Analytical Models of IP Traffic on UMTS Mobile Networks

Jesse Landman

March 17, 2005

Abstract

Mobile communications networks are becoming ubiquitous worldwide. Such networks have achieved high rates of success in the area of connected voice traffic. An example of such a network is the Global System Mobile (GSM) cellular network. It is apparent that these networks are also able to offer packet-switched services such as IP for Internet and email browsing. The GPRS extension to GSM networks is a working example of this. However, it is necessary to completely overhaul such Second Generation (2G) networks in order to provide high data rates for packet services.

In this dissertation we investigated the performance of IP traffic over UMTS networks. UMTS is a Third Generation (3G) replacement for GSM technologies and consists of a land-based Core Network and a UMTS Radio Access Network (UTRAN). UMTS networks utilise Code Division Multiple Access (CDMA) technologies to multiplex the radio resources available to the network. CDMA allows for all users of the network to transmit simultaneously, reducing the need to multiplex using time (TDMA) or frequency (FDMA). CDMA channels do however experience multipath fading as well as multiple access interference.

Our investigation involved deriving two analytical models, one that would model the queueing and processing on the CN, and one that would model the transmission errors on the UTRAN.

In order to model the queueing on the CN, we analysed the UMTS IP network and derived two models that can be approximated by two different queueing systems: a $M^{[X]}/D/1$ queue and a $BMAP/D/1$ queue. In other words, the IP packet inter-arrival times are characterised by a Batch Poisson arrival process ($M^{[X]}$) and Batch Markovian Arrival Process (BMAP), respectively. The service process in our model

is deterministic.

We modelled the physical channel by modifying Hidden Markov Models (HMMs) that exist in other related work. Our model incorporates Ricean fading, which is a type of fading that exists in indoor and low-range outdoor environments. The model also incorporates Turbo Coding, an error correcting technique employed by UMTS networks.

In order to verify the analytical models, we created a discrete event simulation of the UMTS network. We used IP traffic measurements to provide interarrival times.

The results of the experiments showed that the HMM could provide a high level of accuracy when predicting bit and block error rates. The results also showed that Poisson arrival process was an inadequate approximation of the network, whereas the BMAP provided highly accurate results.

Acknowledgements

I would like to acknowledge the following people:

- Professor Kritzinger, who always tried to push me to towards my best, even though I always pushed back. Thank you for your unwavering support.
- The DNA and CVC laboratory students, who would always lend an ear to my complaints and bad jokes. Special thanks to Colette Consani, who helped with proof reading, and Lourens Walters, who inspired much of the simulator and provided valuable insights into the ways of the MSc student.
- Donald Cook, who supported me in many ways and always believed in me.
- My family, both in Cape Town and Johannesburg, who never quite knew what I was doing, or why, but nevertheless kept their faith in me.
- Mandy, Max, Rupert and Sonny, who are not forgotten.
- Elizabeth, who was and still is a friend in need.

Contents

Abstract 1

Acknowledgements 3

1 Introduction 12

1.1 Overview . . . 13

1.2 Goals of this Dissertation . . . 16

1.2.1 Literature Review . . . 18

1.2.2 Contribution and Model Outcomes . . . 20

1.3 Dissertation Outline . . . 21

2 System Under Investigation 22

2.1 Introduction . . . 23

2.2 Overview . . . 23

2.2.1 Core Network . . . 24

2.2.2 UTRAN . . . 27

2.3 CN Packet Domain Procedures . . . 28

2.4 Link Layer Protocols . . . 29

2.4.1 Link Layer Channels . . . 29

2.4.2 PDCP . . . 32

2.4.3 Radio Link Control (RLC) Protocol . . . 32

2.4.4 Medium Access Control (MAC) Protocol . . . 34

2.5 Physical Layer Protocol . . . 36

2.5.1 Channel Error Control . . . 36

2.5.2 Channel Spreading with DS-CDMA 39
2.5.3 Channel Modulation and Transmission 41
2.6 Radio Link Conditions 41
2.6.1 Fading 42
2.6.2 Multiple Access Interference (MAI) 46
2.7 Packet Data Services 47
2.7.1 Packet Data Flow through the CN 47
2.7.2 Packet Data Flow through the UTRAN 48
2.7.3 IP Service Resources and Parameters 48
2.8 Summary 49

3 IP and Link Layer Model **51**
3.1 Queueing Theory Fundamentals 52
3.1.1 Queueing Theory Notation 53
3.1.2 Basic Performance Measures 53
3.2 Applications to UMTS 54
3.2.1 Overview 54
3.2.2 Service Process 55
3.2.3 Arrival Process 58
3.3 The $M^{[X]}/D/1$ Queue 58
3.4 The BMAP/D/1 queue 60
3.4.1 The Batch Markovian Arrival Process (BMAP) 60
3.4.2 Solving The $BMAP/D/1$ Queue 62
3.5 Summary 65

4 Physical Channel Model **66**
4.1 Physical Channel Review 67
4.2 Hidden Markov Model 67
4.2.1 HMM Parameterisation 71
4.2.2 User Signal Model 72
4.2.3 MAI Signal Model 76
4.3 Performance Metrics 76

4.3.1 Integration with Queueing Metrics 78
4.4 Summary 81

5 Simulator **82**
5.1 Introduction 83
5.2 Conceptual Design 83
5.2.1 Overview 83
5.2.2 Network Entities 83
5.3 Implementation 87
5.3.1 Overview 87
5.3.2 Simulation Algorithms 87
5.3.3 Traffic Generation 90
5.3.4 Transmission Errors 92
5.3.5 Turbo Coding 96
5.4 Data Analysis 97
5.4.1 Data Recording 97
5.4.2 Analysis 98
5.5 Summary 100

6 Results **101**
6.1 Introduction 102
6.2 HMM Parameterization 102
6.3 Delay Results 105
6.3.1 $M^X/D/1$ Queue Parameterization 106
6.3.2 $BMAP/D/1$ Queue Parameterization 107
6.3.3 Results 107
6.4 Performance Analysis 114
6.4.1 Radio Channel Performance 114
6.4.2 Packet Network Performance 115
6.5 Summary 117

7 Conclusions **119**

7.1 Physical Channel Models 120

7.2 Packet Network Models 120

7.3 UMTS Packet Services 121

7.4 Conclusions And Future Work 121

7.4.1 Summary 121

7.4.2 Future Work 122

Bibliography **123**

List of Figures

1 The architecture of a UMTS cellular network. 14

2 A general overview of the UTRAN network elements. The UTRAN network elements are the Node B and the RNC. 24

3 The overall packet domain structure (based on [2], page 20). 25

4 The GPRS network structure. 26

5 Overview of the UTRAN. 28

6 The PDCP protocol. 32

7 The RLC protocol. This example shows the attachment of an AM header. The blocks are segmented into 320 bit segments, with a 16 bit header, as is the case for the 384 bps example in Table 1. 34

8 The MAC protocol. This example shows the 18 bit header being attached. 36

9 The UTRA-FDD turbo coder (Figure taken from [4]). 38

10 The spreading process. 40

11 The spreading process on the downlink [5]. 41

12 The channel combination process on the downlink [5]. 42

13 A simple demonstration of Rayleigh fading. 44

14 A simple demonstration of Ricean fading 45

15 Inter and Intra-cellular interference. 47

16 Timeline of data flow through the GPRS PLMN (TX = *Transmission*). GTP-U implies GTP-U over UDP/IP. 48

17 Timeline of data flow through the UTRAN RNC and Node B(TX = *Transmission*). 49

18 An overview of the GPRS/RNC domain of UMTS. 55
19 A simplified overview of the RNC queueing system. 56
20 The model with further simplifications. 57
21 The Poisson Process. 60
22 A BMAP. Once a sojourn has expired in state i, a transition is made either to state j, where no arrivals occur, or to state 0, where a batch arrival occurs, and the process is restarted in state j. 61

23 A Gilbert-Elliot Channel. 68
24 The extended GEC. Each state is either *good* (G) or *bad* (B). 69

25 Conceptual design of the simulation model. 84
26 The entities being simulated. 85
27 An overview of the queues involved in the acknowledgement process. . 87
28 The Object model of the simulator. 88
29 The simulation algorithm. 91
30 A random sample of IP traffic arrival times over 150, 800, and 2300 samples. Each dot on the graph represents a packet arrival. 93
31 The mean IP traffic arrival times over 150, 800, and 2300 samples. . 94
32 A comparison of the theoretical Ricean CDF and the CDF of the simulated Ricean amplitudes (K = 5). 95
33 Bit Error Rates for varying h. 97
34 Bit Error Rates for varying L_c. 98
35 Sample output for each entity. 99

36 The BER (top) and BLER (bottom) as estimated by the HMM and the simulator. 103
37 The BER (top) and BLER (bottom) as estimated by the HMM and the simulator. 104
38 Delay vs. Load for the $M^{[X]}/D/1$ queue for different values of S_0 (top graph, $\Gamma = 1$) and Γ (bottom graph, $S_0 = 0.003$). 108

39 Delay vs. Load for the parameterized $M^{[X]}/D/1$ queue, with simulated results. 109
40 Delay vs. Load for $BMAP/D/1$ queue for different values of S_0 ($\Gamma = 1$).111
41 Delay vs. Load for the parameterized $BMAP/D/1$ queue, with simulated results. 113
42 Delay vs. Load for the parameterized $BMAP/D/1$ and $M^{[X]}/D/1$ models, with simulated results. 114
43 BER results with and without Turbo coding. 115
44 Channel throughput for 1 and 6 simultaneous packet network users. . 116

List of Tables

1 RLC Parameters for a DSCH channel. The two possible channel bit rates, 384 kbps and 2048 kbps, are shown. 34

2 Network parameters and resources for the transport of IP traffic through a typical UMTS network. 50

3 Summary of arrival and service requirements. 55

4 Simplified summary of arrival and service requirements. 56

5 Simulated values for the BER and BLER (presented in Figure 37). . 105

6 Rooted mean-squared error the BER for low j and high j respectively (presented in Figure 37). 106

7 Simulated delay values for the $M^{[X]}/D/1$ queue (presented in Figure 39). 110

8 Rooted mean-squared error for the BMAP/D/1 model (presented in Figure 39). 110

9 Simulated delay values for the $BMAP/D/1$ queue (presented in Figure 41). 112

10 Simulated delay values for the $BMAP/D/1$ queue (presented in Figure 42). 112

Chapter 1

Introduction

1.1 Overview

Mobile communications networks have experienced a rapid evolution over the past few decades. This evolution is due to improved technologies that allow for wider deployment of higher quality network services. Cellular networks in particular have become ubiquitous worldwide and cater for a very a large subscription base. Currently, second generation Global System Mobile networks (2G GSM) provide voice services as well as limited data services such as the Short Message Service (SMS). The demand for more advanced services from mobile networks has caused a global shift towards a new or third generation of cellular networks. This new generation of cellular networks caters for "always-on" Internet access, email, fax, voice and video conferencing, and entertainment via streaming media.

Several standards for a GSM replacement have surfaced. These have merged into two groups: UMTS (Universal Mobile Telecommunications System) and cdma2000. UMTS, backed by the European Telecommunications Standards Institute (ETSI), was submitted to the ITU's IMT-2000 body in 1998 and has since been recognised as an international standard [45]. The cdma2000 standard was developed by the Telecommunications Industry Association (TIA) of the USA and is a similarly recognised international standard. UMTS and cdma2000 are designed as 3G replacements for the existing 2G networks in Europe (GSM) and the USA (IS-95) respectively. We have chosen to focus on UMTS.

The UMTS standard is specified by the Third Generation Project Partnership Group (3GPP). The primary focus of UMTS is to provide higher data transmission rates with better support for packet services, as well as using more efficient technologies to provide basic cellular services. UMTS Terrestrial Radio Access (UTRA) transmission technology standards are divided into frequency division duplex (UTRA-FDD) and time division duplex (UTRA-TDD). UTRA FDD transmissions occur on paired 5 MHz bands in the 1920-1980 MHz range for the uplink, and in the 2110-2170 MHz range for the downlink. UTRA TDD transmission occur in a shared 5 MHz band (with time division duplexing) in the 1900-1920 MHz and 2010-2025 MHz range [22].

UMTS has a mixed-cell architecture, which means that cells are split into different sizes based on a particular set of service requirements. Macrocells provide services to rural and outdoor users, with a peak data rate of 144 kbit/s for rural outdoor scenarios. Microcells provide peak data rates of 384 kbit/s to suburban outdoor users. Picocells provide the highest data rates, 2048 kbit/s, to users that are in low-range outdoor or indoor environments (see Figure 1).

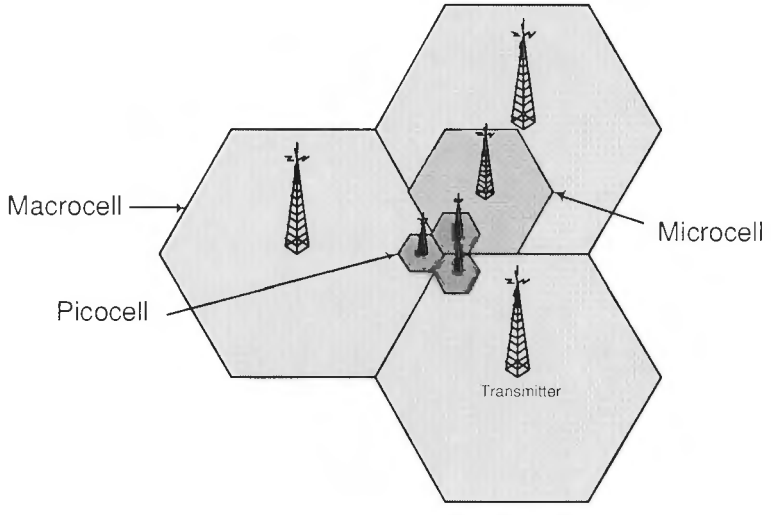

Figure 1: The architecture of a UMTS cellular network.

One of the main differentiators between GSM and UMTS technologies is the multiple access technique that each employs. GSM uses Time Division Multiple Access (TDMA), which multiplexes different channels by assigning a time slice to each one, thus allowing each user to access the network during a predefined set of timeslots. On the other hand, UMTS uses a technology known as Code Division Multiple Access (CDMA). In CDMA systems, narrowband data signals are multiplied by a pseudo-random or pseudo-noise (PN) wideband code [32, 45] known as a *spreading code*. All users are assigned a unique, orthogonal code. Uplink transmissions may occur simultaneously for all users and on the same carrier frequency. The transmitter encodes each user's signal with the code assigned to it. Signals are then encoded with the

unique *scrambling code* which has are assigned to base stations (the scrambling code reduces the interference caused by neighbouring cells).

The encoded signal will appear as noise to all receivers on the channel. The target receiver decodes the transmitted signal by multiplying the 'noisy' signal with its unique PN code and filtering out all wideband information, thus revealing the data.

The use of CDMA is advantageous because it allows for simultaneous use of the transmission frequency. This means that all users can use the full available spectrum, allowing for much greater data rates. CDMA also has its disadvantages. The first arises from the fact that the codes assigned to the users of the system are sometimes not completely orthogonal, which leads to a *Multiple Access Interference* or MAI. This means that the performance of the link for each user increases as the number of users in the cell decreases.

Another issue which confronts designers of CDMA systems is known as the *near-far effect.* In a CDMA system, users that transmit at a high power close to the receiving base station can "drown out" lower power signals that are transmitting from further away. The high power signal increases the noise floor at the base station demodulators and decreases the chance that a distant signal will be received [45]. In order to overcome this problem, base stations implement *power control* by regularly sampling the signal strength of each user and sending power change signals on downlink control channels. This helps to control intra-cellular power. However, *inter-cellular* near-far effects can still be detrimental to low power signals.

Radio communication links pose further problems for network designers because the physical link is often unreliable. There is not always a line of sight from transmitter to receiver, and other radio signals can dramatically affect the quality of a data signal. Furthermore, it cannot be assumed that users of the system are receiving their data via similar quality channels. This fact complicates scheduling decisions.

One particular radio link phenomenon that is highly significant for CDMA systems is known as *multipath fading.* Multipath fading is characterised by rapid fluctuations of a signal over very short distances or time durations. The reason for the fluctuations is that a signal often becomes scattered when transmitted in a densely built

area, which means that different versions of a signal arrive at the receiver from different directions. The contribution from each signal leads to either a strengthening or a cancellation of the signal. Fading also occurs when either the receiver or the transmitter are moving in relation to the one another. This causes different Doppler shifts on each reflected signal, which distorts the received signal. Each reflected signal is known as a *multipath component.* Aside from causing these fluctuations, which deteriorate the quality of the signal, fading disrupts the orthogonality of the PN codes due to the fact that the different multipath components arrive at the receiver out of phase.

There are two general types of fading environments. The first consists of a transmitter that has no line of sight with the receiver, so that the receiver will only receive reflected and scattered multipath signals. The second environment occurs when there is a line-of-sight between the transmitter and the receiver, although multipath components are still present. The first kind is known as a *Rayleigh* fading environment, and the second is known as a *Ricean* fading environment.

In order to correct errors that occur on the wireless link, UMTS systems employ either Convolutional coding or Turbo coding as their Forward Error Correction (FEC) technique. In the case of packet services, Turbo codes are used. These codes are described in detail in Chapter 2 and Chapter 5.

UMTS network designers and engineers need to take all of the previously stated factors - fading, MAI, CDMA - into account. With the rapid evolution of UMTS standards as well as the technologies that they use, designers are required to thoroughly test their designs before implementation. Tools based on analytical models provide simple yet sufficiently accurate methods for investigating various aspects of communications networks such as UMTS networks. This dissertation therefore provides such a tool.

1.2 Goals of this Dissertation

The primary goal of this dissertation is to study the performance of the UMTS networks when providing packet-switched services. We have analysed packet data traffic

because high quality packet based services such as IP are an important and novel[1] aspect of 3G services and thus are poorly understood in terms of 3G networks. Specifically, we have studied traffic on the downlink (from base station to user), for two reasons. The first reason is that IP traffic is asymmetric. This means that traffic on the downlink is of a much larger volume than traffic on the uplink, and therefore a bottleneck on the downlink would lead to poor performance. The second reason is that the downlink caters for multiple users on shared resources, and this fact also leads to further bottlenecks. We have also studied only the FDD scenario for the sake of simplicity, although the work that has been done can be extended to the TDD scenario. The transport and physical channels we have modelled are the DSCH and the physical DSCH (PDSCH) respectively. More specifically, we have chosen to model the 384 Kbps DSCH, as the IP data we will use for simulations are measured from a connection with a similar data rate.

Since we have concentrated on IP traffic, we have also considered a Ricean fading environment. This is because we believe that access to IP traffic is more likely to occur in indoor and low-range outdoor environments due to the availability of higher data rates. In these situations there is a line-of-sight between the receiver and the transmitter, and thus signals will experience Ricean fading.

Finally, we have studied the *characteristics* of IP traffic, by which we mean that we are interested in the effect of bursty, self-similar traffic on a 3G wireless link. This is of interest because packet data has been traditionally accessed over wireline networks, where bit error rates are minimal and negligible. Higher error rates have a significant effect on the delays experienced by users of such a network. The behaviour of TCP falls beyond the scope of this work.

The following section will outline previous work in this field as it pertains to our interests. Section 1.2.2 will describe the contributions of this thesis.

[1]Note that is the high quality nature of the packet service that is novel, and not the packet service itself.

1.2.1 Literature Review

CDMA

Spread spectrum multiple access (SSMA) techniques such as CDMA have been investigated extensively in the available literature. M.B. Pursley [43] provided much of the basic theory required for modelling SSMA communications channels. Pursley and E. Geraniotis, in [25] and [26], have since described the performance of non-coherent Direct Sequence SSMA systems that experience Ricean fading. The results given are derived mathematically and simulated. The results derived in these papers are still relevant, though there have been modifications to counter the shortcomings of the standard Gaussian approximations used. As an example, Cheng and Beaulieu [19] provide an overview of the various techniques available for calculating bit-error probabilities in Rayleigh fading environments, which are also applicable to Ricean fading environments. An excellent reference for such methods is the book by T.S. Rappaport [45].

Markov Models and Fading

Many authors have used analytical models such as Markov models to model CDMA channels, fading channels, and combinations of both. Turin and van Nobelen [48] model Rayleigh fading channels using Hidden Markov Models (HMM), and provide closed form solutions for the fading duration and level-crossing rates[2]. Zorzi *et al.* ([41], [50] , and [51]) have shown the value of Markov Models in modelling radio communications channels, particularly those that experience fading. These results are given greater relevance by Wang and Chang [49], who have verified that the assumption that a Rayleigh fading channel behaves like a first-order Markovian system is valid. Pimentel and Blake [18] use Partitioned Fritchman's Markov Models to analyse burst channels that experience fading. Judge and Takawira [34] use a similar approach to [18]: they extended Gilbert-Elliot models, which are two state Markov models, and created a HMM that approximates the error process on a narrowband

[2]The *level-crossing rate* of a faded signal defines the number of times per second that the signal falls below a certain signal strength.

Rayleigh fading channel. The channel is a CDMA ALOHA-based packet radio channel, and the process above the physical layer is not considered.

UMTS

The chief sources of information about UMTS are the 3GPP specifications (e.g. [4]). Mate, Caldera and Rinne [15] have analysed the Downlink Shared Channel (DSCH) of a Wideband CDMA cell in a fairly high level manner, concentrating on the shared nature of the channel. Furuskär *et al* [17] analyse the High Speed DSCH (HS-DSCH) in a similar way. Both [15] and [17] use discrete-event simulation for the analysis. Gruhl [29] analyses scheduling at the MAC layer on the DSCH in UMTS networks via simulation. A similar study is performed in [47].

Queueing Theory

Klemm *et al* [13] introduce the use of queueing theory, and in particular the Batch Markovian Arrival Process (BMAP), for modelling IP traffic in a UMTS network. They refer to their work in [14], where they show that the BMAP traffic arrival process has applicability to IP traffic modelling. They also provide an algorithm and a software package for parameterising a BMAP from an IP trace. They show results for a BMAP/D/1 queue as it applies to standard IP traffic.

The BMAP was first introduced by Marcel Neuts in [40], although the tutorial by Lucantoni is necessary reading in order to fully understand BMAP/G/1 queues. Abate and Whitt [33] provide the necessary mathematical tools for solving BMAP/G/1 queues.

Turbo Codes

Several resources exist that provide an explanation of Turbo codes, such as the original paper by C. Berrou, A. Glavieux, and P. Thitimajshima that introduced Turbo codes to the world [27]. Another excellent resource is the tutorial by Ryan [46]. Several theoretical aspects of Turbo codes have been extensively investigated, most notable the error floor [24] i.e. the minimum performance provided by the Turbo codes. Lee

and Blahut [36] provide a simple approximation for the performance of Turbo codes which are useful for analysis.

1.2.2 Contribution and Model Outcomes

In this dissertation we analyse the physical channel by extending the model devised by Judge and Takawira ([34]) to cater for Ricean fading and Turbo coding. In order to verify these results, we have written a simulator that uses formulae given by Rappaport [45] and that follows the 3GPP specifications.

In order to model the link layer, we draw extensively on queueing theory, specifically $G/D/1$ queues. We compare the BMAP and the batch Poisson arrival process by modelling the link layer as both an $M^{[X]}$/D/1 queue and a BMAP/D/1 queue. In order to analyse the BMAP, we use the software tool provided by Lindemann et al [37].

Our model of the physical channel is a unique application of existing work. Our BMAP/D/1 queue is also derived from existing theory, but is used to model a system that has not, to the best of our knowledge, been modelled in such a way.

The chief contribution of this dissertation, however, is in the combination of the physical channel model and the link layer model, which provides a greater view of the performance of UMTS networks without adding much complexity. By combining the two models, we have provided a useful method of simultaneously studying the statistical characteristics of IP traffic and the statistical characteristics of radio channels.

Our model analyses the mean delay experienced by an IP packet from the point of entry into the 3G network (the RNC) until the point of transmission. The mean delay is logically dependent on the amount of time spent in the processing queue, which in turn depends on the number of packets in the queue. This number is derived from two sources. Firstly, the number of packets in the queue depends on the arrival rate from outside the network into the RNC. The second factor is the number of retransmissions that are necessary in order to successfully transmit a particular IP packet. To model the arrival process, we draw on queueing theory; to model the retransmissions, we

utilise the physical channel model. The combination of the two yields the overall delay and thus our desired results.

