BLUMTS

- Bluetooth as an access technology to UMTS

Master thesis

KTH, Royal Institute of Technology
Department of Microelectronics and Information Technology

Telia Research AB, Mobile and Wireless Communication

Stockholm, spring 2002

Author: Stefan Kinnestrand
Examiner: Professor Gunnar Karlsson, KTH
Advisor: John Tillberg, Telia Research AB

Abstract

This thesis considers how Bluetooth and UMTS can be used together to give electronic devices access to UMTS packet data services. UMTS provides mobile wireless data access and Bluetooth is used for final delivery to and from local devices. An example of what might become a common scenario is one or several PDAs, laptops or other devices sharing a UMTS terminal via Bluetooth. The UMTS terminal acts as a gateway and provides Internet access to the devices.

The objective of this thesis is to evaluate how Bluetooth and UMTS work together and to identify and analyse fundamental issues for a Bluetooth/UMTS gateway. There are several factors that affect the performance: the UMTS radio access bearer data rate, the Bluetooth polling algorithm, the number of users sharing a common gateway, and the type of Bluetooth packets that are being used.

A simulator has been implemented to evaluate how performance varies for different user scenarios. Simulation results and theoretical results are presented along with an analysis in the report.

The results show that Bluetooth and UMTS are well matched in terms of data rate, and that they work well together for most user scenarios. However, there is no general answer to which of these technologies is the limiting one; it depends on the conditions and settings for the two radio channels involved. Simulation results also show that the round robin polling algorithm, which is currently used in the Bluetooth chips, might be a bottleneck if several users share a common gateway. An exhaustive, or partially exhaustive, polling algorithm performs much better. Furthermore, there is a need to adapt the UMTS and Bluetooth radio bearers to each other in order to achieve efficient utilization of the radio resources. There is an optimal mapping between the Bluetooth packet type and the UMTS data rate such that throughput is maximized and the allocated radio resources are kept to a minimum. This mapping requires signalling between Bluetooth and UMTS. Such signalling is also needed in order to be able to provide quality of service.

Acknowledgments

I am indebted to a number of persons that have contributed to this work. First and foremost I would like to thank my colleagues at Telia Research AB, Mobile and Wireless Communication, for their help during this project. John Tillberg, my advisor at Telia, deserves a special acknowledgment for his assistance and inspiration. He was the one that came up with the idea for this master thesis project and he kept his enthusiasm during the entire work. Furthermore, I am grateful to Ulf Nilsson, Per Ernström, Niklas Wirén, and Anders Dahlén at Telia Research AB for fruitful discussions and for sharing their expertise with me. I would also like to express my appreciation to Professor Gunnar Karlsson, KTH, for his support, advices, and help in other ways. Finally, I am infinitely grateful to my family for their support during my entire education.

Stockholm, January 2002
Stefan Kinnestrand

Table of contents

1 INTRODUCTION **1**

1.1 THE WIRELESS LANDSCAPE 1
1.2 PROJECT OBJECTIVE 2
1.3 PROBLEM DEFINITION 2
1.4 RESEARCH METHODOLOGY 2
1.5 DELIMITATIONS 3

2 BLUETOOTH **5**

2.1 BLUETOOTH TECHNOLOGY 5
2.1.1 Vision and goals 5
2.1.2 Multi-access scheme 5
2.1.3 Transmitter and receiver 7
2.1.4 Medium access control 7
2.1.5 Physical links 8
2.1.6 Bluetooth packets 8
2.1.7 Error correction 9
2.1.8 Protocol architecture 10
2.1.9 Usage models and profiles 11
2.2 BLUETOOTH RESEARCH 12
2.2.1 Bluetooth-based Internet access point 12
2.2.2 Polling algorithms 13
2.2.3 Packet types and SAR policies 14
2.2.4 TCP over Bluetooth 15
2.2.5 Interference between Bluetooth and IEEE802.11 15
2.2.6 Issues of relevance to this project 16

3 UMTS **17**

3.1 SYSTEM OVERVIEW 17
3.1.1 System architecture 17
3.1.2 The UTRA FDD radio interface 19
3.1.3 Error protection and retransmission 19
3.1.4 Transport channels for packet data 20
3.1.5 UE operational modes 21
3.1.6 UMTS QoS 22

4 SIMULATOR IMPLEMENTATION **25**

4.1 SIMULATOR OVERVIEW 25
4.1.1 Simulation environment 25
4.1.2 Simulator architecture 25
4.2 THE BLUETOOTH PART 27
4.2.1 Architecture and protocols 27
4.2.2 Packet encapsulation and segmentation 28

4.2.3 Bluetooth parameters 30
4.3 THE UMTS PART 31
4.3.1 Architecture 31
4.3.2 UMTS parameters 32
4.4 THE NETWORK PART 32
4.5 TRAFFIC MODELS 32
4.6 PERFORMANCE METRICS 34
4.7 SIMULATOR LIMITATIONS 34
5 SIMULATIONS AND ANALYSIS 37
5.1 THEORETICAL ANALYSIS 37
5.2 SIMULATION RESULTS AND ANALYSIS 41
5.2.1 About the results 41
5.2.2 Concurrent web sessions 41
5.2.3 One voice conversation and concurrent web sessions 46
5.2.4 File downloads and web sessions 48
6 CONCLUSIONS 51
6.1 CONCLUSIONS 51
6.2 FURTHER WORK 52
7 REFERENCES 53
7.1 WRITTEN SOURCES 53
7.2 WEB SOURCES 54
APPENDIX A – ACRONYMS 55
APPENDIX B – RANDOM VARIABLES 57
THE LOG-NORMAL DISTRIBUTION 57
THE BOUND PARETO DISTRIBUTION 57
APPENDIX C – ON THE DEFINITION OF THROUGHPUT 59

List of figures

Figure 2.1 Bluetooth frequency hop selection ... 6
Figure 2.2 The FH/TDD scheme applied in Bluetooth ... 6
Figure 2.3 The Bluetooth packet structure. ... 8
Figure 2.4 The Bluetooth protocol stack. ... 11
Figure 2.5 The Bluetooth profile structure. ... 12
Figure 3.1 The European UMTS frequency spectrum ... 17
Figure 3.2 UMTS network architecture ... 18
Figure 3.3 Spreading and scrambling of data in UMTS. ... 19
Figure 3.4 UE modes and the different service states ... 21
Figure 3.5 The UMTS QoS architecture levels. ... 22
Figure 4.1 Simulator architecture. ... 26
Figure 4.2 The simulator objects and the buffer structure. ... 27
Figure 4.3 The LAN Access Profile protocol stack ... 28
Figure 4.4 Data segmentation and encapsulation. ... 29
Figure 5.1 Downlink throughput versus number of slaves for UL:64/DL:384 kbps RAB. ... 42
Figure 5.2 Uplink throughput versus number of slaves for UL:64/DL:384 kbps RAB... 44
Figure 5.3 Downlink throughput versus number of slaves for UL:64/DL:256 kbps RAB. ... 45
Figure 5.4 Uplink throughput versus number of slaves for UL:64/DL:256 kbps RAB... 45
Figure 5.5 Downlink throughput for 128 kbps UMTS RAB and 1-slot packets. ... 46
Figure 5.6 Throughput for web sessions with one voice conversation active. ... 47
Figure 5.7 Throughput for file downloads ... 48
Figure 5.8 Throughput for one file download with concurrent web sessions ... 49

List of tables

Table 2.1 Bluetooth transmitter power classes ... 7
Table 2.2 Bluetooth SCO packets. ... 9
Table 2.3 Bluetooth ACL packets. ... 9
Table 3.1 QoS attributes for the UMTS bearer service for the four different traffic classes. ... 23
Table 4.1 Possible combinations of number of SCO links and the SCO packet interval. 31
Table 4.2 Traffic models for HTTP traffic. ... 33
Table 5.1 Maximum number of slaves that can use the gateway before Bluetooth becomes the bottleneck for RR polling ... 38
Table 5.2 The minimum time required for sending a 1500 byte IP packet over the Bluetooth link when FEP is used ... 39
Table 5.3 A summary of when UMTS or Bluetooth is the limiting technology for fair exhaustive polling. ... 39
Table 5.4 A summary of when UMTS or Bluetooth is the limiting technology for round robin polling. ... 40

1 Introduction

This chapter discusses some major trends in wireless communications and why there is an interest in using Bluetooth in combination with UMTS. The chapter also includes the definition of the research problem and the aim of this project.

1.1 The wireless landscape

Currently there are two major trends in wireless communications. One trend is that cellular network technologies are migrating towards packet switching to bring Internet to mobile terminals. It has already begun with the launch of GPRS and it will continue with the introduction of UMTS. The other trend is the rapid development of short-range wireless technologies such as Bluetooth and IEEE802.11. These technologies are capable of linking devices wirelessly to each other in an ad-hoc manner, and they can also link devices to the Internet.

Associated with these two trends is the rapid growth in the use of portable devices such as mobile phones, PDAs, and laptops. The capabilities of these are increasing in terms of processing power, display quality, and functionality. Most likely, people will have several devices like mobile phones, PDAs and laptops in the future. To combine them all into one is currently not an option since small size is important for mobile phones, whereas sufficient display size is important for PDAs, and finally enough processing power and display size is important for laptops. Some people argue that a combination of all these devices into one is the ultimate solution, but even so, such technology lies many years away. Several types of devices will coexist for a foreseeable future and there will be a need for linking these devices to the Internet and to each other. The combination of short-range wireless technologies and cellular network technologies will be important to satisfy this need.

The combination of Bluetooth and UMTS is attractive for several reasons. Bluetooth has unique features among the available short-range wireless technologies in that it has been designed to be a small, low-power, low complexity, and low-cost technology. The small size and low price of Bluetooth chips allow them to be included in many devices. Bluetooth replaces the need for cables and it supports automatic synchronization and ad-hoc networking. UMTS, on the other hand, will enable wireless access to the Internet independent of location. However, due to the price and size of UMTS modules it will not be attractive to include them into all sorts of devices.

Bluetooth could be used to link different electronic devices to a UMTS equipped device. This approach has several advantages. Typically, only one device will have to be equipped with UMTS functionality. Other devices can gain access to UMTS packet data services via Bluetooth. This reduces the size and the price of the devices, and since Bluetooth is very power efficient compared to UMTS, their batteries will also last longer or could be made smaller. There are several potential user scenarios. Some examples are:

- PDAs, laptops or other devices communicate with a UMTS mobile terminal via Bluetooth. The terminal acts as a Bluetooth/UMTS gateway and the devices gain access to the Internet via the UMTS network.
- Bluetooth-based access points could be set up in trains and buses and UMTS could be used for linking the access point to the Internet.
- A digital camera can transfer pictures or streaming video to a UMTS mobile terminal via Bluetooth. The pictures or video can then be sent to friends and relatives over the UMTS network.

Altogether, using Bluetooth in combination with UMTS opens up new possibilities for mobile communication. These possibilities motivate the current study.

1.2 Project objective

At Telia Research AB a lot of research is done within the area of mobile and wireless communication. Lately multi-access services have received much attention. Within multi-access services the combination of cellular network technologies and short-range wireless technologies is a matter of major interest. Bluetooth together with UMTS opens up new possibilities for mobile communication. It is so far an unexplored area since earlier research has focused on either Bluetooth or UMTS separately.

The objective of this thesis is to evaluate how Bluetooth and UMTS work together and to identify and analyse fundamental issues for a Bluetooth/UMTS gateway.

1.3 Problem definition

The research task is to answer some fundamental questions about the combination of Bluetooth and UMTS: How does performance vary with the number of users simultaneously sharing a Bluetooth/UMTS gateway? What effect will the Bluetooth scheduling algorithm have on performance? How do variables such as the UMTS radio access bearer and the Bluetooth packet type affect performance? Which is the limiting technology, UMTS or Bluetooth? What basic functionality will be required from a Bluetooth/UMTS gateway?

1.4 Research methodology

The project has been divided in two main parts. The first part consists of a literature study and theoretical analysis. The second part consists of an implementation of a simulator and the simulation results. The aim of the first part is to find out how Bluetooth and UMTS can be modelled, what important factors affect the Bluetooth and UMTS links, and what functionality would be required from a Bluetooth/UMTS gateway. The simulations in the second part of the project aim at evaluating the performance for compound Bluetooth/UMTS links for some realistic user scenarios.[1]

[1] No UMTS equipment was available and therefore simulating was the only feasible way to evaluate performance. Furthermore, a simulator had to be implemented from scratch since no appropriate simulator was available.

1.5 Delimitations

- The analysis will be limited to a single Bluetooth piconet. Scatternets will not be considered.
- A complete architecture for how to build a Bluetooth/UMTS gateway will not be presented. Some important functionality will be identified but not analysed in detail.
- It is assumed that when several terminals share a common Bluetooth/UMTS gateway their traffic streams are multiplexed onto one UMTS radio bearer in a FIFO manner.
- The maximum UMTS data rate considered is 384 kbps.
- The simulations only consider non-real time services such as web sessions and file downloads.

2 Bluetooth

This chapter gives a comprehensive introduction to the Bluetooth technology and to prior research on Bluetooth that is of importance to the project.

2.1 Bluetooth technology

2.1.1 Vision and goals

Bluetooth is a short-range radio technology that originally was intended to eliminate the need for cables between electronic devices. The Bluetooth radio system has been designed in order to achieve low power, cost, complexity, to be robust, and to attain worldwide availability. The design enables small and cheap radios that can be embedded in different devices. The vision is that embedded radios will lead towards ubiquitous connectivity. Furthermore, Bluetooth will allow this to occur without explicit user interaction.

The original idea that led to the development of Bluetooth was born at Ericsson in Lund, Sweden, in 1994. In early 1998 Ericsson together with Nokia, IBM, Intel, and Toshiba formed a special interest group (SIG) to promote Bluetooth. Since then 3Com, Lucent, Microsoft, and Motorola have joined the SIG. The purpose is to establish a de facto standard for the air interface and the software that controls it, and thereby ensure interoperability between devices from different manufacturers. The Bluetooth specification is open and free. It is available at http://www.bluetooth.com.

There is a Bluetooth qualification program (BQP) for product certification to control the quality and to ensure interoperability between devices from different manufacturers. A company must become a SIG member and their products must be tested before it is allowed to sell them under the Bluetooth name (Bluetooth Special Interest Group 2001c).

2.1.2 Multi-access scheme

Bluetooth operates in the unlicensed ISM band at 2.4 GHz. The problem with this band is the large number of systems operating there (e.g. IEEE802.11b WLANs, microwave ovens, and baby monitors). Bluetooth applies frequency hopping (FH) to combat interference and fading: 79 hop carriers with a 1 MHz spacing have been defined along with a large number of pseudo-random sequences. The unit that controls the FH channel is called the *master* and determines the particular sequence used. The clock of the master unit defines the phase of the hopping sequence. All participating units on the hopping channel are *slaves*. As shown in Figure 2.1 the slaves use the master identity to select the same hopping sequence and add time offsets to their respective clocks to synchronize to the frequency hopping.

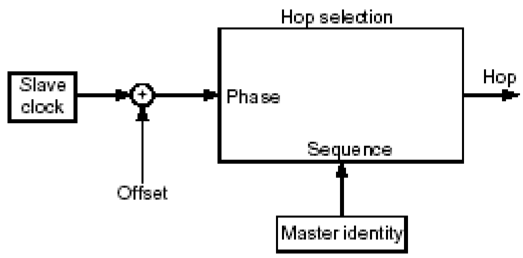

Figure 2.1 Bluetooth frequency hop selection.

In the time domain, the channel is divided into slots. The minimum dwell time is 625 μs and corresponds to a single slot. Full duplex communications is achieved by applying time division duplex (TDD). This means that a unit alternatively transmits and receives. Since transmission and reception takes place at different time slots, transmission and reception also takes place at different hop carriers. Figure 2.2 illustrates the FH/TDD channel applied in Bluetooth.

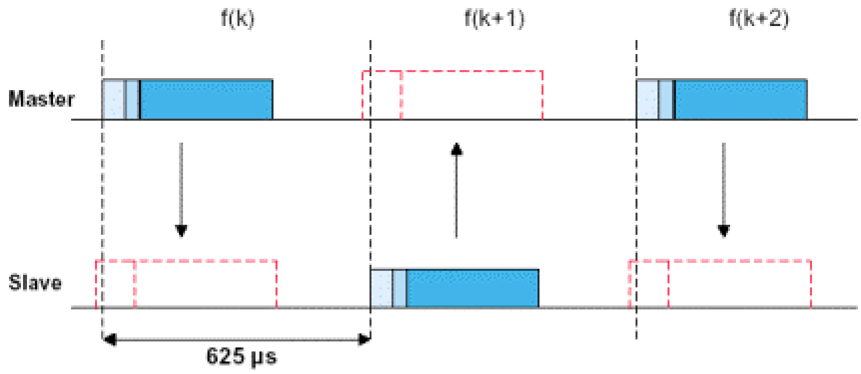

Figure 2.2 The FH/TDD scheme applied in Bluetooth.

An FH Bluetooth channel is associated with a *piconet*. As mentioned above, the piconet channel is defined by the master unit's identity (providing the hop sequence) and system clock (providing the hop phase). All other units participating in the piconet are slaves. Each Bluetooth unit has a free running clock. When a piconet is established the slaves add offsets to their system clocks to synchronize to the master. These offsets are released when the piconet is cancelled, but can be stored for later use. Different channels cannot have the same master and therefore they have different hopping sequences and phases. The maximum number of units that can participate on a common channel is limited to eight (one master and seven slaves). This enables high capacity between all units. The master/slave role is only attributed to a unit for the lifetime of the piconet. When the piconet is cancelled, the master/slave roles are cancelled. Each unit can become a master or a slave. By definition, the unit establishing the piconet becomes the master, but once the piconet has been established the master/slave roles can be exchanged.

Multiple piconets can be interconnected with each other and are referred to as a *scatternet*. It is possible to interconnect different piconets, but a unit can communicate in only one piconet at a time. The different piconets are *not* synchronized in time and each piconet has its own hopping sequence. A unit can jump from one piconet to another by adjusting the piconet channel parameters, and it can change role when jumping. For example, a unit can be master in one piconet at one instant in time, and be a slave in a different piconet at another instant in time. A unit can also be a slave in different piconets, but cannot be master in different piconets since the master parameters specify the piconet FH channel, and two piconets cannot share the same channel (Bluetooth Special Interest Group 2001a, Haartsen 2000).

