	Ethernet		0.001	0.005	0.00
UDP	WCDMA-AM	SF=8	0.003	0.011	0.31
	WCDMA-TM		0.001	0.005	16.07
	Ethernet		0.001	0.006	0.00
TCP	WCDMA-AM	SF=8	0.009	0.020	0.00
	WCDMA-TM		1.114	0.700	0.00

Table 3: Synthetic conversational flow results.

Transport Layer	Data Link Layer	Physical Layer	Delay[s]	Jitter[s]	Packet Lost[%]
	Ethernet		< 0.001	< 0.001	0.00
UDP	WCDMA-AM	SF=8	0.001	0.005	0.39
	WCDMA-TM		< 0.001	< 0.001	12.81

6. CONCLUSION

This paper has presented a testbed emulating a UTRAN segment over a wired infrastructure, including the various UMTS traffic classes and transport channels, and the time-correlated impairments of a wireless radio channel. The testbed has been shown to enable the evaluation of the QoE of services delivered through a cellular network. It helped in identifying the best parameter setting of the emulated UMTS layers.

Acknowledgements

The authors would like to thank Wolfgang Karner from the Institute of Communications and Radio-Frequency Engineering of Vienna for his valuable help.

7. REFERENCES

[1] Technical Specification Group Radio Access Network, "TS 23.107 V7.0.0, Quality of Service (QoS) concept and architecture," 3rd Generation Partnership Project (3GPP), Tech. Rep., Mar. 2006.

[2] H. Van Peteghem and L. Schumacher, "Description of an IPv6 Linux-Based UTRAN Testbed." in *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, Barcelona (Spain), Mar. 2006.

[3] H. Holma and A. Toskala, *WCDMA for UMTS*, 3rd ed. Wiley inter-science, Oct. 2004.

[4] S. Hemminger, "Network Emulation with NetEm," in *Proceedings of the 6th Australia's National Linux Conference (LCA)*, Canberra (Australia), Apr. 2005.

[5] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253–1265, Sept. 1960.

[6] E. O. Elliot, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell Systems Technical Journal*, vol. 42, pp. 1977–1997, Sept. 1963.

[7] W. Karner, "A UMTS DL DCH BLER Model Based on Measurements in Live Networks," COST 273, Tech. Rep., Jan. 2005.

[8] O. M. Teyeb, "Quality of Packet Services in UMTS and Heterogeneous Networks," Ph.D. dissertation, Aalborg University, Oct. 2006.

[9] O. De Mey, L. Schumacher, and X. Dubois, "Optimum Number of RLC Retransmissions for Best TCP Performance in UTRAN," in *Proceedings of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Berlin (Germany), Sept. 2005.

[10] Technical Specification Group Radio Access Network, "TR 25.942 V7.0.0, Radio Frequency (RF) system scenarios (Release 6)," 3rd Generation Partnership Project (3GPP), Tech. Rep., Mar. 2007.

[11] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta, "Packet service in UMTS: delay-throughput performance of the downlink shared channel," *Computer Networks*, vol. 38, no. 1, pp. 43–59, Jan. 2002.

[12] "Special Article on IMT-2000 Services," *NTT DoCoMo Technical Journal*, vol. 3, no. 2, Sept. 2001.

[13] ETSI, "Human Factors (HF):Usability Evaluation for the Design of Telecommunication Systems, Services and Terminals," ETSI EG201.472 v1.1.1, Tech. Rep., Feb. 2002.

[14] "Traffic generator tool," http://www.postel.org/tg/tg.htm, last visited: June 29, 2007.

[15] H. Van Peteghem and L. Schumacher, "Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS," in *Proceedings of the 64th Vehicular Technology Conference (VTC-fall)*, Montreal (Canada), Sept. 2006.