1.3 Dissertation Outline

Chapter 2 describes the UMTS based cellular network that we have modelled. In Chapter 3, we introduce the aspects of queueing theory that are relevant to our work, and apply the theory to the network described in Chapter 2. Chapter 4 provides the details of our radio transmission model, which is also based on the details provided in Chapter 2. Chapter 5 describes the simulator that we produced to verify our analytical model. The results of our experiments, as well as a discussion of the results, are given in Chapter 6. Chapter 6 also discusses the parameterisation of the model. Finally, we give our conclusions in Chapter 7.

Chapter 2

System Under Investigation

2.1 Introduction

This chapter describes the architecture and protocols of the UMTS cellular network that we have analysed. Such detail is required in order to understand the analytical model derived later.

The specifications for UMTS networks are readily available (for example [4, 5, 8]). UMTS (third generation or *3G*) technology is designed to extend existing and well established GSM networks (second generation or *2G*) and the resulting network is essentially the successor to existing cellular networks. UMTS-based networks offer completely integrated packet and circuit switched systems, allowing users to freely and seamlessly switch between voice and data services. UMTS also offers higher data rates as it multiplexes the radio resources using CDMA. Packet based services are provided via a Public Land Mobile Network (PLMN) which is a private IP network utilised specifically for mobile users. The PLMN used by UMTS services is an extension of the General Radio Packet Service (GPRS) PLMN.

The first section of this chapter describes the architecture of the core network (CN) and the radio access network (UTRAN). Section 2.3 describes the procedures that affect packet data in the CN, and Sections 2.4 and 2.5 describe the link layer and physical layer protocols. Section 2.6 describes the characteristics of the radio link that affect the quality of transmissions. Finally, Section 2.7 gives a brief summary of the various stages in the life-cycle of a data packet as it passes through the UMTS network.

2.2 Overview

The UMTS network architecture is similar to that of GSM: it consists of mobile devices known as User Equipment (UE) that communicate with a base station (Node B in UMTS terminology) over a radio air interface. The Node B is controlled by a Radio Network Controller (RNC) via fixed wireline connections. These three elements form the UMTS Radio Access Network or UTRAN. The UTRAN acts as the interface between the mobile user and the Core Network (CN), which controls the

interconnection of mobile users, as well as the connections between mobile users and external networks such as the Internet. In other words, the CN forms the backbone of the cellular network, and the UTRAN provides the radio access points, as is shown in Figure 2.

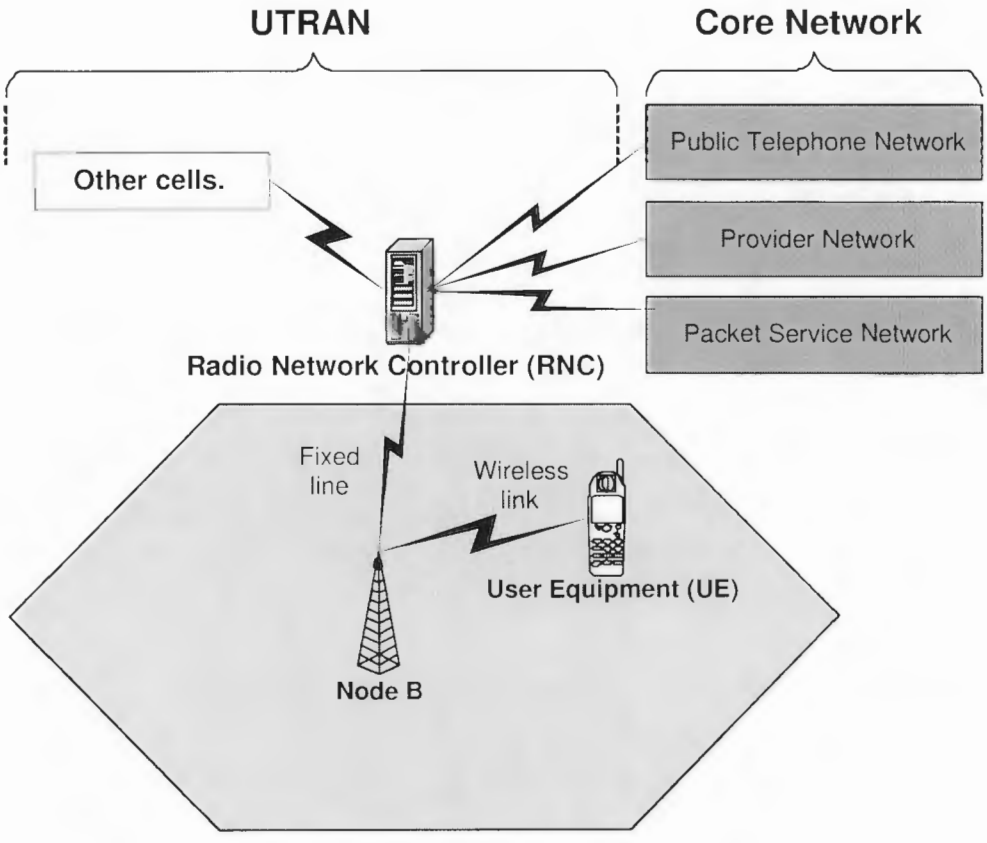

Figure 2: A general overview of the UTRAN network elements. The UTRAN network elements are the Node B and the RNC.

2.2.1 Core Network

The core network (CN) provides the land-based backbone of the mobile network. It provides the links between base stations and public telephone (circuit) networks,

and between base stations and packet networks. It also provides links between one base station and another, and to the cellular providers' own databases, such as Home Location Registers (HLR), which are necessary for accessing subscriber information. Our work is concerned with the links that affect packet data. Packet data makes use of 3G packet service bearers which are designed for point to point and point to multipoint packet data services [1]. 3G packet services use GPRS-based private IP networks to provide these IP services. It must be noted that the GPRS IP network can be thought of as distinct from GPRS itself: GPRS defines an upgrade to GSM that involves the implementation of a private IP network as well as the manipulation of the physical channels to accommodate packet data. UMTS is a further upgrade, which involves either the integration of the UTRAN into existing GSM networks, or, if upgrading is not applicable, the creation of a new UMTS network. It still utilises the GPRS IP network (more specifically, a 3G upgrade thereof) to route packet data. In this dissertation, when we refer to GPRS, we are referring to this GPRS IP network. Figure 3 shows the overall network structure.

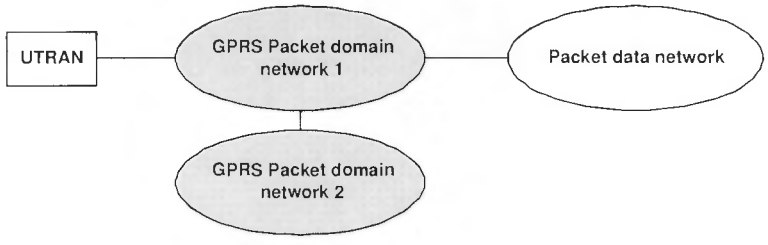

Figure 3: The overall packet domain structure (based on [2], page 20).

Figure 4 shows the core network elements that affect packet data in the GPRS network and introduces the following elements:

- Packet Data Server: Any land-based packet data server, such as an IP server or FTP server.
- PLMN: Public Land Mobile Network. This is a land-based network that services a mobile network. In this case, it is a private GPRS IP network that services

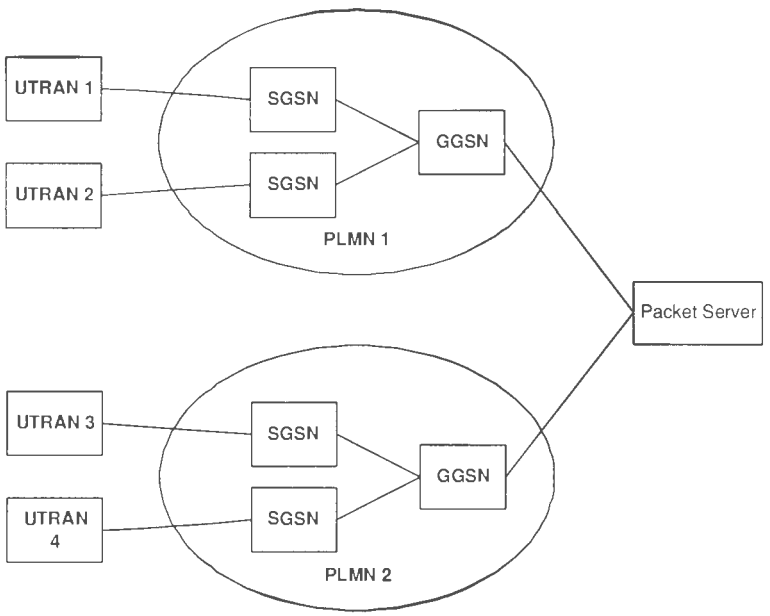

Figure 4: The GPRS network structure.

the UTRAN.

- GGSN: Gateway GPRS Support Node. This acts as a gateway between the external network and the GPRS network.
- SGSN: Serving GPRS Support Node. As its name suggests, these network elements serve individual users, acting as a proxy to allow for transparent data flow between the end user, or user equipment (UE), and the packet server.
- UTRAN: The access network between the CN and the radio link.

A local PLMN is served by a single GGSN, as shown in Figure 4, which can then serve multiple SGSNs, which in turn serve multiple UTRANs. The GGSN and SGSNs can either exist in the same logical unit physically, or can be distributed. The transport of data between the GSNs, as well as between the SGSN and the UTRAN, is performed using best-effort UDP/IP. The GPRS Tunnelling Protocol (GTP-U) is

used on top of the UDP/IP protocol [2, 22]. GTP tunnelling is used to transfer data transparently between the UE and the packet domain network. This means that packets are encoded with GPRS specific data when transported between the GSNs, and stripped of the aforementioned data when it is sent outside of the GPRS domain.

A specific SGSN keeps track of all users that require its service. This definition encompasses all mobile packet network users within the range of the base stations that use the SGSN in question. The SGSN performs basic functions such as security and access control.

2.2.2 UTRAN

The UTRAN provides the link between mobile users and the core network. It consists of a series of Radio Network Subsystems (RNS), each of which consists of one Radio Network Controller (RNC), which in turn controls one or more Node B's. RNC's are responsible for controlling the use of radio resources and ensuring their integrity. Node Bs handle radio transmission and reception to and from the user equipment [10]. Figure 5 shows the basic architecture of the UTRAN.

In terms of the OSI model, the functions of the link control layer, such as the provision of error free transmission, are carried out at the RNC. The main function of the physical layer, namely the transmission of raw bits, is performed at the Node B.

The RNC is responsible for the bulk of the processing required to send the data over the radio link, and for undoing the same processing in order to send data from the mobile user to the CN. There are several protocols that are involved in the RNC as part of the Layer 2 functionality. The first one that affects packet data passed down from layer 3 is the Packet Data Convergence Protocol (PDCP). The PDCP compresses IP headers. The compressed data are then passed to the Radio Link Control protocol (RLC), which segments the packet data into *transport blocks*, and then to the Medium Access Control protocol (MAC). These protocols will be explained in Section 2.4.

The Node B is the final network element that packet data must flow through on the downlink of the UTRAN. Packets are sent from the RNC in transport block

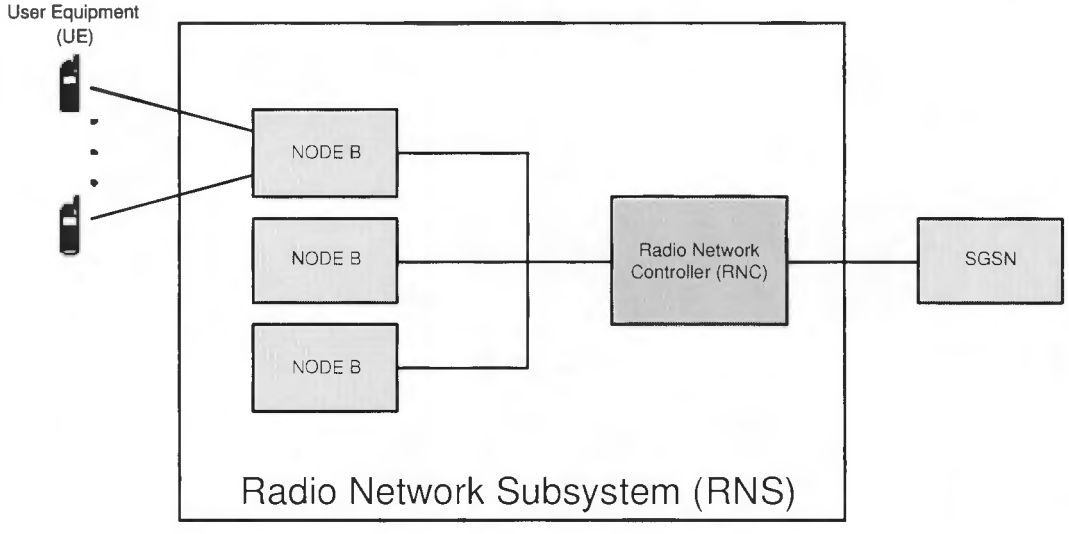

Figure 5: Overview of the UTRAN.

format, ready for transmission. The Node B is responsible for the modulation and transmission of these transport blocks. This will be discussed in Section 2.5.

2.3 CN Packet Domain Procedures

When a UE wishes to use the packet data functions of the cellular network, it first makes its presence known to the network by performing a `GPRS-ATTACH`, which is a network function that registers the presence of a UE with an SGSN that will provide the packet service for the UE. Once attached, the UE is then available for paging, SMS via GPRS, and notification of incoming packet data.

If the UE then wishes to send or receive packet data, it must activate a Packet Data Protocol (PDP) *context*, which determines the protocol that is used for the session. Example PDPs are IP and PPP (Point to Point Protocol) [2]. At PDP context activation, the SGSN will establish a PDP context for the UE, which is characterised by a context index, type, address and state. The PDP address is assigned either dynamically or statically by the GGSN, depending on the network implementation.

The GGSN then contains all routing information for packet delivery from itself to SGSN, which is the point of attachment for the UE to the IP network.

Once the PDP context has been activated, packet data transfer can begin. Data packets are encapsulated in GTP-U (GTP on the User plane) packets and travel from the UE, through the RAN, and finally to the SGSN in the PDP format that has been specified. For our purposes, we assume the PDP context to be IP. The IP packets are then transported via the GPRS PLMN to the GGSN that forms the gateway between the PLMN and the packet network that the UE wishes to access. The GTP (de-)encapsulation takes place at the GSNs and at the RNC, but the tunnelling information is removed once the packet has been sent outside of the GPRS network, using the IP address of the SGSN as its return address [22].

On the reverse or downlink, packets arrive at the GGSN, which contains routing information for the SGSN that is serving the UE. The packets are then encapsulated, and sent through the SGSN to the UE in much the same way as the uplink using GTP-U.

Once a packet has been processed by the SGSN, it is forwarded to the UTRAN, which utilises radio resources to send the packets to the mobile UE.

2.4 Link Layer Protocols

Packet data arrive at the RNC from the packet data network via the SGSN. The data is processed by the link layer protocols at the RNC, which carry out various functions such as channel mapping, segmentation and the allocation of transmission resources.

2.4.1 Link Layer Channels

Three types of channels are defined in the link layer: Logical, Transport, and Physical. These channels provide different views of the service provided at different levels of the protocol stack. Logical channels define the general service offered; transport channels define how these services are provided; and physical channels define how the data are physically transmitted.

Logical Channels

Logical channels are the channels that directly map data such as packet data to a data transfer service that is offered by the UTRAN. Each type of channel is defined by the type of information transferred, and is either a Control or a Traffic channel [7]. These channels are assigned by higher layer control protocols such as the Radio Resource Control Protocol (RRC), which also controls how the channels are mapped to transport channels.

There are two subgroups of logical channels, those for *traffic*, and those for *control*. Two types of logical traffic channels are provided, the Dedicated Traffic Channel (DTCH) and the Common Traffic Channel (CTCH). The CTCH is used for point-to-multipoint traffic for a group of UEs. The DTCH is a point-to-point channel dedicated to one UE. Since we are focusing on data being sent to individual users and not on data being broadcast to multiple users, the DTCH is of more interest than the CTCH.

Transport Channels

Logical channels must be mapped to transport channels. Transport channels are defined by the characteristics of the data being transferred. Transport channels are relevant at a lower level than the logical channels. In other words, they lie between the logical channel and the physical layer.

There are many kinds of transport channels, depending on the requirement of the user. Some are dedicated, such as the Dedicated Channel (DCH), some are shared, and used for small-scale data, such as the Random Access Channel (RACH), and some are shared and are used for bulk data like IP traffic, such as the Downlink Shared Channel (DSCH). The Broadcast Channel (BCH) is used for broadcasting over entire cells. All channels that carry data for one user are mapped from the DTCH. The BCH, for instance, would map from the CTCH.

Physical Channels

Physical channels are mapped from transport channels based on the requirements of the transport channel. These requirements will alter the spreading factor, the coding type, and the transport block size.

In UMTS, channels are defined by specific carrier frequencies, scrambling codes and spreading codes [3]. All downlink FDD channels are carried on the same carrier frequency. A *scrambling code* is a code similar to a *spreading code*, but is applied to all channels in one cell, in order to allow effective re-use of channel codes in neighbouring cells.

The transmitted channel takes the form of a 10 ms *frame.* Each frame is divided into 15 *slots*, each of which contains a transport block. Each frame in a specific channel is dedicated to one user, or, in the case of a shared channel, one user per frame.

Channel Mappings

Channels are mapped from logical to transport to physical in a very specific manner in the UTRAN. We narrow our focus to the mapping of channels that are involved in packet data transport. Since the packet data we are interested in is intended for a unique user and is not broadcast data, the choice for logical channel is the DTCH.

The case study in [11] outlines the issues which must be considered when choosing a transport channel for packet data flows. The two channels that are applicable to downlink packet data traffic are the DCH and the DSCH. Both have their advantages and disadvantages, and find uses in different areas of packet services: the DCH, because of its dedicated nature, can be used as a constant bit-rate carrier of packet traffic, and would thus be well suited to streaming, realtime traffic. On the other hand, the DSCH has been designed to be an efficient means of sharing valuable resources amongst multiple non-realtime, bursty, asymmetric traffic flows. The DSCH has high bit-rate codes assigned to it to be shared amongst such users [15]. The DSCH is therefore the channel of choice for non-realtime IP traffic, which is our chosen area of investigation. Therefore, the mapping from logical to transport channel is DTCH

to DSCH for IP traffic. The DSCH maps directly to a Physical DSCH (PDSCH), which, depending on the data rate, has a pre-defined spreading factor, as well as a predefined transport block size and coding algorithms. These details are explained in Section 2.7.3, Table 2.

2.4.2 PDCP

The PDCP protocol provides header compression and decompression of IP data streams at both the transmitting and receiving entities [9]. The compression is used in order to streamline packet services, and can compress UDP/IP headers of 48 bytes to less than 10 bytes [11, 22]. It does so by using the IETF compression protocol defined in RFC 2507 [39]. This protocol keeps track of all packets belonging to any one flow, i.e. packets that are all coming from the same source. Since the packets have similar, if not identical, data in their headers, certain elements can be discarded. Figure 6 shows the data (PDU) formats in the transfer of data between Layer 3 and Layer 2.

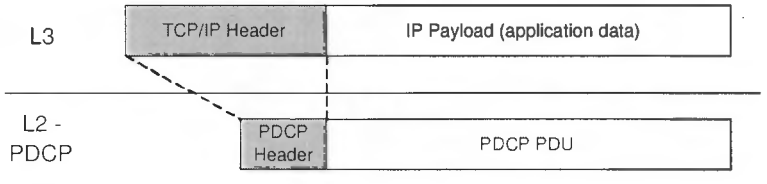

Figure 6: The PDCP protocol.

2.4.3 Radio Link Control (RLC) Protocol

The Radio Link Control protocol is responsible for a number of important functions:

- Segmentation/Reassembly of IP packets into RLC PDUs (Protocol Data Units);
- Error Correction;
- In-sequence Delivery of upper layer PDU's;

- Duplicate Detection and Flow Control;
- Logical channel assignment.

RLC Modes

The RLC can perform its functions in one of three modes: transparent mode (TM), unacknowledged mode (UM), and acknowledged mode (AM). In all three modes, basic segmentation and reassembly is supported. In TM, no additional protocol information is appended to the upper layer PDUs. UM provides basic protocol functions such as error detection. It does not, however, provide guaranteed delivery, and thus will only pass on to upper layers those PDUs, which are error free. When using AM, the RLC guarantees delivery of PDUs, as well as the service encompassed by UM and TM. The PDUs are guaranteed not only to be delivered, but also to be delivered error free. Thus, Automated Repeat-Request, or ARQ, is used. ARQ utilises Forward Error correction (FEC) and retransmissions to manage errors that occur on the physical channel. FEC is discussed in detail in Section 2.5, page 36.

Of the three transfer modes, only AM and UM are supported by packet services [2]. Of these two, UM is more suited to delay sensitive traffic, as it does not provide costly error correction, while AM is more suited to error sensitive traffic as it utilises in-order and error free service [11].

RLC Segmentation

RLC segmentation occurs according to certain rules specified by the 3GPP specifications [12]. An example of this is shown in Table 1. The Protocol Data Unit (PDU) size is shown for the Downlink Shared Channel (DSCH). This segmentation will produce the data blocks that will eventually become transport blocks. Table 1 includes the size of each RLC PDU, and how long the header is. It can be noted that the DSCH is used in AM, and therefore has a larger header than UM or TM based channels.

Figure 7 shows the data transfer from the PDCP to the RLC protocol.

	384 Kbps	2048 Kbps
Logical channel type	DTCH	DTCH
RLC mode	AM	AM
Payload sizes, bit	320	640
PDU header, bit	16	16

Table 1: RLC Parameters for a DSCH channel. The two possible channel bit rates, 384 kbps and 2048 kbps, are shown.

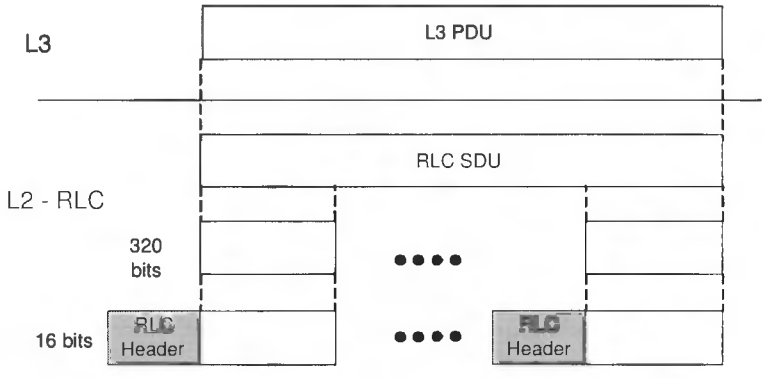

Figure 7: The RLC protocol. This example shows the attachment of an AM header. The blocks are segmented into 320 bit segments, with a 16 bit header, as is the case for the 384 bps example in Table 1.

2.4.4 Medium Access Control (MAC) Protocol

The MAC protocol is divided into several MAC entities:

- MAC-b: Controls the broadcast channel (BCH).
- MAC-c/sh: Controls all common and shared channels, such as the DSCH.
- MAC-d: Controls the dedicated channel (DCH).
- MAC-hs: controls the high speed packet service.

The MAC protocol is responsible for the following functions [6]:

- **Logical Channel to Transport Channel mapping:** Data flows, having been assigned a Logical Channel type in the RLC protocol, must be mapped to a Transport Channel.

- **Transport Format selection and implementation:** Transport Formats are defined as the formats offered by the Physical Layer to the MAC protocol and vice versa for the delivery of Transport Blocks during a transmission time interval (TTI) on a transport channel. Transport Formats define the parameters of the channel and are assigned according to the traffic class of the data flow. The class to which a traffic flow belongs is indicative of the nature of the traffic: the *Conversational Class* includes traffic that is sensitive to delays and has a conversational pattern, such as voice and video traffic; the *Streaming Class* includes packet-switched voice and video traffic; the *Interactive Class* includes all services that follow a request-response pattern, such as web browsing traffic; and the *Background Class* includes all data that does not need to be received within a specific time frame, such as downloading emails. By organising traffic into classes, radio resources can be better managed in order to cater for each class.

- **Scheduling:** The MAC-d and MAC-c/sh entities are responsible for scheduling traffic on dedicated and shared channels respectively.

The MAC layer has other functions that fall beyond the scope of our work, such as traffic measurement.

Figure 8 shows the effect of the MAC protocol on the upper layer PDU. The contents of the MAC header vary according to the type of transport channel assigned to the data flow, but MAC headers are invariably used for UE identification. Dedicated Channels (DCH's) used without multiplexing require no MAC header, as the identification of the UE is implicit and is based on the spreading code (described is Section 2.5). A DSCH is shared and thus UE's cannot distinguish their data from the data of other UE's, and therefore require an identification field in the MAC header.