2.1.3 Transmitter and receiver

The Bluetooth transmitter is classified into three power classes (see Table 2.1). It is expected that implementations using power class 3 (1 mW) will be most common. The range for a 1 mW transceiver is about 10 meters and it is enough for most uses. With power class 3 the implementation could be small since RF amplifiers are not needed. In addition, the power consumption is low which means that batteries last longer or could be made smaller. Power control is required for transmit powers above 4dBm.

Power class	Max output power	Min output power	Power control
1	100 mW (20dBm)	1 mW (0 dBm)	Yes, for powers > 4 dBm. Otherwise optional.
2	2.5 mW (4 dBm)	0.25 mW (-6 dBm)	Optional
3	1 mW (0 dBm)	N/A	Optional

Table 2.1 Bluetooth transmitter power classes.

The modulation used is GFSK, a simple modulation scheme that allows low complexity in the receiver. The transmitted data has a symbol rate of 1 Msymbols/s. The required sensitivity of the receiver is –70 dBm or better. This is easy to achieve and means that the receiver implementation can be cheap (Bluetooth Special Interest Group 2001a).

2.1.4 Medium access control

The master controls the traffic on the piconet and takes care of access control. Access control is centralized and completely contention free. Only communication between the master and one or more slaves is possible (i.e. communication between two slaves has to go through the master). Time slots are alternatively used for master transmission and slave transmission. In the master transmission, the master includes an address of the slave unit for which the information is intended. In order to prevent collisions on the channel due to multiple slave transmissions, the master uses polling. For each slave-to-master slot, the master decides which slave is allowed to transmit. Only the slave addressed in the master-to-slave slot directly preceding the slave-to-master slot is allowed to transmit (Bluetooth Special Interest Group 2001a, Haartsen 2000).

The polling algorithm is the set of rules that determines when the master switches from one slave to another, which is the next slave to be polled, the number of successive times that slave is polled etc (Capone et al 2001). The polling algorithm will affect performance in the piconet. The overall goal is to achieve fair sharing of bandwidth and high link utilization. In the first version of the Bluetooth specification a round robin (RR) scheme was specified. Round robin polling simply means that the master polls the slaves one after another in a cyclic fashion. Unfortunately this polling algorithm performs poorly since it is not well suited for a TDD system where uplink and downlink scheduling are tightly coupled. The simple RR scheme wastes slots, and thus bandwidth, since the master polls the slaves in a cyclic fashion even when the master or the slave has nothing to send. In the current version of the Bluetooth specification the polling algorithm is unspecified, leaving the issue to be solved by the implementer.

2.1.5 Physical links

Bluetooth supports synchronous services such as voice and asynchronous services such as bursty data. Two types of physical links have been defined:

- Synchronous connection oriented (SCO) links.
- Asynchronous connectionless (ACL) links.

The SCO link is a symmetric, point-to-point link between the master and a single slave. The link is established by reservation of duplex slots at regular intervals, and can thus be considered a circuit. The master will send SCO packets at regular intervals in the master-to-slave slots to the particular slave, and the slave is always allowed to respond with an SCO packet in the following slave-to-master slot. The ACL link is a point-to-multipoint link between the master and all the slaves in the piconet. In slots that are not reserved for SCO links, the master can establish an ACL link to any of the slaves. The ACL link is established on a per slot basis and can thus be considered a datagram connection. A slave is permitted to return an ACL packet in the slave-to-master slot if and only if it has been addressed in the preceding master-to-slave slot. ACL packets not addressed to a specific slave are considered broadcast packets and are read by every slave in the piconet (Bluetooth Special Interest Group 2001a).

2.1.6 Bluetooth packets

Bluetooth uses packet-based communication. Only a single packet can be sent in each slot. All packets have the format shown in Figure 2.3.

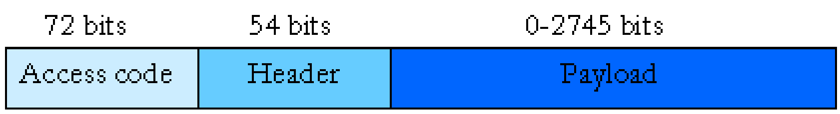

Figure 2.3 The Bluetooth packet structure.

There are four control packets defined in Bluetooth:

- ID packet: Consists of the access code only. Used for signalling.
- NULL packet: Consists of access code and packet header. Used if link control information carried by the packet header has to be conveyed.
- POLL packet: Similar to the NULL packet. Used by the master to force slaves to return a response.

- FHS packet: A synchronization packet used to exchange clock and identity information. Contains all the information necessary to get two units hop synchronized.

There are several other packet types in Bluetooth used for synchronous and asynchronous services. For SCO links there are four packet types, all intended for 64 kbps voice connections (see Table 2.2). Three of those, the HV (high quality voice) packets, are intended for conversations. The fourth SCO packet type, DV (data voice), is a combined voice and data packet. The SCO packets do not include any CRC and are never retransmitted.

Packet type	User payload (bytes)	FEC	CRC	Symmetric max rate (kbps)
HV1	10	1/3	No	64
HV2	20	2/3	No	64
HV3	30	No	No	64
DV	10 + (0 – 9)	(2/3)	(Yes)	64 + (57.6)

Table 2.2 Bluetooth SCO packets.

Text and numbers within parentheses refer to data field for DV.

There are seven ACL packets of three categories (see Table 2.3): single slot packets, 3-slot packets, and 5-slot packets. Multi-slot packets are sent on a single hop carrier, i.e. there is no frequency switch in the middle of the packet. The frequency that is valid in the beginning of the packet is used throughout. Six of the ACL packets, the DM (data medium) and DH (data high) packets, contain CRC code and retransmission is applied if no acknowledgement is received. The 7th ACL packet, the AUX1 packet, has no CRC and is not retransmitted (Bluetooth Special Interest Group 2001a).

Packet type	Payload header (bytes)	User payload (bytes)	FEC	CRC	Symmetric max rate (kbps)	Asymmetric max rate (kbps)	
						Forward	Reverse
DM1	1	0-17	2/3	Yes	108.8	108.8	108.8
DH1	1	0-27	No	Yes	172.8	172.8	172.8
DM3	2	0-121	2/3	Yes	256.0	384.0	54.4
DH3	2	0-183	No	Yes	384.0	576.0	86.4
DM5	2	0-224	2/3	Yes	286.7	477.8	36.3
DH5	2	0-339	No	Yes	432.6	721.0	57.6
AUX1	1	0-29	No	No	185.6	185.6	185.6

Table 2.3 Bluetooth ACL packets.

2.1.7 Error correction

Bluetooth includes both FEC and packet retransmission schemes. The FEC schemes are very simple which is a requirement because of the limited processing time between receive and transmit. The 1/3 FEC merely uses a three-bit repetition code with majority decision at the receiver. This FEC is used for packet headers, and can additionally be applied to the payload of SCO packets. For the 2/3 FEC, a shortened Hamming code is used. It can be applied to the payload of both SCO and ACL packets.

For ACL links a fast-ARQ scheme is applied, which means that the sender is notified of the packet reception in the receive slot directly following the transmit slot in which the packet was sent. Each payload contains a CRC code to check for errors, and the acknowledgement is piggybacked on the packet header of the return packet. The fast-ARQ scheme only protects the payload of the packet (and only on those which have CRC). Packet headers and voice payloads are not protected by fast-ARQ. Broadcast packets are checked for errors using the CRC, but no fast-ARQ is applied and they are never acknowledged (Bluetooth Special Interest Group 2001a, Haartsen 2000).

2.1.8 Protocol architecture

The Bluetooth protocol stack is shown in Figure 2.4. The lower layers of the protocols, i.e. the data link layer and the physical layer, are designed to provide a flexible base for further protocol development. Other protocols, such as RFCOMM, are adopted from existing protocols and are slightly modified. The upper layers are existing protocols without modifications. Different applications may run over the protocol stacks, but to achieve interoperability, matching applications in remote devices must run over identical protocol stacks (Bluetooth Special Interest Group 2001a, Haartsen 2000). The basic functionality of the Bluetooth specific protocols is:

- RF layer: specifies the radio parameters.
- Baseband layer: specifies the lower level operations at the bit and packet levels, e.g. FEC operations, encryption, CRC calculations, and fast-ARQ.
- Link Manager Protocol (LMP): specifies link set-up, security and control. LMP messages have higher priority than user data. Among the more important functionalities are connection establishment and release, connection and release of SCO and ACL channels, traffic scheduling, link supervision, and power management.
- Logical Link Control and Adaptation Protocol (L2CAP): forms an interface between standard data transport protocols and the Bluetooth protocol. It handles multiplexing of higher-layer protocols and segmentation and reassembly (SAR) of large packets.
- Service Discovery Protocol (SDP): enables a Bluetooth unit to find the capabilities of other Bluetooth units in range.
- RFCOMM: provides emulation of serial ports. It is based on the ETSI standard TS07.10, and is intended to cover applications that make use of the serial ports of the devices in which they reside.

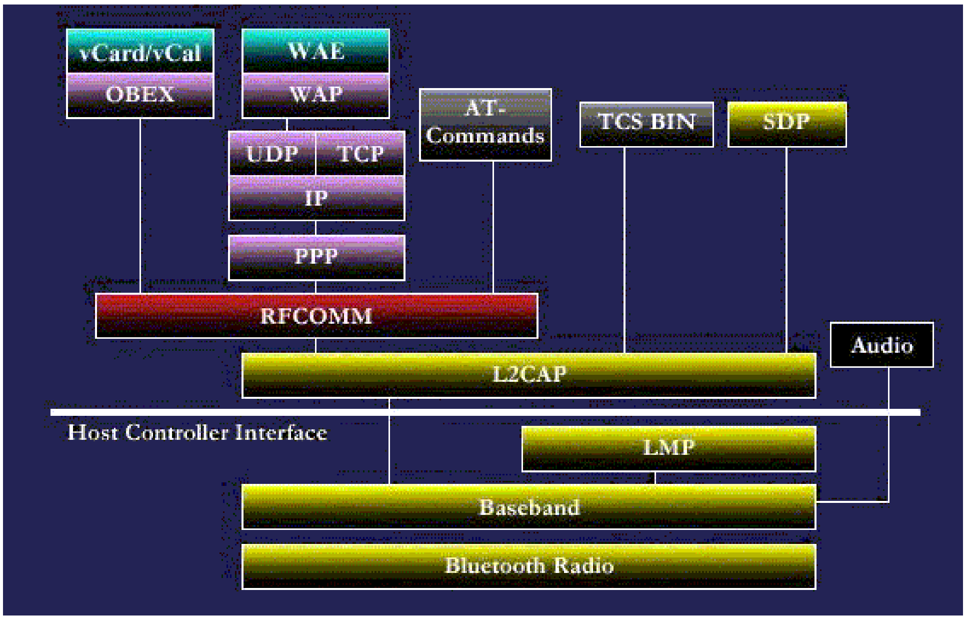

Figure 2.4 The Bluetooth protocol stack.

2.1.9 Usage models and profiles

The Bluetooth Special Interest Group has developed a number of usage models that describe the primary Bluetooth applications and the intended devices (Bluetooth Special Interest Group 2001c). The usage models are:

- The Internet bridge. This scenario gives laptops access to the Internet through a mobile terminal or a wired connection.
- The interactive conference. In meetings and conferences, documents can be transferred instantly between selected participants, and electronic business cards can be exchanged automatically.
- The headset. Connects a wireless headset to the mobile phone, laptop or any wired connection in order to keep the hands free.
- The automatic synchronizer. Automatic synchronization of information in the users' desktop, laptop, PDA and mobile phone.
- The three-in-one phone. With Bluetooth the mobile phone can be used in different scenarios: as a cordless phone (fixed line charge), as a mobile phone (cellular charge), or as a walkie-talkie (no charge).

While usage models describe the primary Bluetooth applications and the intended devices, a profile specifies how interoperability is achieved (Bluetooth Special Interest Group 2001b). A profile defines a selection of messages and procedures as basis for interoperability. Devices must have the same profiles implemented to be able to communicate with each other, e.g. to set up a voice connection both devices must have the cordless telephony profile implemented. A profile is dependent on another profile if it reuses part of that profile. The Bluetooth profile structure and the dependencies are depicted in Figure 2.5.

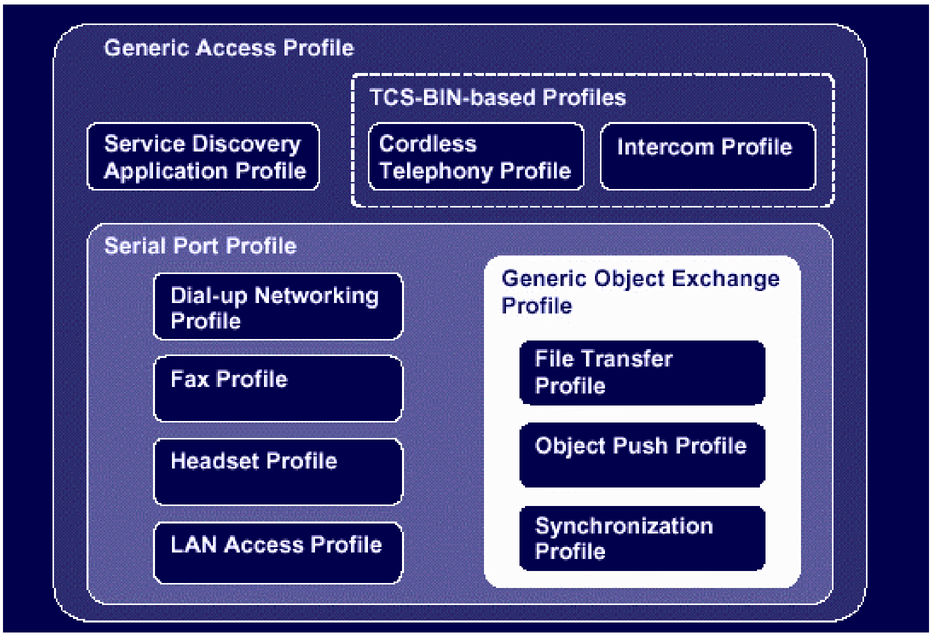

Figure 2.5 The Bluetooth profile structure.

2.2 Bluetooth research

Bluetooth is a quite new technology and it is an area of intensive research. Reflecting this is the fact that during the time period of this project the number of research articles about Bluetooth in the IEEE Xplore database approximately doubled. There are, however, already quite extensive research done on the performance of Bluetooth piconets and scatternets, some of which deal with issues relevant to this project.

Most research papers on Bluetooth concern polling algorithms, segmentation and reassembly (SAR) policies, TCP over Bluetooth, the ability of Bluetooth piconets to handle different user scenarios and traffic types, and finally the interference between Bluetooth and IEEE802.11. For the scenarios of this project, where Bluetooth is intended as an access technology to UMTS, all of these issues are important. However, most research aims at optimising details of the Bluetooth technology, while little research is done on practical networking issues. Most research only concerns communication within piconets or scatternets. A couple of research papers consider the case of using Bluetooth as an access method to the wired network for Internet access. To the best of the author's knowledge this project is the first that deals with the interaction between Bluetooth and UMTS.

2.2.1 Bluetooth-based Internet access point

Prior research closest to this project is that of (Kim et al 2001). They address the performance of a Bluetooth-based Internet access point. The access point has a single Bluetooth radio unit that acts as a master and up to seven connected Bluetooth equipped notebooks (acting as slaves). Simulations are made for web traffic, while varying the values of various parameters including users' think time, message lengths, and the Internet access delay. It is shown that a Bluetooth-based access point is feasible in terms of both throughput and delay, providing throughput from tens of kbps to hundreds of kbps depending on the number of slaves. For heavy load the throughput decreases as 1/n, where n is the number of slaves. However, even for seven slaves and heavy load throughput is shown to be better than the fastest dial-up modem (56 kbps). Unfortunately, throughput is measured only for the simple RR polling scheme. As mentioned above, the RR scheme is far from optimal. It would have been interesting to compare throughput for other polling schemes as well.

Kim et al (2001) simulate three different polling algorithms; round robin (RR), weighted RR, and multi-level RR, and compare them in terms of delay. The idea behind the modified RR schemes is to give priority to slaves, which actually have data waiting at the master. In the weighted RR scheme slaves with data are polled several times successively, while slaves with no data are polled only once. In multi-level RR the slaves are divided into two classes, one consisting of slaves with download data and the other consisting of slaves with no data. Slaves in the first class are polled several times successively, prior to slaves in the second class, which are polled only once. It is shown that the modified versions of the RR scheme perform better than the simple RR scheme in terms of completion delay. Furthermore, it is shown that the multi-level RR performs better than the weighted RR. Kim et al (2001) also analyse how the performance can be improved by using multiple Bluetooth units (masters) in the access point. The conclusion is that the performance improves as the number of Bluetooth units increase up to 40. After that frequency collisions on the different channels become too frequent and the performance deteriorates.

2.2.2 Polling algorithms

The polling algorithm will significantly affect the performance of a Bluetooth piconet. There are two facts that complicate the design of efficient polling algorithms for Bluetooth piconets (Capone et al 2001).

- A master-to-slave transmission is *always* following a slave-to-master transmission.
- The master only has partial knowledge of the piconet's queue status. The master knows the status of all the master-to-slave queues, but cannot have updated information on the status of the slave-to-master queues even if explicit signalling is provided. This is because the slaves can only communicate their queue status when they are polled.

There are several papers written with proposals and analyses of different polling algorithms. Some main ideas can be identified. To achieve fairness most of the proposed algorithms are based on the RR scheme. An improvement of this scheme in order to minimize delay is to poll each slave with data several times successively. Exhaustive polling algorithms poll a slave until it has no more data to send. However, to prevent continuously transmitting or receiving slaves from capturing the channel a limit for the maximum number of successive polls is often introduced. Such polling algorithms are called partially exhaustive. A further improvement is to avoid polling slaves with no data to send or receive. This is done by somehow incorporating the "activeness" of the different slaves. Depending on how much the slaves send or receive they are assigned different weights, and hence different priorities.