Appendix B

Scientific Award
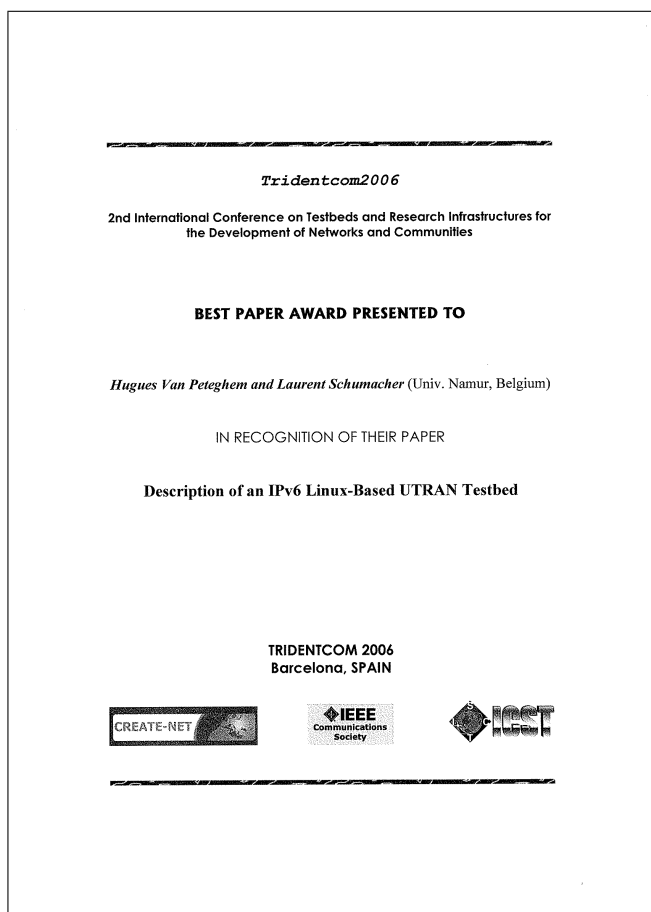
Appendix C

Stochastic Distributions

C.1   Poisson Distribution

The Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time since the last event. The Poisson distribution can also be used for the number of events in other specified intervals such as distance, area or volume.

The work focused on certain random variables N that count, among other things, a number of discrete occurrences (sometimes called "arrivals") that take place during a time-interval of given length. If the expected number of occurrences in this interval is λ, then the probability that there are exactly k occurrences (k being a non-negative integer, $k = 0, 1, 2, ...$) is equal to:

$$f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{C.1}$$

where k is the number of occurrences of an event - the probability of which is given by the function and λ is a positive real number, equal to the expected number of occurrences that occur during the given interval.
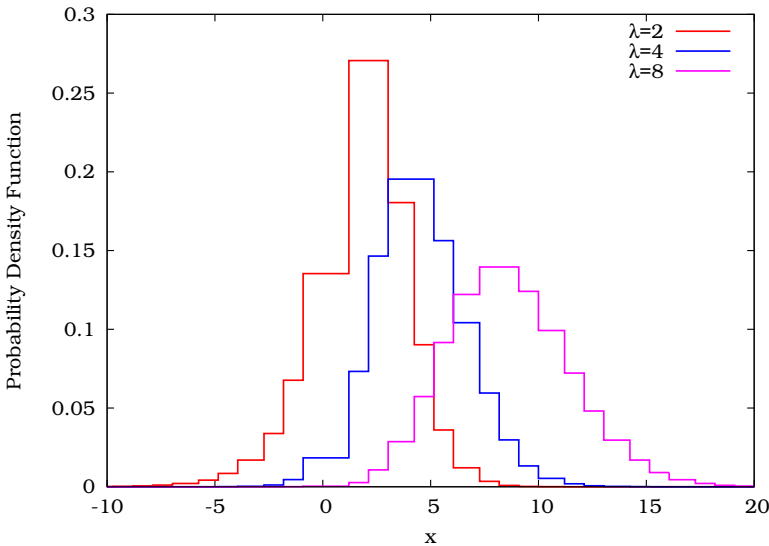

Figure C.1: Poisson distribution.

C.2 Exponential Distribution

The exponential distributions are a class of continuous probability distribution. An exponential distribution arises naturally when modelling the time between independent events that happen at a constant average rate.

The probability density function of an exponential distribution has the form:

$$f(x, \lambda) = \begin{cases} \lambda e^{-\lambda x} & , \ x \geq 0, \\ 0 & , \ x < 0. \end{cases} \quad (C.2)$$

where $\lambda > 0$ is a parameter of the distribution, often called the rate parameter. The distribution is supported on the interval $[0, +\infty)$.
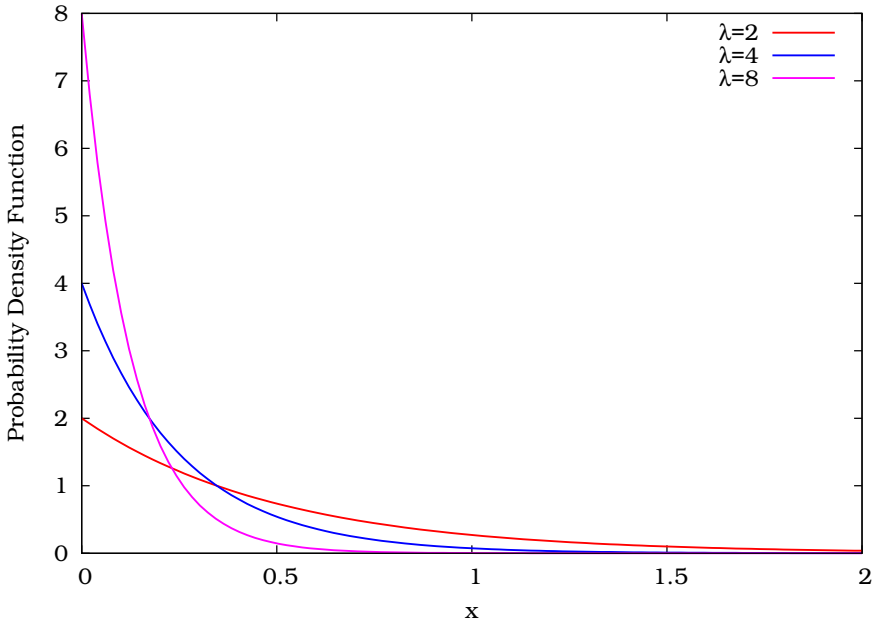