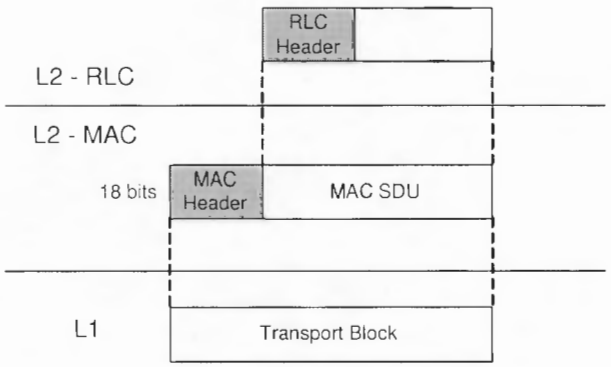

Figure 8: The MAC protocol. This example shows the 18 bit header being attached.

2.5 Physical Layer Protocol

The Physical Layer (PHY) is responsible for the transmission of bit streams on the physical medium [7]. It handles error detection, forward error correction encoding, power control, and the modulation and spreading of channels.

2.5.1 Channel Error Control

Automated Repeat-Request (ARQ) error-control schemes are used in most mobile communications networks to compensate for the fact that errors can occur frequently on radio channels. Most schemes are simple acknowledge-retransmission schemes that resend data that has been corrupted by waiting for acknowledgements or non-acknowledgements from the receiver. Other schemes use Forward Error Correction (FEC)[1], which encodes packets with redundant data that allows for the detection and correction of multiple errors.

While the actual encoding and decoding of the transport blocks takes place in the physical layer, it is the RLC protocol that controls the ARQ process. The physical layer encodes data only if the RLC protocol demands it. If the RLC protocol is operating in Transparent Mode (TM), no encoding will take place in the physical

[1]Also known as *Feed-Forward Error Correction*

later. If it is operating in Unacknowledged Mode (UM), encoding will occur, but the RLC protocol will not initiate the retransmission of data that has not been restored by the FEC. In Acknowledged Mode (AM), both encoding and retransmission will occur.

Error detection occurs with the use of a CRC attachment. In UMTS, two types of error *correcting* codes are prescribed, *convolutional coding* and *turbo coding*. They both employ methods of FEC, i.e. encoding redundant data into a data block. The parameters of the encoding schemes are fixed in UMTS [4], but there is flexibility: one can employ $\frac{1}{2}$ or $\frac{1}{3}$ rate convolutional coding, or $\frac{1}{3}$ rate turbo coding. The designation ($\frac{1}{2}$, $\frac{1}{3}$) of the coding scheme refers the *coding rate*, which determines how much redundant data each scheme encodes into the transmitted messages. If K is the length of a block of data before encoding, and if Y is the length of the block after encoding, then the following formulae can be applied for the different encoding schemes:

- $\frac{1}{2}$ rate convolutional coding: $Y = 2K + 16$
- $\frac{1}{3}$ rate convolutional coding: $Y = 3K + 24$
- $\frac{1}{3}$ rate turbo coding: $Y = 3K + 12$

Turbo Coding

The UMTS specifications note that the DSCH employs only $\frac{1}{3}$-rate Turbo coding. Turbo codes were first introduced in 1993, and are a very simple yet powerful extension of Recursive Systematic Convolutional (RSC) codes, which use generator polynomials to encode redundant data into transmissions. Turbo encoders employ two RSC encoders, one that encodes the original data, and one that encodes data that has been passed through an *interleaver*. The interleaver alters the sequence of bits in the original data block in a manner that is deterministic and is easily reversed by the decoder, such as shifting bits by one place, or applying a pseudo-random process that produces an almost random interleaving of bits. This pseudo-randomness allows for Turbo codes to approach the theoretical channel capacity limit proposed by Shannon.

The turbo coder specified for UTRA-FDD channels is shown in Figure 9. The input data bits are $\mathbf{x_K} = \{x_1, x_2, x_3, ..., x_K\}$, where K is the size of the input block. The data is passed through the interleaver to produce two sets of data to pass through the encoders, $\mathbf{x_K}$ and $\mathbf{x'_K}$.

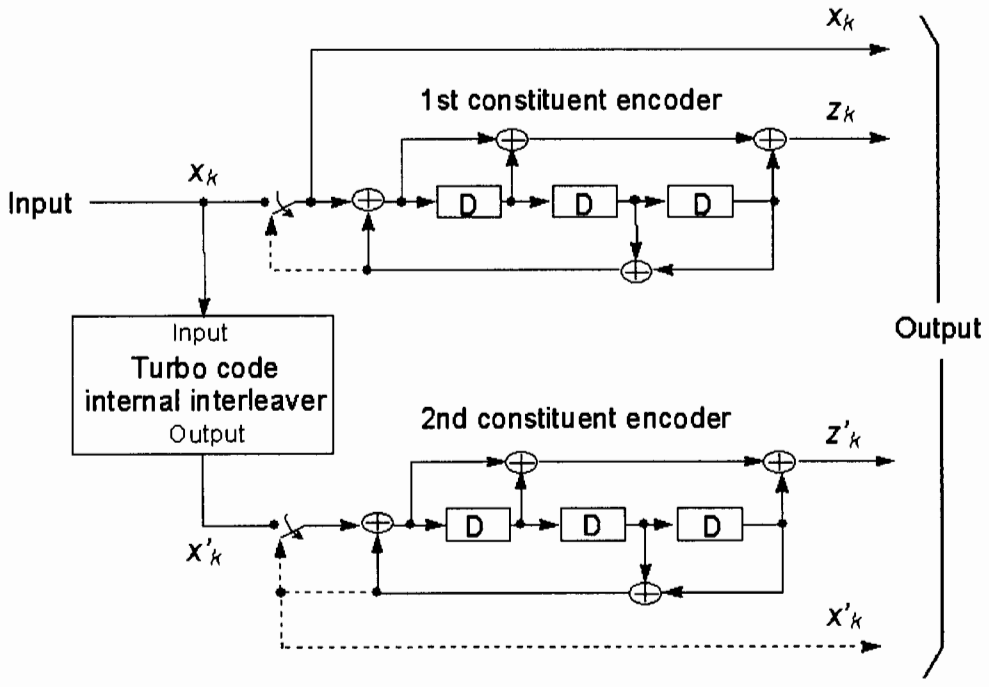

Figure 9: The UTRA-FDD turbo coder (Figure taken from [4]).

The generator function for the first convolutional encoder is $g_0(D) = 1 + D^2 + D^3$, and the generator for the second encoder is $g_1(D) = 1 + D + D^3$. These two encoders produce output bits $\mathbf{z_K} = \{z_1, z_2, z_3, ..., z_K\}$ and $\mathbf{z'_K} = \{z'_1, z'_2, z'_3, ..., z'_K\}$. Also required are the *trellis termination bit sequence*, which is the sequence of bits that is required to zero the encoders after the data has been passed through it. These bits are denoted by

$$x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}$$

The combined output of the encoder, and thus the data to be transmitted, is

$$\mathbf{x_{TX,K}} = \{x_1, z_1, z'_1, ...x_K, z_K, z'_K, x_{K+1}, z_{K+1}, ...x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, ..., x'_{K+3}, z'_{K+3}\}$$

It is obvious from this that given K input data bits, $3K + 12$ bits will be transmitted. The details of the interleaver are given in TS 25.212 [4].

2.5.2 Channel Spreading with DS-CDMA

Possibly the most important aspect of UMTS-based mobile communications systems is the multiple access technique that has been specified by the standardisation bodies. UMTS uses Wideband Direct Sequence CDMA (W-DSCDMA), which is a multiple access technique that differentiates separate signals based on individual codes rather than timeslots (TDMA) or frequencies (FDMA).

The CDMA method requires that each user signal be assigned a code, known as a *spreading code* or *Pseudo-Noise code (PN code)*, which is unique to that signal, and is orthogonal to all other spreading codes. When a user's signal is multiplied by the code, the effective bandwidth is widened, because the spreading code is a wideband signal, while the user's signal is narrowband. The duration of the PN code is known as the *chip length.* This new signal appears as noise to all other users, but, when multiplied by the same code as was used to spread the signal, the original signal is retrieved.

The following example from Haykin [32] illustrates this concept. Suppose that our user signal, *b(t)*, is a signal that has values between 1 and -1, such as in Figure 10 (a). The PN code can be represented by the signal *c(t)* (Figure 10 (b)). When the signal $b(t)$ is multiplied with $c(t)$, the transmitted signal $m(t)$ is created, which has a spectrum similar to that of $c(t)$ as in Figure 10 (c). This process is known as *spreading* because the original signal has been spread over a higher spectrum range.

The received signal, $r(t)$, will be $m(t)$, plus additive interference $i(t)$. The received signal is therefore:

$$r(t) \quad = \quad m(t) + i(t) = c(t)\, b(t) + i(t) \tag{1}$$

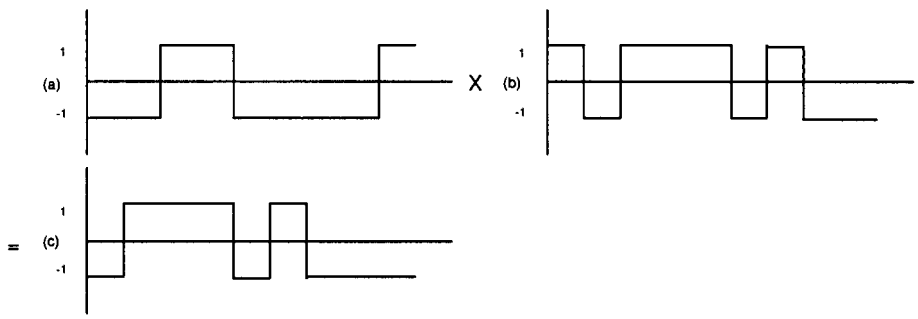

Figure 10: The spreading process.

This signal is then multiplied by the PN code ($c(t)$), producing the following:

$$z(t) = c(t)\, r(t) = c^2(t)\, b(t) + c(t)\, i(t)$$

Since $c(t)$ is a signal that only has values 1 or -1, $c^2(t) = 1\ \forall t$. Therefore,

$$z(t) = b(t) + c(t)\, i(t)$$

In order to extract the signal $b(t)$, note that $b(t)$ is a narrowband signal, and that, because it is multiplied by $c(t)$, the interference is wideband. A baseband filter with a wide enough bandwidth can therefore be used to recover $b(t)$.

The transmitted signal will appear as noise to any receiver that does not know the PN code $c(t)$. It must also be noted that the interference part $i(t)$ is the limiting factor for the system, because as it grows, or becomes increasingly powerful, the ability of the filter to extract $b(t)$ decreases. Using CDMA therefore means that our system is *interference limited* and not noise limited [32, 45]. The application of spreading in UMTS is shown in figure 11.

On the downlink (as on the uplink), most channels are modulated using Quadriphase-shift Keying (QPSK) which splits the signal into two components, known as the (I, Q) pair. The components are multiplied by the channelisation code $C_{ch,SF,m}$ which is the m^{th} channelisation code with *spreading factor* SF. The SF of a code defines the degree to which the bandwidth of the original signal is spread. For instance, in the

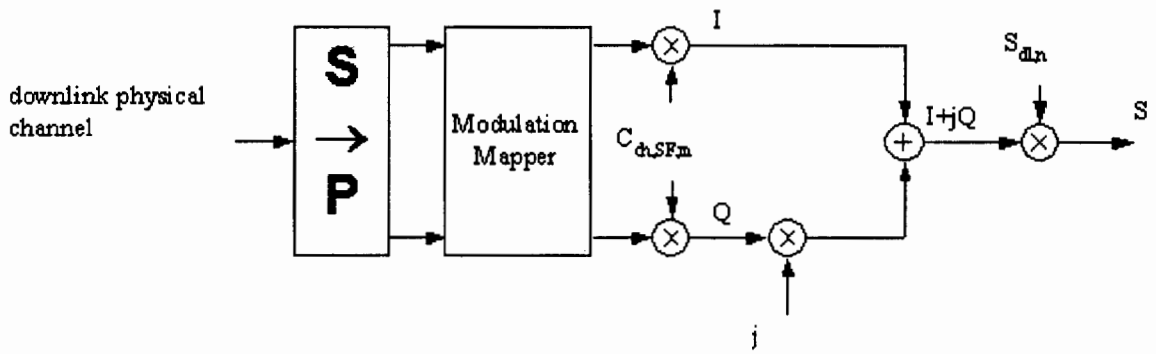

Figure 11: The spreading process on the downlink [5].

example in Figure 10, $c(t)$ has a bandwidth twice that of $b(t)$, and thus a spreading factor of 2.

The Q component of the QPSK is multiplied by $j = \sqrt{-1}$. The I and jQ components of the QPSK are then added and treated as one signal. This signal is multiplied by the *scrambling code*, $C_{dl,n}$, for the downlink Node B serving the UE. Different scrambling codes are used for different base station transmitters to limit inter-cell interference. This process has as its result the signal S, as is shown in Figure 11.

2.5.3 Channel Modulation and Transmission

The signal S in Figure 11 is combined with the other signals being transmitted. Each signal S_i is first weighted by a weight factor G_i, and then summed with all signals. The summed signal, point T in Figure 12, is then modulated and transmitted.

2.6 Radio Link Conditions

The final link in the chain stretching from an IP server to the end user is the mobile radio link. This link provides some of the more challenging problems for network designers due to the fact that errors occur randomly and with far greater frequency than

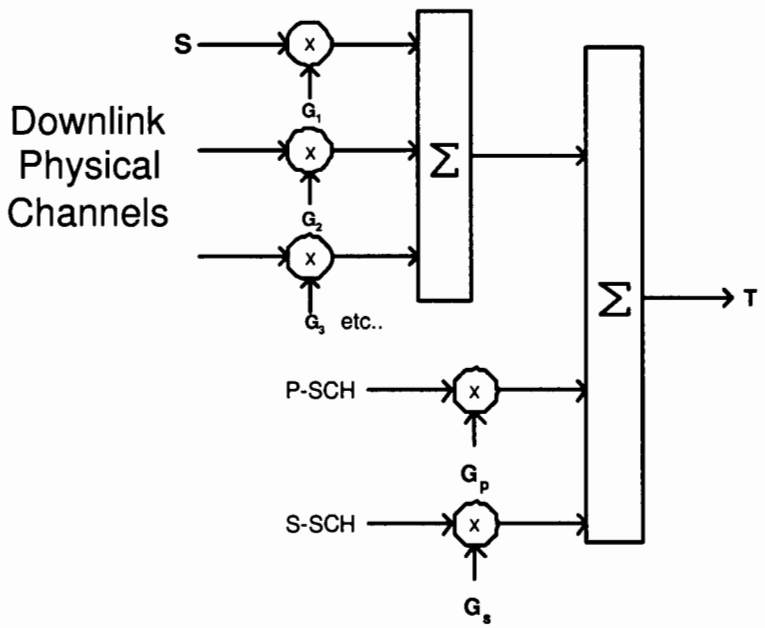

Figure 12: The channel combination process on the downlink [5].

wired links. Radio signals undergo distortion, which is due to a number of factors. However, a type of distortion known as *fading* dominates, and can cause high bit error rates. Another factor influencing transmissions is *Multiple Access Interference* (MAI). This occurs due to the fact that spreading factors are not always orthogonal. The next two sections deal with these phenomena.

2.6.1 Fading

When multiple versions of the same signal interfere with one another, there will be rapid fluctuations of amplitudes and phases of the received signal. This phenomenon, and its effect on the received signal, is known as *small scale, multipath fading.*

There are several factors that directly influence fading in radio channels [45]:

- **Multipath propagation:** this describes multiple copies of a signal arriving at a receiver due to reflections off obstructions in the line of sight between the

transmitter and the receiver.

- **Speed of the mobile device:** if a mobile UE is moving at high speed, this will cause Doppler shifts on the components of the multipath signals.
- **Speed of objects in the environment:** if objects in the line of sight of the transmission are moving, then signals that are reflected off of them will have distortions due to Doppler shifts.

Fading effects can be characterised by multipath time delay spread and Doppler spread. Multipath time delay spread determines whether a channel experiences *flat* or *frequency selective* fading. Doppler spread determines whether a channel experience *slow* or *fast* fading.

Flat fading occurs when all of the spectral components of a signal are equally affected by fading. The *coherence bandwidth* of a channel defines the bandwidth over which a channel will experience flat fading. If the bandwidth of a transmission is less than the coherence bandwidth, the channel experiences flat fading. However, if the transmission bandwidth is *greater* than the coherence bandwidth, the channel will experience *frequency-selective* fading. Frequency-selective fading is more difficult to model, as it does not affect all parts of the transmission equally. Flat fading channels are often referred to as narrowband channels, and frequency-selective fading channels are known as wideband channels.

The rate at which fading fluctuates determines whether or not the channel experience fast or slow fading. If the bit period is greater than the average period of a fade, the channel is fast fading. Otherwise, it is slow fading.

As its name implies, WCDMA generally experiences frequency-selective fading. However, Dinan *et al* [23] explain that the radio-link conditions in an indoor or pico-cell are such that the signal experiences flat fading. Higher UMTS data rates also dictate that, since the bit period is far smaller than the fading duration, the channel will experience slow-fading. Since we are concerned with the behaviour of the network in pico- and micro-cells at high data rates, we have modelled flat, slow fading.

Flat fading channels are also known as *amplitude varying channels.* The distribution of the variation of the amplitudes in these channels can be modelled with one of

two statistical distributions, depending on the environment. If all reflected signals are received with similar signal strengths, then the fading is known as *Rayleigh fading*, as the envelope of a single multipath component can be characterised by a Rayleigh distribution. Figure 13 illustrates this phenomenon: the signal sent from the Node B is reflected off of surrounding obstacles, in this case buildings. Multiple reflections therefore arrive at the UE.

The Rayleigh probability density function (PDF) is given by the following formula:

$$p(r) = \frac{r}{\sigma^2} exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{2}$$

That is, the above equation describes the probability that a Rayleigh faded signal has amplitude r, given that σ^2 is the time-average power of the received signal. Rayleigh fading is likely to occur in an outdoor environment where the line of sight between transmitter and receiver is obstructed, and the receiver is most likely moving.

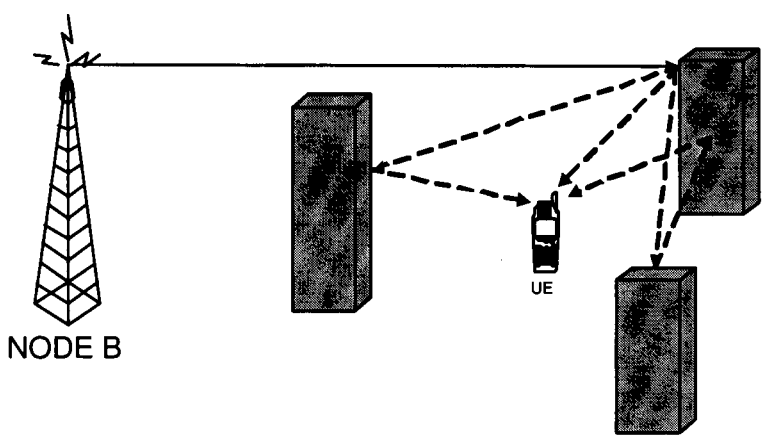

Figure 13: A simple demonstration of Rayleigh fading.

When all signals are received with equal power except for one signal, which has a superior signal strength and is dominant over other reflected signals, the fading is known as *Ricean fading*. This is because the envelope of a single multipath component can be characterised by a Ricean fading distribution. The Ricean probability density function is given by the following formula:

$$p(r) = \frac{r}{\sigma^2} e^{\left(-\frac{(r^2+A^2)}{2\sigma^2}\right)} I_0\left(\frac{Ar}{\sigma^2}\right) \quad (3)$$

Here, A is defined as the peak amplitude of the dominant signal, and $I_0(x)$ is the Bessel function of the first kind of order zero. Ricean fading is most likely to occur in an indoor or pico-cell environment, where there is a line of sight between the transmitter and the receiver, and the receiver is not moving. This is illustrated by Figure 14: the signal sent from the Node B is reflected off of surrounding buildings; multiple reflections therefore arrive at the UE, all with similar signal strength, except for the signal at point A, which has a direct line of site to the UE.

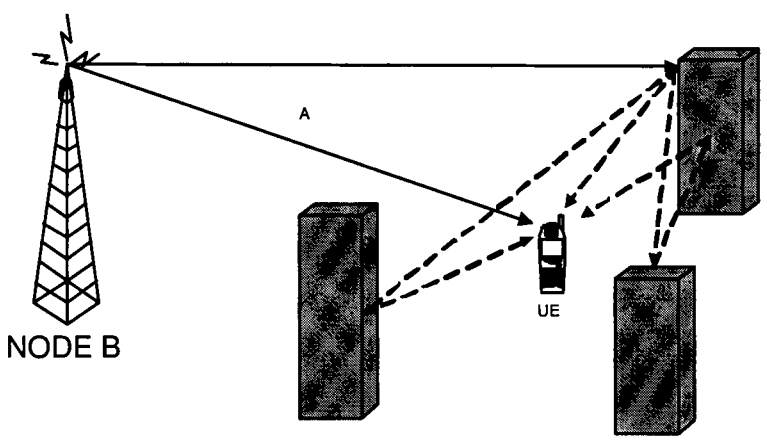

Figure 14: A simple demonstration of Ricean fading

When one considers the type of environment being examined in this study, it is obvious that Ricean fading would occur in the majority of cases. This follows when one considers that IP traffic users (i.e. HTTP, email, FTP) would most likely not be moving, and would in fact be stationary and located in an area where the data rate would be most beneficial, which would be in an indoor or pico-cell environment.

The importance of Equations 2 and 3 will become evident in Chapter 4, where our model of the radio link is discussed.

2.6.2 Multiple Access Interference (MAI)

Multiple Access Interference (MAI) arises when signal-spreading codes are not completely orthogonal. There are two types of MAI on the downlink in cellular systems: *inter-cellular* and *intra-cellular* interference [22].

When the transmission of multiple, simultaneous channels takes place on the downlink, two steps are defined [22]. In the first step, the base station applies spreading codes to each channel, making sure that orthogonality is maintained. However, due to multipath fading, this orthogonality is not always perfect, and therefore intra-cellular MAI will occur. It will increase as the number of users increases (Section 2.5.3, and in particular Figure 11, has already mentioned this). The second step involves combining the individual signals into one signal, and multiplying the result by a *scrambling code*, which helps to ensure that inter-cellular interference, that is, the interference that is experienced due to the presence of other Node B's, is kept to a minimum. Each Node B has a unique scrambling code.

Inter-cellular interference is minimal due to the use of scrambling codes. Intra-cellular interference, however, is of great concern, as it increases as the number of users increases. This fact will have to be taken into account. Figure 15 illustrates inter- and intra-cellular MAI.

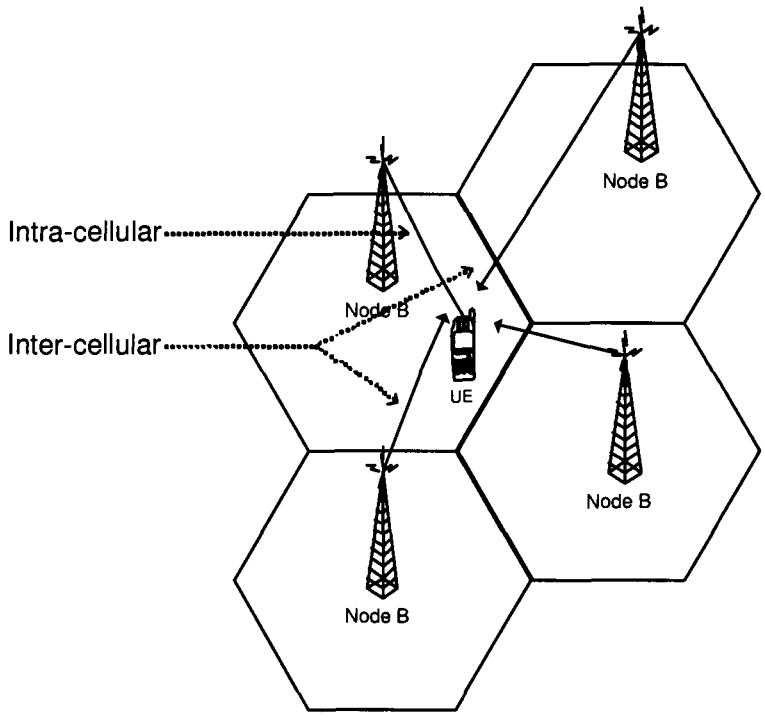

Figure 15: Inter and Intra-cellular interference.