All the proposals of (Kim et al 2001), (Das et al 2001), (Johansson et al 1999), (Bruno et al 2001), and (Capone et al 2001) follow the ideas above. Johansson et al (1999), for example, propose a polling algorithm called fair exhaustive polling (FEP) and show that it performs much better than RR. In FEP slaves are divided into two complementary sets of those in active and inactive state. A polling cycle starts with the master moving all the slaves to the active state, and then begins one of possibly several polling sub cycles. All active slaves are polled in a RR fashion during a polling sub cycle. A slave is moved from the active to the inactive state when it has no data to send *and* the master has no data to send to the specific slave. The iterative process continues until the active set is emptied. Then a new polling cycle starts. An additional parameter is added to limit the polling interval of any slave to some predetermined maximum time. This prevents continuously transmitting slaves from capturing the channel.

FEP is attractive because of its good performance and its simplicity, especially with regards to implementation. Others like Das et al (2001) and Kalia et al (1999, 2000) have proposed more sophisticated schemes. Their idea is to send information to the master about the slaves' queue status to allow optimal polling decisions. There are bits in the packet headers available for this, but these schemes are far more complicated and thus more expensive to implement than FEP. A basic problem with the research on polling algorithms is that they are all compared to the simple RR scheme. No comparisons are made between the many new proposals and it is therefore hard to draw conclusions about which algorithm is the best. This is further complicated by the fact that different polling algorithms are best for different traffic types.

2.2.3 Packet types and SAR policies

Many have studied the impact on performance of different packet types. Elg et al (1999) have studied the performance of data traffic over Bluetooth piconets. One of their conclusions is that Bluetooth has very good support for handling a mix of voice traffic and data traffic. Others like Das et al (2001) have noticed the same, but have also concluded that the presence of voice conversations, i.e. SCO links, significantly degrades the data traffic performance. The reason is that in the presence of voice conversations only 3-slot packets or single slot packets can be used. In general, the performance of data traffic increases when allowing data packets to be sent in multi-slots (Johansson et al 1999). The explanation is that the transmission overhead decreases when multi-slot packets are used and that more time can be used for transmission when frequency hops only occur once over multiple slots instead of in every slot.[2] To see why the packet type has such significant impact on performance note that the payload size per slot for DH5 packets (339 bytes in 5 slots or 67.8 bytes per slot) is significantly larger than that of DH3 packets (183 bytes in 3 slots or 61 bytes per slot) and DH1 packets (27 bytes per slot).

SAR policies are implemented by the L2CAP to support a larger maximum transmission unit (MTU) than the largest baseband packet. The SAR policy affects performance since it determines which packet type to use. The choice of SAR policy is mainly a trade-off

[2] The maximum packet lengths are approximately 250μs shorter than the respective multiple of the slot duration to allow for synthesizer re-tuning. For a DH1 packet 366 μs is available for transmission, for a DH3 packet 1622 μs, and for a DH5 packet 2870 μs (Haartsen et al 2000).

between link utilization and delay. For example: a 549 byte packet can be segmented in several ways. To maximize link utilization the appropriate segmentation is a 5-slot packet, a 3-slot packet, and a 1-slot packet. However, to minimize delay the best segmentation is two 5-slot packets.

Das et al (2001) have performed an extensive study of several schemes designed to improve the performance of asynchronous data traffic over a piconet with many active slaves. They study the impact of two different SAR policies. Their conclusion is that an SAR policy called Optimum Slot Utilization (OSU) that segments the higher layer packets into as few, but hence large, baseband packets as possible is optimal in terms of throughput, delay as well as link utilization. Since the evaluation is done only for RR this conclusion may not be valid for different polling algorithms and different traffic characteristics. However, the simplicity of OSU makes it attractive as opposed to many of the much more complicated SAR policies suggested by others.

2.2.4 TCP over Bluetooth

Carrying TCP/IP traffic within Bluetooth piconets does not cause any serious problems. Johansson et el (2000a, 2000b) have shown by simulations of Bluetooth piconets that TCP throughput can be kept high and end-to-end delays reasonably low. This is mainly due to the effective fast-ARQ scheme used in Bluetooth, where retransmissions are made immediately after a packet error and hence no TCP timeouts occur. Das et al (2001) have investigated the performance of different versions of TCP (Tahoe, Reno, New Reno, and Sack) over Bluetooth. Their conclusion is that the difference in throughput is insignificant. This is due to the fact that the fast-ARQ scheme at the Bluetooth link layer eliminates the need for modifications of the error recovery at the transport layer.

The work of Bruno et al (2001) has similarities to this project. They consider a scenario where a Bluetooth master is used as a wireless access point to the Internet. They simulate seven slaves with a mix of TCP and UDP traffic and study the role that the Maximum Segment Size (MSS) of a TCP connection has in determining the TCP throughput when using TCP over Bluetooth. The conclusion is that the selection of MSS value for a TCP connection affects the Bluetooth performance. This is due to the fact that each TCP packet has to be segmented into fixed sized lower layer packets. However, this is not something specific for Bluetooth, but holds for all time slotted radio links.

2.2.5 Interference between Bluetooth and IEEE802.11

Several papers concern the interference between Bluetooth and IEEE802.11 WLANs. Agrawal et al (2000) have measured the performance degradation for both Bluetooth and IEEE802.11 DS WLAN when they are used together in an embedded system, which could be considered a worst-case scenario. Their measurements show that the reduction in Bluetooth throughput due to WLAN interference can be significant (much more than 50% in some cases). Also, it is shown that multi-slot packets fare much worse than 1-slot packets. This is attributed to the fact that the probability is greater that a 1-slot packet can be transmitted during the WLAN transmitter idle time. The measurements also show that the packet error ratio due to WLAN interference is not very different for medium (DM) and high (DH) rate packets. That is a sign that in the presence of WLAN interference the FEC in Bluetooth is not capable of providing enough redundancy. The explanation is that the FEC only can handle single bit errors. Interference, on the other hand, occurs in bursts and since there is no interleaving in Bluetooth this will result in burst errors in the packets.

2.2.6 Issues of relevance to this project

Having discussed earlier research on Bluetooth we can identify a few issues that will be of importance to this project and that will have to be considered:

- The number of users sharing a Bluetooth/UMTS gateway.
- The traffic characteristics of the users.
- The polling algorithm of the master.
- The SAR policy.
- The packet type used for data communication.
- The impact of SCO links (voice) on data traffic performance.
- The interaction between TCP and the lower layers of the Bluetooth protocol stack.
- The interference from other Bluetooth units when several piconets are active.
- The interference from non Bluetooth users (mainly IEEE802.11) in the ISM band.

3 UMTS

This chapter gives a brief introduction to selected areas of UMTS that are of importance to this project.

3.1 System overview

The Universal Mobile Telecommunications System, UMTS, is the name of the European 3rd generation mobile system standard. It differs from earlier generations of mobile systems mainly in that it has been designed to provide all kinds of services. Besides speech and SMS these include email, video telephony, multimedia conferencing, and more.

UMTS is based on a developed GSM core network and uses wideband code division multiple access (WCDMA) at the radio interface. The UMTS radio interface, or UMTS terrestrial radio access (UTRA), consists of two modes of operation: the frequency division duplex (FDD) mode and the time division duplex (TDD) mode. In Sweden both FDD and TDD bandwidth has been allocated, but only FDD will be used in the first phase. Each FDD band, i.e. uplink and downlink band, has been divided into four slots of 15 MHz each. Hence, in total there is 120 MHz frequency spectrum allocated for UTRA FDD as shown in Figure 3.1. In Sweden four operators have been assigned one license each for the 15 MHz uplink and downlink bands: Europolitan, Hi3G, Orange, and Tele2 (Holma et al 2000, Post och Telestyrelsen 2001).

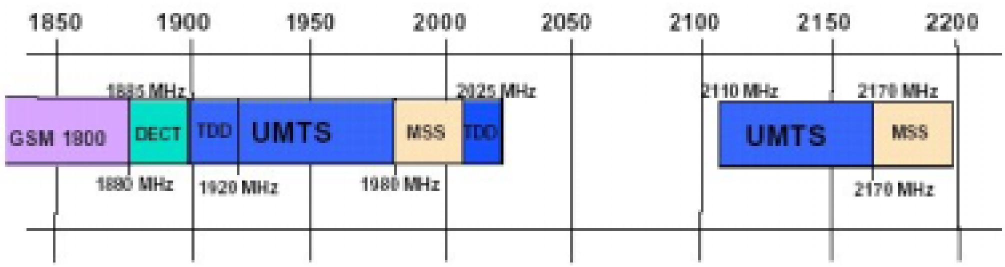

Figure 3.1 The European UMTS frequency spectrum.

3.1.1 System architecture

To get an overview of the UMTS architecture it is helpful to look at the different network elements and their interfaces. Figure 3.2 illustrates this. The network architecture can be divided into four elements at a high level: the user equipment (UE, i.e. mobile terminal in UMTS terminology), the UMTS terrestrial radio access network (UTRAN), the UMTS core network (UMTS CN), and beyond that other external networks. The first three elements constitute the UMTS public land mobile network (UMTS PLMN).

The UE consists of a radio terminal and a UMTS subscriber identity module (USIM), a smart card used for identification, authentication and encryption. The U_u interface connects the UE to the UTRAN. The UTRAN consists of radio network subsystems (RNS) interconnected to each other via the I_{ur} interface. Each RNS consists of a radio network controller (RNC) and several Node-Bs, i.e. base stations, which are associated to the RNC over the I_{ub} interface. The RNC controls the radio resources for the Node-Bs connected to it and provides services to the CN. The Node-B converts the data flow between the U_u and the I_{ub} interfaces, and participates in radio resource management.

The I_u interface separates the UTRAN and the UMTS CN. Note that there are two different I_u interfaces, one for the packet switched domain and one for the circuit switched. The CN transfers traffic to external networks and handles higher-level intelligence and functionality of the system. In order to reuse 2^{nd} generation systems and make roaming possible between GSM and UMTS, the UMTS CN is based on the CN for GSM/GPRS.

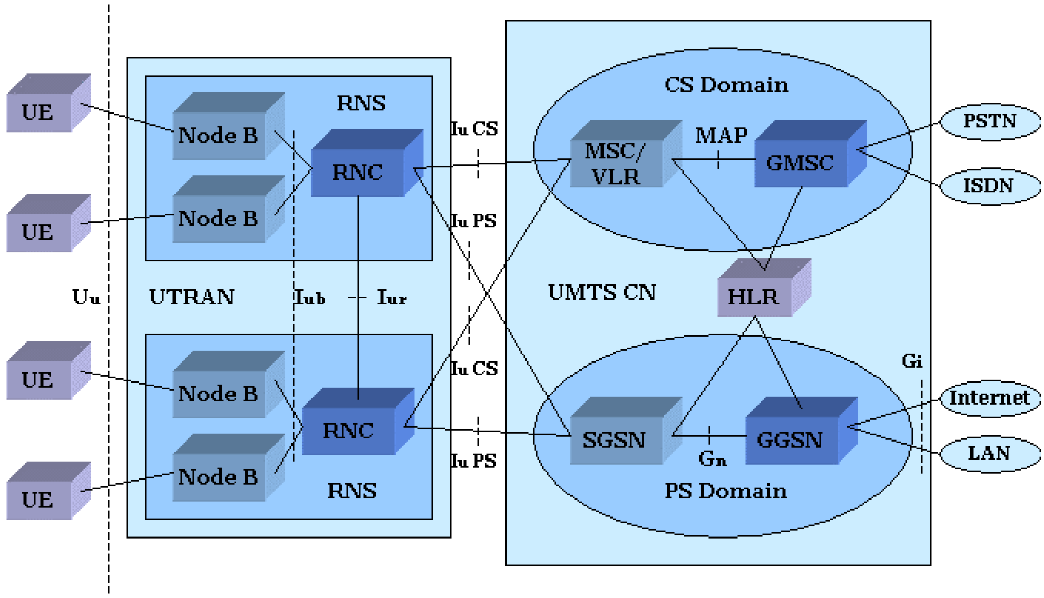

Figure 3.2 UMTS network architecture.

The UMTS core network is divided into a circuit switched domain and a packet switched domain and it has five main elements according to Figure 3.2. The home location register (HLR) is a database located in the user's home system and stores the user's service profile, i.e. what he is allowed to do. The main entities in the GSM/UMTS circuit switched core network are the mobile services switching centre/visitor location register (MSC/VLR) and the gateway MSC (GMSC). The MSC switches circuits and the VLR holds information about the location of the UE as well as a copy of the user's service profile. The GMSC is the access point to external circuit switched networks such as the PSTN.

The main entities in the GSM/UMTS packet switched core network domain are the serving GPRS support node (SGSN) and the gateway GPRS support node (GGSN), connected via the G_n interface. These two packet switched entities are the counterparts of the MSC/VLR and the GMSC in the circuit switched domain. In addition to the five elements mentioned all other higher-level functionality such as billing systems are located in, or at least in connection to, the CN (Holma et al 2000).

3.1.2 The UTRA FDD radio interface

UTRA FDD is based on DS-CDMA. Hence, the basic physical resource is the code-frequency-time plane. The symbols are QPSK modulated and spread with a chip rate of 3.84 Mcps (mega chips per second). This yields a bandwidth of roughly 5 MHz for the modulated signal. The bit stream is divided into 10 ms radio frames, which in turn are subdivided into 15 slots of 2560 chips each.

Two different types of code sequences are used in UTRA FDD to spread the signals. They are referred to as channelisation codes and scrambling codes (see Figure 3.3). The channelisation codes are so called orthogonal variable spreading factor (OSVF) codes with a spreading factor (SF) that varies from 4 to 512 chips. The use of OVSF codes allows the spreading factor to be changed and the orthogonality between different spreading codes of different lengths to be maintained. Orthogonality is necessary in order to reduce interference between users. The channelisation codes are used in the downlink to separate connections to different users within one cell. In the uplink they are used to separate data and control channels from the same terminal.

The scrambling codes are Gold codes truncated to 38400 chips. In the downlink scrambling codes are used to separate different cells and sectors from each other. Since the same scrambling code is used for the entire cell (sector) in the downlink, separation between users is in the hands of the channelisation code. In the uplink scrambling codes are used to separate transmission from different sources.

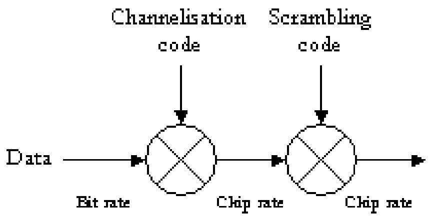

Figure 3.3 Spreading and scrambling of data in UMTS.

UTRA FDD shall support services with different requirements in terms of data rate. The most important instrument is the variable spreading factor. However, the spreading factor can only be increased or decreased by multiples of two. Therefore, an advanced rate matching and multiplexing scheme is used in order to achieve a low granularity of data rates (Holma et al 2000).

3.1.3 Error protection and retransmission

The radio interface provides three different means for error protection and detection: *checksums*, *interleaving* and *channel coding*. Error detection may be provided on the transport blocks through CRC of 8 to 24 bits. The interleaving is based on the 10 ms frame structure. Depending on the characteristics of the desired service the interleaving depth can be varied. The interleaving depth is determined by the transmission time interval (TTI), which can be set to 10, 20, 40 or 80 ms. The adaptive multi-rate (AMR) speech codec is based on 20 ms frames. For packet data and other services, which rely on retransmission when data is corrupted, interleaving of 10ms or 20 ms will be used. The longer interleaving periods are intended for real time services that do not use retransmission. FEC is provided by convolutional codes with rate 1/2 and turbo codes with rate 1/2 or 1/3.

In UMTS the radio link control (RLC) protocol provides segmentation and retransmission services. Each RLC instance can operate in one of three modes: transparent mode, unacknowledged mode or acknowledged mode. In *transparent mode* no protocol overhead is added to higher layer data. Higher layer data is not segmented and received erroneous data can only be discarded or marked erroneous. An example of user services that could utilise transparent mode is streaming media. In *unacknowledged mode* retransmission is not used and data delivery is not guaranteed. Depending on the configuration, received erroneous data is marked or discarded but is not retransmitted. Segmentation and concatenation of higher layer data is supported in this mode. Unacknowledged mode could, for example, be used by voice over IP or cell broadcast service. An automatic repeat request (ARQ) mechanism is used in *acknowledged mode* for error correction. An indication of the status of the received data is piggybacked to the sender. The quality vs. delay performance can be controlled through configuration of the maximum number of retransmissions allowed. If data cannot be delivered correctly within the maximum number of retransmissions the data is discarded and higher layers are notified. The RLC can be configured for both in-sequence delivery and out-of-sequence delivery. With in-sequence delivery the order of higher layer packet data units (PDUs) is maintained, whereas out-of-sequence delivery forwards higher layer PDUs as soon as they are completely received. The acknowledged mode is the normal RLC mode for packet-based services, such as Internet browsing or file download (Holma et al 2000).

3.1.4 Transport channels for packet data

In UTRA the data generated at higher layers is carried over the air with transport channels. After channel coding and interleaving the transport channels are mapped onto physical channels. The radio resources of the physical layer are divided into a set of physical channels defined by the physical-layer parameters frequency, time, scrambling code and channelisation code. The mapping between transport channels and physical channels is almost one to one. There are, however, a number of physical channels that are used only internally by the physical layer for transfer of control information.

The transport channels can be grouped into dedicated channels and common channels. The main difference between them is that a common channel is a resource divided between all or a group of users in a cell, whereas a dedicated channel is reserved for a single user. The packet scheduler in the RNC selects the transport channel to be used for packet data.

The only dedicated transport channel is the dedicated channel, for which the term DCH is used. The dedicated transport channel carries both user data as well as higher layer control information. It is characterized by fast power control, fast data rate change on a frame-by-frame basis and support for soft handovers. In the first phase of UMTS there are two common transport channels: the forward access channel (FACH) in the downlink and the random access channel (RACH) in the uplink. They are used to carry control information, but also to carry small amounts of user data, e.g. a packet burst from a web page download (Holma et al 2000).