Figure C.2: Exponential distribution.

C.3 Pareto Distribution

The Pareto distribution is a power law probability distribution that coincides with social, scientific, geophysical, actuarial, and many other types of observable phenomena.

The following examples are sometimes seen as approximately Pareto-distributed:

- File size distribution of Internet traffic which uses the TCP protocol (many smaller files, few larger ones), or

- The length distribution in jobs assigned supercomputers (a few large ones, many small ones).

The probability density function of the Pareto distribution has the form:

$$f(x, a, b) = \begin{cases} \frac{ba^b}{x^{b+1}} & , \ x \geq a, \\ 0 & , \ x < a. \end{cases} \tag{C.3}$$

where a is the (necessarily positive) minimum possible value of x, and b is a positive parameter.
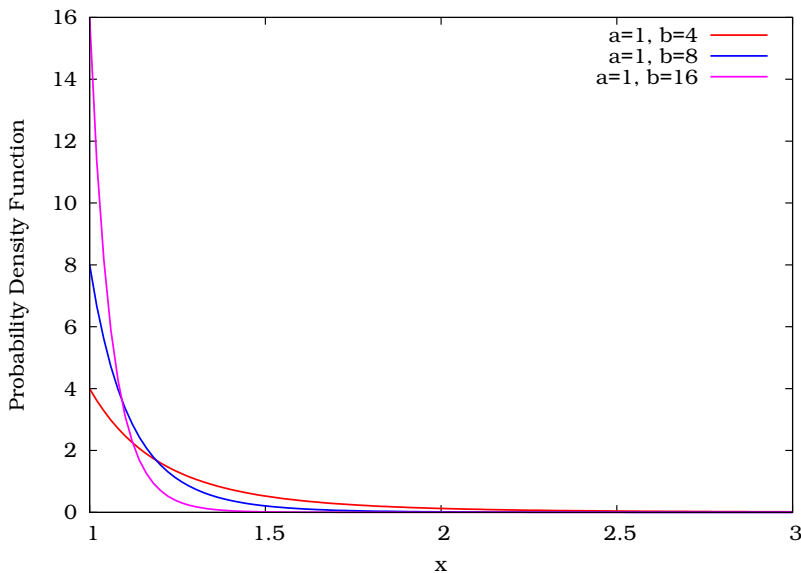

Figure C.3: Pareto distribution.

C.4 Gamma Distribution

The gamma distribution is a two-parameter family of continuous probability distributions. It has a scale parameter λ and a shape parameter ρ. If ρ is an integer then the distribution represents the sum of ρ exponentially distributed random variables, each of which has parameter $\frac{1}{\lambda}$.

The probability density function of the gamma distribution can be expressed in terms of the gamma function:

$$f(x, \rho, \lambda) = \begin{cases} x^{\rho-1} \frac{e^{-x/\lambda}}{\lambda^\rho \, \Gamma(\rho)} & , \ x > 0, \\ \rho & , \ \lambda > 0. \end{cases} \tag{C.4}$$
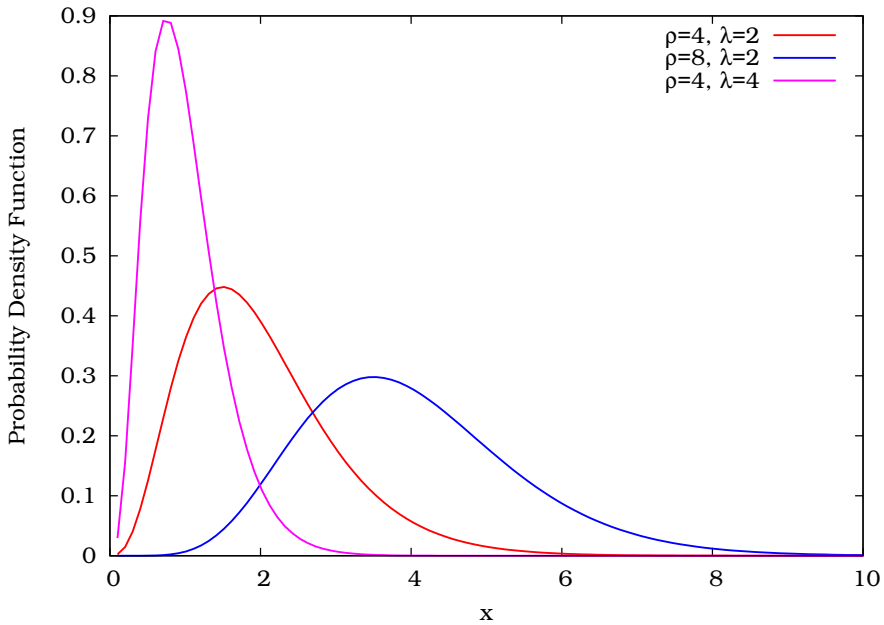