2.7 Packet Data Services

Our interest in the packet data services available in UMTS networks has already been mentioned. This section shows how packet data is transported through the various stages of a typical UMTS CN and UTRAN. Section 2.7.3 will also give the various parameters of the channels and resources involved in the transport of IP traffic.

2.7.1 Packet Data Flow through the CN

Figure 16 shows which network elements in the GPRS PLMN through which packets must pass on the downlink.

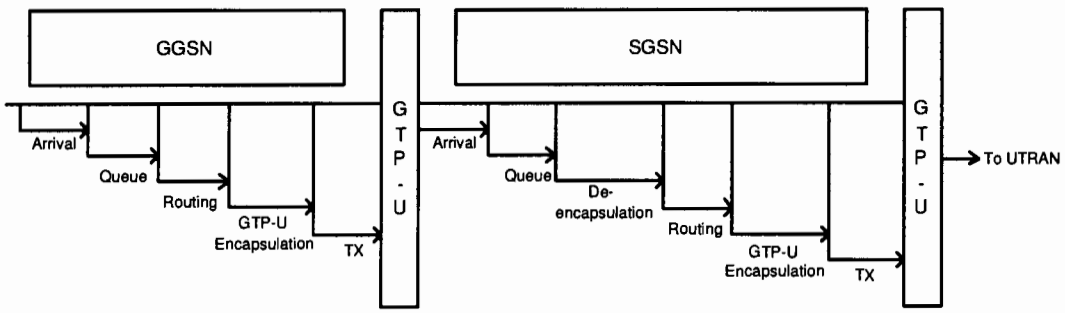

Figure 16: Timeline of data flow through the GPRS PLMN (TX = *Transmission*). GTP-U implies GTP-U over UDP/IP.

2.7.2 Packet Data Flow through the UTRAN

The diagrams in Figure 17 summarise the flow of packet data through the UTRAN RNC and Node B respectively, and where time is spent in each stage.

2.7.3 IP Service Resources and Parameters

IP data follows the path defined in Figures 16 and 17. It is mapped from a DTCH logical channel to a DSCH transport channel, which is then mapped to a PDSCH physical channel. The various parameters of this process are given in Table 2.

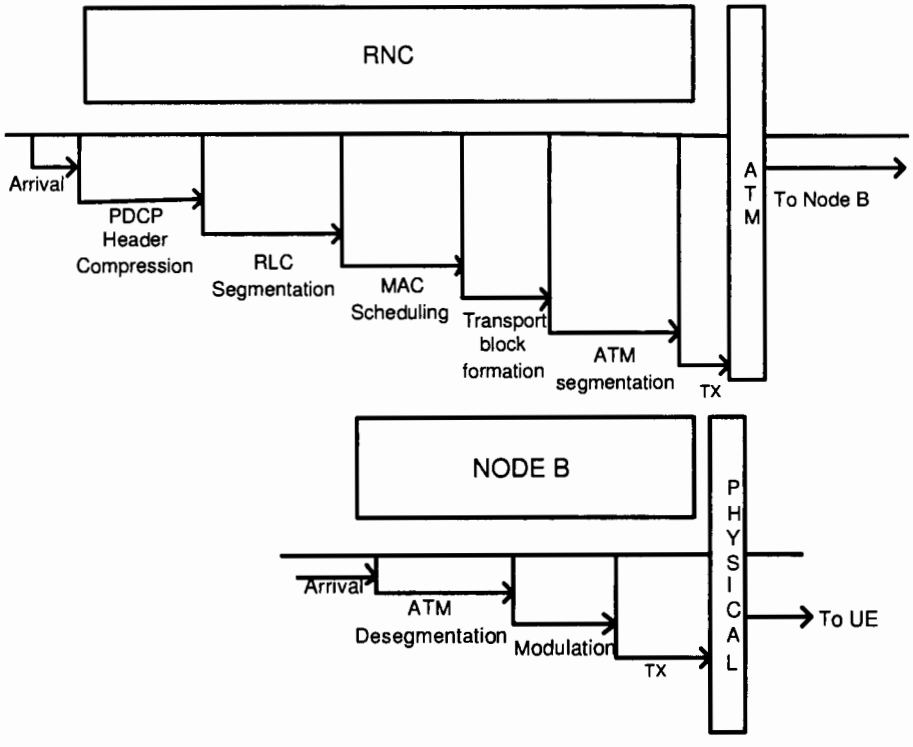

Figure 17: Timeline of data flow through the UTRAN RNC and Node B(TX = *Transmission*).

2.8 Summary

We have described the UMTS network that we have modelled. We have focused on the transport of IP traffic and the network elements and protocols that affect IP traffic flow. IP traffic first passes through the GPRS PLMN, which handles routing of data between the Internet and end users. The data is then forwarded to the RNC, which controls the PDCP, RLC and MAC protocols, which are responsible for the compression, segmentation, and scheduling of the IP packets. The IP data segments, called transport blocks, are sent to the Node B that is serving the end user in order to be transmitted. The transport blocks are first encoded using Turbo coding, which are then spread using spreading codes. The spreading codes arise due to the fact that the multiple access technique employed by UMTS is WCDMA. Transmissions

PLMN Transport Protocol	GTP-U over UDP/IP
Logical Channel	DTCH
Transport Channel	DSCH
PDP Context	IP
RLC Mode	AM
RLC Payload Size	320 bits (384 Kbps) or 640 bits (2048 Kbps)
RLC Header	16 bits
MAC Header	18 bits
MAC Multiplexing	Frame by frame basis
Coding Type	Turbo coding
Spreading Factor (SF)	8 (384 Kbps) or 4 (2048 Kbps)

Table 2: Network parameters and resources for the transport of IP traffic through a typical UMTS network.

on the radio link are affected by flat, slow, Ricean fading as well as multiple access interference (MAI), the latter being due to a loss of orthogonality of the spreading codes.

Chapter 3

IP and Link Layer Model

In order to discuss the aspects of queueing theory that will be used in our analytical model, certain fundamentals of queueing theory must be defined. The discussion in this text is minimal; for a thorough treatment of the subject, see the books written by Gross and Harris [28], Kleinrock [35] and Tijms [31].

3.1 Queueing Theory Fundamentals

Queueing theory describes the set of problems that involve *customers* waiting in a *queue* for a certain *service*. This fairly broad field can be applied to the modelling of a variety of problems, such as scheduling in a router, or traffic flow on a busy road system. One way to look at a queueing system is as a *system of flows* [35], in which some commodity moves through one or more channels to get from one point to another. These flows can either be *deterministic* (steady rates of flow) or *stochastic* (random flow patterns). The following are fundamental descriptors of queueing systems:

- **Customer arrival process:** The arrival of customers into the queue may be stochastic. Therefore, the arrival distributions must be known in order to model the system. Customers can arrive singly or simultaneously (known as *batch* or *bulk* arrivals) with a specific distribution for the batch sizes. The behaviour of the customer on reaching the queue can also be studied, as the customer can decide not to enter the queue upon arrival (*balking*), or can leave after a short while after losing patience (*reneging*). An example of a stochastic arrival process is the Poisson process.

- **Service process:** The length of a queue depends not only on the arrival rate of customers *into* the queue, but also on the rate at which customers *leave* the queue. This rate is determined by the service pattern of the system, which describes the distribution of the time taken to serve customers. This is stochastic and, as with the arrival pattern, can be single or bulk.

- **Queueing discipline:** Queueing disciplines define the manner in which customers are selected from the queue for service. The queueing discipline plays

an important role in the overall performance of the system, as it has to provide the best possible service for the particular situation in which it is placed. For example, an IP server that is implementing stringent QoS will have to select users based on priority, whereas a telephone exchange may select users based on when they arrived in the queue (First Come First Served or FCFS).

3.1.1 Queueing Theory Notation

It is convenient to describe a queueing system using a uniform notation. This notation takes the form **A/B/C/D/E**, where **A** describes the arrival pattern, **B** describes the service pattern, **C** describes the number of servers available, **D** describes the system capacity, and **E** describes the queueing discipline. Not all of the symbols are necessary in order to specify a queueing system. For example, if no capacity is specified, then it is assumed to be infinite, and if no queueing discipline is specified, it is assumed to be a FCFS queue.

In order to understand the notation, consider the M/M/1 queueing model: this model has a Poisson arrival and service process (denoted as M), and is served by a single server. The arrival and service processes could also be Erlang (E_k), Deterministic (D), Phase type (PH) or General (G).

3.1.2 Basic Performance Measures

The basic performance measures that follow can be applied to queues of the class G/G/1. They are measures that therefore apply to a broad spectrum of queueing models.

We denote the average rate at which customers enter the queue by λ, and the average service rate (or the rate at which customers leave the server) by μ. The utilisation factor of the system, ρ, is

$$\rho = \frac{\lambda}{\mu} \tag{4}$$

The parameter ρ can also be described as a measure of traffic congestion.

Of interest to us are the long run averages of the system, which are the average waiting times, W_q (the waiting time in the queue) and W (the waiting time in the queue plus the service time), as well as the average number of customers, L_q and L.

An important result, known as *Little's Formula*, comes from the relationship between the average waiting time in the queue, W_q, and the average number of customers in the queue, L_q. This relationship states that the average number of customers in the queue is equal to the average rate of arrival multiplied by the average waiting time in the queue, and can be written as

$$L_q = \lambda W_q$$

this can be extended to the system as a whole:

$$L = \lambda W$$

3.2 Applications to UMTS

3.2.1 Overview

We have stated that the objective of this work was to use queuing theory to model the flow of packet data from an external network through to a base station or Node B (which has been define in Chapter 2). Figure 18 shows a basic overview of the network to be modelled, with the queueing-relevant details highlighted. The arrival processes are shown at points (a), (b), and (c), with service processes at points (d), (e), and (f).

The arrival process at *(a)* is characterised as a packet data flow arriving at the gateway node (GGSN). Since the GGSN serves only one SGSN, the arrival process at *(b)* is characterised as the *departure process* of the GGSN. In the same way, the arrival process at the RNC is the departure process at the SGSN. Processing at (d) and (e) are basic routing and packet session management as described in Chapter 2. Processing occurs at the RNC in the PDCP and RLC layers before entering a queue

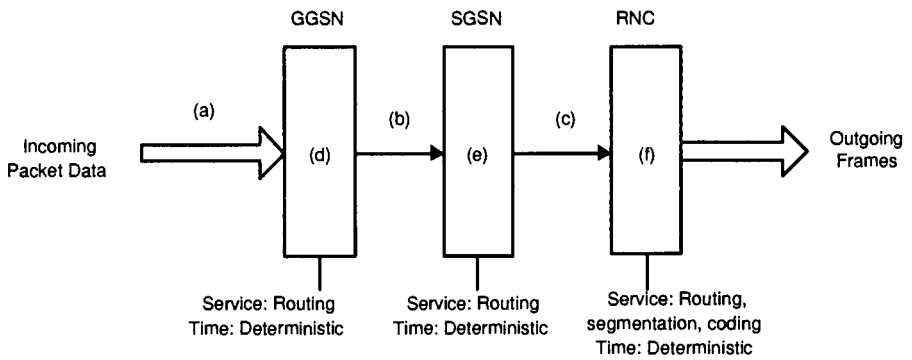

Figure 18: An overview of the GPRS/RNC domain of UMTS.

in the MAC layer that handles the scheduling of the packets onto the shared channel. The details are summarised in Table 3.

Network Element	**Arrival Process**	**Service**
GGSN	IP Arrival process into UMTS network	Routing
SGSN	GGSN Departure process	Routing Authentication Session Management
RNC	SGSN Departure process	PDCP RLC MAC

Table 3: Summary of arrival and service requirements.

3.2.2 Service Process

In order to simplify the above scenario, we note that the GGSN and SGSN perform simple routing functions. Since the processing that occurs in the RNC is significantly more complex, the service delays at the GSN elements can, for our purposes, be considered negligible. The GSN queues are therefore $G/0/1$ queues. Thus we can

assume that the arrival process at points (a), (b) and (c) are the same. The simplified system is shown in Figure 19.

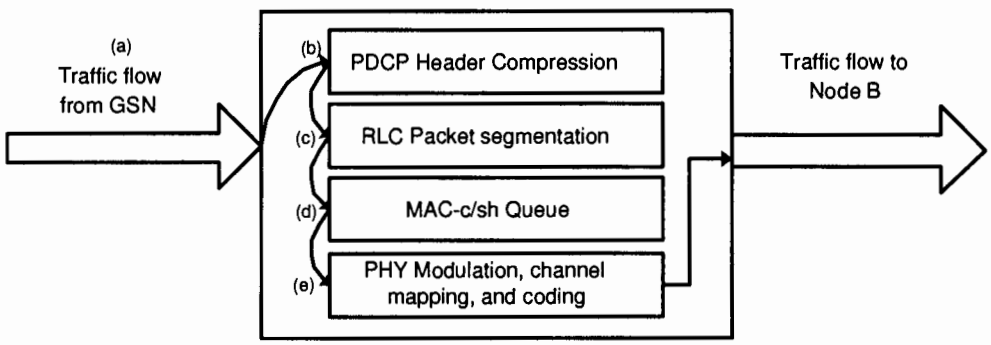

Figure 19: A simplified overview of the RNC queueing system.

IP packets arrive at point (a) from the GSN network. Processing occurs at points (b) and (c), which are the PDCP and RLC protocols respectively. The RLC protocol segments the IP packet into a batch of transport blocks, which arrive at the MAC-c/sh queue for scheduling and transmission. This is summarised in Table 4.

Network Element	**Arrival Process**	**Data Unit**	**Service**
PDCP	IP Arrivals	IP Packet	Header Compressions
RLC	PDCP Departures	PDCP SDU	Segmentation
MAC	RLC Departures	Transport Blocks	Scheduling

Table 4: Simplified summary of arrival and service requirements.

It must be noted that the bulk of the processing occurs *after* the packets have been processed by the PDCP and the RLC: the PDCP protocol simply decides whether or not to omit certain header fields, and the RLC protocol segments data into blocks according to a predefined pattern. On the other hand, all of the channel and error control coding occurs on a block by block basis in the MAC and PHY layers. Therefore, the processing in the PDCP and RLC layers can be treated in the same manner as the GSN network elements. However, they do have a subtle effect on performance

due to the fact that each layer applies a header to the packet, decreasing the effective throughput of the system.

We have therefore further simplified the model: IP packets arrive at the MAC layer as transport blocks of equal size; the number of blocks is dependent on the size of the packets. The blocks are then processed and transmitted. This is shown in Figure 20.

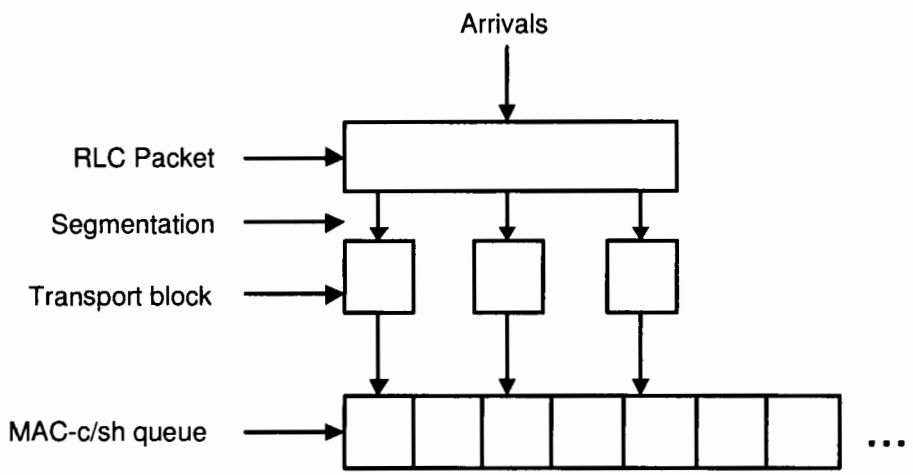

Figure 20: The model with further simplifications.

Analysing the processing in the MAC layer poses interesting problems. The basic processing that occurs for each block is the same as for each other block, which implies a deterministic service pattern. However, complications arise due to the fact that blocks are transmitted at 10ms intervals, which means that the delay experienced by each block must be rounded to the nearest 10. It must be noted then that the service process is therefore not strictly deterministic, but for the sake of simplicity we will model the system under this assumption.

Following from these arguments, the UMTS packet service network can be modelled as a $G/D/1$ queueing system. The rationale for our arrival process(es) follows.

3.2.3 Arrival Process

It is tempting to model the arrivals with a batch arrival process, due to the fact that each IP packet arrives at the MAC layer after being segmented into blocks. However, once all of these blocks have been processed, they are transmitted simultaneously, because they are part of the same IP packet and thus are destined for the same end user. If the number of blocks is small, and if the processing delay is small, then it can be safely assumed that an entire IP packet will be transmitted per frame. Thus the arrival process is equivalent to a batch arrival process with all batches equal to 1.

Given the preceding argument, we have modelled the system as a batch arrival system in order to determine experimentally whether or not this assumption is valid. We have modelled the arrivals with two processes, a batch Poisson process and a Batch Markovian Arrival Process (BMAP). BMAP's have been shown to model IP traffic better than Poisson processes, but are much more difficult to analyse. The following sections deal with the $M^{[X]}$/D/1 queue and the BMAP/D/1 queue.

3.3 The $M^{[X]}/D/1$ Queue

The $M^{[X]}/D/1$ queue is a specialisation of the more general $M^{[X]}/G/1$ queue, where customers arrive in batches rather than singly [31]. The rate with which each batch arrives is governed by a Poisson process with parameter λ, and the batch size X is distributed according to a predefined distribution β_j where j is the size of the batch. Due to the specialised nature of such a queue, the basic formulae which apply to the queue are somewhat different: if $E(X)$ is the mean batch size, then:

$$\rho = \frac{\lambda E(X)}{\mu} \tag{5}$$

where μ is the average service rate. The $M^{[X]}/G/1$ queue is generally intractable except for the special cases of exponential and deterministic service times, i.e. the $M^{[X]}/M/1$ and $M^{[X]}/D/1$ queues.

For the $M^{[X]}/D/1$ queue, we consider a batch Poisson arrival process where the service time is a constant S_0, and p_j is the steady state probability that there are j

customers present in the queue. A full treatment on this queueing model is given by Tijms [31].

The relevant quantities are those that are used in Little's Law, which are the waiting time and queue length. As previously mentioned, the formulae are modified in the case of the batch arrival system:

$$\begin{aligned} L &= \lambda\, E(X)\, W \\ L_q &= \lambda\, E(X)\, W_q \end{aligned} \tag{6}$$

The formula for L_q is given in [31]:

$$L_q = \frac{1}{2(1-\rho)}\left[\rho^2 + \rho\left\{\frac{E(X^2)}{E(X)} - 1\right\}\right] \tag{7}$$

Thus, using Equations 6 and 7, the formula for W_q can be derived:

$$W_q = \frac{1}{2(1-\rho)\lambda E(X)}\left[\rho^2 + \rho\left\{\frac{E(X^2)}{E(X)} - 1\right\}\right] \tag{8}$$

Since $L - L_q$ is defined as being equal to λ/μ, and with $S_0 = 1/\mu$, we have that

$$L = L_q + \frac{\lambda}{\mu} = L_q + \lambda \cdot \frac{1}{\mu} = L_q + \lambda \cdot S_0 \tag{9}$$

Similarly, since $W = W_q + 1/\mu$, we have:

$$W = W_q + S_0 \tag{10}$$

We can use the above equations to calculate basic properties of the queue, which influence a wide variety of design decisions such as required buffer sizes and quality of service.

3.4 The BMAP/D/1 queue

3.4.1 The Batch Markovian Arrival Process (BMAP)

The BMAP was first introduced by Neuts [40] in 1981 as a versatile Markovian point process. The notation was later modified and the process was introduced as the Batch Markovian Arrival Process (BMAP). The BMAP was envisioned as a process that would maintain the tractability of the simpler Poisson processes while generalising inter-arrival times [21].

The tutorial by Lucantoni [21] motivates the BMAP by generalising the Poisson process. The generalisation relaxes the requirement that interarrival times are exponentially distributed, which is the case in the Poisson process. To do this, one must focus on the definition of a Poisson process. A Poisson process is a continuous-time Markov chain with one state, which is visited successively. The sojourn time of the Markov process is exponential with rate λ, and after each sojourn time expires, the state is revisited, which coincides with an arrival. This model is shown in Figure 21.

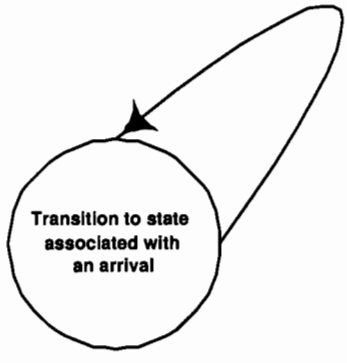

Figure 21: The Poisson Process.

In order to generalise the Poisson process, we add auxiliary states to the point process and only associate arrivals with a subset of these states. Consider a Markov chain with N states. The sojourn time in state i is exponentially distributed with parameter λ_i. When the sojourn time has expired, there is a transition to any of the

N states which may or may not correspond to an arrival. If a transition occurs from state i to state j, then there will be an arrival of batch size k with probability $P(k)_{i,j}$. Conversely, there will be a transition to state j with *no arrivals* with probability $P(0)_{i,j}$. Therefore, according to [21], we have the following for $1 \leq i \leq N$:

$$\sum_{j=1,j\neq i}^{N} P(0)_{i,j} + \sum_{k=1}^{\infty} \sum_{j=1}^{N} P(k)_{i,j} = 1 \tag{11}$$

Another way to understand the process is to define an absorbing state 0, in which batch arrivals occur. With probability $P(k)_{i,j}$, the process will enter the absorbing state from transient state i, where a batch arrival of size k occurs, and the process is immediately restarted in state j. However, the process could also experience a transition to state j without entering the absorbing state with probability $P(0)_{i,j}$. This is shown in Figure 22.

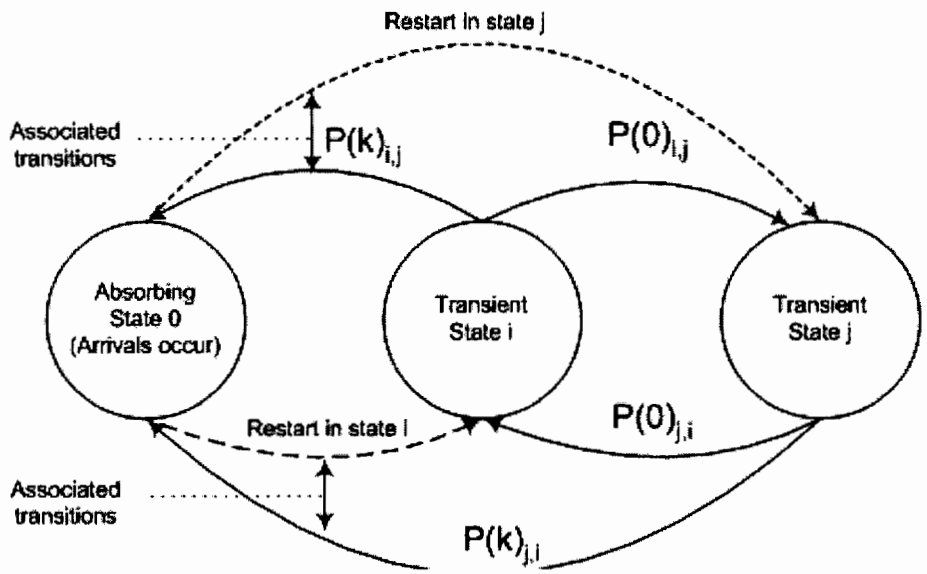

Figure 22: A BMAP. Once a sojourn has expired in state i, a transition is made either to state j, where no arrivals occur, or to state 0, where a batch arrival occurs, and the process is restarted in state j.

To summarise, the difference between a Poisson and a batch Markovian arrival

process is that, while both have exponentially distributed transitions between states, Poisson processes are a single state system with arrivals coinciding with every transition, whereas BMAP's are N state systems, with arrivals coinciding with only a subset of all possible transitions.