3.1.5 UE operational modes

The two basic operational modes of a UE are *idle mode* and *connected mode*. The connected mode can be further divided into service states, which define what kind of physical channel a UE is using. Figure 3.4 shows the service states in the connected mode. It also shows the transitions between idle mode and connected mode and the possible transitions within the connected mode.

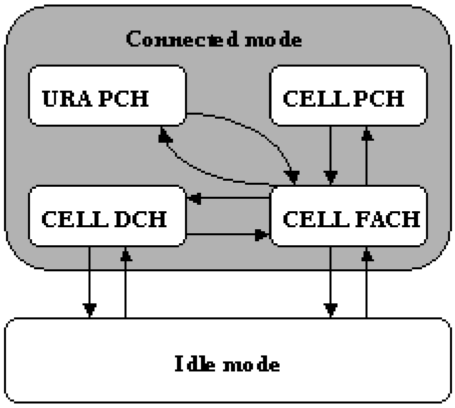

Figure 3.4 UE modes and the different service states.

In idle mode the UTRAN has no information about the individual UEs and can only address all UEs in a cell. The Cell_DCH state is a "normal" connected state of GSM type, where a dedicated physical channel is allocated to the UE and the UE is known on a cell level by the UTRAN. Dedicated channels can have high data rates and transfer large amounts of user data, but they also take up a lot of radio capacity in a cell. UMTS is supposed to support new types of services, such as web browsing, for which periods of complete inactivity are interleaved with traffic bursts. The Cell_FACH, Cell_PCH, and URA_PCH states have been devised for decreasing level of activity. In these states radio and battery capacity is saved at the cost of increasing delays. In the Cell_FACH state no dedicated physical channel is allocated to the UE, but RACH and FACH channels are used for both signalling messages and small amounts of user data. The RACH and FACH have an advantage over the DCH in that their connection delay is very low, i.e. they can send small amounts of data immediately. However, as they can only transfer small amounts of data the UE services state is often changed to Cell_DCH during a web page download. As shown in (Bergström 2001) the delay associated with the physical channel reconfigurations and transport channel reconfigurations affects the delay and throughput for web traffic.

In the Cell_PCH state the UE is known on a cell level and can be reached only via the paging channel (PCH). The Cell_URA state is the very similar to the Cell_PCH state except that the UE is known by the UTRAN only on the URA level, which is the level above the cell level in the network area hierarchy.

3.1.6 UMTS QoS

The UMTS standard supports both voice calls and high-speed packet data transfer. It differs significantly from current GPRS in that it has QoS capabilities. Current GPRS only support best effort traffic whereas UMTS has been designed to match user application requirements to network capabilities. This means that different applications may be treated differently by UMTS in terms of QoS.

To enable QoS requires co-operation of all network layers from top to bottom, as well as every network element from end to end. Network services are considered end-to-end, i.e. from one terminal equipment (TE) to another TE. To realize a certain network QoS for end-to-end services a bearer service with clearly defined characteristics and functionality has to set up from the source to the destination of a service. A bearer service includes all aspects to enable the provision of a contracted QoS. The UMTS bearer service architecture is depicted in Figure 3.5. Each bearer service on a specific layer offers its individual service using services provided by the layers below (3GPP 2001a).

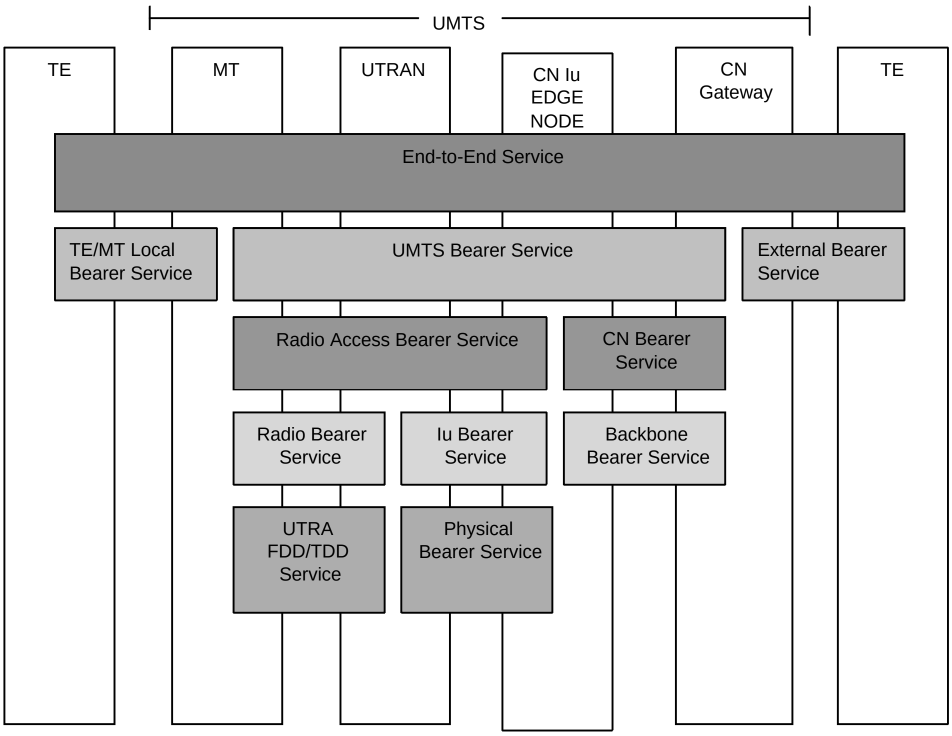

Figure 3.5 The UMTS QoS architecture levels.

The bearer services have been standardised in 3GPP in order to be able to support transport of many different types of applications. Four end-user traffic classes for UMTS have been defined to support applications that require different QoS:

- Conversational class: Low delay is required as well as low delay variation. Typical services are telephony and video conferencing.
- Streaming class: Transfer delay does not have to be low. Delay variation should be low. Typical services are music and video on demand.
- Interactive class: Low bit error rate required. Low round trip delay. Typical services are web browsing and remote surveillance.
- Background class: No special requirement on delay. Low bit error rate is required. Typical services are email delivery, SMS, and file transfer.

UMTS bearer services are specified by a set of QoS attributes. Table 3.1 shows some of the UMTS bearer service attributes and their respective value for each of the four different traffic classes. UMTS will according to the standards be able to support packet switched downlink (DL) user data up to 2 Mbit/s, but in the first phase of UMTS a DL data rate of 384 kbps is probably more realistic. In the uplink (UL) 64 kbps packet switched user data is likely to be supported in the first phase.

	Conversational class	**Streaming class**	**Interactive class**	**Background class**
Maximum bit rate (kbps)	<2048	<2048	< 2048 – overhead	< 2048 – overhead
Delivery order	Yes/No	Yes/No	Yes/No	Yes/No
Maximum SDU size (octets)	<=1500 or 1502	<=1500 or 1502	<=1500 or 1502	<=1500 or 1502
SDU error rate	10^{-2}, $7*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-5}	10^{-1}, 10^{-2}, $7*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-6}	10^{-3}, 10^{-4}, 10^{-6}
Residual bit error rate	$5*10^{-2}$, 10^{-2}, $5*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-6}	$5*10^{-2}$, 10^{-2}, $5*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}	$4*10^{-3}$, 10^{-5}, $6*10^{-8}$	$4*10^{-3}$, 10^{-5}, $6*10^{-8}$
Delivery of erroneous SDUs	Yes/No/-	Yes/No/-	Yes/No/-	Yes/No/-
Transfer delay (ms)	100 (80) – maximum value	250 – maximum value		
Guaranteed bit rate	<2048	<2048		

Table 3.1 QoS attributes for the UMTS bearer service for the four different traffic classes. Numbers within parentheses refer to attributes for UMTS radio access bearer service.

4 Simulator implementation

This chapter explains how Bluetooth and UMTS have been modelled. It also explains how the simulator is implemented, what functionality is included, and what its limitations are.

4.1 Simulator overview

4.1.1 Simulation environment

The performance of a Bluetooth/UMTS gateway has been evaluated by means of computer simulations. Since no equipment for UMTS was available this was the only feasible way to evaluate compound Bluetooth/UMTS links. The goal with the simulations was to find out how performance, in terms of throughput and delay, depend on variables such as the number of users simultaneously sharing a gateway, the Bluetooth packet type, and the UMTS radio access bearer (RAB).[3]

The simulation program was implemented in C++ as an event driven simulator. The reason for implementing an own simulator is simply that, to the best of the author's knowledge, no simulator exists that allows simulations of Bluetooth *and* UMTS in a reasonably easy way. IBM has developed an open source simulator for simulating Bluetooth piconets, called BlueHoc, which uses Network Simulator 2.[4] However, it was concluded that neither Network Simulator 2 nor BlueHoc were suited for the simulation purposes of this project and that it would require too much time to modify them. Instead it was decided to develop an own simulator. This allowed much more flexibility in the design of the simulator and also full control of all simulator details. However, due to the time constraints of this project it also meant that many of the advanced features of a simulator like Network Simulator 2 could not be implemented.

C++ was as chosen as the programming language for a number of reasons. First and foremost it is fast and this is important since one often has to simulate long time periods in order to get good statistics and reliable results. Furthermore, the C++ programming environment is available for free under UNIX. Finally, C++ is a language that the author has previous experience with and feel comfortable with.

4.1.2 Simulator architecture

The simulator consists of three parts: the Bluetooth part, the UMTS part, and the network part as depicted in Figure 4.1. The Bluetooth part models the Bluetooth radio link, the UMTS part models the UTRAN and the CN, and the network part models the end network. Both the Bluetooth radio link and the UTRAN represent dynamic links, i.e. the packet delays incurred over these links depend on radio channel conditions and the

[3] A UMTS RAB is set up between the UE and the CN I_u edge node and is an interface towards higher layer services. It is specified by its QoS attributes and is set up with a specific data rate in the uplink and downlink.

[4] See http://www.isi.edu/nsnam/ns for NS-2 and http://www-124.ibm.com/developerworks/opensource/bluehoc for BlueHoc.

link traffic load. Both the Bluetooth radio link and the UTRAN have been modelled in a detailed way in order to capture their dynamic behaviour. The CN and the end network, on the other hand, represent more or less static links, i.e. the packet delays incurred over these links mainly depend on the overall network traffic load. For normal traffic conditions these networks can be modelled in a static way as a simple packet delay and this is the approach taken.

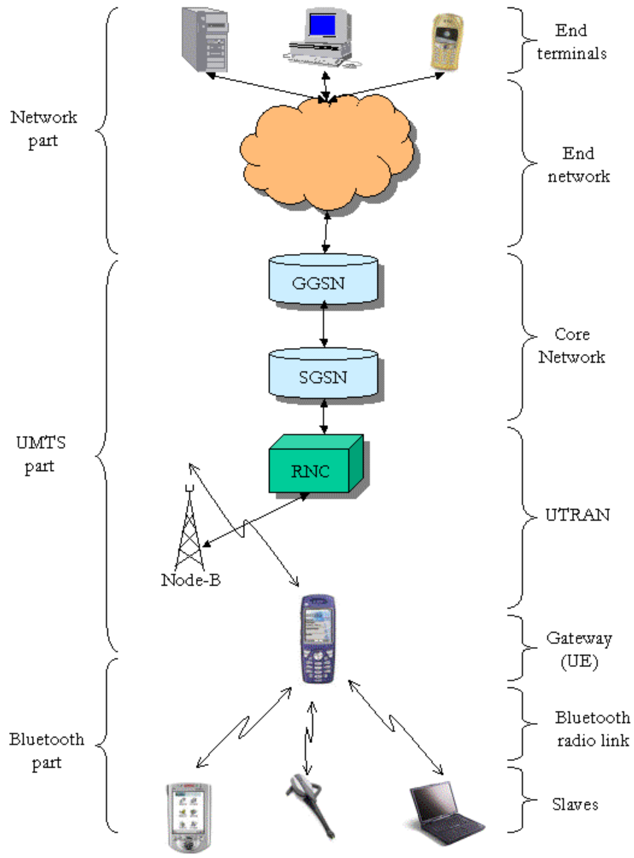

Figure 4.1 Simulator architecture.

The simulator consists of four main types of objects that are implemented in C++: Bluetooth slaves, a Bluetooth/UMTS gateway (acting as Bluetooth master and UMTS UE), an RNC, and end terminals, according to Figure 4.1. The slaves and the end terminals represent the communicating peers. The slaves represent whatever access terminals are used, e.g. PDAs or laptops. The end terminals represent the remote peers and could for example be web servers or PCs. The Node-B had not been implemented as an object, but instead the RNC object includes the Node-B functionality. As mentioned above, the CN and the end network are modelled as packet delays and therefore they are not implemented as objects.

4.2 The Bluetooth part

4.2.1 Architecture and protocols

The Bluetooth part of the simulator consists of the slaves and the gateway, which is acting as the master. The gateway includes both a Bluetooth and a UMTS part. It communicates with the slaves via Bluetooth. One to seven slaves can communicate with the gateway simultaneously. The gateway has several buffers. In the Bluetooth domain it has one transmit and one receive buffer for each slave it communicates with. These buffers hold Bluetooth baseband packets. All buffers have been modelled with infinite size. Each slave has two buffers at the baseband level, one for outgoing packets and one for incoming packets. These buffers have also been modelled with infinite size. Figure 4.2 shows the implemented objects and the buffer structure.

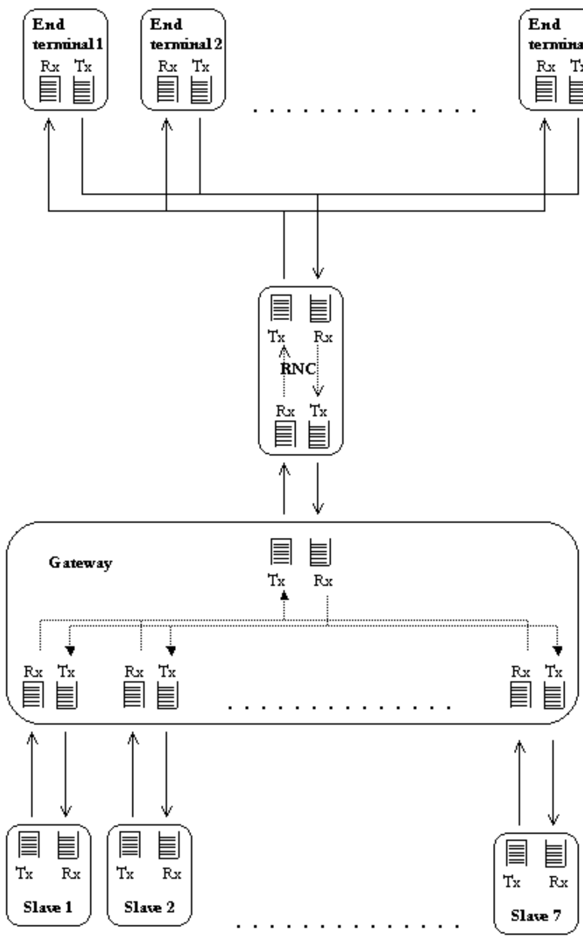

Figure 4.2 The simulator objects and the buffer structure.

The Bluetooth/UMTS gateway has been assumed to use the LAN Access Profile specified in (Bluetooth Special Interest Group 2001b). This profile covers user scenarios where one or several Bluetooth devices use a local access point for wireless connection to a LAN or another network. The use of a profile specified by the Bluetooth SIG is important in order to ensure interoperability between Bluetooth devices from different manufacturers. As long as the devices (the gateway and the slaves) have the same profile implemented they will be able to communicate with each other. The LAN Access Profile specifies a protocol stack according to Figure 4.3, and this protocol stack has been implemented in the simulator. (3GPP 2001c) also specify PPP for connecting terminal equipment (Bluetooth slaves in this case) to a mobile termination (the gateway in this case). However, should another future profile be used only the PPP and RFCOMM layers will change.[5] This will not affect the simulation results noticeably since these two layers have only been modelled by adding packet headers. Their actual protocol functionality does not affect simulations and have therefore been excluded.

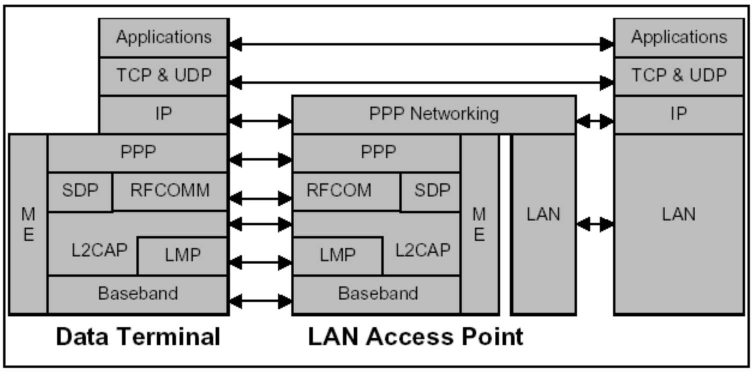

Figure 4.3 The LAN Access Profile protocol stack.
The Data Terminal corresponds to one of the slaves and the
LAN Access Point corresponds to the gateway.

4.2.2 Packet encapsulation and segmentation

In the slave units, data is generated at the application level according to traffic models that can be set individually for each slave. Application data is segmented into packets and encapsulated according to Figure 4.4. The MSS of TCP can be specified, but in the simulations a size of 1460 bytes has been used. Including a 20 bytes TCP header and a 20 bytes IP header this results in an IP datagram size of 1500 bytes, which is the standard size.[6] L2CAP has been assumed to have an MTU of 672 bytes, which is the default value according to (Bluetooth Special Interest Group 2001a). The L2CAP packets are segmented into baseband packets according the OSU policy recommended by (Das et al 2001), see section 2.2.3. The baseband packets are placed in the slave's transmit buffer. In reality, the size of the Bluetooth baseband buffers is small and therefore only small amounts of data will be sent from higher layers to the baseband level at a time. However, for the purposes of this simulator it does not matter whether packets are being queued at higher layers or at the baseband layer.

---

[5] There are discussions within the Bluetooth SIG PAN working group to replace the PPP and RFCOMM layers with a protocol called Bluetooth Network Encapsulation Protocol (BNEP) in the next version of the Bluetooth specification (Johansson et al 2001).