Figure C.4: Gamma distribution.

C.5 Log Normal Distribution

The log normal distribution is the single-tailed probability distribution of any random variable whose logarithm is normally distributed. If Y is a random variable with a normal distribution, then $X = exp(Y)$ has a log normal distribution; likewise, if X is log normally distributed, then $log(X)$ is normally distributed.

A variable might be modelled as log normal if it can be thought of as the multiplicative product of many small independent factors. For example the long-term return rate on a stock investment can be considered to be the product of the daily return rates.

The log-normal distribution has the probability density function:

$$f(x, \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \ x > 0. \qquad \text{(C.5)}$$

where μ and σ are the mean and standard deviation of the variable's logarithm (by definition, the variable's logarithm is normally distributed).
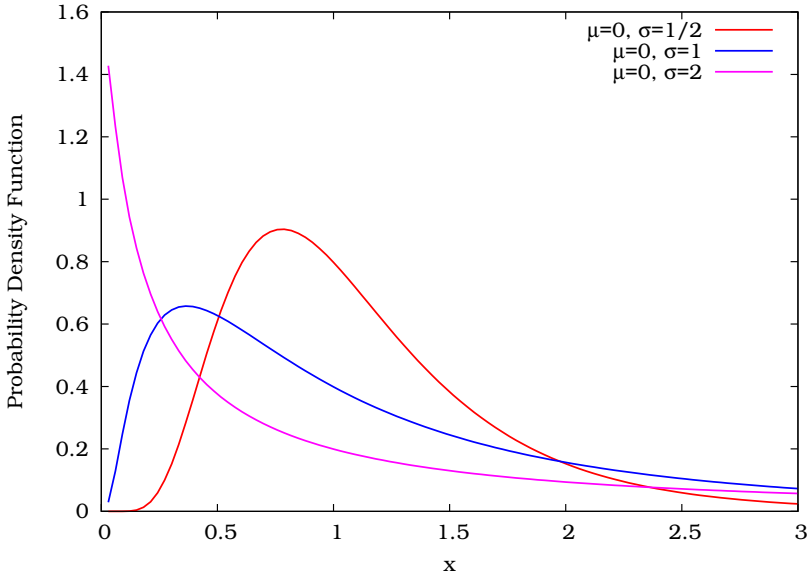

Figure C.5: Log normal distribution.

C.6 Geometric Distribution

The geometric distribution is either of two discrete probability distributions: the probability distribution of the number X of Bernoulli trials needed to get one success, supported on the set $\{1, 2, 3, ...\}$, or the probability distribution of the number $Y = X - 1$ of failures before the first success, supported on the set $\{0, 1, 2, 3, ...\}$.

If the probability of success on each trial is p, then the probability that k trials are needed to get one success is

$$\Pr(X = k) = (1 - p)^{k-1} p, \ \ k = 1, 2, 3, ... \tag{C.6}$$

Equivalently, if the probability of success on each trial is p, then the probability that there are k failures before the first success is