The transition rates can be represented by the matrix $\mathbf{D}$. We define $\mathbf{D(0)_{i,j}} = \lambda \cdot \mathbf{P(0)_{i,j}}$ for $i \neq j$, $\mathbf{D(0)_{i,i}} = -\lambda$ and $\mathbf{D(k)_{i,j}} = \lambda \cdot \mathbf{P(k)_{i,j}}$ [13]. $\mathbf{D(0)}$ is the rate matrix for all transitions without arrivals, and $\mathbf{D(k)}$ is the rate matrix for all arrivals of batch size k. This implies that

$$\mathbf{D} = \mathbf{D(0)} + \sum_{\mathbf{k=1}}^{\mathbf{N}} \mathbf{D(k)} = \sum_{\mathbf{k=0}}^{\mathbf{N}} \mathbf{D(k)} \tag{12}$$

The BMAP traffic arrival process is difficult to use analytically, and its important metrics are often calculated numerically. However, the BMAP provides a far more accurate view of IP traffic, because it captures two important statistical properties of IP traffic, namely *self-similarity* and *burstiness* [13]. Burstiness refers to the tendency of the arrival process to occur in 'bursts', where the packets arrive with a rate far above the mean, and self-similarity refers to the similarity of flows over a wide range of time scales. Poisson models are widely regarded as inadequate for modelling such types of traffic. The BMAP, however, is versatile enough to model this behaviour, while still being analytically tractable.

3.4.2 Solving The $BMAP/D/1$ Queue

This section describes the BMAP-based queue that has a generic service time distribution. We will specialise these results in order to find expressions for the BMAP/D/1 queue.

Preliminary Definitions

The BMAP/G/1 queue is a single server queue governed by a BMAP arrival process. The BMAP is specified by the matrix $\mathbf{D}$, as defined in Equation 12. The service time has an arbitrary distribution function $H(x)$ with mean α and Laplace-Stieltjes

transform $h(s)$. The Laplace-Stieltjes transform, $h(s)$, of a function $H(t)$, is defined as [31]:

$$h(s) = \int_0^\infty e^{-st} dH(t) \tag{13}$$

Our service time is deterministic with mean S_0, which simplifies the calculation. From [16], we have:

$$h(s) = e^{-sS_0} \tag{14}$$

Several quantities need to be defined before the waiting time distribution can be given: if we define $\boldsymbol{\pi}$ as the stationary probability vector of the Markov process ($\boldsymbol{\pi}\mathbf{D} = \mathbf{0}$)and $\boldsymbol{\eta} = \sum_{k=0}^{N} k\, \mathbf{D_k}\, \mathbf{e}$ ($\mathbf{e}$ is a column vector of 1's), then λ is defined as $\lambda = \boldsymbol{\pi}\boldsymbol{\eta}$. The traffic intensity is defined as $\rho = \lambda\, \alpha$, where α is the mean of $H(x)$ [21].

The final quantity that is required for the waiting time distribution of the BMAP/G/1 queue is a result that stems from the definition of the *busy period.* The busy period is defined by $G(z, s)$, which is the two-dimensional transform of the number served during and the duration of the busy period. It is given by Lucantoni ([21]) as

$$G(z, s) = z \int_0^\infty e^{-sx} e^{D[G(z,s)]x} dH(x) \tag{15}$$

We then define the matrices $G(s) = G(1, s)$ and $G = G(0)$, where the latter satisfies

$$G(z, s) = \int_0^\infty e^{D[G]x} dH(x) \tag{16}$$

where $D[G(z, s)] = \sum_{k=0}^{\infty} D_k G(z, s)^k$.

Finally we define $\mathbf{g}$, which is the invariant probability vector of G, which satisfies

$$\mathbf{gG} = \mathbf{g}, \quad \mathbf{ge} = \mathbf{1} \tag{17}$$

We can now define the waiting time distribution for the BMAP/G/1 queue.

The Waiting Time Distribution

The distribution for the waiting time (or workload) is given by Lucantoni in [21]. We define $\bar{\mathbf{W}}(\mathbf{x}) = (\bar{\mathbf{W}}_\mathbf{1}(\mathbf{x}), ..., \bar{\mathbf{W}}_\mathbf{m}(\mathbf{x}))$, where $\bar{W}_j(x)$ is the joint probability that the arrival process is in phase j and a customer waits at most for a time x before entering service. If the Laplace-Stieltjes transform of $\bar{\mathbf{W}}(\mathbf{x})$ is $\mathbf{W}(\mathbf{s}) = \int_0^\infty \mathbf{e}^{-\mathbf{sx}}\mathbf{d}\bar{\mathbf{W}}(\mathbf{x})$, then

$$\mathbf{W}(\mathbf{s}) = \mathbf{s}(\mathbf{1}-\rho)\mathbf{g}[\mathbf{sI} + \mathbf{D}(\mathbf{h}(\mathbf{s}))]^{-\mathbf{1}}, \quad \mathbf{W}(\mathbf{0}) = \pi \tag{18}$$

where $\mathbf{I}$ is the identity matrix.

According to [20], the waiting time distributions is given by

$$\bar{w}(x) = \bar{\mathbf{W}}(\mathbf{x})\mathbf{e} \tag{19}$$

It follows that the mean waiting time $\bar{W}$ is

$$\bar{W} = \int_0^\infty x\bar{w}(x)dx \tag{20}$$

Numerical Computation

Several methods exist for solving Equation 18. These methods are fairly technical and involve many separate techniques and theories. For example, the methods described by Lucantoni in [20] draw on several sources, and require methods such as those defined by Abate and Whitt [33] in order to invert the Laplace-Steltjies transform in Equation 18.

However, we have not solved the analytic equations for the waiting time, but have rather simulated the queue from the parameterised BMAP. In order to do this, we have utilised a software package provided by Lindemann et al [37] known as `IP2BMAP`. The package parameterises a BMAP from a given IP packet trace in the format (`arrival-time packet-size`). The algorithms used are described in detail in [14].

The package consists of several tools, which we have used in the following way:

- `abs2rel`: transforms an the absolute timestamp of an arrival to a relative or *interarrival* time.

- `byte2batch`: converts the packet sizes into batch sizes.
- `bmapem`: parameterises the BMAP using the EM algorithm, described in [14]. This tool uses a trace file consisting of relative timestamps and batch sizes, which requires the use of `abs2rel` and `byte2batch`.
- `bmaptrace`: generates a trace file from the BMAP parameterisation.

The IP trace is generated from real traffic, and is provided by our simulation tool (described in Chapter 5). Once the `IP2BMAP` tool has generated the BMAP trace, we run a simple simulator that calculates the delays on packets (or blocks) that have use the BMAP trace for interarrival times, and experience a deterministic service time.

3.5 Summary

In this chapter we have discussed certain aspects of queueing theory and how we have used this theory to model the system described in Chapter 2. We simplified the model of the UMTS packet network to allow us to focus on the performance of queueing at the link layer, where transport blocks arrive in batches. Our queueing models of choice were therefore batch arrival models, one with a Poisson arrival process, the other with a Markovian arrival process. Both queues have a deterministic service time. The Poisson queue ($M^{[X]}/D/1$) provides simplicity, while the Markovian queue (BMAP/D/1) provides a more accurate view of IP traffic flow. Despite the complexity of the BMAP/D/1 queue, there exist algorithms and methods for solving for the waiting time distribution. We have defined the simplest, which uses tools to parameterise the BMAP and simulate the queue. We will compare the two arrival processes in Chapter 6, where we will also analyse the parameters of the model.

Chapter 4

Physical Channel Model

4.1 Physical Channel Review

The physical channels used by UMTS mobile networks are defined by the multiplexing technique that they employ, namely CDMA. CDMA systems share the transmission medium by allowing all signals to be transmitted simultaneously and at any time. This is achieved by multiplying each signal by a unique, orthogonal spreading code (discussed in detail in Chapter 2). Due to the multi-user nature of CDMA systems, the transmitted signals can be affected by *Multiple Access Interference* or MAI. This occurs when multiple, non-orthogonal signals are transmitted simultaneously. MAI is worsened by *multipath fading* which is a signal distortion that is characteristic of any nearly radio transmission channel. Fading causes multiple version of a signal to exist, usually out of phase with the original, which can render the orthogonality useless.

It is important to consider these factors when modelling a network that is based on such technology. The following sections detail the model that we have developed for the purpose of analysing a CDMA radio channel on an indoor, downlink, shared UTRAN channel that experiences Ricean fading.

4.2 Hidden Markov Model

Hidden Markov Models (HMMs) can be defined as stochastic, finite-state automata that consist of a set of finite states $\mathbf{Q} = \{\mathbf{q_1}, \mathbf{q_2}, .., \mathbf{q_n}\}$, each of which is associated with a specific probability distribution [30]. The transitions between states (p_{ij}) are defined as the probability of moving to state q_j, given that the chain was previously in state q_i.

HMMs define two simultaneous stochastic processes: the sequence of HMM states, and the observable output from each state. The latter is visible, whereas the former is hidden (hence the name).

Hidden Markov models have been used extensively to model radio channels because of the need to model the *memory* inherent in radio channels; this memory is introduced by the fact that errors often occur in bursts, and are thus statistically

dependent [18]. Hidden Markov models belong to a class of models known as finite-state channels(FSCs), which are characterised by an underlying Markov chain. Given a certain channel input X_k at time interval k, and given that the Markov chain was in state S_{k-1} during the previous interval, we can determine the output of the channel, y_k, based on the conditional probability $P(y_k, s_k|x_k, s_{k-1})$. The states of the HMM are associated with channel conditions, while the difference between y_k and x_k is defined as the error sequence. Thus, the HMM is capable of modelling the error sequence and thus the channel memory, despite the fact that it has as a major component a Markov chain, which is defined as being memoryless.

We have chosen to adopt the approach of Judge and Takawira [34]. In their paper, they extend a popular HMM known as the Gilbert-Elliot Channel (GEC). The GEC is a two-state HMM, one state defined as a *good* channel and the other as a *bad* channel. The transitions, p_{01} and p_{10}, between the states are defined, as well as the probabilities that an error will occur in either state. Figure 23 shows an example of a GEC.

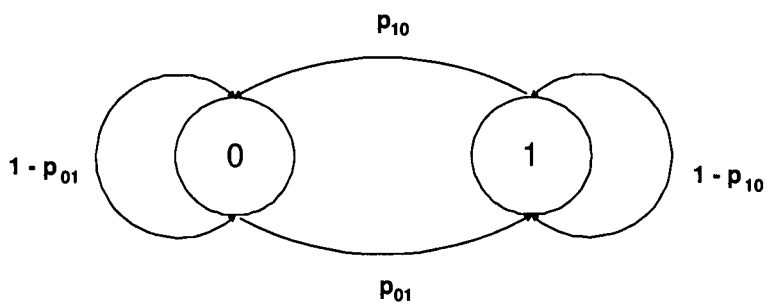

Figure 23: A Gilbert-Elliot Channel.

Judge and Takawira extend this model by creating N *good* states and N *bad* states. Each state is defined as being either *good* or *bad* (where $\Omega \in \{good, bad\}$), conditioned on the number of transmitting users, j, $0 \leq j \leq N$. The probability of being in state Ω with j interfering users is $P_j(\Omega)$. The model therefore incorporates MAI. Figure 24 shows the extended model.

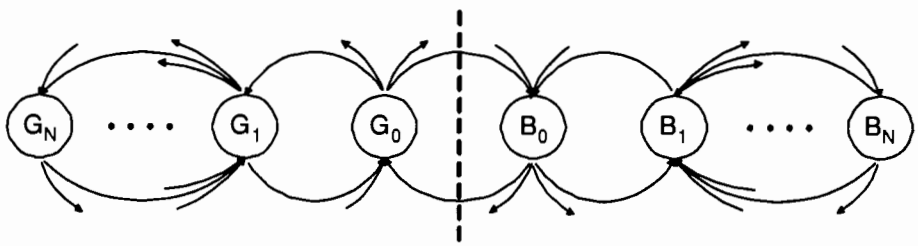

Figure 24: The extended GEC. Each state is either *good* (G) or *bad* (B).

The above model can be described mathematically in the following way:

$$P(G_j) = \sum_{i=0}^{\infty} P(G_i)\omega_{ij}^{GG} + \sum_{i=0}^{\infty} P(B_i)\omega_{ij}^{BG} \tag{21}$$

$$P(B_j) = \sum_{i=0}^{\infty} P(G_i)\omega_{ij}^{GB} + \sum_{i=0}^{\infty} P(B_i)\omega_{ij}^{BB} \tag{22}$$

$$\sum_{i=0}^{\infty}[P(G_i) + P(B_i)] = 1 \tag{23}$$

where ω_{ij}^{CD} is the probability of moving from state C_i to state D_j $(C, D \in \Omega)$.

We define π_{ij} as the steady state probability of moving from a state with i interfering signals to a state with j interfering signals; π_{ij} is the ij^{th} element of the matrix $\boldsymbol{\pi}$. We can also define p_{int}^{j} as the probability that there are j interfering signals being transmitted during the current timeslot.

The probability that the HMM is in a channel state C with j interfering signals is defined as $P_{channel}^{C_j} = P\{\Omega = C, x = j\}$. Therefore, $P_{channel}^{C|j} = P\{\Omega = C|x = j\}$ is the probability that the HMM is in channel state C *conditioned on* there being j interfering signals.

Another transition probability that is useful to us is the joint state probability that the HMM will move from a state C with i interfering signals to a state D with j interfering signals. We define this as

$$P\{\Omega_{n+1} = D \mid \Omega_n = C,\ x_{n+1} = j,\ x_n = i\} = \Pi_{channel}^{D_j|C_i} \tag{24}$$

Using these definitions, and applying Bayes' Theorem, we have:

$$P(C_j) = P^{C_j}_{channel} = P^{C|j}_{channel} \cdot p^j_{int} \tag{25}$$

and

$$\omega^{CD}_{ij} = \Pi^{D_j|C_i}_{channel} \cdot \pi_{ij} \tag{26}$$

The quantity p^j_{int} is defined as

$$p^j_{int} = \sum_{i=0}^{\infty} p^i_{int} \cdot \pi_{ij} \tag{27}$$

The other quantities described can be defined after considering the objectives of the model. Our goal was to model a system where interfering signals are transmitted at random, with randomly varying durations; more specifically, we have modelled cellular telephone calls that are active for a random period of time. Following the example set in [34], we have assumed a Poisson distribution for the arrival process that governs the interfering signals, while the call durations are geometrically distributed. The number of calls that terminate per timeslot is governed by a binomial distribution. We can therefore define π_{ij}, which is our interference transition variable, as

$$\pi_{ij} = \sum_{k=0}^{i} \cdot F_{binom}(i,k) \cdot F_{Poisson}(j-i+k) \tag{28}$$

where

$$F_{binom}(i,\ k) = \left(\begin{array}{c} i \\ k \end{array} \right) . \gamma_{geom} u^k . (1-\gamma_{geom})^{i-k}$$

and

$$F_{Poisson}(m) = \frac{e^G G^m}{m!}$$

The parameters γ_{geom} and G are the mean transmission termination and arrival rates respectively. The duration of each call is define by

$$F_{geom}(l) = (1 - \gamma_{geom})^{l-1}\gamma_{geom} = L(l)$$

In other words, the probability that the HMM moves from a state with i interfering signals to a state with j interfering signals is equal to probability that k signals will leave and $k + (j - i)$ will arrive, summed over k.

4.2.1 HMM Parameterisation

In order to find expressions for $\Pi^{D_j|C_i}_{channel}$ and $P^{C|j}_{channel}$, we note that the greater the level of MAI experienced, the more likely it is that our signal will contain errors. We can define θ as the ratio between an arbitrary signal and the combined MAI signal. We denote the envelope of the transmitted amplitude as u, and the envelope of the MAI amplitude is y. Then, if $\frac{y}{u} \leq \theta$, $\Omega = G$. Conversely, if $\frac{y}{u} \geq \theta$, then $\Omega = B$.

Given that $\Pi^{D_j|C_i}_{channel}$ and $P^{C|j}_{channel}$ are both dependent on the channel condition, we can use θ to derive expressions for them.

We define the probability density functions (PDFs) of the amplitude envelopes of an arbitrary signal as $U_{sig}(u)$. The PDF of the MAI signal is defined as $U_{MAI}(y)$. If $P^{G|j}_{channel}$ is the probability that the HMM is in a *good* state in the presence of j transmitting signals, then

$$P^{G|j}_{channel} = \int_0^{\infty}\int_{\frac{y}{\theta}}^{\infty} U_{MAI}(y|j) \cdot U_{sig}(u)dudy \tag{29}$$

where $\frac{y}{\theta}$ is the limit above which our transmitted amplitude is unaffected by the MAI. $P^{B|j}_{channel}$ is defined as $P^{B|j}_{channel} = 1 - P^{G|j}_{channel}$.

In order to find $\Pi^{D_j|C_i}_{channel}$, we first define $\Pi^{D_j,C_i}_{channel}$, which is given by

$$\Pi^{D_j,C_i}_{channel} = P\{\Omega_{n+1} = D, \Omega_n = C | x_{n+1} = j, x_n = i\}$$

$\Pi^{D_j,C_i}_{channel}$ is the probability that the HMM is in states C_i and D_j in consecutive time slots. Therefore, via Bayes' Theorem:

$$\Pi_{channel}^{D_j|C_i} = \frac{\Pi_{channel}^{D_j,C_i}}{P_{channel}^{C|j}} \tag{30}$$

If we wish to compute $\Pi_{channel}^{D_j,C_i}$, we have to integrate over the instantaneous amplitudes of our transmitted signal and the MAI respectively, both in the current timeslot and in the previous timeslot. For example, the probability that the HMM has moved from a good state to another good state, given i and j interfering signals in each state, is given by

$$\begin{aligned}\Pi_{channel}^{G_j,G_i} &= \int_0^\infty \int_0^\infty U_{sig}(u_n).U_{MAI}(y_n < \theta u_n|i) \\ &\quad . \left[1 - \int_0^{\frac{y_{n+1}}{\theta}} U_{sig}(u_{n+1}|u_n)du_{n+1}\right] U_{MAI}(y_n < \theta u_n|j)du_n dy_{n+1}\end{aligned} \tag{31}$$

where

$$U_{MAI}(y < \phi|j) = \int_0^\phi U_{MAI}(y|j)dy \tag{32}$$

$U_{sig}(u_{n+1}|u_n)$ is defined as the PDF of the envelope of the instantaneous fading amplitudes of the user signal in the current timeslot, conditioned on its amplitude in the previous timeslot. Similar expressions can be found for $\Pi_{channel}^{G_j,B_i}$, $\Pi_{channel}^{B_j,G_i}$, and $\Pi_{channel}^{B_j,B_i}$. We describe the expressions for $U_{sig}(u_n)$, $U_{sig}(U_{n+1}, U_n)$ and $U_{MAI}(y|j)$ in the following sections.

4.2.2 User Signal Model

In this section, we provide expressions for $U_{sig}(u)$, which we will refer to as the *user signal* because it is an arbitrary signal that is transmitted by the user that we are modelling.

If we define $x(t)$ as the signal transmitted by our user, then the received signal, $y(t)$, is given as [48]:

$$y(t) = c(t)x(t) + n(t) \tag{33}$$

where $c(t)$ and $n(t)$ are complex random processes governing the fading and noise distortions respectively. $n(t)$, usually modelled as additive white Gaussian noise, will be ignored for our purposes, because of the fact that this system is interference limited (see Chapter 2, Section 2.5.2, p 40).

According to [48], the PDF of a sequence of fading amplitudes $\mathbf{c_k}$ is:

$$f(\mathbf{c}_k) = (2\pi)^{-k}|\mathbf{D}|exp\left(\frac{1}{2}\mathbf{c}_k\mathbf{D}c_k^T\right) \tag{34}$$

The $(i,j)^{th}$ element of the matrix $\mathbf{D}$ is given by

$$D = \frac{1}{[R(t_j - t_i)cos(\theta_i - \theta_j)]_{k,k}} \tag{35}$$

and $R(\tau)$ is the autocorrelation function of $c(t)$, which, according to [18], is:

$$R(\tau) = \sigma_u^2 J_0(2\pi f_D|\tau|) \tag{36}$$

In the above equation, J_0 is the Bessel function of the first kind and of order zero; σ_u^2 is the variance of $u(t)$, which is the envelope of $c(t)$. If the product $f_D|\tau|$ is small (< 0.1), the fading process is "slow". If it is large (> 0.2), the process is "fast". If the duration of the fade is denoted as t_s, we can define the correlation coefficient of two consecutive channel samples:

$$\rho = \frac{R(t_s)}{R(0)} = J_0(2\pi f_D t_s) \tag{37}$$

The fading signal $c(t)$ can be represented by its in-phase and quadrature components, $c_I(t)$ and $c_Q(t)$. In the case of Rayleigh fading,

$$c(t) = c_I(t)cos(2\pi f_D t) - c_Q(t)sin(2\pi f_D t) \tag{38}$$

However, in a Ricean fading environment [32],

$$c(t) = c_I'(t)\, cos(2\pi f_D t) - c_Q(t)\, sin(2\pi f_D t) \tag{39}$$

where

$$c_I'(t) = A + c_I(t) \tag{40}$$

The quantity A represents the amplitude of the dominant signal which occurs as a result of the line of sight between the transmitter and the receiver.

In order to simplify, we note that $c(t)$ can be defined in terms of its envelope $u(t)$ and its phase $\theta(t)$. Transforming to these co-ordinates $((u, \theta))$, the following derivation of the PDF of the sequence u_k can be produced (as used by [48]):

$$f(\mathbf{u}_k) = (2\pi)^{-k}|\mathbf{D}|u_1u_2 \ldots u_k \int_0^{2\pi}\int_0^{2\pi} \ldots \int_0^{2\pi} exp\left(\frac{1}{2}\mathbf{u_k}\mathbf{G}\mathbf{u_k^T}\right) d\theta_1 d\theta_2 \ldots d\theta_k \tag{41}$$

where $\mathbf{u}_k = (w(u_1), w(u_2), \ldots, w(u_k))$. $\mathbf{G}$ is defined by $g_{ij} = d_{ij}$, and d_{ij} is the ijth element of $\mathbf{D}$. The function $w(u_i)$ is defined as

$$w(u_i) = \sqrt{A^2 + u_i^2 + 2u_iA\,cos(\theta_i)} \tag{42}$$

Equation 41 provides a means of calculating the behaviour of Ricean fading variables over k consecutive intervals. We are only interested in the cases $k = 1$ and $k = 2$, as the HMM does not require higher dimensions (see Equation 31).

For $k = 1$, we have

$$f(u_1) = \frac{u_1}{\mu} e^{-\frac{u_1^2+A^2}{2\mu}} I_0\left(\frac{Au_1}{\mu}\right) \tag{43}$$

where $\mu = R(0)$ is the local mean scattered power. Equation 43 is the *Ricean PDF* [45]. $I_0(x)$ is the modified Bessel function of the first kind (order zero).

The distribution can be expressed in terms of one unknown parameter by defining $K = \frac{A^2}{2\mu}$ [45], [51]. The Ricean PDF is therefore:

$$U_{sig}(u) = \frac{(1+K)}{\bar{p}} e^{-K} u e^{-\frac{(1+K)}{\bar{p}}\frac{u^2}{2}} I_0\left(\sqrt{2K\frac{(1+K)}{\bar{p}}}u\right), u >= 0 \tag{44}$$

K is known as the *Rice Factor*, and is defined as the ratio between the Line of Sight (LOS) component of the signal and the scattered component [51]. If $K = 0$,

Equation 44 becomes a Rayleigh distribution. K is adequate in order to completely specify the Ricean distribution [45]. The local mean power is given by $\bar{p} = \frac{1}{2}A^2 + \mu$.