[6] The IP datagram standard size normally used when nothing has been negotiated is either 576 bytes or 1500 bytes. Here the latter one has been used.

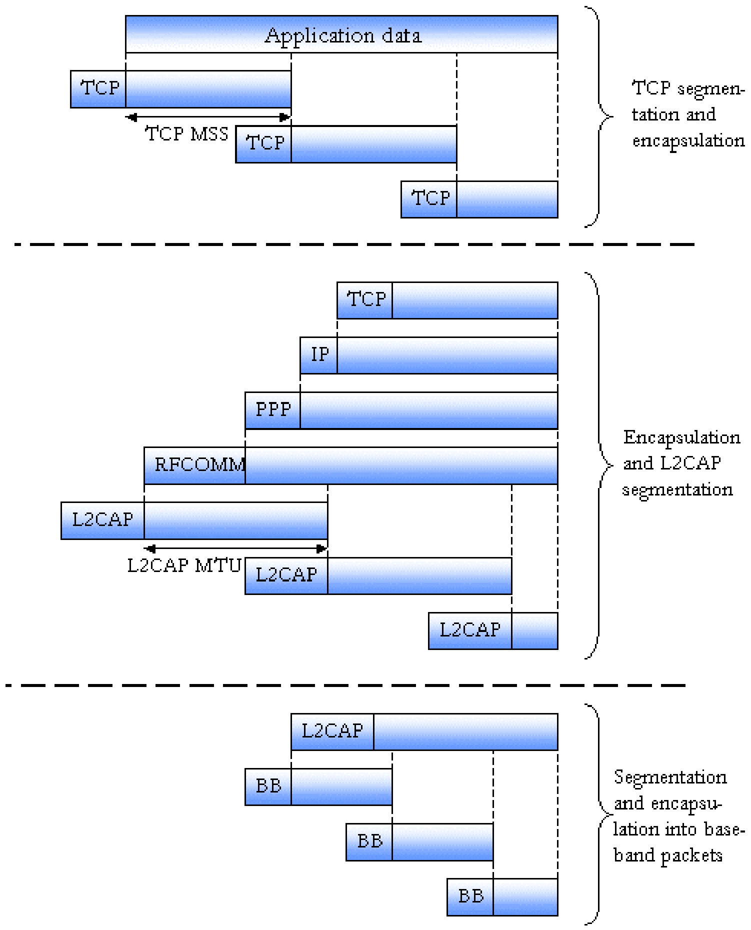

Figure 4.4 Data segmentation and encapsulation.

In the gateway, packet encapsulation and segmentation is done the same way as in the slaves. However, no data is generated in the gateway. Instead, the gateway receives data from the end terminals via the UMTS link or from the slaves via the Bluetooth links. In the gateway, packets are transferred between the UMTS and the Bluetooth domain at the IP level. When the gateway receives data from UMTS it puts the data in a UMTS receive buffer until it has received an entire IP packet. When an entire IP packet has been received it is relayed to the Bluetooth domain. There it is encapsulated and segmented into Bluetooth baseband packets in the same way IP packets are encapsulated and segmented into baseband packets in the slaves, see Figure 4.4. The baseband packets are placed in the Bluetooth transmit buffer for the destined slave (in the gateway there is one transmit buffer for each slave). As mentioned above, in reality all baseband packets will not be sent to the baseband level immediately, but the same argument as above applies here too.

Bluetooth baseband packets sent from the master and received at the slave are forwarded to higher layers where packet headers are stripped off and the lower layer packets are reassembled to higher layer packets. In the simulator, packet reassembly is not explicitly performed, instead statistics, such as delay and throughput, is calculated when a packet is received.

Bluetooth baseband packets sent from a slave and received at the gateway are forwarded to higher layers where packet headers are stripped off and they are reassembled to higher layer packets. When an entire IP packet has been reassembled it is relayed to the UMTS domain and put in the UMTS transmit buffer.

4.2.3 Bluetooth parameters

The simulator includes an error model for the Bluetooth radio link. The packet error ratio (PER) can be specified and erroneous packets are retransmitted as specified in (Bluetooth SIG 2001a). Only packets containing data are assumed to be erroneous. This is motivated by the fact that NULL packets and POLL packets are short (126 bits) and have no payload. Their information is contained in the packet header, which is always protected by a 1/3 FEC code. The packet error ratio for NULL and POLL packets should thus be much lower than the packet error ratio for packets containing data.

Two polling algorithms have been implemented: round robin and fair exhaustive polling. Round robin has been chosen for a number of reasons. It is the simplest of all polling algorithms and it serves as a good reference when comparing performance for other polling algorithms. Furthermore, the point-to-multipoint Bluetooth chips currently available on the market use the round robin polling algorithm.[7] Future Bluetooth chips will probably use more sophisticated polling algorithms, most likely exhaustive or partially exhaustive polling algorithms. It was considered important to include such a polling algorithm in the simulator in order to analyse the impact of the polling algorithm on overall performance. Fair exhaustive polling was chosen because of its simplicity and good performance.

All types of Bluetooth data packets are included in the simulator. It can be specified whether FEC should be used or not. Additionally, a parameter called slot limit can be specified to 1, 3 or 5, in order to limit the maximum length of the Bluetooth packets. As mentioned above the SAR policy called OSU, recommended by Das et al (2001), is used in the simulator.

The simulator is intended for simulating ACL links since the simulation scenarios of interest concern data traffic. However, SCO links can be included in order to analyse their impact on ACL link performance. A parameter N_{SCO}, representing the number of active SCO links, can be specified to 0, 1, or 2 (the case of 3 SCO links has been excluded since no ACL links can be set up at the same time as three SCO links). For N_{SCO} equal to 1, an additionally parameter T_{SCO}, representing the SCO packet interval measured in Bluetooth time slots can be specified to 4 or 6. T_{SCO} equal to 4 means that the SCO packets are protected by 2/3 FEC, and T_{SCO} equal to 6 means that the SCO packets are not protected by FEC. For N_{SCO} equal to 2, T_{SCO} is assumed to be 6, i.e. no FEC is used, since otherwise no ACL links can be set up. Table 4.1 summarizes the possible combinations of N_{SCO} and T_{SCO}. Note that for N_{SCO} equal to 1 and T_{SCO} equal to

[7] The author has spoken to Ericsson Microelectronics and Cambridge Silicon Radio, two of the leading Bluetooth chip manufacturers. Both of them use the round robin polling algorithm in their point-to-multipoint chips.

6, four consecutive slots are available for data on ACL links (followed by two SCO slots). This means that slot limit three is used in the downlink and slot limit one in the uplink. For N_{SCO} equal to 1 and T_{SCO} equal to 4, or for N_{SCO} equal to 2 and T_{SCO} equal to 6, only two consecutive slots are available for data traffic (followed by four SCO slots) and hence the slot limit is one in the downlink and uplink.

	T_{SCO}=4	T_{SCO}=6
N_{SCO}=1	2/3 FEC	No FEC
N_{SCO}=2	-	No FEC

Table 4.1 Possible combinations of number of SCO links and the SCO packet interval.

4.3 The UMTS part

4.3.1 Architecture

The UMTS part consists of the gateway (the UE in UMTS terminology), the RNC and the CN. The CN has been modelled at the IP level as a simple packet delay for the I_u and the G_n interfaces (see Figure 3.2 for a description of these interfaces). This is motivated by the fact that packet delays in the CN mainly depend on the overall network traffic load. For normal traffic conditions, i.e. when the CN is not congested, the IP packet delay will not vary much. The delays for the I_u and G_n interfaces were taken from simulations of Telia Mobile AB's CN done at Telia Research AB. These simulations were done in a simulator called OPNET and generated log files containing IP packet delays for the I_u and the G_n interface for interactive/background bearer class services. These log files were used to model the CN in the C++ simulator.

In order to simplify the model of the UTRAN it has been assumed that the UE is in Cell_DCH connected state, i.e. a dedicated physical channel is allocated to the UE. Furthermore, it has been assumed that the traffic streams belonging to different slaves are multiplexed onto the same UMTS radio bearer (RB) in a FIFO manner.[8] This means that there is no priority handling between the different traffic streams. The UTRAN model consists of the UE (the gateway) and the RNC. Both of these have been modelled with two buffers for IP packets, one for incoming packets and one for outgoing. IP packets from the end network arrive at the RNC after passing through the CN. After being queued in the RNC's transmit buffer, the IP packets are segmented or concatenated into radio frames. The product of the radio frame length, 10 ms, and the DL data rate gives the number of data bits in each radio frame. Depending on the IP packet length and the DL data rate each radio frame may contain a varying number of IP packets. Since IP packets arriving at the RNC may originate from different end terminals each radio frame may contain IP packets from many sources. The reason for this is that the transmit buffer in the RNC has been implemented as a simple FIFO queue. Transmission of radio frames from the RNC to the gateway occurs every DL TTI seconds. The TTI thus determines how many radio frames that are sent at a time. Radio

[8] A UMTS RB is set up between the UE and the RNC. A UMTS RB service covers all aspects of the radio interface transport such as transport channel, coding, RLC mode etc.

frames received at the gateway are reassembled to IP packets and relayed to the Bluetooth domain.

In the gateway IP packets relayed from the Bluetooth domain to the UMTS domain are placed in the gateway's UMTS transmit buffer. The IP packets are segmented or concatenated into radio frames and sent to the RNC in the same way as explained above. The only difference is that the TTI and the data rate might differ in the UL and DL. Radio frames received at the RNC are reassembled to IP packets and transferred to the CN.

4.3.2 UMTS parameters

The main parameters that can be specified are the TTI and the data rate for the UL and the DL respectively. An error model for the UMTS radio interface has been included and the block error rate (BLER) can be specified. A block is the quantity that is sent over the UMTS radio interface every TTI seconds. For implementation simplicity it is assumed that the RLC is operated in acknowledged mode with in-sequence delivery (see section 3.1.3). This means that erroneous blocks are retransmitted and the order of higher layer PDUs is preserved. To simplify implementation a radio block can only be lost once.

4.4 The network part

The network part has been modelled at the IP level. The network part consists of the end terminals and the end network. Each end terminal communicates with a specific slave. The end terminals represent the remote peers and could for example be web servers or PCs. In the end terminals data is generated at the application level according to traffic models that can be set individually for each end terminal. The application data is segmented into packets with a size equal to TCP MSS. Each packet is encapsulated with TCP and IP headers and put in the end terminal's send buffer where the IP packets are being transmitted with an interval that can be specified. IP packets received at an end terminal from the end network are reassembled into higher layer packets.

The end network represents the G_i interface, i.e. the connection between the end terminal and the GGSN (see Figure 3.2). It has been modelled as a simple IP packet delay. In the simulations this delay has been set to either 0 or 50 ms. The first value is used for studying the performance without the influence of an external network. The latter value comes from measurements made by Khaunte et al (1998) and represents the mean delay to a remote server.

4.5 Traffic models

The choice of traffic model is mainly a trade-off between implementation simplicity and how detailed the model is. For this project a web traffic model that captures the coarse behaviour of HTTP traffic was desired, but it was not considered necessary to model all the details of an individual web page. The chosen model is a single user model of mobile HTTP that is being used at Telia Research AB. In the simulator the traffic model can be specified individually for each slave and end terminal.

A web page is modelled as a burst of downlink application data. The burst represents all objects that a web page may consist of. Inline objects in the web page are thus not modelled individually, but they are lumped together into a downlink burst. A downlink burst arrives with an interarrival time that is included in the model. The interarrival time models what is sometimes called the user think time or reading time. The requests for web pages and inline objects are not modelled individually, but they are lumped together into an uplink burst. The uplink burst is modelled as a burst of uplink application data whose size is a fraction of the corresponding downlink burst size. The uplink bursts arrive at the same time as the downlink bursts.

The random number generation is based on the C library function drand48(). This function generates double precision floating point pseudo-random numbers uniformly distributed over the interval [0, 1]. For arbitrarily distributions with cumulative distribution function $F_X(x) = P(X \le x)$ the pseudo-random number X is generated according to the formula:

$$X = F_X^{-1}(u), \qquad \textbf{(1)}$$

where *u* is a random variable uniformly distributed over the interval [0, 1].

The burst interarrival time for downlink HTTP data is generated according to the bound Pareto model. If X is the random variable representing the interarrival time its cumulative distribution function is given by:

$$F_X(x) = \frac{1-(k/x)^\alpha}{1-(k/T)^\alpha}, k \le x \le T. \qquad \textbf{(2)}$$

X is confined to the interval [k, T] and α is the shape parameter.

The downlink HTTP data burst size is generated according to the lognormal distribution. A random variable X is said to have a lognormal distribution if the random variable $Y = \ln(X)$ has a normal distribution. If X is the random variable representing the burst size its cumulative distribution function is given by:

$$F_X(x) = \frac{1}{2}\left[1 + erf\left(\frac{\ln(x) - m}{s\sqrt{2}}\right)\right], x \ge 0, s > 0. \qquad \textbf{(3)}$$

The parameter values used in the simulations are listed in Table 4.2. These parameter values correspond to a mean HTTP DL burst size of 18 kB and a mean HTTP DL/UL interarrival time of 23 seconds. For more information about the two distributions see Appendix B – Random variables.

	Model	**Parameters**
HTTP DL burst size [bytes]	Log-normal	m=8.23, s=1.77
HTTP UL burst size [bytes]	Log-normal	0.2 * DL burst size
HTTP DL/UL interarrival time [s]	Bounded Pareto	α=0.75, k=5.80, T=129 s

Table 4.2 Traffic models for HTTP traffic.

4.6 Performance metrics

There are several performance metrics that are of interest when evaluating link performance. The four most important metrics are:

- Transfer delay
- Throughput
- Bit error ratio or packet error ratio
- Transfer delay variation

In this project two performance metrics have been studied: transfer delay and throughput.

Transfer delay is defined as the total delay (in seconds) incurred from the time application data is generated at the source (slave or end terminal) to when it is received at recipient (end terminal or slave). Transfer delay is thus measured end to end.

Throughput is an indication of how much data the user can receive per second. Throughput is defined as the size of the application data burst (in bits) divided by the transfer delay. With this definition throughput is actually a lower limit of how much data the user can receive per second. This definition has been used since it allows straightforward measurements in the simulations and it is a reasonably good approximation of how much data a user can receive per second. For further discussion about throughput see Appendix C – On the definition of throughput.

All performance metrics represent average values from the simulation. In order to get reliable results an outer loop has been placed around an inner loop (batch loop) in the simulation program. The idea is to simulate a very large number of packet arrivals in each batch loop. At the end of each batch loop the average values of the performance metrics are calculated. When a sufficient number of batches have been simulated these values are used to calculate the standard deviations for each performance metric. The number of batches has then been adjusted to ensure a low standard deviation and reliable results.

4.7 Simulator limitations

In the simulator important features of Bluetooth and UMTS have been modelled in a detailed way while other, less important, features have not been included at all.

TCP functionality has not been included in the simulator. The flow control is taken care of by simply adjusting the send rate of the sources. The effects of TCP congestion control algorithms have thus not been considered. In Bluetooth the ARQ scheme applied is fast enough to efficiently hide channel errors to the transport layer and this result in very few TCP timeouts and retransmissions (Johansson et el 2000a, 2000b). In UMTS the same is not true. The delay introduced by retransmissions at the RLC layer frequently cause TCP timeouts and retransmissions (Sachs et al 2001). However, this is a general problem for TCP over UMTS that has to be solved, and it has been considered beyond the scope of this project to take this effect into account.

Processing delays have not been included in the simulation model. In reality processing delays in all the nodes will affect performance. However, since processing delays only affect the arrival time of packets and do *not* affect the interarrival time between packets this simplification does not affect the simulation results noticeably.

The link manager protocol, which is responsible for link set-up and control in Bluetooth, has not been included in the simulator. LMP messages have higher priority than user data and will affect the performance for Bluetooth user data. In the simulator it has been assumed that the Bluetooth links are already set up and that they do not have to be changed during the simulation time. In reality some LMP messages will have to be sent even after the Bluetooth links are set up. However, the effect of LMP messages on user data performance should be negligible under normal circumstances.

The UMTS model assumes that the UE is in Cell_DCH connected mode all the time. This means that a dedicated physical channel is allocated to the UE. In reality the UE might be in any of the four service states mentioned in section 3.1.5. As shown in (Bergström 2001) this will affect the delay and throughput for web traffic since it takes some time to change the service state and perform physical channel reconfigurations or transport channel reconfigurations. However, the aggregate traffic through the gateway might very well be large enough for it to stay in the Cell_DCH connected mode all the time.

5 Simulations and analysis

This chapter presents a theoretical analysis and the simulation scenarios and simulation results. Furthermore, it analyses and discusses the results.

5.1 Theoretical analysis

Before turning the attention to the simulation results it is helpful to do some theoretical analysis of a Bluetooth/UMTS gateway. One of the purposes with the simulations was to find out for which parameter settings UMTS or Bluetooth is the limiting technology[9]. In addition, the objective was to find out how performance is affected by the limiting technology (or bottleneck). Whether UMTS or Bluetooth is the bottleneck can actually be estimated by some simple calculations. This provides us with insight into how UMTS and Bluetooth interact, and it also allows us to verify the simulation results. However, even if calculations can tell us *which* technology will be the limiting one only the simulation results can tell us *how* performance is affected by the limiting technology.

Focusing on the IP packets in the gateway we can analyse whether UMTS or Bluetooth is the bottleneck. This is of course dependent on parameters such as the UMTS RAB data rate, the Bluetooth polling algorithm, the number of slaves, and the Bluetooth packet type. As an example we can assume that an interactive or background, UL of 64 kbps and DL of 384 kbps, UMTS RAB is used. When a burst of downlink data is transferred from the sender to the receiver an IP packet with a size of 1500 bytes is relayed from the UMTS domain to the Bluetooth domain in the gateway every 1500*8/384000=31.25 ms (assuming a continuous stream of IP packets arriving at the gateway). The question now arises as to how long time it takes for this IP packet to be sent over the Bluetooth link. If the time required for this is more than 31.25 ms we can conclude that Bluetooth is the bottleneck, otherwise that UMTS is the bottleneck.[10]

The time required to send an IP packet over the Bluetooth link can be estimated. Using slot limit five and no FEC, a 1500 byte IP packet requires four DH5 packets and one DH3 packet with OSU segmentation. If we ignore the uplink traffic (since there is much less traffic in the uplink than in the downlink) and assume that Bluetooth is not the bottleneck, i.e. the Bluetooth buffers are emptied before the next IP packet arrives, we can get a lower limit for the time required to send an IP packet over the Bluetooth link. With round robin and N slaves simultaneously sharing the gateway the average time required, T, can be expressed as:

[9] With limiting technology or bottleneck we mean which one of Bluetooth or UMTS that is limiting in terms of data rate.