$$\Pr(Y = k) = (1 - p)^{k} p, \ \ k = 0, 1, 2, 3, ... \tag{C.7}$$

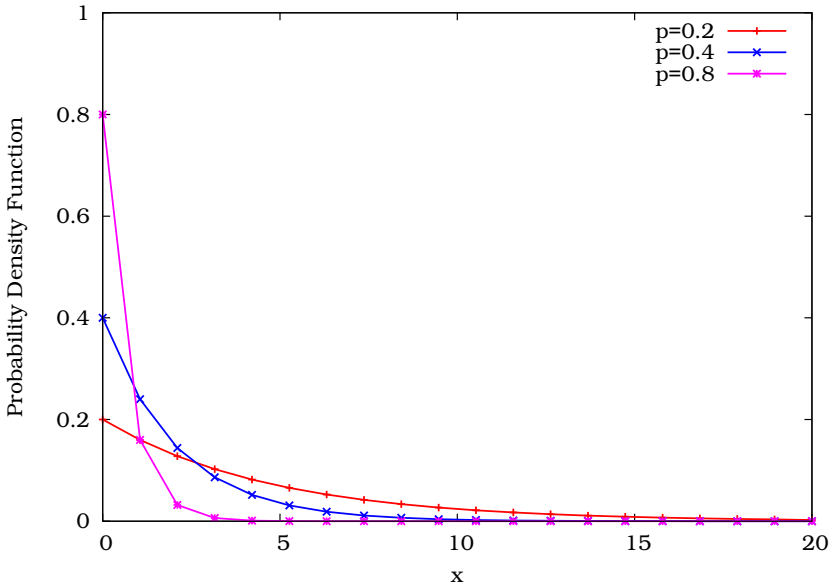In either case, the sequence of probabilities is a geometric sequence.

Figure C.6: Geometric distribution.

C.7 Weibull Distribution

The Weibull distribution is a continuous probability distribution with the probability density function:

$$f(x, k, \lambda) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & , \ x \geq 0, \\ 0 & , \ x < 0. \end{cases} \tag{C.8}$$

where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution.

The Weibull distribution is often used in the field of life data analysis due to its flexibility, it can mimic the behaviour of other statistical distributions such as the normal and the exponential. If the failure rate decreases over time, then $k < 1$. If the failure rate is constant over time, then $k = 1$. If the failure rate increases over time, then $k > 1$.
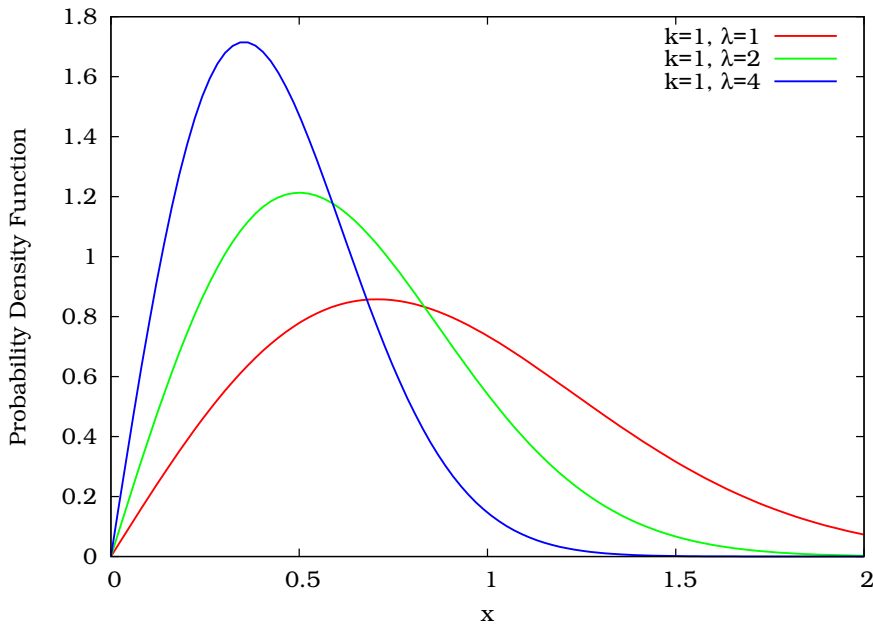

Figure C.7: Weibull distribution.

Appendix D

TG's Pareto Implementation

D.1 Added Code

The added code is based on the GNU Scientific Library (GSL) numerical library for C programmer[1]. This library provides a wide range of mathematical routines such as random number generators, special functions, and least-squares fitting. Instead of including the whole library to compute the Pareto distribution, which would be too heavy for TG, only the relevant parts of the GSL code were included.

D.1.1 tg.y & scan.l

The first step in supporting the Pareto distribution is to enable the notification of this distribution in the command line of TG. This is done by adding the new distribution in the tg.y (lines 192 and 746^+) and in the scan.l (line 121) files to indicate to the Lex parser the existence of this newly added distribution.

```
0     /* tg.y */
(...)
192   %token (...) DIST_PARETO
(...)
746   | DIST_PARETO number number number   /* alpha min max */
747   {
748   char    *cp;
```

---

[1]The GNU Scientific Library: http://www.gnu.org/software/gsl/, last visited: March 13, 2008

```
749     if ((cp = dist_pareto_init(&$$.tmpdist, $2.d, $3.d, $4.d)) != NULL)
750     yyerror(cp);
751     }
752     ;
(...)


0     /* scan.l */
(...)
121   "pareto"    { count(); BEGIN LEX_NORMAL; return(DIST_PARETO); }
(...)
```