Equation 44 provides the expression for $U_{sig}(u)$ which can be used in Equations 29 and 30. In order to find an expression for $U_{sig}(u_{n+1}|u_n)$, we must consider the case where $k = 2$ is Equation 41. The solution to the resulting expression can be found in [42]:

$$f(u_1, u_2) = \frac{u_1\, u_2}{\mu^2(1-\rho^2)} exp\left[-\frac{2K(1-\rho)}{1-\rho^2} - \frac{u_1^2 + u_2^2}{2\mu(1-\rho^2)}\right] \cdot \sum_{m=0}^{\infty} \epsilon_m\, I_m\left(\frac{\rho u_1 u_2}{\mu(1-\rho^2)}\right) I_m(\eta\, u_1) Im(\eta\, u_2) \tag{45}$$

where:

$$\begin{aligned} \eta &= \frac{\sqrt{2K(1+\rho^2)}}{\sqrt{\mu}(1-\rho^2)} \\ \epsilon_0 &= 1 \\ \epsilon_m &= 2,\ m > 0. \end{aligned}$$

and I_m is the modified Bessel function of the first kind and order m [32]:

$$I_m(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{(x\, cos\, \theta)}\, cos(m\theta)\, d\theta \tag{46}$$

The modified Bessel function can be approximated by:

$$I_m(x) = \sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}x\right)^{(n+2m)}}{m!\,(n+m)!} \tag{47}$$

In order to get $U_{sig}(u_{n+1}|u_n) = f(u_1|u_2)$, we use Bayes' Theorem:

$$U_{sig}(u_{n+1}|u_n) = f(u_1|u_2) = \frac{f(u_1, u_2)}{f(u_1)} \tag{48}$$

Therefore:

$$U_{sig}(u_{n+1}|u_n) = \frac{u_{n+1} exp\left[\left(-\frac{2K(1-\rho)}{1-\rho^2} - \frac{u_{n+1}^2+u_n^2}{2\mu(1-\rho^2)}\right) + \left(K + \frac{(1+K)}{2\bar{p}}u_n^2\right)\right]}{\mu^2(1-\rho^2)\frac{(1+K)}{\bar{p}} I_0\left(\sqrt{2K\frac{(1+K)}{\bar{p}}}u_n\right)} \cdot \sum_{m=0}^{\infty} \epsilon_m I_m\left(\frac{\rho u_{n+1}u_n}{\mu(1-\rho^2)}\right) I_m(\eta\, u_{n+1}) Im(\eta\, u_n) \quad (49)$$

4.2.3 MAI Signal Model

In order to obtain an expression for the MAI, one would have to take into account a summation of multiple Ricean random variables, and such an expression (in closed form) is not easy to obtain. However, it is possible to approximate the distribution of the MAI amplitudes with a Gaussian distribution, as is demonstrated in [19], [25] and [26]. The approximation can be found in [45]:

$$U_{MAI}(y|1) = U_{sig}(y) \quad (50)$$

$$U_{MAI}(y|j) = \frac{1}{\sqrt{2\pi\sigma_u^2}} exp\left(\frac{-(y-j\mu_u)^2}{2j\sigma_u^2}\right), \quad j > 1 \quad (51)$$

Simply put, this is the PDF of the interference amplitude conditioned on the number of users present, j. When $j = 1$, a Ricean distribution can be used (Equation 34).

4.3 Performance Metrics

The physical channel model, represented by the HMM, fits into the overall analytical model by providing a means of measuring the bit error rate (BER) and Block Error Rate (BLER).

We use the definition for $P_{channel}^{G|j}$ (Equation 29) to define the BER. We denote the BER given j simultaneous interfering signals as α_j.

The probability that an error occurs in a *good* channel is defined as p_{err}^{good}, while p_{err}^{bad} is the probability that an error occurs in a *bad* channel. Thus,

$$\alpha_j = 1 - \left(P_{channel}^{G|j}(1 - p_{err}^{good}) + P_{channel}^{B|j}(1 - p_{err}^{bad})\right) \quad (52)$$

In [34], p_{err}^{good} and p_{err}^{bad} are constants. However, this simplistic model does not take Turbo coding into account, and thus provides a poor fit to out simulated data. We have instead conditioned p_{err}^{good} on j, the number of interfering users. To do this, we have used formulae presented in [36] for the upper and lower bound of Turbo code performance. The error rate of turbo-coded signals lies between two asymptotes, one of which is relevant at a high Signal to Noise ratio (SNR), and one of which is relevant at a low SNR. The formulae for each asymptote, as well as a transition probability between the two asymptotes, are given in [36]. We denote the lower bound estimate as P_e^l, and the upper bound as P_e^u. In order to use these expressions we must define the following:

- d_{free}, the free distance of the Turbo code words (i.e. the transmitted binary sequences after Turbo coding).
- N_{free}, the average multiplicity of the code words (i.e. the number of code words with weight [1] d_{free}).
- w_{free}, the average weight of the code words.
- R_c, the code rate (i.e. $\frac{1}{3}$ for UMTS).
- k, the length of the Turbo code interleaver.
- $Q(z)$,which is defined as $Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy = \frac{1}{2} erfc\left(\frac{z}{\sqrt{2}}\right)$
- N, the spreading factor used for transmission.

Using these parameters, the following expressions can be used [36]:

[1] The *weight* of a codeword can be defined as the number of 1's in the binary representation of the word.

$$P_e^l(j) = \frac{w_{free}N_{free}}{k}Q\left(\sqrt{2d_{free}R_c\frac{3N}{j}}\right) \tag{53}$$

$$P_e^u(j) = Q\left(\sqrt{2R_c\frac{3N}{j}}\right) \tag{54}$$

If we apply Equations 53 and 54 to a UTRAN DSCH, then $R_c = \frac{1}{3}$, $N = 8$ and $k = 320$. The values for d_{free}, N_{free} and w_{free} are found using software provided by R. Garello [24], which is presented in detail in [44]. The values for the UMTS Turbo coder are $d_{free} = 24$, $N_{free} = 1$ and $w_{free} = 24$.

P_e^l and P_e^u can also be described as the BER during low interference and during high interference, respectively. If γ is the transition probability of the BER curve from P_e^l to P_e^h, then we can define $p_{err}^{bad}(j)$ [2] as

$$p_{err}^{bad}(j) = \gamma P_e^l(j) + (1-\gamma)P_e^h(j) \tag{55}$$

We will parameterise γ experimentally in Chapter 6. The HMM estimate of the BER is therefore defined as

$$\alpha_j = 1 - \left(P_{channel}^{G|j}(1 - p_{err}^{good}(j)) + P_{channel}^{B|j}(1 - p_{err}^{bad}(j))\right) \tag{56}$$

Using α_j, we can also define the block error rate β_j (BLER), which is the error rate for each transport block. Given that there are B_{block} bits in a block, this probability can be obtained by

$$\beta_j = 1 - (1 - \alpha_j)^{B_{block}} \tag{57}$$

This quantity can be used to determine the retransmission requirements of the system.

4.3.1 Integration with Queueing Metrics

The metrics provided by the HMM must be integrated with the metrics provided by the queueing models provided in Chapter 3. These must combine to provide a

[2] We will assume that no errors occur in a *good* state.

means of analysing the performance of the overall system. The most obvious metric to concentrate on is that of delay. In the link layer model, delay is found via the mean waiting time of an IP packet, W. The purpose of the physical layer model is to determine the number of times that a particular block needs to be retransmitted due to errors incurred on the radio channel. Using these metrics, we can determine the average total delay experienced by each IP packet.

First, we define R_j^n as the probability that a sequence of n transmissions are successful, given j interferers (we assume that M is the maximum number of transport blocks per IP packets).

$$R_j^n = \sum_{i=0}^{\infty} R_i^{n-1} \pi_{ij} \Pi_{channel}^{G_j,G_i} \tag{58}$$

$$R_j^0 = p_{int}^0 \beta_j \tag{59}$$

where p_{int}^j, π_{ij}, β_j and $\Pi_{channel}^{G_j,G_i}$ are defined in Equations 27, 28, 57, and 31 respectively.

We therefore define R^n as the probability that all n consecutive transmissions succeed, averaged over all interfering users:

$$R^n = \sum_{i=0}^{\infty} R_i^n \tag{60}$$

In order to determine the probability of success for the m^{th} individual block in the n block sequence (R_j[3], we simply divide the probability that m blocks succeed by the probability that $m-1$ succeed:

$$R_m = \frac{R^m}{R^{m-1}} \tag{61}$$

Therefore, the average probability of success for an individual block, R is [4]

[3]Note the subscript R_j as opposed to the superscript R^n.

[4]Although it may seem that β_j and R are identical, they are in fact distinct: β_j defines the block error rate conditioned on the number of interfering users; R defines the long term average probability of success for a block, given that the number of interfering users is governed by a certain probability distribution. In this case, the distribution is Poisson (Eq. 28).

$$R = \frac{1}{M} \sum_{m=1}^{M} R_m \tag{62}$$

Given Equation 58, we can calculate the number of erroneous transmissions per n transmitted blocks, N_{err}:

$$N_{err}(n) = \lfloor (1 - R) \cdot n \rfloor$$

The total number of transmissions required such that all n blocks are successfully received can be calculated via the following recursion:

$$N_{TX}(n) = n + N_{TX}(N_{err}(n)) \tag{63}$$

$$N_{TX}(1) = 1 \tag{64}$$

The definition of $N_{TX}(n)$ is fairly intuitive: the number of transmissions required to successfully transmit n blocks must be equal to n plus the number of retransmissions. The retransmissions themselves are subject to error, which is the reason for the recursion in Equation 63.

Since we have discovered the number of transmissions required, we multiply $N_{TX}(n)$ by the delay experienced by a block, namely W, which is defined in Chapter 3, Equation 10 ($M^{[X]}$/D/1) and in Equation 20 (BMAP/D/1). Note that W must be rounded to the nearest 0.01 seconds in order to capture the effect of the 10ms frame length. The rounded W is given as W_{round}. Therefore, the total delay experienced by a batch of n blocks, $T_{\Delta}(n)$, is

$$T_{\Delta}(n) = W_{round} \cdot N_{TX}(n) \tag{65}$$

An appropriate value for n is the average number of blocks per IP packet, which is defined as γ_{geom} for both the $BMAP/D/1$ queue as well as the $M^{[X]}/D/1$ queue.

4.4 Summary

In this Chapter, we have described a model for the physical channel based on Hidden Markov Models. The model is conditioned on the channel conditions (to account for fading) as well as the number of interfering users present (to account for MAI). We have provided expressions for the amplitude envelope of the Ricean fading signal and the MAI, as well as an expression for two consecutive fading amplitudes. Finally, we have provided an algorithm for calculating the performance of the channel based on the model output.

Chapter 5

Simulator

5.1 Introduction

In this chapter we describe the tool that was created to simulate the system described in Chapter 2. The analytic model of this system is described in Chapters 3 and 4. In order to validate the analytic model, we have created an application to simulate the events that it models. The simulator models all of the network entities described in Chapter 2, using realistic IP traces as traffic sources and applying Gaussian approximations for determining errors as they occur on the physical wireless channel. Everything that occurs between the traffic generation and the transmission of radio frames is simulated according to 3GPP specifications.

5.2 Conceptual Design

5.2.1 Overview

The simulator is conceptually divided into three parts: the traffic generation facility, the network itself, and the transmission module. The traffic generation produces packet interarrival times and sizes, which must then be routed through the network described in Chapter 2, i.e. the packet-switched network architecture for UMTS. This network consists of two CN elements, the GGSN and the SGSN, as well as two UTRAN elements, the RNC and the Node B. We have chosen to model the CN as two objects, namely the GGSN and the SGSN. However, we felt that it is conceptually simpler to model the RNC and Node B using the protocols that are used in each, namely the PDCP, RLC, MAC, and PHY protocols. The transmission module simulates the radio link, taking into account fading and interference. The conceptual design is shown in Figure 25, and the entities are shown in the class diagram in Figure 26.

5.2.2 Network Entities

Each entity is modular, although they are all controlled by a single controlling entity, which maintains the network parameters and oversees the simulation process. The

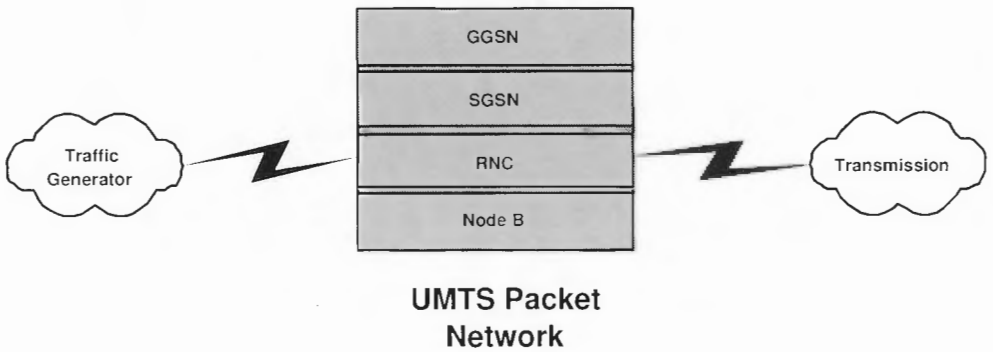

Figure 25: Conceptual design of the simulation model.

flow of data through the network is presented in Chapter 2 in Figures 16 and 17. Below, we briefly discuss each entity and the simplifications made in each. Each entity records all transactions that take place within itself.

Traffic Generator

The traffic generator produces a random flow of IP traffic based on real IP traces (explained in detail in Section 5.3). Traffic is generated for a specific number of traffic sources. The traffic generator therefore simulates an external IP network, which gains access to the UMTS IP network via the GGSN. The rate at which the traffic is generated is governed by the controlling entity.

GSNs

The GGSN is the entry point into the private UMTS IP network. Each IP packet header defines its destination, which the GGSN maps to a specific SGSN. The IP packet is tunnelled using the GTP-U protocol to the SGSN. The SGSN maps the IP packet to the RNC, which controls the flow of data to the UE for which the packet is intended. The IP packet is tunnelled to the RNC in question. The simulator must therefore simulate the lookup process in each entity, as well as the GTP-U tunnelling.

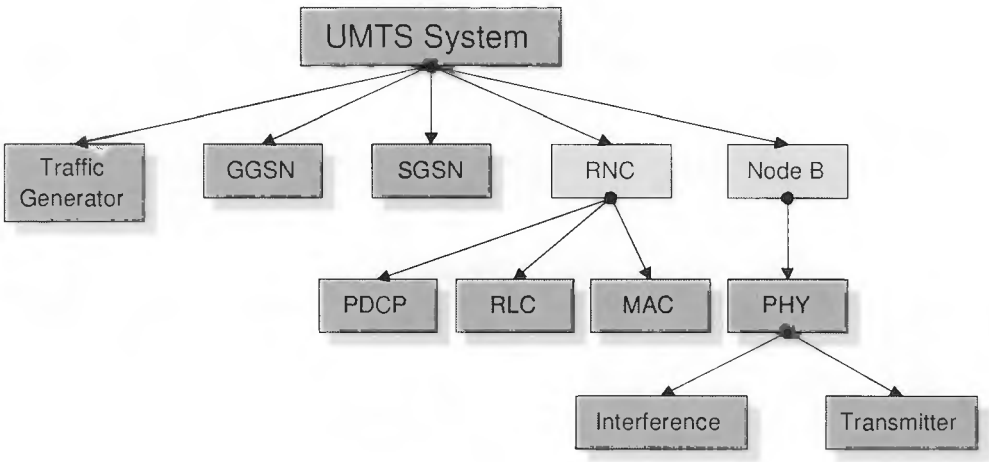

Figure 26: The entities being simulated.

PDCP

The PDCP protocol is the first sub-layer of the link layer. It compresses the header of the IP packet and attaches a PDCP header to the resulting packet. This is then passed down to the RLC protocol.

RLC

The RLC entity is responsible for segmenting the PDCP SDU into RLC PDU's, otherwise known as transport blocks. The size of the PDU is governed by the choice of transport channel. As we have only modelled one channel, the DSCH, there is no need to cater for other options. The choice of the DSCH also means that the RLC protocol must be in Acknowledged Transfer Mode (AM), meaning that the RLC entity must keep track of all RLC PDUs that are transferred to the MAC sub-layer in order to account for each transmission. If an error is recorded in a PDU, it must be retransmitted. The RLC protocol adds a header to each PDU and transfers them to the MAC protocol. The simulator must be able to keep track of all blocks and perform the necessary steps required for retransmissions to occur.

MAC

The MAC sub-layer is responsible for scheduling transport blocks onto the DSCH. However, we have decided to defer this responsibility to the PHY layer, as it is easier to model all systems relating to transmission in one entity. The MAC layer adds a header to each block.

PHY and Transmission

The PHY layer performs the majority of the processing required to send the transport blocks over the physical medium. The first process that must occur is the attachment of a cyclic redundancy code (CRC) to each transport block. Once this has happened, the blocks must undergo Turbo coding. The encoded blocks are then spread according to a specific spreading factor, which is determined by the controlling entity. The blocks will then be ready for transmission.

Transmission occurs at 10ms intervals. The PHY entity keeps track of all blocks that are waiting for transmission. It also keeps track of the users to whom the packets belong, arranging them in a First Come First Serve (FCFS) queue. The packets belonging to the user at the top of the FCFS queue at each transmission interval are transmitted.

Transmission is simulated by applying a random number of bit changes to the transmitted blocks. The block is the decoded using a Turbo Decoder. A comparison is made between the original and the decoded block to determine whether the block was correctly received.

Figure 27 shows all of the queues involved in the acknowledgement and retransmission process described above.

Interference

An important aspect of the radio system is the number of interfering signals that are present during transmission, as this affects the signal quality. The Interference entity generates interfering signals, which are typically voice calls. These interfering signals are generated at a specific time and for a specific duration according to a specific

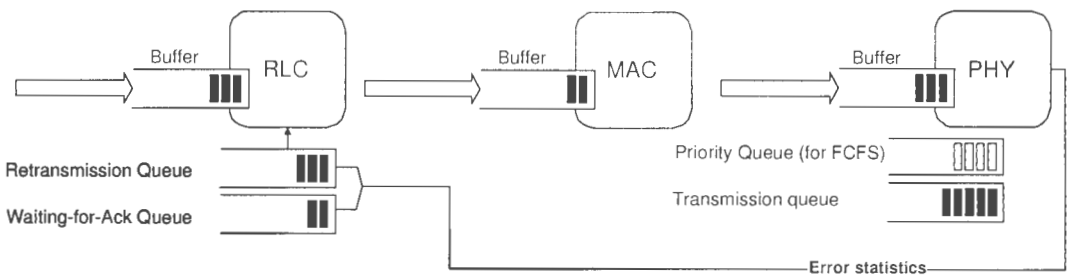

Figure 27: An overview of the queues involved in the acknowledgement process.

distribution.

5.3 Implementation

5.3.1 Overview

The simulator was implemented in Java, which was chosen because it is well suited to the modular design that was laid out in the previous section. Figure 28 shows the object model of the simulator.

The object model shows a controlling entity `UMTSSystem`, which contains various `Entity` objects. These objects are specialised into a `GGSN`, `SGSN`, `PDCP`, `RLC`, `MAC`, and `PHY` entity. The PHY makes use of the `TurboCoder` object.

Other important objects are the `InterferenceSystem`, `DataAnalyser`, and `TrafficSource`. Also shown are the various data structures that are used.

5.3.2 Simulation Algorithms

The simulator entities all perform the tasks laid out in the 3GPP specifications. These tasks have already been described. This section provides details of the algorithms that are not specified and thus need to be approximated or simulated. Specifically, we describe the simulation of the data transfer through the UMTS network, as well as traffic generation, transmission errors, Turbo (de-)coding, and data analysis.

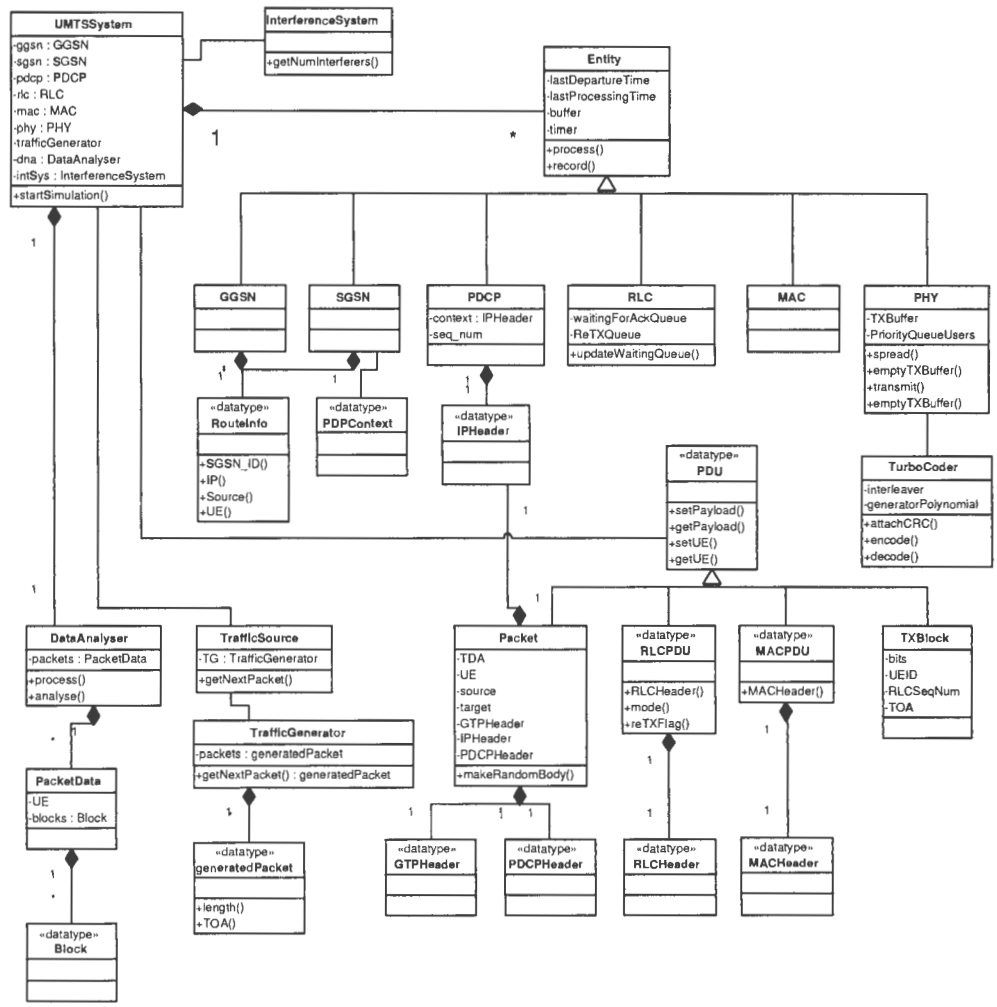

Figure 28: The Object model of the simulator.

System Timer

There are several methods that can be used to simulate time, such as modelling the system in real time, or by incrementing time by fixed steps. We have chosen to use packet interarrivals to drive the simulator. When the first packet arrives at the GGSN, the system clock is set to the arrival time of the packet. The packet is then routed through the network via the network entities, where processing occurs, after which the data is transmitted. The arrival of the packet at each entity is recorded as well as the time that it leaves. The time of transmission is also recorded. The timing of subsequent packets is as follows:

Define A_j^{GGSN} as the time at which a packet j arrives at the GGSN. The time at which it leaves the GGSN is $D_{p_j}^{GGSN}$. In order to determine the time at which packet j enters the SGSN service unit, A_j^{SGSN}, we must take into consideration the packet $j-1$:

$$A_j^{SGSN} = \begin{cases} D_j^{GGSN}, & D_j^{GGSN} > D_{j-1}^{SGSN} \\ D_{j-1}^{SGSN}, & otherwise \end{cases} \tag{66}$$

By applying Equation 66 to all of the entities, and by noticing that the time of arrival (TOA) of a packet into any entity is dependent on the time of departure (TOD) of each preceding packet, we can implement queueing buffers without having to explicitly store any packets.

Another time sensitive aspect of the simulator is the transmission interval. Given that the simulator does not proceed in fixed time increments, it must first check whether or not any transmissions are pending upon arrival of a new packet. If the previous transmission time is TX_{j-1}, and given that our transmission interval is 10ms, we define the number of transmissions that should have taken place between the time of the previous transmission and the arrival of the new packet as

$$TX_{num} = \left\lfloor \frac{(A_j^{GGSN} - TX_{j-1})}{10} \right\rfloor$$

Transmission Scheduling

The simulator performs the transmission simulation TX_{num} times for each new packet arrival. During each of these transmission iterations, it must determine which transport blocks to transmit. The first criterion is that the block must belong to the entity at the front of the FCFS queue. Secondly, no more than 15 blocks can be transmitted at any one time. The third factor to consider is the time at which the block became available for transmission; if we are looking at the i^{th} of TX_{num} transmissions, then the block should only be considered if $TX_j(i) < D^{PHY}_{j-1+i}$. In other words, a transport block can only be considered for transmission if it arrived in the transmission queue before the current transmission time. Otherwise, it must wait until it reaches the top of the FCFS queue again.

Once the blocks have been selected and the transmission simulated, the PHY entity must let the RLC entity know which blocks have been acknowledged. The RLC updates its queue, as does the PHY.

Simulation Process

The simulation algorithm - at a fairly high level of abstraction - is given in Figure 29.