[10] The analysis can be done for the other IP packet standard size 576 bytes. However, the outcome of the analysis is not dependent on the IP packet size.

$$T=(T_{POLL}+T_{NULL})\times\frac{N}{2}+[T_{DH5}+T_{NULL}+(T_{POLL}+T_{NULL})\times(N-1)]\times 4+T_{DH3}=$$
$$\left((1+1)\times\frac{N}{2}+[5+1+(1+1)\times(N-1)]\times 4+3\right)\times 0.625= \qquad (4)$$
$$(9\times N+19)\times 0.625ms$$

The first term represents the average time it takes before the first baseband packet belonging to the IP packet can be sent from the gateway. The second term represents the time it takes to send four DH5 packets, and the third term the time required to send one DH3 packet. For Bluetooth not to be the bottleneck we require:

$$T\le 31.25ms \Leftrightarrow (9\times N+19)\times 0.625\le 31.25 \Leftrightarrow N\le\left(\frac{31.25}{0.625}-19\right)/9=3.4 \qquad (5)$$

Since T represents a lower limit for the time required, N represents an upper limit for the number of slaves that the gateway can handle simultaneously without Bluetooth being the bottleneck. With uplink traffic and the possibility of several slaves receiving data at the same time we would require N to be somewhat lower than stated above. Since we can only have an integer number of slaves our analysis tells us that for the considered case Bluetooth becomes the bottleneck for more than three slaves.

The same analysis can be done for different packet types. Table 5.1 summarizes the maximum number of slaves that can use the gateway simultaneously before Bluetooth becomes the limiting technology when round robin polling is used. Since we only can have an integer number of slaves the numbers in the table have been rounded downwards to the nearest integer.

	DH5	DM5	DH3	DM3	DH1	DM1
Max slaves	3	1	1	<1	<1	<1

Table 5.1 Maximum number of slaves that can use the gateway before Bluetooth becomes the bottleneck for RR polling.

A similar analysis can be done with the polling algorithm changed to FEP. Ignoring the uplink traffic and the possibility of several slaves receiving data at the same time the minimum average time T required for sending 4 DH5 packets and one DH3 packet with FEP is:

$$T=(T_{DH5}+T_{NULL})\times 4+T_{DH3}=[(5+1)\times 4+3]\times 0.625=16.875ms \qquad (6)$$

This assumes that it takes no time before the first baseband packet belonging to a newly arrived IP packet can be sent from the gateway. This is equivalent to the assumption that the traffic is bursty enough for only one slave to receive data at a time. A consequence of that assumption is that for FEP the time required to send an IP packet over the Bluetooth link is independent of the number of slaves sharing the gateway. Keeping in mind that one 1500 byte IP packet is relayed to the Bluetooth domain in the gateway every 31.25 ms we conclude that UMTS is the bottleneck. Again, the analysis can be done for different packet types. Table 5.2 summarizes the time required to send a 1500 byte IP packet over the Bluetooth link for different packet types when FEP is used. These numbers represent lower limits and should be adjusted upwards when taking uplink traffic and the possibility of several slaves receiving data simultaneously into account. Since an IP packet is relayed from the UMTS domain to the Bluetooth domain in the gateway every 31.25 ms we conclude that UMTS is the limiting technology for DH5, DM5, and DH3. For all other packet types Bluetooth is the limiting technology. Note that for FEP, unlike RR, this result does not depend on the number of slaves sharing the gateway.

	DH5	DM5	DH3	DM3	DH1	DM1
Time [ms]	16.875	25.625	21.875	31.875	69.375	110.625

Table 5.2 The minimum time required for sending a 1500 byte IP packet over the Bluetooth link when FEP is used.

The analysis above can be done for UMTS RABs with different DL data rates. Table 5.3 and Table 5.4 summarize the results of such an analysis for fair exhaustive polling and round robin.[11] The tables show when Bluetooth or UMTS is the limiting technology.

	DH5	DM5	DH3	DM3	DH1	DM1
384 kbps	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth
256 kbps	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
128 kbps	UMTS	UMTS	UMTS	UMTS	UMTS	Bluetooth
64 kbps	UMTS	UMTS	UMTS	UMTS	UMTS	UMTS

Table 5.3 A summary of when UMTS or Bluetooth is the limiting technology for fair exhaustive polling.

[11] Again, the results are not dependent on the IP packet size.

UMTS DL data rate 384 kbps

	DH5	DM5	DH3	DM3	DH1	DM1
1 slave	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth
2 slaves	UMTS	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
3 slaves	UMTS	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
4 slaves	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
5 slaves	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
6 slaves	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
7 slaves	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth

UMTS DL data rate 256 kbps

	DH5	DM5	DH3	DM3	DH1	DM1
1 slave	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
2 slaves	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth
3 slaves	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth
4 slaves	UMTS	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
5 slaves	UMTS	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
6 slaves	UMTS	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth
7 slaves	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth	Bluetooth

UMTS DL data rate 128 kbps

	DH5	DM5	DH3	DM3	DH1	DM1
1 slave	UMTS	UMTS	UMTS	UMTS	UMTS	Bluetooth
2 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
3 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
4 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
5 slaves	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth
6 slaves	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth
7 slaves	UMTS	UMTS	UMTS	Bluetooth	Bluetooth	Bluetooth

UMTS DL data rate 64 kbps

	DH5	DM5	DH3	DM3	DH1	DM1
1 slave	UMTS	UMTS	UMTS	UMTS	UMTS	UMTS
2 slaves	UMTS	UMTS	UMTS	UMTS	UMTS	Bluetooth
3 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
4 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
5 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
6 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth
7 slaves	UMTS	UMTS	UMTS	UMTS	Bluetooth	Bluetooth

Table 5.4 A summary of when UMTS or Bluetooth is the limiting technology for round robin polling.

5.2 Simulation results and analysis

5.2.1 About the results

The figures in this section show throughput versus the number of slaves sharing the gateway. As mentioned in section 4.6 the transfer delay has been measured as well. However, the graphs including transfer delay provide little additional information to the graphs displaying throughput. The latter ones are also easier to interpret and are therefore the only graphs presented here.

In each figure the Bluetooth parameter setting for the simulation scenario is specified. In the figures the different curves are labelled XXX-DYZ, where XXX indicates the polling algorithm used (RR or FEP), Y indicates whether FEC is used or not (H means that FEC is not used, M that FEC is used), and Z indicates the slot limit (1, 3 or 5). FEP-DM3, for example, means that the corresponding curve shows the simulation results for Fair Exhaustive Polling where the Bluetooth packets are maximum three slots long and protected by FEC. The packet error ratio (PER) for Bluetooth has been set to 0.01 in the simulation scenarios presented in this section. This is believed to be a realistic value for normal Bluetooth radio channel conditions.

At the top of each figure the UMTS settings are also specified. In the simulation scenarios presented in this section the TTI has been set to 20 ms according to (3GPP 2001c). The block error rate (BLER) has been set to 0.02 since this is believed to be a realistic value. In the simulation scenarios presented in this section the UMTS RAB data rate is 64 kbps in the uplink and 384, 256 or 128 kbps in the downlink.

5.2.2 Concurrent web sessions

This scenario considers one to seven slaves that share the gateway and run web sessions concurrently. The web traffic has been modelled individually for each slave according to the single user traffic model described in section 4.5. UMTS RABs with different downlink data rates, specified in (3GPP 2001c), have been considered and simulations are done for both fair exhaustive polling and round robin as well as for different Bluetooth packet types.

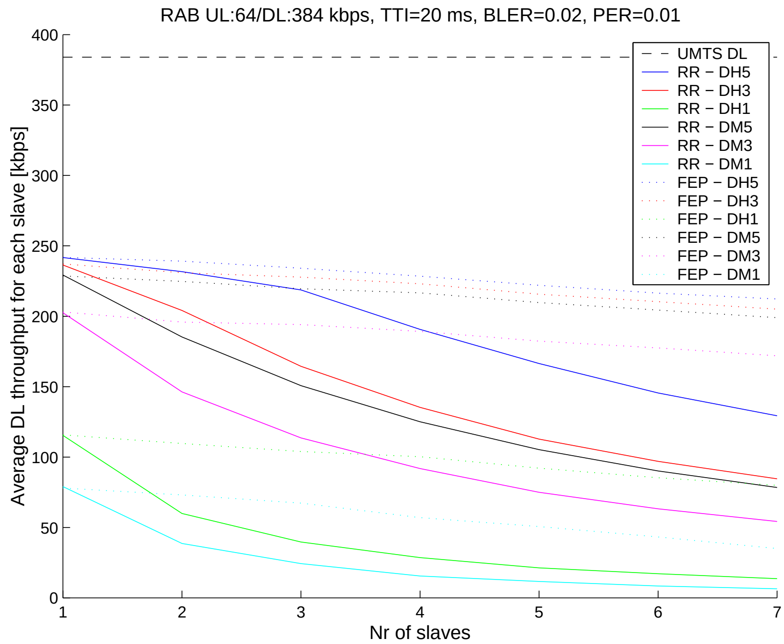

Figure 5.1 Downlink throughput versus number of slaves for UL:64/DL:384 kbps RAB.

Figure 5.1 shows the simulation results for a PS RAB, interactive or background class, with data rate UL:64/DL:384 kbps. This will probably be the highest available data rate in the first phase of UMTS. From the figure a number of important conclusions can be drawn.

It is obvious that when more than one slave is communicating with the gateway the polling algorithm used by the gateway has a crucial impact on performance. For RR throughput decreases rapidly as the number of slaves sharing the gateway increases. For FEP, on the other hand, throughput is hardly affected at all by the number of slaves sharing the gateway. Only a slow, linear degradation can be noticed.

We also notice that the Bluetooth packet type, determined by the slot limit and whether FEC is used or not, affects performance significantly. For slot limit one, throughput decreases considerably compared to slot limit five. However, even for slot limit five we see that throughput is below the UMTS DL data rate. This is partly due to overhead introduced by segmentation and encapsulation in different protocol layers, but mostly due to how throughput has been measured and how the web traffic has been modelled (for further information see Appendix C – On the definition of throughput). Overall, we see that a Bluetooth/UMTS gateway is able to handle web traffic well, even with several users sharing the same gateway. Only for slot limit one and several users throughput falls below what can be considered acceptable for web browsing with UMTS.

To isolate the effect that the Bluetooth packet type has on throughput we can look at the FEP curves where throughput is almost independent of the number of slaves sharing the gateway. Changing the packet type between DH5, DM5, and DH3 has no appreciable effect on throughput. This means that the Bluetooth link is not fully utilized for any of these packet types and that UMTS is the bottleneck. Changing to any of the other packet types makes Bluetooth the bottleneck and throughput decreases considerably. Comparing with Table 5.3 we see that this is what our analysis predicted. This behaviour leads us to the insight that when FEP is used, there is an optimal mapping between the Bluetooth packet type and the UMTS DL data rate.[12] For DH5, DM5, and DH3 the optimal mapping is a UMTS DL data rate of 384 kbps. This maximizes throughput and fully utilizes the available UMTS DL data rate. However, for the other packet types Bluetooth is the bottleneck when the UMTS DL data rate is 384 kbps. This means that more UMTS radio resources, in the form of OVSF codes, than necessary have been allocated. An optimal mapping for these packet types (DM3, DH1, and DM1) is thus a UMTS RAB with a lower DL data rate. The simulation results also tell us *how* the limiting technology affects throughput. When FEP is used the limiting technology determines "the height" of the throughput curve. As the number of slaves increases a slow, linear degradation in throughput is noticed.

For RR the analysis is not as straightforward. Here both the Bluetooth packet type and the number of slaves sharing the gateway affect whether Bluetooth or UMTS is the limiting technology. This means that there is no simple optimal mapping between the Bluetooth packet type and the UMTS DL data rate in this case. Focusing on the RR curves we see that throughput seems to decrease exponentially as the number of slaves increase. This seems to be the case for all RR curves except for the RR-DH5 curve. In this case throughput seems to decrease linearly as the number of slaves increases to three, and for additional slaves throughput seems to decrease exponentially. The analysis in section 5.1 can explain this behaviour. Table 5.4 tells us that for RR-DH5, Bluetooth becomes the bottleneck for more than three slaves. This is what we can see in the figure. For up to three slaves UMTS is the bottleneck and throughput decreases linearly as the number of slaves increase. For more than three slaves Bluetooth is the bottleneck and throughput decreases exponentially with the number of slaves.

From the discussion above we can conclude that independent of the polling algorithm, the Bluetooth packet type affects whether UMTS or Bluetooth is the limiting technology. For FEP there is an optimal mapping between the Bluetooth packet type and the UMTS DL data rate. For RR polling the same mapping is true only for one slave using the gateway. The existence of such a mapping has important implications. If, for example, the Bluetooth radio channel is fading, or if the operating distance for Bluetooth is close to 10 meters the packets might have to be protected by FEC. This can be taken care of by the Link Manager, which automatically can perform a change between DH and DM packets when the radio link conditions change. A change from DH packets to DM packets may change throughput significantly and make Bluetooth the bottleneck as seen in Figure 5.1. In such cases it would be desirable to change the UMTS RAB to one with lower DL data rate in order to get a better match between the UMTS DL data rate and the Bluetooth data rate. This would minimize the OVSF codes allocated by the gateway. Another example of such a situation is when Bluetooth is operated in an environment with WLAN interference. From the work of (Agrawal et al 2000) we know that 1-slot

[12] With optimal mapping we mean that throughput end to end is maximized and the UMTS RAB has a data rate as low as possible. The lower the UMTS RAB data rate, the less of the available OVSF codes in a cell will be allocated.

packets perform best under such conditions. We now know that using 1-slot packets means that Bluetooth will be the bottleneck and that UMTS is being underutilized if the UMTS RAB DL data rate is 384 kbps. We should thus change the UMTS RAB to one with lower DL data rate in such a situation in order not to allocate more OVSF codes than necessary.

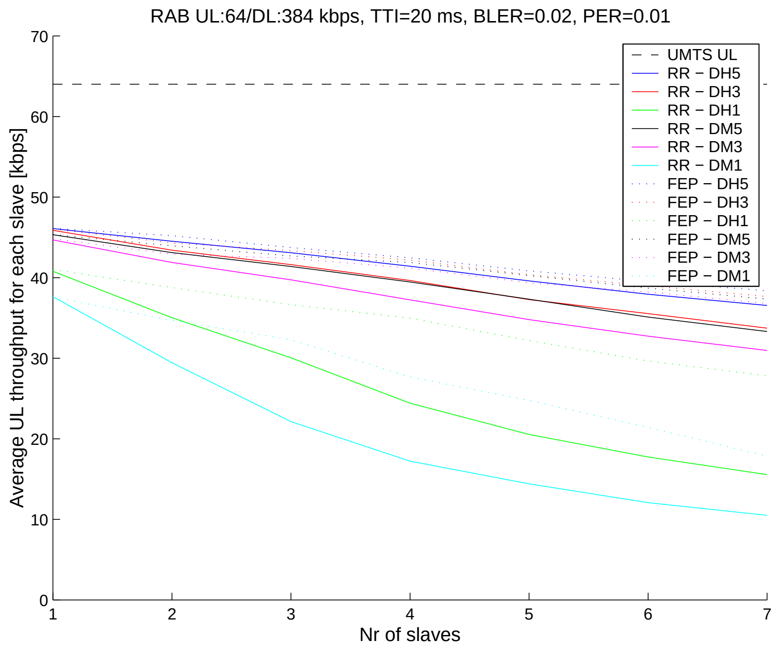

Figure 5.2 Uplink throughput versus number of slaves for UL:64/DL:384 kbps RAB.

Figure 5.2 shows the results for the same simulation scenario as Figure 5.1, but for the uplink instead of the downlink. The uplink performance has a different behaviour than the downlink performance. The explanation for this is that UMTS (64 kbps) is the bottleneck and throughput is hardly affected at all by the polling algorithm, the Bluetooth packet type, or the number of slaves sharing the gateway. It is only for slot limit one that Bluetooth becomes the limiting technology and affects performance appreciably. Thus, the uplink traffic performance is almost entirely determined by the UMTS uplink data rate.

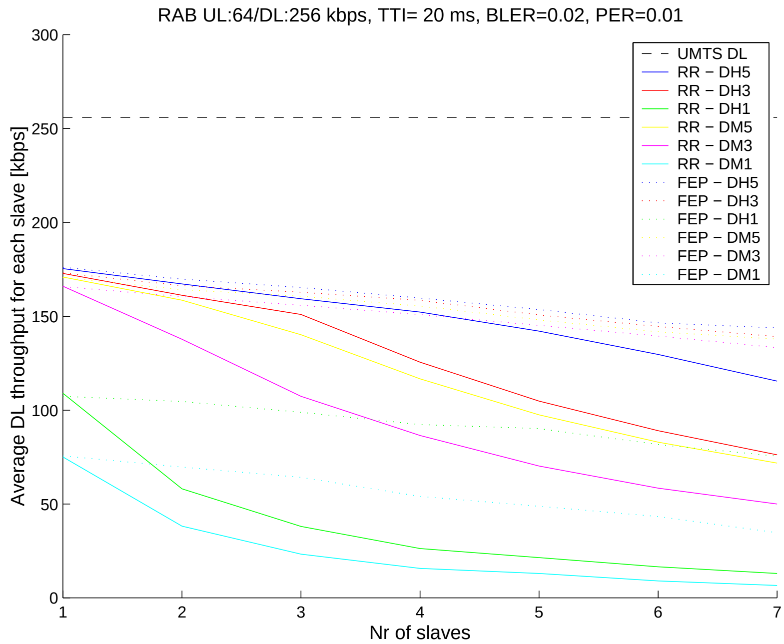

Figure 5.3 Downlink throughput versus number of slaves for UL:64/DL:256 kbps RAB.