D.1.2   distribution.c & distribution.h

The second step consists in implementing the body of the distribution. All the distributions are implemented into the `distribution.c` file and exported using `distribution.h`. This implies introducing into these two files the Pareto distribution by adding three new functions in the first file:

- `proba()`: returns a randomly generated real number between 0 and 1 (both excluded),

- `dist_pareto_gen(dist)`: returns a Pareto distributed random variable,

- `dist_pareto_init(dist, alpha, min, max)`: initialises the Pareto distributed random variable within the interval [min, max].

```
0     /* distribution.c */
1     double
2     proba ()
3       {
4       double res = (1.0*random()/(RAND_MAX+1.0));
5       while (res==0.0 || res == 1.0) res = (1.0*random()/(RAND_MAX+1.0));
6       return(res);
7       }
8
9     /* distribution.c */\end{listing}
10    double
11    dist_pareto_gen(dist)
12
13      distribution *dist;
14      {
15      double x = proba();
16      double z = pow (x, -1/(dist->par1));
17      double res = (dist->par2)*z;
18      while (res>dist->par3)
19        {
20        x = proba();
21        z = pow (x, -1/(dist->par1));
22        res = (dist->par2)*z;
23        }
24      return(res);
25      }
26
27    char *
28    dist_pareto_init(dist, alpha, min, max)
29
```

```
30      distribution *dist;
31      double      alpha;
32      double      min;
33      double      max;
34      {
35      dist->generate = dist_pareto_gen;
36      dist->par1 = alpha;
37      dist->par2 = min;
38      dist->par3 = max;
39      return (NULL);
40      }
(...)
```

The final step is to modify the `distribution.h` (line 43) file to export the new `dist_pareto_init(dist, alpha, min, max)` function:

```
0    /* distribution.h */
(...)
43   extern char              *dist_pareto_init();
(...)
```

D.2 Usage

Once all the modifications mentioned above are completed, it is possible to use the Pareto distribution like the other ones. The Pareto distribution needs three parameters: exponent (α), scale (β) and maximum value (m).

The Pareto distribution might be, as all the other distributions, used in both `arrival` or `length` argument:

```
# client1.tg
on 0:15 tcp 10.0.14.2.81
at 2 setup
at 0.247940 arrival exponential 0.0083 length pareto 1.1 81.5 66666
packet 12
wait 22.284628
(...)
```

```
# client2.tg
on 0:15 tcp 10.0.14.2.81
at 2 setup
at 13.596063 arrival pareto 2 3 5 length 634
packet 82
wait 11.670262
(...)
```

D.3 Proof

In Fig. D.1, the line represents the GSL's implementation of the Pareto function with the following parameters (representing

the packet length distribution of an Interactive session as pre-sented in Table 3.3): $\alpha = 1.1$, $beta = 81.5$ and $m = 66,666$, whereas the points show a graphical representation of a set of $10,000$ random generated numbers base on the Pareto distribution implemented into TG.
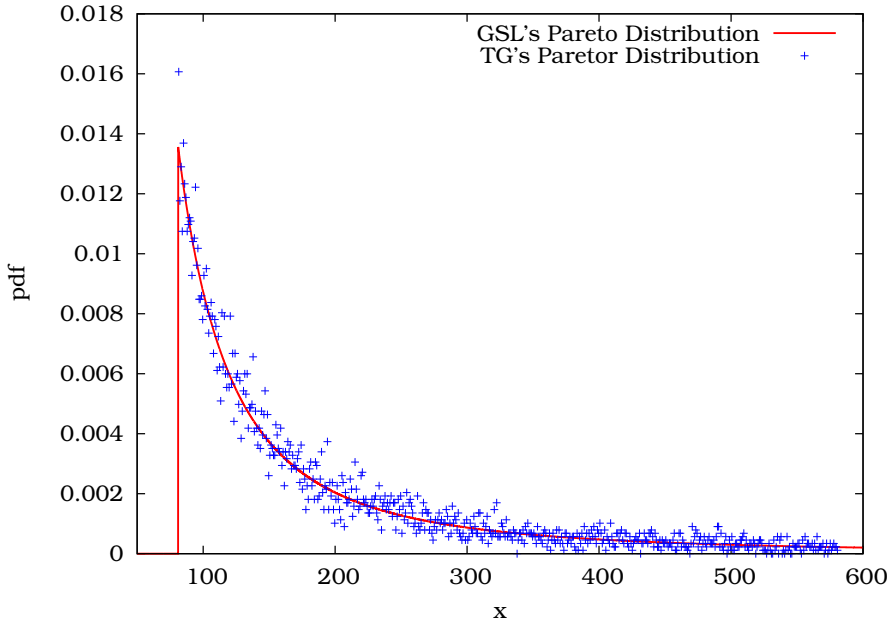

Figure D.1: TG's Pareto distribution.