5.3.3 Traffic Generation

In order to model the flow of IP traffic as accurately as possible, we have designed the simulator to produce packet inter-arrival times and sizes according to empirical trace data. The data we use is provided by Walters in [38] on pages 132-134. The data is in the form of HTTP arrival times and sizes. The data is also described in terms of *Web Users*, which define the human internet users, and *Web Clients*, which define the software used by the human users to access the internet. Of the variables that are described, the following have proven relevant to our work:

- Browsing Inter-session Time (T_B): this defines the inter-arrival time of browsing sessions as initiated by a human user.

```
for j = 1: numPackets
    newpacket = trafficsource.getNewPacket
    numTX = (newpacket.TOA - oldTXtime) / 10

    // do the transmissions
    for i = 1 : numTX
        errors [] = phy.transmit
        rlc.updateAcknowledgmentQueue(errors)

    //6 entities: GGSN, SGSN, PDCP, RLC, MAC, PHY
    for k = 1:6
        // time of departure = processing time plus time of arrival
        TOD = entity[k].process(newpacket) + TOA

        if entity[k-1].lastdeparturetime > TOD
          TOD = entity[k-1].lastdeparturetime

        newpacket.setDepartureTime(TOD)
        entity[k].record

// after completion of the simulation:
dataAnalyser.analyse
```

Figure 29: The simulation algorithm.

- Web-User Requests per Session (N_{user}): the number of "clicks" per user per session.
- Web-Client Requests per Web-User Request (N_{client}): The number of requests sent by the browser per click.
- Web-User Request Inter-arrival Time (T_{user}): the inter-arrival time of web-user requests.
- Web-Client Request Inter-arrival Time (T_{client}): the inter-arrival time of web-client requests.
- Web-Client Response Size (X): the size of packets returned to the browser from the IP server.

The above variables must be manipulated to extract the inter-arrival times and the sizes of the HTTP packets on the downlink, which correspond to web-client responses. We will assume a negligible amount of time between request and response. Therefore we can say that the response inter-arrival times are equal to the web-client request inter-arrival times. In order to calculate our HTTP response arrival times,

we randomly select values from the data in [38] for T_B and N_{user}. We then randomly select an inter-arrival time (T_{user}) for each web-user request, and add T_B to it, to produce an arrival time (as opposed to inter-arrival time). For each web-user request, we generate a random number of web-client requests from the data provided in [38] by selecting a value from the tables provided using a uniformly distributed random variable. We randomly select an inter-arrival time T_{client} for each of these and add it to the arrival time of the web-request. We therefore have a list $\mathbf{T}$ of web-client arrival times for this particular session. For each element of $\mathbf{T}$ we generate a size X from the data.

In order to convert the HTTP traffic to IP traffic, we simply segment the HTTP traffic into L IP packets, where $L = \lceil \frac{X}{1500} \rceil$, and 1500 is approximately the largest possible IP packet size. We assume a small, non-negligible delay for each IP packet (which accounts for processing and transmission) add it to the arrival time of the HTTP packet, and assign it to the IP packet. The resulting list is then sorted into ascending order.

Figure 30 shows a sample of randomly generated traffic at different time scales. The dots on the graph represent packet arrivals (or transmission, depending on the observation point). It can be observed that the traffic exhibits bursty behaviour, since the transmissions occur in batches that are well spaced. The traffic exhibits similar patterns at different time scales, indicating self-similarity. Figure 31 shows the mean of the generated traffic at the same time scales.

5.3.4 Transmission Errors

We have used the Standard Gaussian Approximation (SGA) to simulate transmission errors. The SGA has been widely used and its derivation can be found in great detail in [45]. The SGA can be applied as follows:

Assume that a signal is received with power P_0. Given a spreading factor N, we can determine the bit error rate P_e if it experiences interference from K other signals, each received with power P_k $(1 \leq k \leq K)$:

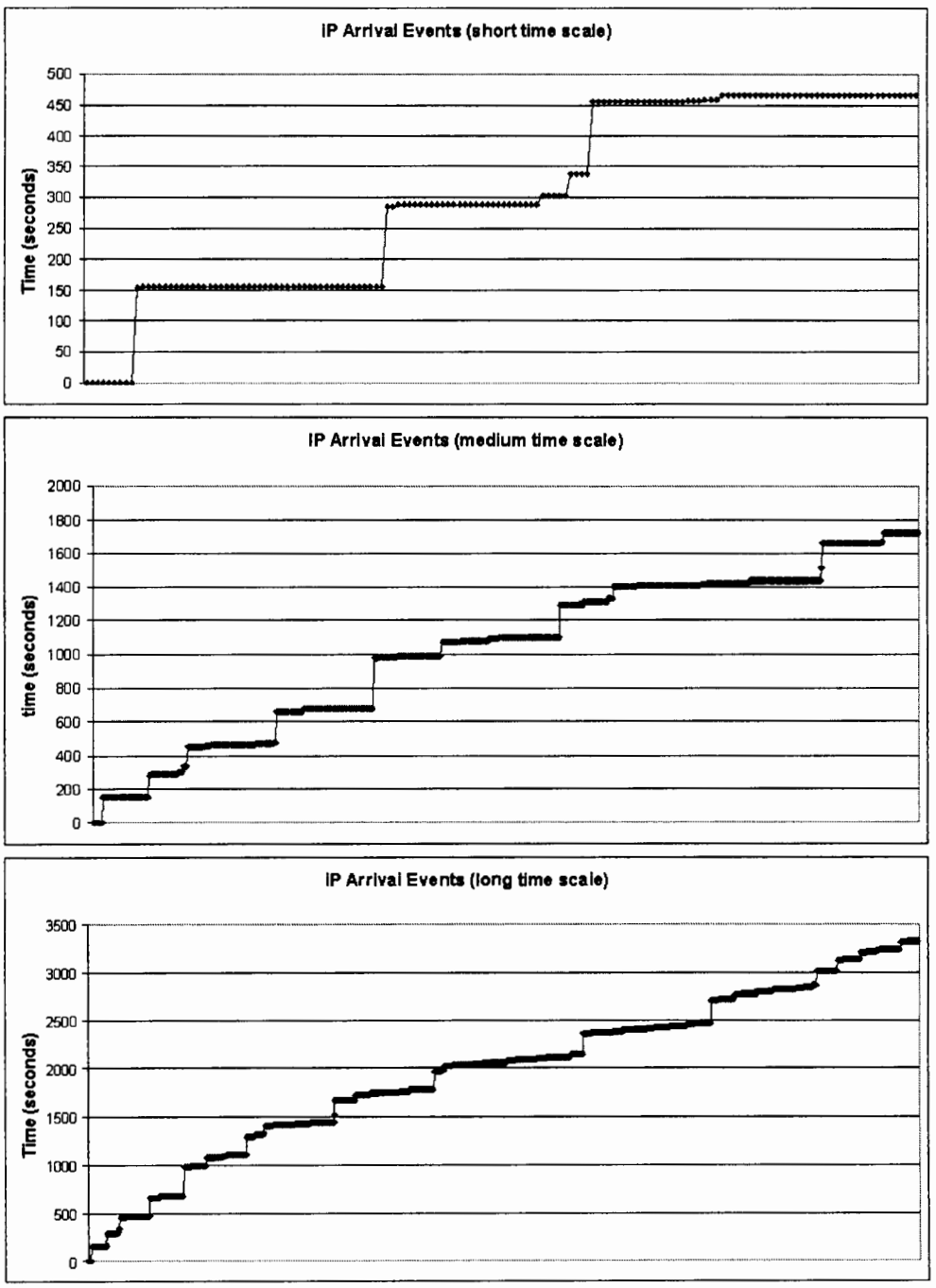

Figure 30: A random sample of IP traffic arrival times over 150, 800, and 2300 samples. Each dot on the graph represents a packet arrival.

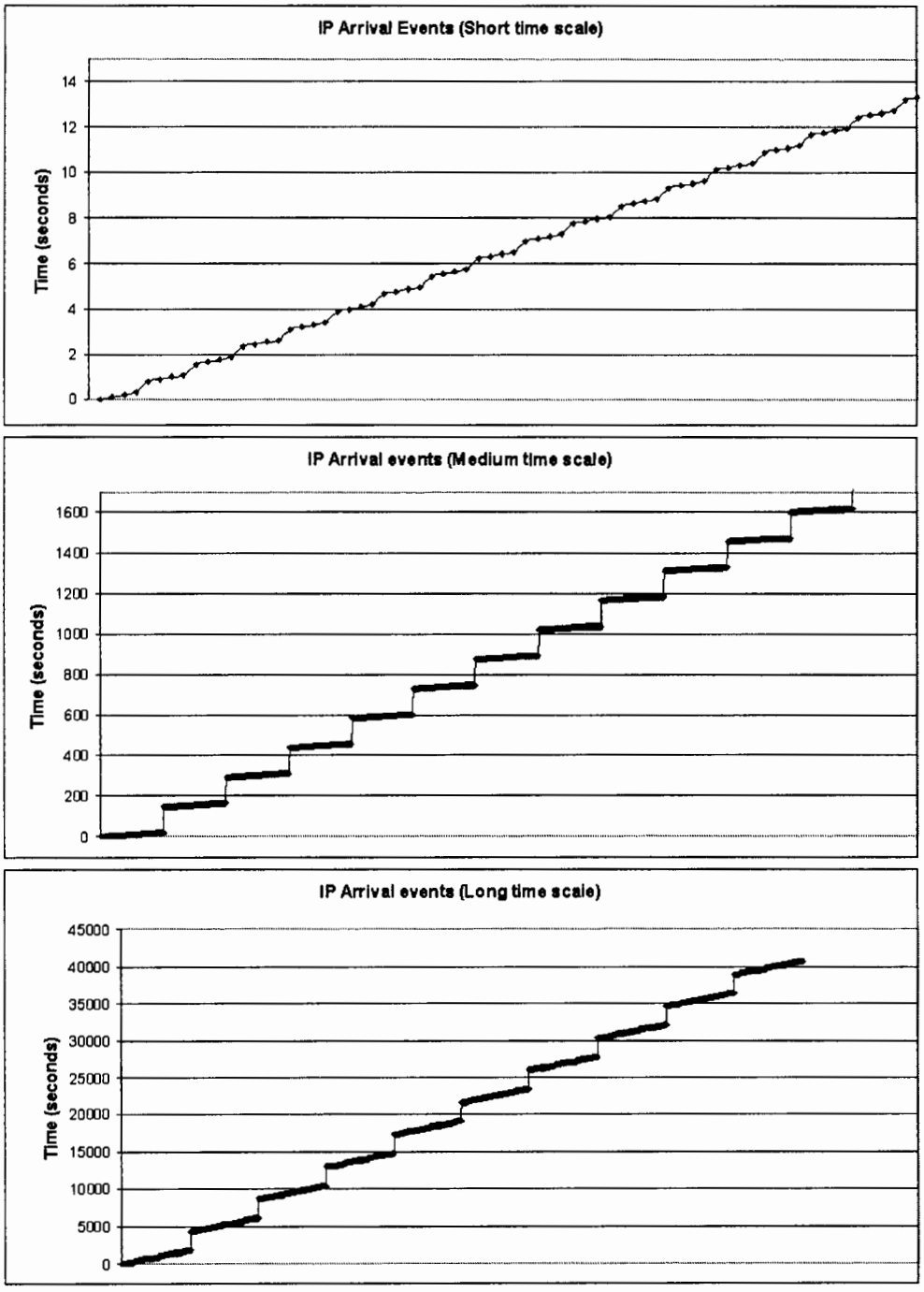

Figure 31: The mean IP traffic arrival times over 150, 800, and 2300 samples.

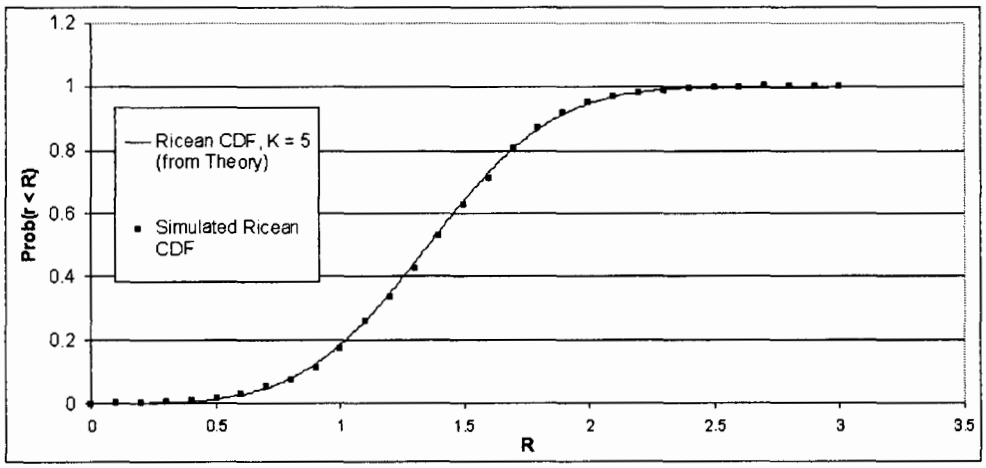

Figure 32: A comparison of the theoretical Ricean CDF and the CDF of the simulated Ricean amplitudes (K = 5).

$$P_e = Q\left(\sqrt{\frac{3P_0N}{\sum_{k=1}^{K-1} P_k}}\right) \tag{67}$$

where

$$Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy = \frac{1}{2} erfc\left(\frac{z}{\sqrt{2}}\right) \tag{68}$$

For our purposes, the spreading factor is equal to the spreading factor of a typical DSCH: $N = 8$ for a 384 kbps channel, and $N = 4$ for a 2048 kbps channel [4].

In order to calculate the received power of signal k, P_k, we can use the relationship $P_k = \frac{1}{2}A_k^2$, where A_k is the received amplitude of the signal. This amplitude can be calculated using the Ricean CDF (Cumulative Distribution Function), which is obtained using a numerical integration of the Ricean PDF (Chapter 4, page 74, Equation 44). Figure 32 shows a comparison of the theoretical values for the Ricean CDF, and the simulated values.

5.3.5 Turbo Coding

The PHY entity is responsible for the Forward-Error Correction (FEC) encoding in the UTRAN. The choice of FEC on the DSCH is Turbo coding. As described in Chapter 2, Turbo codes are very powerful FEC codes. The DSCH employs $\frac{1}{3}$-rate coding, which means that a block of data will be roughly three times its original size after it has been encoded. Turbo codes are a simple extension to Recursive Systematic Convolutional (RSC) codes. The Turbo coding process involves applying an RSC code to two different versions of the input block of data, one that is in correct order, and another that has been pseudo-randomly permuted. This will produce two blocks of parity data, which are transmitted along with the original data.

The randomization of the data is performed by an *interleaver*. The interleaver algorithm, as well as the RSC codes, are given in 3GPP TS 25.212 [4]. One need only attempt to invert the interleaving process to determine whether or not the instructions have been followed correctly, which is true in the case of our simulator.

The *decoding* process requires a far more complex algorithm. An algorithm known as the modified BCJR[1] algorithm is given by Ryan in [46]. The details of the algorithm are beyond the scope of this work, although they have been implemented in the simulator.

The decoding algorithm requires some parameterisation. The two parameters that are required are the number of iterations h for which the algorithm must run, and the value of $L_c = 4\frac{E_b}{N_0}$. The first parameter, h, exists because of the fact that turbo decoding is an iterative process; the decoding of one set of parity bits is dependent on information generated by the decoding of the second set of parity bits, and vice versa. After each iteration of this "information sharing" process, the decoding performance improves until a certain limit. By experimentation, we have determined that $h \geq 12$ provides no significant change in the decoder performance. An example set of results from a simulation run is shown in Figure 33. clearly indicating that the decoder does not perform noticeably better for $h \geq 12$.

The second parameter, L_c, was also experimented with to determine the best

[1]BCJR stands for Bahl, Cocke, Jelinek and Raviv, who derived the algorithm.

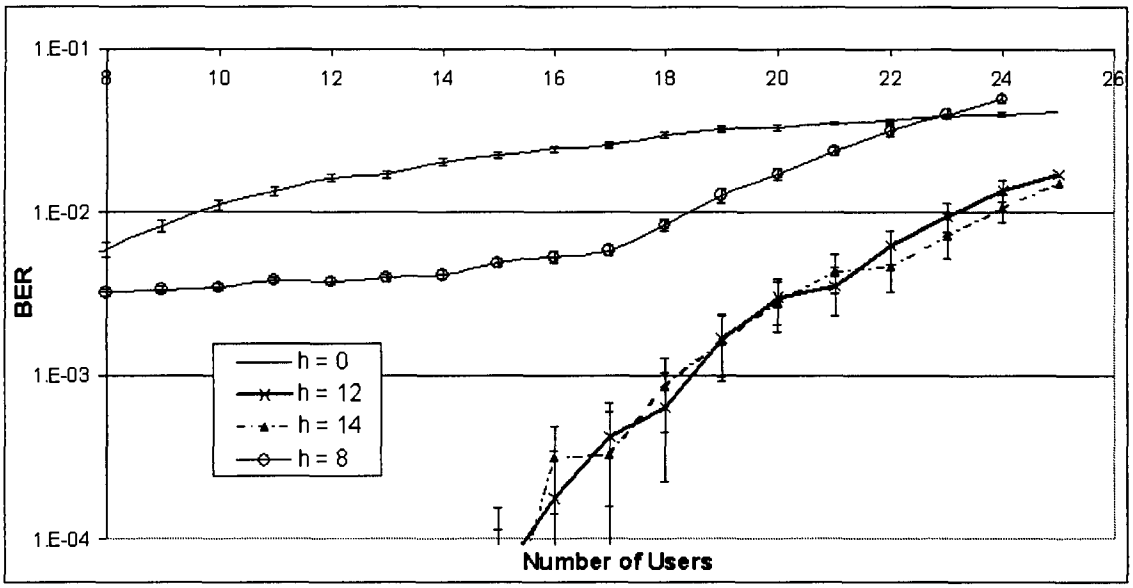

Figure 33: Bit Error Rates for varying h.

performance. Figure 34 shows another example simulation run, which indicates that $L_c = 3$ provides the best error-correcting performance.

5.4 Data Analysis

5.4.1 Data Recording

Each entity records the events that it handles. In other words, the GGSN entity records all packets that arrive from the traffic source; the PHY entity records both the PHY processing events as well as the transmission events. Figure 35 gives a sample of the output for each entity.

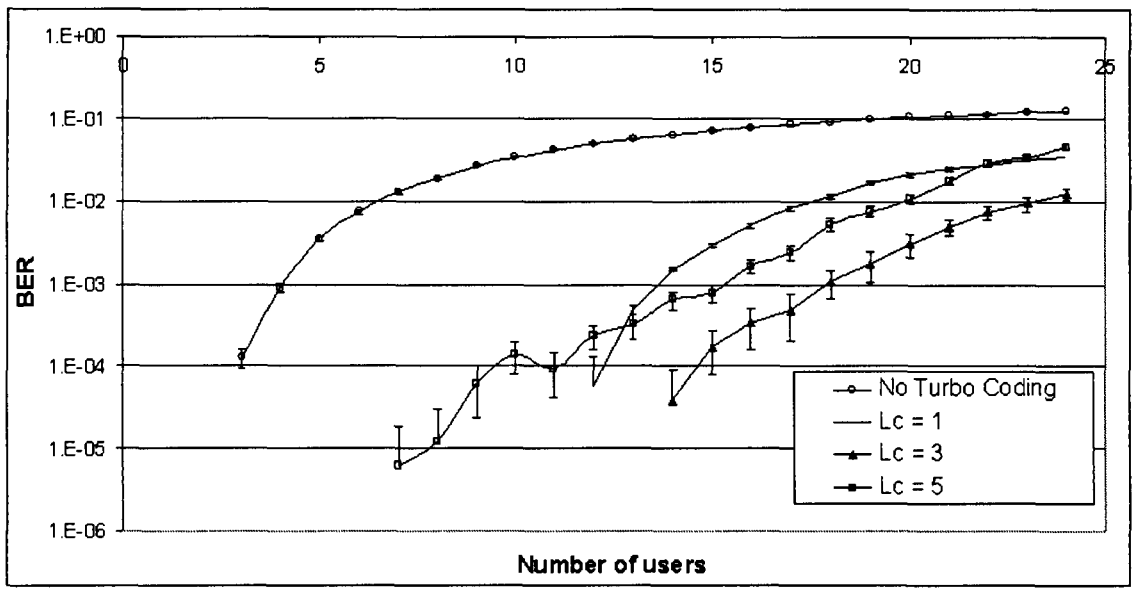

Figure 34: Bit Error Rates for varying L_c.

5.4.2 Analysis

After a completed simulation run, the recorded data is analysed to produce performance metrics. The metrics described below are all presented within 95 % confidence intervals, which are calculated using standard statistical methods.

Delay

The data analysis algorithm consists of simply grouping all details concerning each packet into one logical data structure. The `PacketData` class in Figure 28 provides this functionality. Each `PacketData` object contains several `Block` objects, which hold the relevant information for the packet segments.

We were only interested in extracting the total end-to-end delay of each packet. This can be calculated fairly easily once all relevant blocks pertaining to a packet are extracted from the output file. If packet P_j was split into n blocks $\{B_j, B_{j+1}, \cdots, B_{j+n}\}$, and if each block has an associated time-of-acknowledgement t , the time at which

GGSN:

<PACKETID> 0 <SIZE> 1180 <UEID> 5 <TOA> 0.0 <BUFSIZE> 1 <WAITT> 0.0 <PROCT> 6.7E-7
<DEPT> 6.74E-7

SGSN:

<PACKETID> 0 <SIZE> 1180 <UEID> 5 <TOA> 6.75E-7 <BUFSIZE> 1 <WAITT> 0.0 <PROCT> 1.94E-6
<DEPT> 2.6178E-6

PDCP:

<PACKETID> 0 <SIZE> 1180 <UEID> 5 <TOA> 0.0 <BUFSIZE> 1
<WAITT> 0.0 <COMP> FALSE <PID> 1 <PDCPSEQNUM> 0 <PROCT> 2.6555305758961667E-5 <DEPT> 2.917E-5

RLC:

<PACKETID> 1 <SIZE> 1036 <UEID> 3 <TOA> 109091.692 <BUFSIZE> 1
<WAITT> 0.0 <PID> 0 <PDCPSEQNUM> 1 <BLOCKS> 2
<RLCBLOCK> 1 <MODE> AM <BUFF> FALSE <RLCSEQNUM> 4 <BLOCKLEN> 320
<RLCBLOCK> 2 <MODE> AM <BUFF> FALSE <RLCSEQNUM> 5 <BLOCKLEN> 320
<RETXQL> 0 <WQL> 8 <PROCT> 0.0 <DEPT> 109091.692
<ACK> 0 <TOA> 10.0
<ACK> 1 <TOA> 10.0

MAC:

<TOA> 2.91731E-5 <WAITT> 0.0 <BUFSIZE> 1 <MACBLOCK> 1 <UE> 5 <RLCSEQNUM> 0
<MACBLOCK> 2 <UE> 5 <RLCSEQNUM> 1 <PROCT> 0.0853 <DEPT> 0.0854

PHY:

<TOA> 0.08541 <NUMTB> 4 <WAITT> 0.0 <TXBUFFERLEN> 0 <PHYBLOCK> 1 <RLCSEQNUM> 0
<PHYBLOCK> 2 <RLCSEQNUM> 1
<PROCT> 0.04439 <DEPT> 0.12981

TX:

<TOA> 1.815 <NUMBLOCKS> 2
<BLOCK> 1 <RLCSEQNUM> 0 <BITL> 1170 <NUMERRS> 0 <NUMUSERS> 10 <SUCCESS> TRUE <TCITER> 2
<BLOCK> 2 <RLCSEQNUM> 1 <BITL> 1170 <NUMERRS> 0 <NUMUSERS> 10 <SUCCESS> TRUE <TCITER> 1
<ERRTOTAL> 0

Figure 35: Sample output for each entity.

the block was correctly received, then the total delay experienced by the packet is the difference between its earliest acknowledged block and the latest acknowledged block.

Error Rates

We calculated two error rates: the bit error rate (BER) and the block error rate (BLER). The former is the probability that a single bit is in error. The latter is the probability that an entire block is incorrectly received after Turbo coding. The BER is calculated by dividing the number of erroneous bits (which are recorded as a result

of the equations presented in Section 5.3.4) by the total number of bits transmitted. Similarly, the BLER is the ration between the number of failed blocks, and the total number of transmitted blocks.

5.5 Summary

In this Chapter, we have described the simulator that was used to validate the analytical models of Chapter 3 and 4. We have described the conceptual model, and provided implementation specific details regarding the non-standard aspects of the model, such as traffic generation and transmission error calculations.