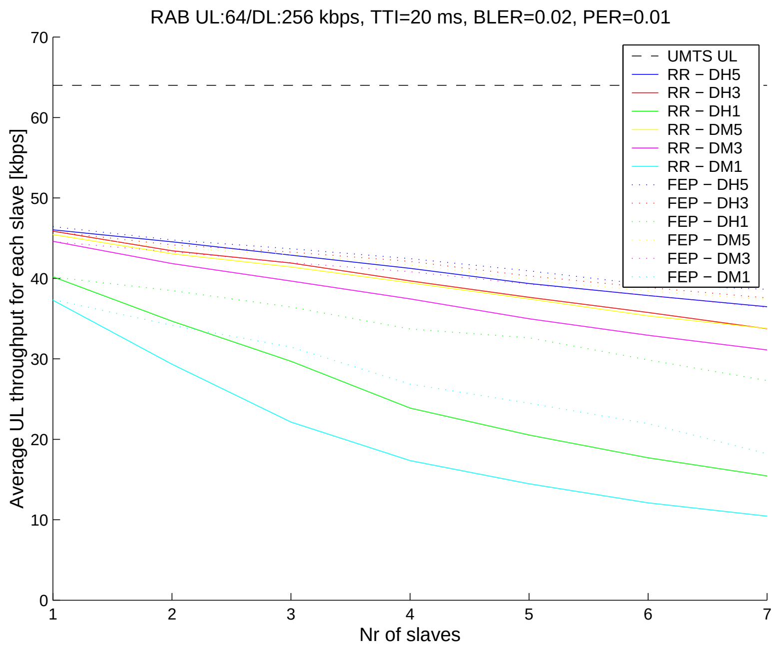

Figure 5.4 Uplink throughput versus number of slaves for UL:64/DL:256 kbps RAB.

In Figure 5.3 and Figure 5.4 the UMTS DL data rate has been changed to 256 kbps. Compared to the 384 kbps curves there are a few differences to be noticed. Overall throughput is lower due to the lower UMTS DL data rate. We also see that for FEP, UMTS is the limiting technology for all packet types except DH1 and DM1. For FEP-

DM3, UMTS is the bottleneck. This was not the case for a UMTS DL data rate of 384 kbps. We thus conclude that when FEP is used the optimal mapping for DM3 is a UMTS DL data rate of 256 kbps.

For RR we notice that the curves differ from those in Figure 5.1. Here the RR-DH5 curve has almost the same shape as the FEP-DH5 curve. This is because UMTS is the limiting technology for up to five slaves and this results in a linearly decreasing RR curve. For more than five slaves Bluetooth becomes the limiting technology and the decrease in throughput becomes exponential. The RR-DH3, RR-DM5, and RR-DM3 curves also show a linear decrease followed by an exponential decrease. This can be attributed to the lower UMTS DL data rate and is consistent with what we anticipated in Table 5.4. The uplink behaviour is the same as in Figure 5.2 since the uplink data rate is the same.

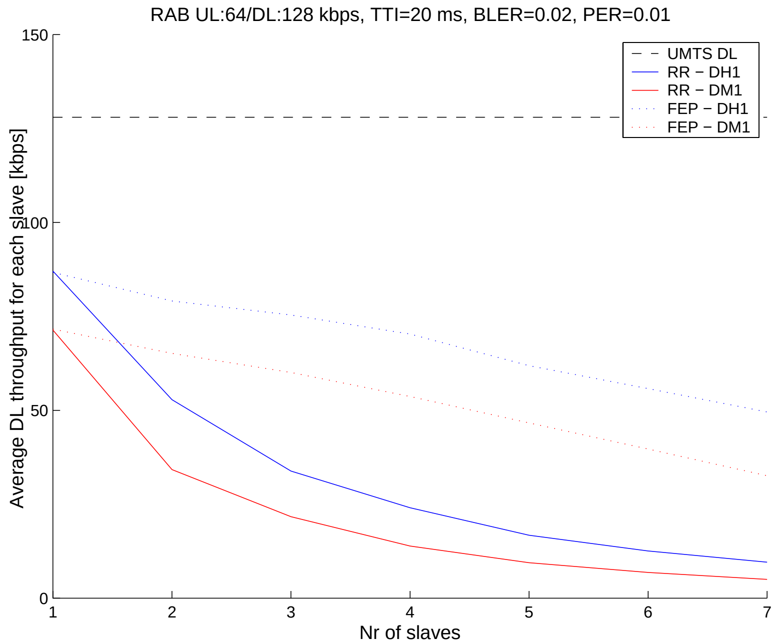

Figure 5.5 Downlink throughput for 128 kbps UMTS RAB and 1-slot packets.

The simulation results for a UMTS RAB with data rate UL:64/DL:128 kbps are shown in Figure 5.5. Only 1-slot packets have been considered since for other packet types UMTS clearly will be the bottleneck. We notice that the throughput for 1-slot packets is lower than for the higher data rate UMTS RABs, but not much lower. When only 1-slot packets can be used in Bluetooth a 128 kbps DL data rate UMTS RAB is thus an optimal mapping. Uplink performance does not differ appreciably from the earlier uplink result curves and is therefore not shown here.

5.2.3 One voice conversation and concurrent web sessions

This scenario studies the impact of a voice conversation running concurrently with the web sessions. It is assumed that the voice conversation uses a Bluetooth SCO link. An example of such a user scenario is when a Bluetooth headset is used for a phone call at the same time as one or several slaves have web sessions running. Two different combinations of UMTS RABs were considered. The first one was a CS RAB, conversational/speech, UL:12.2 kbps DL:12.2 kbps and a PS RAB, interactive or background, UL:64 kbps DL:384 kbps. The second RAB differed from the first in that the PS RAB DL data rate was 128 kbps instead of 384 kbps.

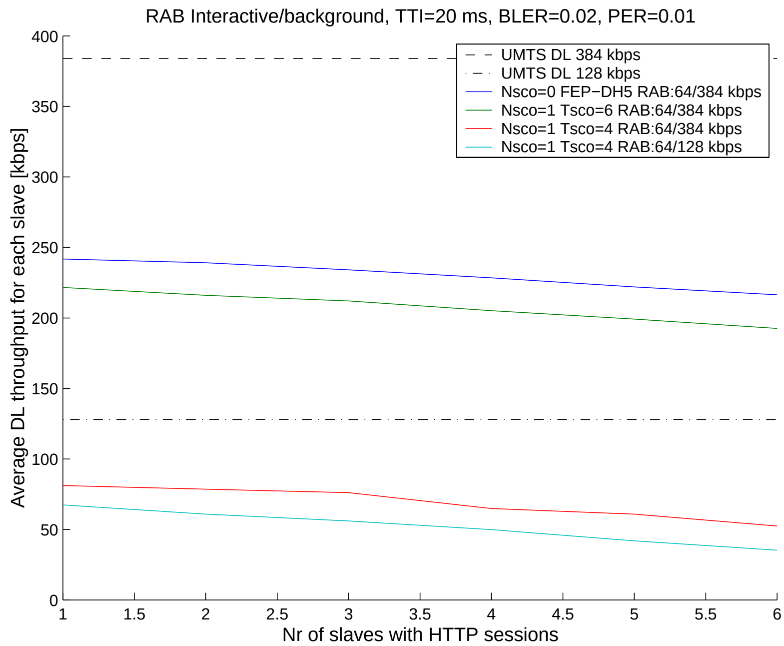

Figure 5.6 Throughput for web sessions with one voice conversation active.

Figure 5.6 shows the downlink throughput for the web sessions. Results are only shown for FEP. N_{SCO} denotes the number of active voice conversations (0 or 1) and T_{SCO} denotes the SCO packet interval. Recall that T_{SCO} equal to 6 means that the SCO packets are not protected by FEC, whereas T_{SCO} equal to 4 means that the SCO packets are protected by FEC (see section 4.2.3 for a further explanation). The Bluetooth data packets (ACL packets) were not protected by FEC. The curve with N_{SCO} equal to 0 shows throughput for FEP-DH5 (compare Figure 5.1).

From the figure we see that throughput for the web sessions is not affected much by a voice conversation with T_{SCO} equal to 6. Even though a voice conversation represents a 64 kbps circuit, adding a voice conversation does not make throughput for the web sessions drop by 64 kbps. This is due to the fact that without a voice conversation UMTS is the limiting technology and Bluetooth is not being fully utilized. Adding a voice conversation, on the other hand, makes Bluetooth the limiting technology and throughput for the web sessions is slightly reduced.

Changing T_{SCO} to 4 for the voice conversation reduces throughput for the web sessions significantly. This is because Bluetooth really becomes a bottleneck when only two consecutive slots are available for data. Changing the UMTS PS RAB to one with DL data rate 128 kbps does not affect throughput much in this case. We thus conclude that when Bluetooth suffers from bad radio channel conditions and FEC has to be used for the SCO packets, the optimal UMTS RAB DL data rate is 128 kbps.

5.2.4 File downloads and web sessions

This scenario studies the performance for file downloads, but also for a mix of file downloads and web sessions. The file downloads were modelled as 1 MB files requested once every minute. The web traffic was modelled according to the traffic model in section 4.5. For the Bluetooth packets no FEC was used and the slot limit was set to five.

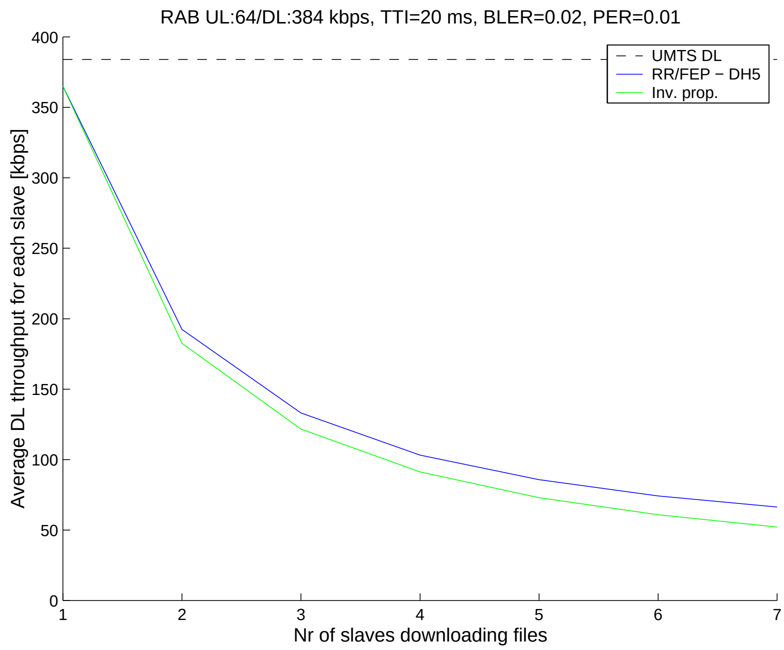

Figure 5.7 Throughput for file downloads.

Figure 5.7 shows how throughput varies when several slaves are downloading files at the same time. As seen in the figure throughput is inversely proportional to the number of slaves downloading files.[13] This is because UMTS is the bottleneck. Due to the non-bursty traffic the slaves have to share the available UMTS bandwidth. Bluetooth is being underutilized and there is no difference in performance between the two polling algorithms. It is obvious that for a demanding traffic scenario such as this, UMTS represents a bottleneck. This indicates that it might not be an attractive alternative to multiplex several traffic streams onto one UMTS RB. An alternative would be to set up one UMTS RB for each slave's traffic stream.

[13] The reason that the RR/FEP – DH5 curve shows a slightly better throughput than the inversely proportional curve is that the file download traffic has some degree of burstiness. For example, most of the time when seven slaves share the gateway all seven slaves will receive data at the same time and share the available bandwidth. However, due to the fact that the file download traffic has some degree of burstiness there will be short time periods when fewer slaves receive data at the same time and thus each of them will get a higher bandwidth.

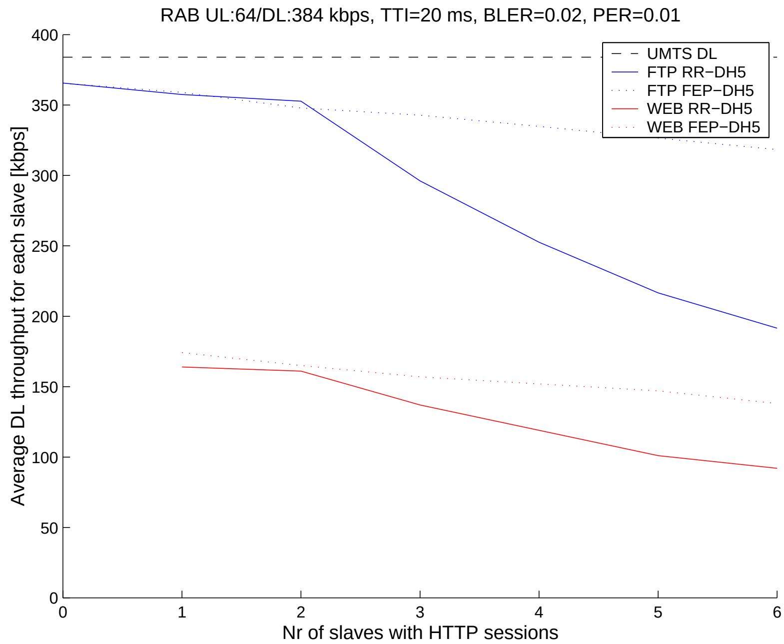

Figure 5.8 Throughput for one file download with concurrent web sessions.

Figure 5.8 shows the results when one slave is downloading files and the others have web sessions running. The number of slaves with web sessions is varied from 0 to 6. For FEP, throughput for the file downloads and the HTTP sessions is almost independent of the number of slaves running web sessions. However, it is obvious that the file download "steals" a lot of bandwidth from the web sessions. Comparing the web traffic curves with those in Figure 5.1 we see that throughput is much lower when there is a file download running concurrently with the web sessions. The file download, on the other hand, is not affected noticeably by the web sessions. This behaviour is explained by the bursty nature of the web traffic. The web traffic and the file download traffic have to share the available bandwidth. During a burst of web traffic they get roughly half of the available bandwidth each. This affects throughput for the web session significantly. For the file download, on the other hand, the duration of the web traffic bursts are short compared to the total file download time and the average throughput is thus hardly affected at all.

The RR curves look a bit different. For less than or equal to two slaves running web sessions the RR curves show the same linear decrease as the FEP curves. For additional slaves running web sessions we see an exponential decrease in throughput. From Table 5.4 we know that for more than three slaves sharing the gateway Bluetooth becomes the bottleneck and this explains the observed behaviour. Altogether, we conclude that the gateway is able to handle this mix of file downloads and HTTP traffic well.

6 Conclusions

This chapter presents the conclusion and gives some suggestions for further work.

6.1 Conclusions

Bluetooth and UMTS are complementary technologies that could be used together in order to provide wireless packet data access from mobile terminals. UMTS provides mobile wireless packet data access and Bluetooth can be used for final delivery to local devices. This is a very likely scenario and the work in this project shows that the two technologies are well suited to be used together.

Bluetooth and UMTS are well matched in terms of data rate, but when considering Bluetooth/UMTS links there is no general answer to which one of these technologies is the limiting one. It mainly depends on the UMTS RAB data rate and the Bluetooth packet type. If several users share the same gateway it also depends on the number of users sharing the gateway and the Bluetooth polling algorithm.

For bursty traffic and more than one user communicating with a Bluetooth/UMTS gateway the polling algorithm implemented in Bluetooth has a crucial impact on performance. An exhaustive or partially exhaustive polling algorithm, such as fair exhaustive polling, results in throughput that only slowly decreases as the number of users sharing the gateway increases. Round robin polling, which is currently used in the Bluetooth chips, performs equally well compared to fair exhaustive polling as long as UMTS is the limiting technology. However, when Bluetooth is the limiting technology round robin polling results in throughput that decreases exponentially with the number of users sharing the gateway.

Round robin polling is used in the first generation Bluetooth chips currently available on the market. This algorithm represents a bottleneck if Bluetooth is used together with UMTS. However, UMTS will not be commercially launched for yet some time and by then it is very likely that a better polling algorithm, such as for example fair exhaustive polling, will be used in Bluetooth. Hence, the RR bottleneck will probably be removed by the time UMTS becomes commercially available.

Bluetooth and UMTS represent dynamic radio links and their operating environments might differ greatly. If the Bluetooth radio channel is fading or suffers from interference FEC has to be used and/or the packet length has to be limited. This means that the data rate for Bluetooth will be limited. On such occasions it would be desirable to adjust the UMTS RAB DL data rate after which type of Bluetooth packet that is being used. The reason for this is that exists an optimal mapping between the Bluetooth packet type and the UMTS RAB DL data rate. This mapping makes sure that throughput is maximized at the same time as the allocated OVSF codes are kept to a minimum. The mapping is always optimal when only one slave uses the gateway. If several slaves use the gateway the mapping is optimal when FEP is used.

Overall, a Bluetooth/UMTS gateway, where traffic streams to and from different slaves are multiplexed onto a single UMTS radio bearer in a FIFO manner, works well for web traffic in terms of throughput. Performance is acceptable even for web sessions running concurrently with a voice conversation or a file download. However, for a more demanding traffic scenarios, such as several simultaneous file downloads, UMTS is a bottleneck and throughput is inversely proportional to the number of slaves sharing the gateway. In such cases, it is advisable to multiplex traffic streams to and from different slaves onto different UMTS radio bearers. Overall, the simulation results in this project show that several devices can use a Bluetooth/UMTS gateway simultaneously. The combination of Bluetooth and UMTS is therefore a good solution for Bluetooth-based access points in trains and buses.