The points set sticks to the line, suggesting that the implementation of the TG's Pareto distribution is acceptable.

Appendix E

TG's Exponential Implementation

E.1 Added Code

To fit as much as possible the GSL code, changes have been integrated to the exponential distribution in the `distribution.c` file. The `dist_exp_gen(dist)` function has been transformed so as it interacts with the previously presented `proba()` function:

```
0     /* distribution.c */
1
2     double
3     dist_exp_gen(dist)
4
5       distribution    *dist;
6       {
7       double u = proba();
8       double res = -dist->par1*log(u);
9       while (res>dist->par3 || res<dist->par2)
10        {
11        u = proba();
12        res = -dist->par1*log(u);
13        }
14      return(res);
15      }
(...)
```

E.2 Usage

The usage of the exponential distribution has not changed at all. It can still be used in both `arrival` or `length` argument, with one (mean) or three (mean, min, max) arguments. See TG's documentation for more details.

E.3   Proof

In Fig. E.1, the line represents the GSL's implementation of the exponential function centred on 12, whereas the points show a graphical representation of a set of $100,000$ random generated numbers based on the exponential distribution implemented into TG.
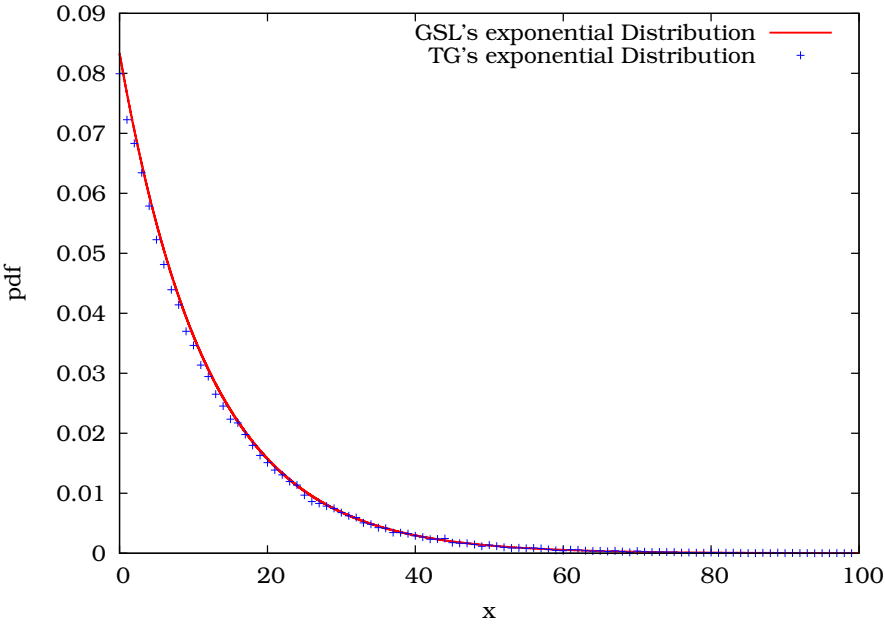

Figure E.1: TG's exponential distribution.

The line and the set of points show a good match, suggesting that the implementation of the exponential distribution is acceptable.

Appendix F

Detailed Iproute2 and NetEm Modifications

F.1   Iproute2 Modifications

The modification applied on Iproute2 are mainly focused on the interpretation of the command line arguments. Only two files have been modified: `pkt_sched.h` and `q_netem.c`.

Three lines have to be added to the first file in order to add three entries to the `tc_netem_qopt`:

- the BLER experienced,

- the RLC mode used, and

- the SF allocated to the channel.

```
0    /* pkt_sched.h */
(...)
447  __u32  bler;           /* indicates the BLER undergone (PDU level) (en 0/00) */
448  __u32  rlc_mode;       /* indicates the RLC mode (0=TM, 1=UM, 2=AM)        */
449  __u32  sf;             /* indicates the SF used (4, 16, ..., 512)          */
(...)
```

The major change in the `q_netem.c` files can be found in the argument parsing method `netem_parse_opt`. The following lines initialise the three above presented entries with the right values, coming from the command line.

```
0     /* q_netem.c */
(...)
202   dist_data = alloca(MAXDIST);
203   dist_size = get_distribution("weibull", dist_data);
204   corrupt.probability=1;
205   NEXT_ARG();
206   /* bler */
207   NEXT_ARG();
208   get_u32(&opt.bler, *argv, 0);
209   NEXT_ARG();
210   /* mode */
211   NEXT_ARG();
212   get_u32(&opt.rlc_mode, *argv, 0);
213   NEXT_ARG();
214   /* sf */
215   NEXT_ARG();
216   get_u32(&opt.sf, *argv, 0);
(...)
```

F.2   NetEm Modifications

The real challenge in this implementation lies with the NetEm kernel module modification. This involves three different files: `skbuff.h`, `pkt_sched.h` and `sch_netem.c`.

The modification in the first of these files includes two additional fields in the `sk_buff` structure, which represents a packet within the kernel implementation. The idea here is to stick two more variable to a packet: `tti_score` which represents the number of error-free TTIs and `rlc_try` which indicates the number of retransmission attempts as presented in Section 5.4.4.

```
0     /* skbuff.h */
(...)
273   unsigned int    len,
(...)
278                   tti_score,
279                   rlc_try;
(...)
```

The code added in the second file is the link between the arguments parsed in the command line and the implementation of the wireless link behaviour into the `sch_netem.c` file. It adds three entries into the `tc_netem_qopt` structure to store the values previously presented:

```
0     /* pkt_sched.h */
(...)
447   __u32   bler;
448   __u32   rlc_mode;
449   __u32   sf;
(...)
```

Finally, the most important part of the implementation lies in the last file. The first step is to implement a PDU divider coupled with a PDU corrupter based on the Weibull distribution:

```
0      /* sch_netem.c */
(...)
155    static short gilbert_weibull_corrupt(struct sk_buff *skb, struct netem_sched_data
                                                                            *qdisc_q)
156    {
157    int     nb_tti, pdu_length=336, cps=3840000, bit_per_tti, pdu_per_tti, j;
158    int  TTI_length=100; /* 10ms */
159    int  rate_coding=2; /* 1/2 */
160    int  MAC_head=18;   /* in bytes */
161    unsigned long  q, max, alea;
162
163    if(qdisc_q->bler==0) return(0); /*No BLER = packet passes withOUT corruption*/
164
165    max   = 4294967295;
166    q   = (max / 18) * 10;
167    bit_per_tti = (((cps/(int)qdisc_q->sf)/rate_coding)/TTI_length);
168    pdu_per_tti = (bit_per_tti/pdu_length)+1;
169    nb_tti  = (((skb->len-MAC_head)*8)/(bit_per_tti-qdisc_q->rlc_mode*8*pdu_per_tti));
170    if (((skb->len-MAC_head)*8)%(bit_per_tti-qdisc_q->rlc_mode*8*pdu_per_tti)!=0) nb_tti++;
171
172    for(j=skb->tti_score; j<nb_tti; j++)
173     {
174     if(qdisc_q->gilb_etat==TTI_NO_ERROR)
175      {
176      if(qdisc_q->gap_length>0)
177       {
178       qdisc_q->gap_length--;
179       skb->tti_score++;
180       }
181      else qdisc_q->gilb_etat=TTI_ERROR;
182      }
183     else {
184      alea = net_random();
185      if(alea < q)
186       {
187       qdisc_q->gilb_etat=TTI_NO_ERROR;
188       qdisc_q->gap_length=(int)tabledist(659/(int)qdisc_q->bler, 1185/(int)qdisc_q->bler,
                                             &qdisc_q->delay_cor,qdisc_q->delay_dist);
189       skb->tti_score++;
190       }
191      }
192     }
193
194    if(nb_tti<=skb->tti_score)                        return(0); /*packet passes
                                                                      withOUT corruption*/
195    else if(qdisc_q->rlc_mode==2 && skb->rlc_try<=maxDAT) return(1); /*packet has to
                                                                      be requeued*/
196    else                                             return(2); /*packet has to
                                                                      be corrupted*/
197    }
(...)
```

Based on this procedure, it is now possible to compute, for each packet, whether it will be corrupted or not during its transmission over the air interface:

```
0     /* sch_netem.c */
(...)
251   if (q->corrupt)
252     {
253     short gilb_res=gilbert_weibull_corrupt(skb, q);
254     while (gilb_res==1) /* re-waiting queue */
255      {
256      skb->rlc_try++;
257      gilb_res=gilbert_weibull_corrupt(skb, q);
```

```
258      }
259      if(gilb_res==2)  /* corruption */
260       {
261       skb->data[skb_headlen(skb)-2] ^= 1<<(net_random() % 8);
262       }
263      }
(...)
```

The modified code (Iproute2 and NetEm kernel module) has been made publicly available on the Internet[1].

---

[1]New version of Iproute2 and Netem:
`http://www.info.fundp.ac.be/~hvp/rech/doc/docs/6/netem_2.0.tar.gz`, last visited: March 13, 2008