6.2 Further work

This project has not been exhaustive and more research remains to be done in the area. Many issues have not been considered at all in this report while others have only been discussed briefly. Some suggestions for future work are:

- In this project it has been assumed that the slaves' traffic streams are multiplexed onto a single UMTS RB and FIFO queues in the gateway and RNC model this. Instead of FIFO queues, some other queuing algorithm, e.g. weighted fair queuing, could be used. This would be necessary in order to offer QoS to the slaves, and would make it possible to give different priorities to different slaves. Another approach would be to set up one UMTS RB for each slave. Both these approaches require further investigation.
- For other services than best effort HTTP, QoS will have to be considered. In the current version of the Bluetooth specification there is limited support for QoS, as L2CAP implementations are only required to support best effort services. There are, however, some baseband parameters such as the maximum polling interval for a slave, which can be negotiated. UMTS, on the other hand, has four different QoS classes with specified QoS attributes. What should the mapping between the Bluetooth baseband parameters and the UMTS QoS attributes look like for different services? Which services will a Bluetooth/UMTS gateway be able to handle without violating the QoS requirements?
- There are many practical issues that will have to be considered when designing a Bluetooth/UMTS gateway. One of the more important issues, especially if the gateway is to be included in a mobile phone, is the buffer space required in the gateway. How does the available buffer size in the gateway affect performance and what is the minimum buffer size required?
- To establish, maintain, and release radio bearers in UMTS and Bluetooth requires signalling. Furthermore, the work in this project shows that for efficient operation a mapping between the Bluetooth packet type and the UMTS RAB DL data rate is required. For example, if the Bluetooth radio bearer operates in an environment with interference and only 1-slot packets can be used, there is little need to set up a high data rate UMTS RAB, but a lower data rate UMTS RAB would be sufficient. There is thus a need for signalling between Bluetooth and UMTS. Also, if QoS should be offered there is a need for signalling of QoS requirements between Bluetooth and UMTS. How should such signalling be done?

7 References

7.1 Written sources

3GPP (2001a), *3GPP TS 23.107: QoS Concepts and Architecture.* V5.1.0, June 2001.

3GPP (2001b), *3GPP TS 27.060: Mobile Station (MS) supporting Packet Switched Services.* V3.5.0, March 2001.

3GPP (2001c), *3GPP TS34.108: Common Test Environments for User Equipment (UE) Conformance Testing.* V4.0.0, September 2001.

Agrawal, Prathima, Famolari, David (2000), "Architecture and Performance of an Embedded IP Bluetooth Personal Area Network". *Proceedings of 2000 IEEE International Conference on Personal Wireless Communications*, 17-20 December, 2000, Hyderabad, India.

Bergström, Paul. (2001), *Packet channel Optimisation Strategies in UMTS*. Master thesis, Department of Teleinformatics, KTH, spring 2001.

Bluetooth Special Interest Group (2001a), *Specification of the Bluetooth System, Specification Volume 1: Core.* Version 1.1.

Bluetooth Special Interest Group (2001b), *Specification of the Bluetooth System, Specification Volume 2: Profiles.* Version 1.1.

Bruno, Raffaele, Conti, Marco, Gregori, Enrico (2001), "Wireless Access to Internet via Bluetooth: Performance Evaluation of the EDC Scheduling Algorithm". *Proceedings of the first Workshop on Wireless Mobile Internet,* July 21, 2001, Rome, Italy.

Capone, Antonio, Gerla, Mario, Kapoor, Rohit (2001), "Efficient Polling Schemes for Bluetooth picocells". *Proceedings of ICC 2001. IEEE International Conference on Communications, 2001*, volume 7, page 1990-1994.

Das, Abhishek, Ghose, Abhishek, Razdan, Ashu, Saran, Huzur, Shorey, Rajeev (2001), "Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network". *Proceedings of IEEE INFOCOM 2001.* page 591-600.

Elg, Johannes, Johansson, Niklas, Johansson, Per, Körner, Ulf, Svennarp, Göran (1999), "Short Range Radio Based Ad-hoc Networking: Performance and Properties". *Proceedings of International Conference on Communications,* Vancouver, Canada, June 1999.

Haartsen, Jaap C (2000). "The Bluetooth Radio System". *IEEE Personal Communications*, February 2000, page 28-36.

Haartsen, Jaap, Matheus, Kirsten, Stahl, Wolfgang, Zürbes, Stefan (2000), "Radio Network Performance of Bluetooth". *Proceedings of ICC 2000, IEEE International Conference on Communications, 2000,* volume 3, page 1563–1567.

Holma, Harri, Toskala, Antti (2000), *WCDMA for UMTS*. John Wiley & Sons, 2000.

Johansson, Niklas, Johansson, Per, Körner, Ulf (1999), "Performance Evaluation of Scheduling Algorithms for Bluetooth". *Proceedings of IFIP Broadband Communications,* Hong Kong, November 1999.

Johansson, Niklas, Kihl, Maria, Körner, Ulf (2000a), "Performance Simulation of TCP/IP over a Bluetooth Ad-hoc Network". *Proceedings of IFIP Networking 2000,* Paris, May 2000.

Johansson, Niklas, Kihl, Maria, Körner, Ulf (2000b), " TCP/IP over the Bluetooth Wireless Ad-hoc Network". *Proceedings of ITC Specialist Seminar on Mobile Systems and Mobility,* Lillehammer, March 2000.

Johansson, Per, Kazantzidis, Manthos, Gerla, Mario (2001), "Bluetooth: An enabler for Personal Area Networking". *IEEE Network,* September/October 2001.

Kalia, Manish, Bansal, Deepak, Shorey, Rajeev (2000), "Data Scheduling and SAR for Bluetooth MAC". *Proceedings of VTC 2000-Spring Tokyo, 2000 IEEE 51st Vehicular Technology Conference Proceedings,* volume 2, page 716 –720.

Kalia, Manish, Bansal, Deepak, Shorey, Rajeev (1999), "MAC Scheduling and SAR policies for Bluetooth: A Master Driven TDD Pico-Cellular Wireless System". *Proceedings of 1999 IEEE International Workshop on Mobile Multimedia Communications*, page 384-388.

Khaunte, S.U., Limb. J.O., *Statistical Characterisation of a World Wide Web Browsing Session*, GA 30332-0280, Georgia Institute of Technology, Atlanta, USA, 1998.

Kim, Jesung, Lim, Yujin, Ma, Joong Soo, Min, Sang Lyul (2001), "Performance Evaluation of the Bluetooth-based Public Internet Access Point". *Proceedings of 15th International Conference on Information Networking*, 2001, page 643-648.

Sachs, Joakim, Meyer, Michael (2001), "Mobile Internet – Performance Issues Beyond the Radio Interface". *Proceedings of 10th Aachen Symposium on Signal Theory,* September 20-21, 2001, Aachen.

7.2 Web sources

Bluetooth Special Interest Group (2001c). www.bluetooth.com. Information acquired 2001-10-01 at 10:47 am.

Post och Telestyrelsen (2001). www.pts.se. Information acquired 2002-01-06 at 3:33 pm.

All 3GPP specifications are available at www.3gpp.org.

Appendix A – Acronyms

3GPP – 3rd Generation Partnership Program
ACL – Asynchronous Connection Less
AMR – Adaptive Multi-Rate
ARQ – Automatic Repeat Request
BCH – Broadcast channel
BLER – Block Error Rate
BNEP – Bluetooth Network Encapsulation Protocol
BQP – Bluetooth Qualification Program
CDF – Cumulative Distribution Function
CDMA – Code Division Multiple Access
CN – Core Network
CPCH – Common Packet Channel
CRC – Cyclic Redundancy Check
CS – Circuit Switched
DCH – Dedicated Channel
DH – Data High
DL – Downlink
DM – Data Medium
DS – Direct Sequence
DSCH – Downlink Shared Channel
DV – Data Voice
FACH – Forward Access Channel
FDD – Frequency Division Duplex
FEC – Forward Error Correction
FEP – Fair Exhaustive Polling
FH – Frequency Hopping
FIFO – First In First Out
FTP – File Transfer Protocol
GGSN – Gateway GPRS Support Node
GMSC – Gateway MSC
GSFK – Gaussian Frequency Shift Keying
GSM – Global System for Mobile communication
GPRS – General Packet Radio Service
HLR – Home Location Register
HSCD – High Speed Circuit switched Data
HSDPA – High Speed Downlink Packet Access
HTTP – Hyper Text Transfer Protocol
IP – Internet Protocol
ISM – Industrial, Scientific, and Medical
L2CAP – Logical Link Control and Adaptation Protocol
LAN – Local Area Network
LMP – Link Manager Protocol
MSC – Mobile services Switching Center
MSS – Maximum Segment Size

MTU – Maximum Transmission Unit
OSU – Optimum Slot Utilization
OVSF – Orthogonal Variable Spreading Factor
PAN – Personal Area Network
PCH – Paging Channel
PDA – Personal Digital Assistant
PDU – Protocol Data Unit
PDF – Probability Density Function
PER – Packet Error Ratio
PLMN – Public Land Mobile Network
PS – Packet Switched
RACH – Random Access Channel
QoS – Quality of Service
QPSK – Quadrature Phase Shift Keying
RAB – Radio Access Bearer
RB – Radio Bearer
RF – Radio Frequency
RFC – Request For Comments
RLC – Radio Link Control
RNC – Radio Network Controller
RR – Round Robin
RSSI – Received Signal Strength Indicator
RTT – Round Trip Time
SAR – Segmentation And Reassembly
SCO – Synchronous Connection Oriented
SDP – Service Discovery Protocol
SDU – Service Data Unit
SF – Spreading Factor
SIG – Special Interest Group
SGSN – Serving GPRS Support Node
TDD – Time Division Duplex
TTI – Transmission Time Interval
UE – User Equipment
UL - Uplink
UMTS - Universal Mobile Telecommunication System
URA – UMTS Registration Area
USIM – UMTS Subscriber Identity Module
UTRA – UMTS Terrestrial Radio Access
UTRAN – UMTS Terrestrial Radio Access Network
VLR – Visitor Location Register
VoIP –Voice over IP
WCDMA – Wideband Code Division Multiple Access
WLAN – Wireless Local Area Network
WPAN – Wireless Personal Area Network

Appendix B – Random variables

The log-normal distribution

A random variable X is said to have a lognormal distribution if the random variable $Y = \ln(X)$ has a normal distribution. X can be computed as e^Y, where Y is normally distributed with mean m and standard deviation s. The cumulative distribution function for X is given by:

$$F_X(x) = \frac{1}{2}\left[1 + erf\left(\frac{\ln(x) - m}{s\sqrt{2}}\right)\right], x \geq 0, s > 0,$$

where the error function is given by:

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt.$$

The corresponding probability density function is given by:

$$f_X(x) = \frac{1}{xs\sqrt{2\pi}}e^{-(\ln x - m)^2/2s^2}.$$

The expectation value and the standard deviation for X are given by:

$$\mu = e^{m+s^2/2} \quad \sigma^2 = e^{2m+s}(e^{s^2} - 1).$$

The bound Pareto distribution

The bound Pareto distribution is derived from the Pareto distribution by confining it to a finite interval [k, T]. A random variable X has a bound Pareto distribution if its cumulative distribution function as given by:

$$F_X(x) = \frac{1-(k/x)^\alpha}{1-(k/T)^\alpha}, k \leq x \leq T.$$

X is confined to the interval [k, T] and α is the shape parameter.
The corresponding probability density function is given by:

$$f_X(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}\left(1-(k/T)^\alpha\right)}.$$

The expectation value and the standard deviation for X are given by:

$$\mu = \frac{k\alpha}{(\alpha-1)} \times \frac{1-(k/T)^{\alpha-1}}{1-(k/T)^\alpha}, \alpha \neq 1$$

$$\sigma^2 = \left(\frac{k}{T}\right)^{\alpha-2}\left(\frac{k\alpha}{\alpha-1}\right)^2\left\{\frac{1}{\alpha(2-\alpha)} \times \frac{1-(k/T)^{2-\alpha}}{1-(k/T)^\alpha} - \left(\frac{1-k/T}{1-(k/T)^\alpha}\right)^2\right\}, \alpha \neq 1,2$$

Appendix C – On the definition of throughput

When considering several consecutive links as in this project (external network, UMTS CN, UMTS radio link, and Bluetooth radio link) there is no unambiguous definition of throughput. In this project throughput has been defined as the application data burst size divided by the end-to-end transfer delay. This definition of throughput has merits, mainly that it is easy to measure and understand, but also disadvantages in that it gives a very conservative measure of throughput as experienced by a user. Still, this definition has been used since the absolute value of the throughput (i.e. the height of the throughput curve) was not considered a priority in this project. The absolute value is much affected by mechanisms not included in the simulator such as TCP congestion avoidance algorithms and processing delays. The priority in this project was to see how throughput varies with different parameters and to see when UMTS or Bluetooth is the limiting technology (i.e. the shape of the throughput curve).

As noted in Figure 5.1 to Figure 5.5 the throughput for web traffic is lower than what one might have expected, i.e. there is a gap between the UMTS RAB data rate and the measured throughput. Part of this gap is due to overhead introduced by encapsulation and segmentation in different protocol layers (remember that throughput is measured at the application layer). However, most of it is due to how throughput is defined and the way the web traffic is modelled. For other traffic than web traffic the gap is much smaller. In Figure 5.7 we see that for a file download and only one slave using the gateway throughput is almost equal to the UMTS RAB data rate.

The fact that the measured throughput is relatively low for web traffic can be explained by looking at the web traffic model in detail. The downlink burst size is log-normally distributed with an expectation value of 18 kB. However, the median value for the lognormal distribution is only 3.8 kB. This means that most of the bursts will have a size much smaller than the expectation value. 50% of them will be smaller than 3.8 kB and 81% of them will be smaller than 18 kB, see Figure C.1. This is indeed a realistic model of mobile HTTP traffic. However, with this model and throughput defined as the burst size divided by the end-to-end transfer delay some problems arise. To see why, lets look at an example.

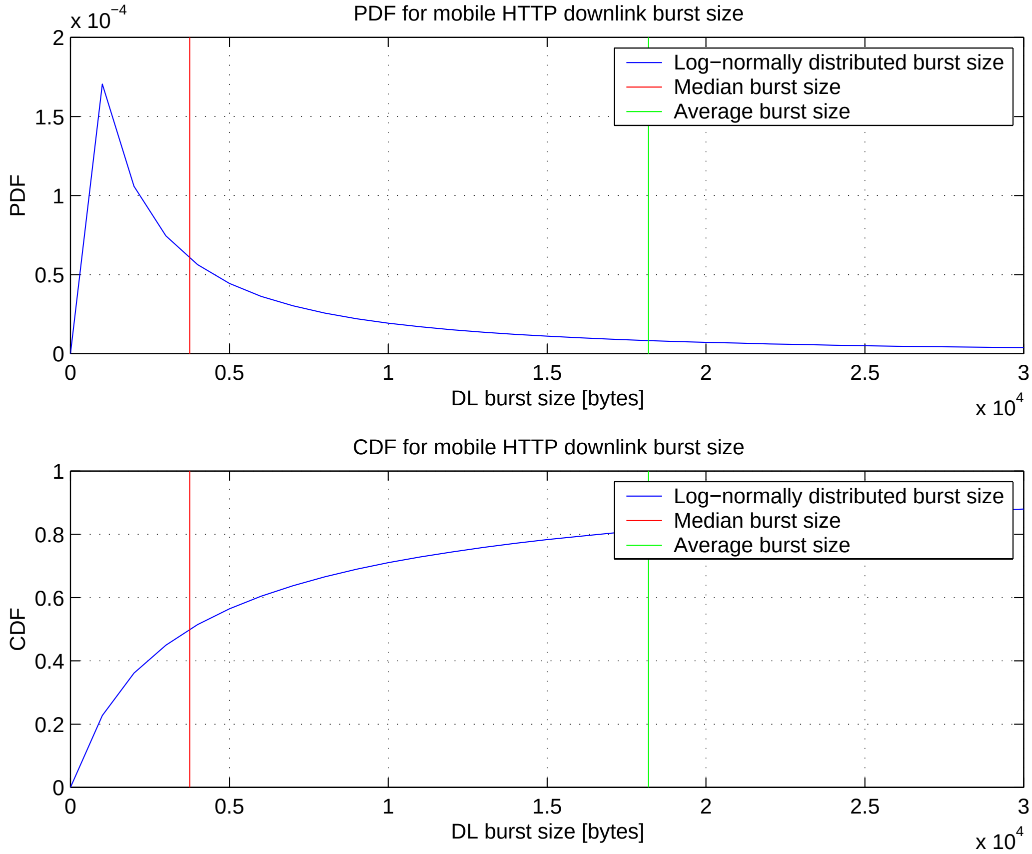

Figure C.1 PDF and CDF for mobile HTTP downlink burst size.

Assume that a downlink data burst generated according to the traffic model has a size of 3 kB (45% of the bursts have a size less than or equal to 3 kB). This burst will be segmented into three IP packets, two with size 1500 bytes and one with size 120 bytes (assuming a TCP MSS of 1460 bytes and TCP and IP headers with a size of 20 bytes each). The delay for these packets incurred over the G_i interface is 50ms. The delay incurred over the UMTS CN (the G_n and the I_u interface) is on average 20 ms. When the first IP packet arrives at the RNC it cannot be sent directly over the radio interface. On average the delay is TTI/2. If we assume TTI=20 ms the average delay is 10 ms. Furthermore, if we assume a DL data rate of 384 kbps, each radio block (the quantity sent every TTI second) can hold 384000*0.020=7680 bits=960 bytes. The number of radio blocks required is thus $\lceil(2*1500+120)/960\rceil=\lceil 3.2083\rceil=4$, i.e. it takes 4*20=80 ms to send the entire burst over the UMTS radio interface (assuming that no block errors and retransmissions occur which would increase the time required). If we assume that Bluetooth is not the bottleneck the delay incurred over the Bluetooth link is negligible. Adding up the delays we get a throughput of 3000*8/(0.050+0.020+0.010+0.080)=150 kbps which is well below the UMTS RAB DL data rate. The same calculations for a burst size of 18 kB results in a throughput of 18000*8/(0.050+0.020+0.010+20*0.020)=300 kbps. For a file download with a file size equal to 1MB throughput is 1000,000*8/(0.050+0.020+0.010+1071*0.20)=372 kbps. We thus conclude that with this definition throughput for an individual burst is dependent on the burst size. Since most of the generated web traffic bursts have a size much smaller than 18 kB, according to Figure C.1, the average throughput for all burst will be well below the UMTS RAB data rate. This is explains the gap between the UMTS RAB data rate and the measured throughput in Figure 5.1 to Figure 5.5.