Chapter 6

Results

6.1 Introduction

In this chapter we present the results of the study. We first discuss how we calculated numerical values for the Hidden Markov Model by comparing the analytical results with simulated results. We then compare the results of the analytical model with simulated values for both the $M^{[X]}/D/1$ and $BMAP/D/1$ models. Note that all graphs are presented using a logarithmic scale unless otherwise stated, and confidence intervals are computed using 95% confidence.

6.2 HMM Parameterization

In Chapter 4, we provided the details of a Hidden Markov Model (HMM) that predicts the Bit Error Rate (BER) and Block Error Rate (BLER) of the system discussed in Chapter 2. However, no values were chosen for θ (Equation 29, Section 4.2.1) and γ (Equation 55, Section 4.3).

In order to find a value for θ that provides accurate results, we have followed the example set in [34]: we plotted BER and BLER results for a range of values of θ, making sure that the range included sets that both underestimated and overestimated the simulated values for the BER and BLER. We then compared the plotted data visually, and altered θ until we found a value that was the closest to the simulated results. We also verified the accuracy of θ using a root mean squared-error test.

To parameterize γ, we noticed the following: if the number of interfering users is represented by j, then if j is greater than or equal to some integer κ, the FEC performance matches the upper bound asymptote, P_e^u, as defined in Equation 54 (Section 4.3); when $j < \kappa$, the FEC performance tends towards P_e^u starting from P_e^l (Equation 53) as j increases. In the BLER plot in Figure 36, we can see that the simulated data for $\theta = 35$ is very close to the upper bound for $j \geq 20$. When $j < 10$, the simulated data tends to zero, as does the lower bound.

In order to approximate the behaviour of the simulated BLER for $10 < h < 20$, we have attempted to fit the data to as simple an equation as possible, namely a polynomial of the form $1 - \left(\frac{j}{\kappa}\right)^{\phi}$. We were able to approximate κ by noticing that

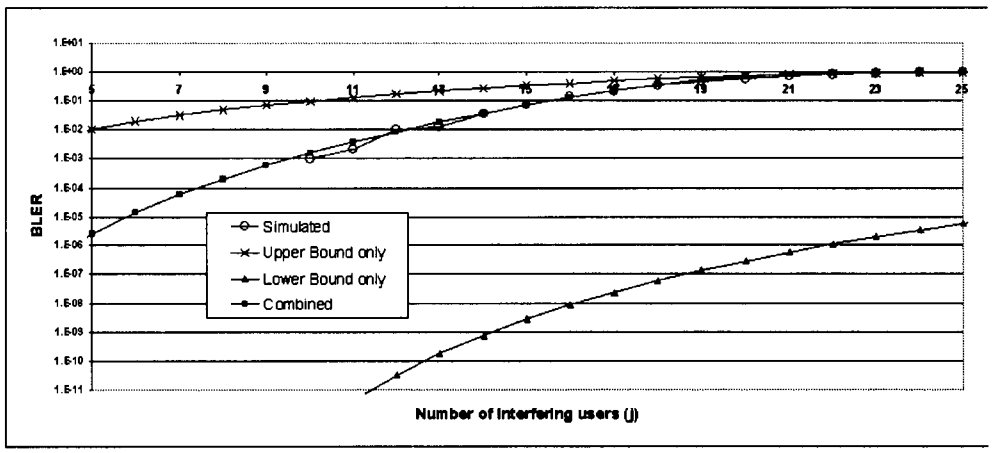

Figure 36: The BER (top) and BLER (bottom) as estimated by the HMM and the simulator.

the simulated data tends towards the upper bound when $j > 20$. The value of ϕ was found by visual analysis, in much the same way as the value for θ was found. Since the value of ϕ determines the rate at which the BER curve changes its shape from that of the lower bound to that of the upper bound, we were able to choose the value for ϕ that approximated this change. The above analysis led to the following expression for γ:

$$\gamma(j) = \begin{cases} 1 - \left(\frac{j}{20}\right)^6 & j < \kappa \\ 0 & j \geq \kappa \end{cases} \tag{69}$$

Figure 37 shows the results for the BER and BLER studies. The tabulated values are presented in Table 5.

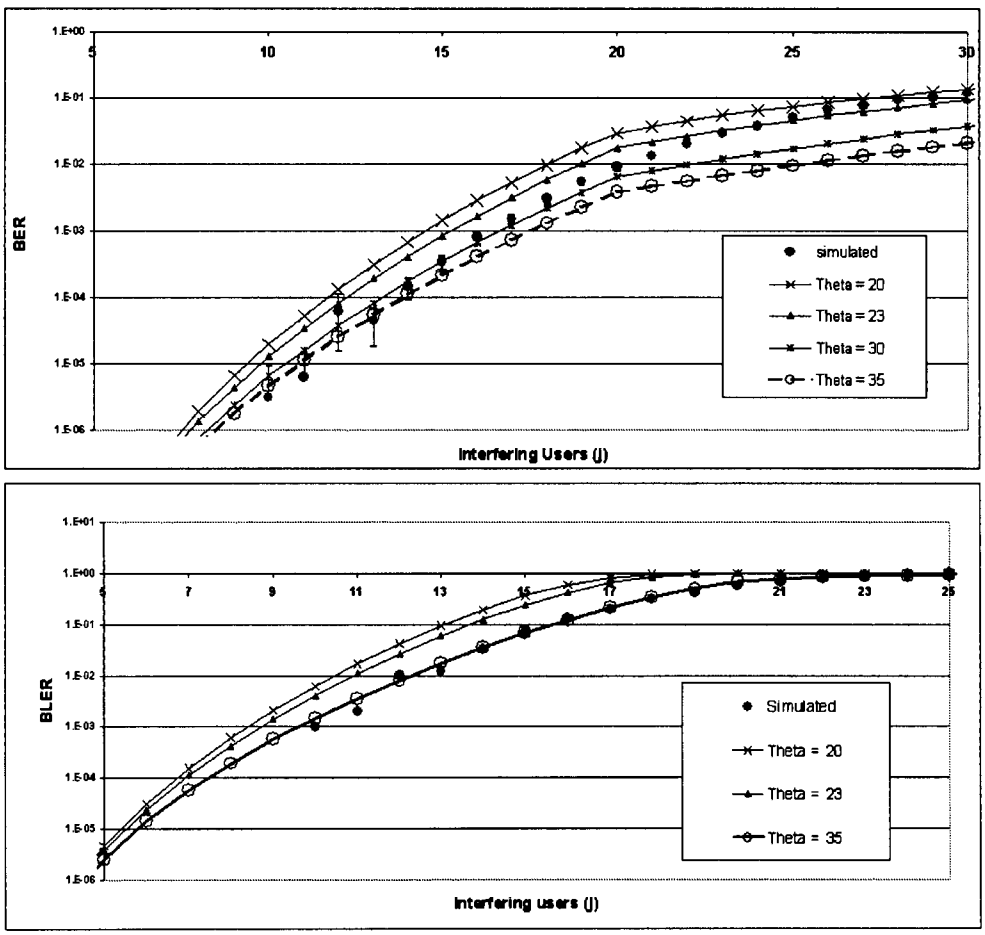

Figure 37: The BER (top) and BLER (bottom) as estimated by the HMM and the simulator.

The simulated BER values are all zero for $j < 10$, due to the fact that the FEC coding corrects all errors in this range. For small values of j, $\theta = 35$ provides the best match, and at high j, $\theta = 23$ provides the best match. This is more accurately shown in Table 6.

On the face of this, we have to choose whether or not we want our HMM to match well for low or high j. However, this choice is made easier when one considers the second plot in Figure 37, the BLER. For small j ($j \leq 20$), $\theta = 35$ provides the best

users (j)	BER	BLER
11	6.25E-06 ± 8.66E-06	0.002 ± 0.00339
12	6.25E-05 ± 4.65E-05	0.01 ± 0.00585
13	4.38E-05 ± 2.59E-05	0.012 ± 0.00753
14	1.44E-04 ± 5.20E-05	0.034 ± 0.01074
15	3.37E-04 ± 9.62E-05	0.067 ± 0.01623
16	8.09E-04 ± 1.69E-04	0.13 ± 0.01860
17	0.00149 ± 2.64E-04	0.201 ± 0.02447
18	0.00299 ± 4.13E-04	0.326 ± 0.02756
19	0.00535 ± 7.13E-04	0.427 ± 0.03049
20	0.0089 ± 9.29E-04	0.56 ± 0.03094
21	0.01324 ± 0.00115	0.68 ± 0.02971
22	0.01993 ± 0.00144	0.783 ± 0.02681
23	0.0287 ± 0.00190	0.867 ± 0.0222
24	0.03678 ± 0.00214	0.913 ± 0.01643
25	0.0499 ± 0.00248	0.961 ± 0.01299
26	0.06343 ± 0.00276	0.978 ± 0.00868
27	0.07591 ± 0.00290	0.989 ± 0.00728
28	0.09118 ± 0.00313	0.992 ± 0.00437
29	0.10251 ± 0.00326	0.996 ± 0.00277
30	0.11741 ± 0.00323	1

Table 5: Simulated values for the BER and BLER (presented in Figure 37).

fit. For $j > 21$, $BLER = 1$ for $\theta = 35$ (as well as for the simulation). In fact, the mean squared-error for the BLER in Table 6 shows that $\theta = 35$ provides the most accurate approximation for the BLER. This means that the difference in BER estimates when $\theta = 35$ and $j > 21$ are irrelevant, as only the BLER is actually used in the performance metric calculations in Section 4.3.1, while the BER is used to calculate the BLER. Thus, we will henceforth assume that $\theta = 35$.

6.3 Delay Results

This section presents the results that have been obtained for average packet delays. These results are derived from the model integration described in Chapter 4, Section

θ	BER ($j \leq 12$)	BER ($j > 12$)	BLER (all j)
20	0.00027	0.078525	1.3443
23	0.00014	0.082239	1.1061
30	4.71E-05	0.270319	0.576
35	3.88E-05	0.331728	0.1903

Table 6: Rooted mean-squared error the BER for low j and high j respectively (presented in Figure 37).

4.3.1.

6.3.1 $M^X/D/1$ Queue Parameterization

The results for the $M^X/D/1$ waiting time W_q stem from Equation 8, which defines the mean waiting time of a transport block in the queue. In order to use the equation, we have to calculate values for the parameter $E(X)$. $E(X)$ can be better described as the mean size of the batch process i.e. the average number of arrivals per batch. We assume that the packet lengths in our model are geometrically distributed, that is:

$$P_{len}(l) = (1-\Gamma)^{l-1}\Gamma \tag{70}$$

where $P_{len}(l)$ is the probability that a packet will be divided into l transport blocks, and $\frac{1}{\Gamma}$ is the mean packet length. It follows that $E(X) = \frac{1}{\Gamma}$.

The parameters ρ and λ, which are the utilisation and parameters respectively, must be considered jointly, as ρ is defined in terms of λ:

$$\rho = \frac{\lambda E(X)}{\mu}$$

We have that $\mu = \frac{1}{S_0}$ (Section 3.3, Page 59), where S_0 is the processing delay.

A transport block that has completed its wait in the queue will wait a further S_0 milliseconds to be processed and transmitted. Given our assumption that processing delays are small, and that the 10ms transmission interval dominates the service

distribution, we have that $0ms \leq S_0 \leq 10ms$. Approximate values of S_0 and Γ can be found through experimentation and validated using the simulated results (Section 6.3.3).

6.3.2 $BMAP/D/1$ Queue Parameterization

The $BMAP/D/1$ queue is parameterized using the `IP2BMAP` package [37], which is discussed in Chapter 3, Section 3.4.2. This tool greatly simplifies the parameterization of the BMAP, as all that is required is an IP trace file - consisting of a series of logged timestamps and packet sizes - and an average batch size. The $BMAP/D/1$ queue also requires a value for the mean service delay.

The IP trace files are discussed in Chapters 3 and 5, and are generated from measurements taken from an existing IP network. The batch size and mean service delay correspond to $\frac{1}{\Gamma}$ and S_0 in the previous section, and are dealt with in the same manner. The results are given in Section 6.3.3.

6.3.3 Results

$M^{[X]}/D/1$ Queue

Figure 38 shows the waiting time W when varying the parameters S_0 and λ (x-axis). The effect of increasing λ and S_0 is evident, and the performance eventually deteriorates beyond recovery for sufficient λ, due to the fact that the arrival rate grows beyond the departure rate. Figure 38 also shows W when varying the parameter Γ (or the mean batch size).

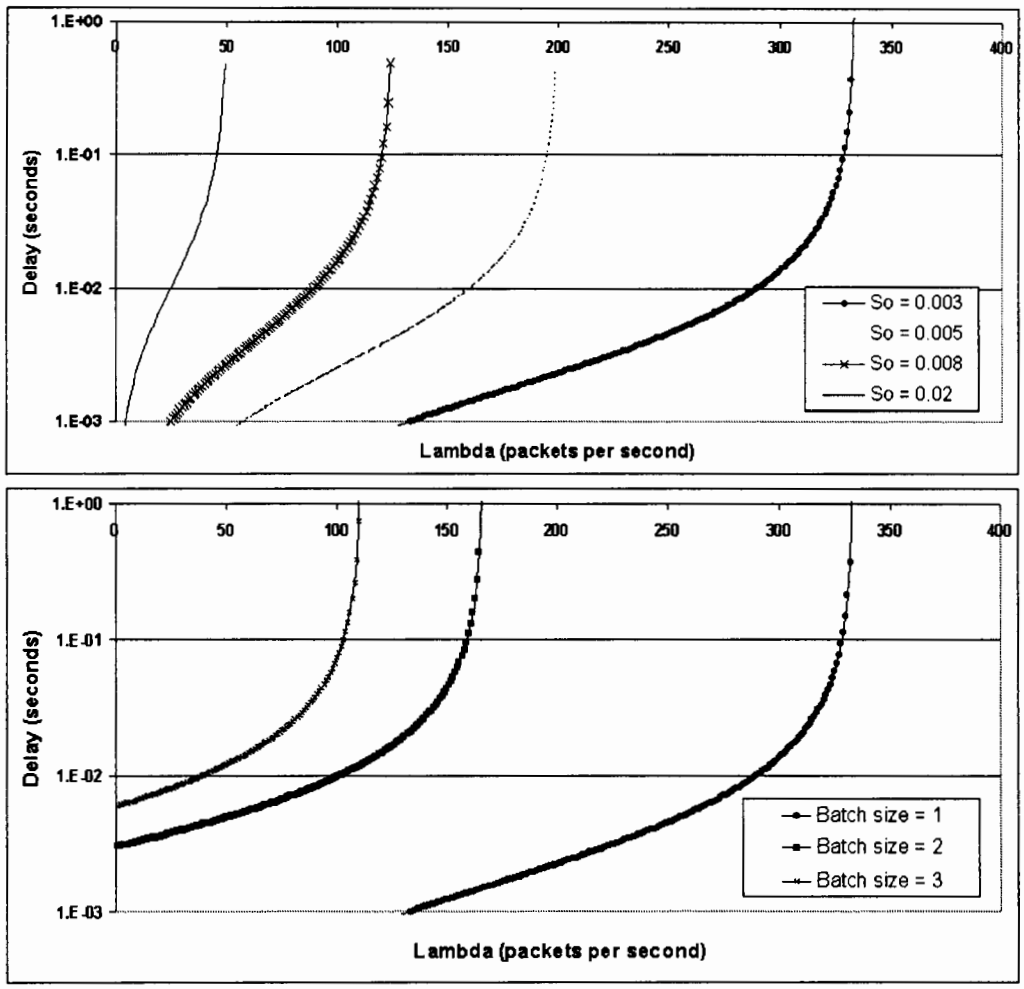

Figure 38: Delay vs. Load for the $M^{[X]}/D/1$ queue for different values of S_0 (top graph, $\Gamma = 1$) and Γ (bottom graph, $S_0 = 0.003$).

In order to ascertain the best (or most correct) parameters, we plotted the results of the analytical model with the results of the simulator. Figure 39, along with the tabulated results in Table 7, present this comparison. Note that the simulated results in Figure 39 are derived using a Poisson distribution for the traffic load. We have used this as a means of validating the model assumptions, such as the deterministic service time and the batch size. The adequate correlation achieved in Figure 39 serves

as a validation of the basic assumptions of the analytical model.

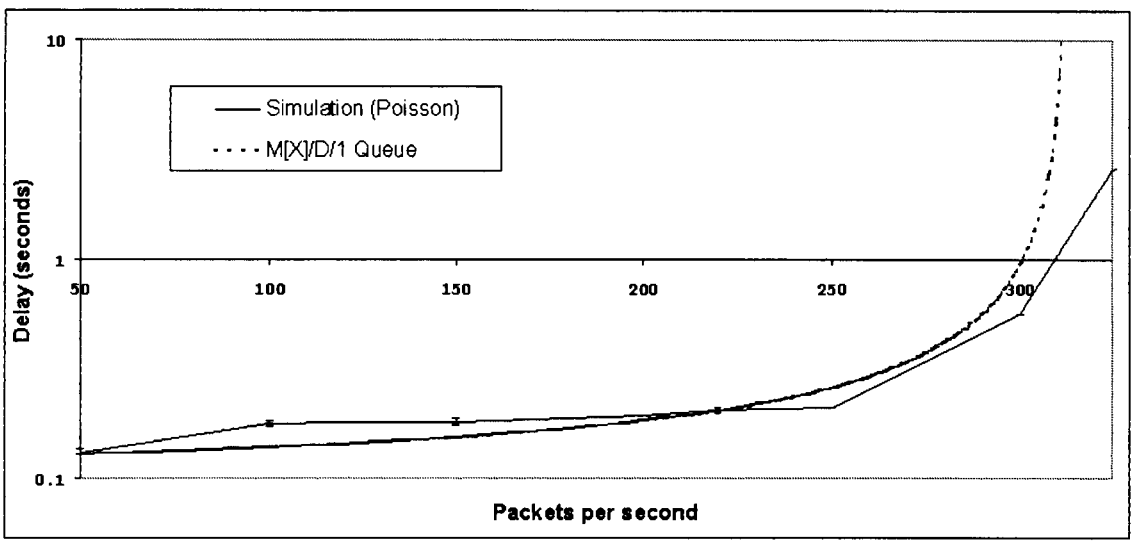

Figure 39: Delay vs. Load for the parameterized $M^{[X]}/D/1$ queue, with simulated results.

The results presented in Figure 39 are for $S_0 = 0.0035$ and $\Gamma = 1$. $\Gamma = 1$ provides fairly accurate results that support our assumption in Chapter 3 that it is more accurate to model the average batch size as being equal to one. The $M^{[X]}/D/1$ model is therefore overly complex, as an $M/D/1$ model will suffice.

$BMAP/D/1$ Queue

Figure 40 shows the $BMAP/D/1$ delay for increasing λ (x-axis), and for different values of S_0. The behaviour of the system due to changes in these parameters is, naturally, identical to the behaviour of the $M^{[X]}/D/1$ queue.

Packets per second	Delay (seconds)
50	0.1319 ± 0.00391
100	0.1782 ± 0.00505
150	0.1817 ± 0.0053
220	0.2044 ± 0.005697
250	0.2119 ± 0.00933
300	0.5609 ± 0.02456
400	4.4192 ± 0.10532

Table 7: Simulated delay values for the $M^{[X]}/D/1$ queue (presented in Figure 39).

As with Figure 39, Figure 41 shows the best-fit $BMAP/D/1$ model, validated with our simulation model. The results are also given in Table 9. Our determining factor for goodness of fit is the Mean-Square Error, the results of which are shown in Table 8. Note that the error is computed for all load values up until the point where the delay tends to infinity. The value of 0.0035 for S_0 proves to be the most accurate.

S_0	Mean Square Error
0.002	0.9831
0.0035	0.0671
0.004	0.4523
0.005	0.4273
0.006	230.6897
0.01	25458.4

Table 8: Rooted mean-squared error for the BMAP/D/1 model (presented in Figure 39).

The simulated arrivals are generated using IP trace data, and are thus more interesting results. They are also our main source of validation, as this data comes from empirical sources (see Chapter 5). The results shown in Figure 41 are for varying numbers of interferers and simultaneous IP users.

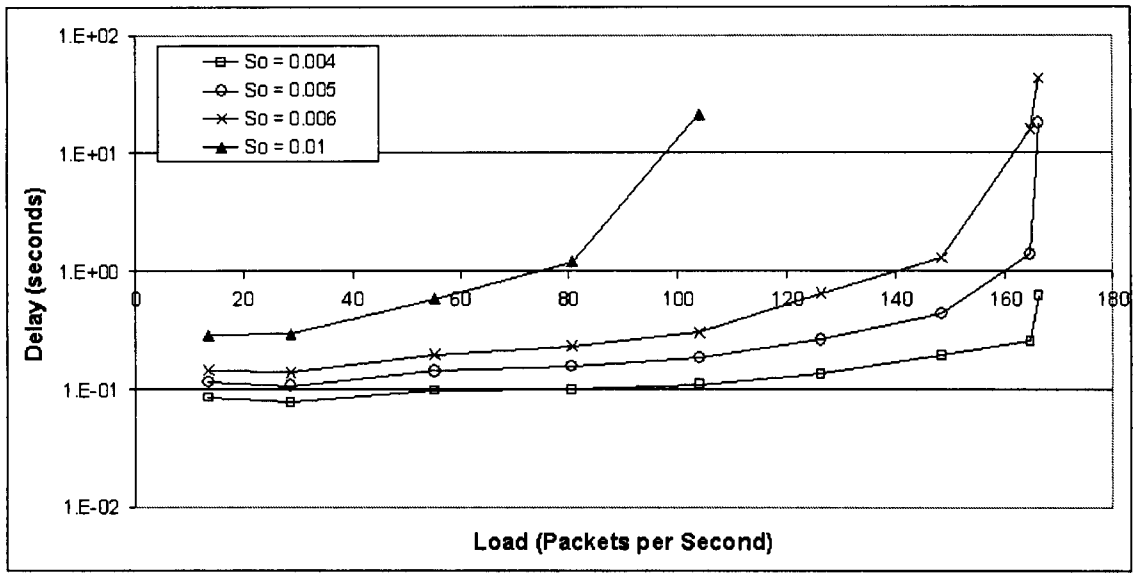

Figure 40: Delay vs. Load for $BMAP/D/1$ queue for different values of S_0 ($\Gamma = 1$).

As with Figure 39, the results in Figure 41 are derived for $S_0 = 0.0035$ and $\Gamma = 1$. This consistency is obviously encouraging, and serves as another validation of the modelling techniques used. It must be noted that when we refer to the load (in packets per second) is not per user, but is instead the overall load experienced by the system. In other words, a load of 200 packets per second arises due to a transmission rate of $\frac{200}{x}$ packets per second from each of the x IP users.

Comparison of $M^{[X]}/D/1$ and $BMAP/D/1$ Results

Figure 42 shows a comparison of the results obtained from the $BMAP/D/1$ and $M^{[X]}/D/1$ models. The simulation results for measured IP traffic are shown as well, and these values are presented in Table 10. This particular comparison is for $j = 10$ interfering users, and 6 IP users.

The difference between the Poisson and Markovian arrival processes is fairly well emphasised in Figure 42; while the BMAP model closely correlates with simulated values, the Poisson model grossly overestimates the performance of the system. The

Packets per second	Delay (seconds)- 6 IP users, 10 Interferers
13.47766	0.2296 ± 0.005055
28.53881	0.2317 ± 0.004628
55.21353	0.2435 ± 0.005294
80.58433	0.262 ± 0.005922
104.10352	0.2745 ± 0.005848
126.17180	0.2962 ± 0.006322
148.35587	0.3350 ± 0.006997
164.74246	0.7892 ± 0.018362
166.222	5.8762 ± 0.1307

Packets per second	Delay (seconds)- 1 IP user, 10 Interferers
13.47766	0.0981 ± 0.001378
28.53881	0.0967 ± 0.001351
55.21353	0.0979 ± 0.001375
80.58433	0.0985 ± 0.001370
104.10352	0.0976 ± 0.001397
126.1718	0.102 ± 0.001518
148.35587	0.1138 ± 0.00183
164.74246	0.5737 ± 0.01558
166.222	6.2573 ± 0.135964

Table 9: Simulated delay values for the $BMAP/D/1$ queue (presented in Figure 41).

Packets per second	Delay (seconds)
13.47766	0.2296 ± 0.005055
28.53881	0.2317 ± 0.004628
55.21353	0.2435 ± 0.005294
80.58433	0.262 ± 0.005922
104.10352	0.2745 ± 0.005848
126.17180	0.2962 ± 0.006322
148.35587	0.3350 ± 0.006997
164.74246	0.7892 ± 0.018362
166.222	5.8762 ± 0.1307

Table 10: Simulated delay values for the $BMAP/D/1$ queue (presented in Figure 42).

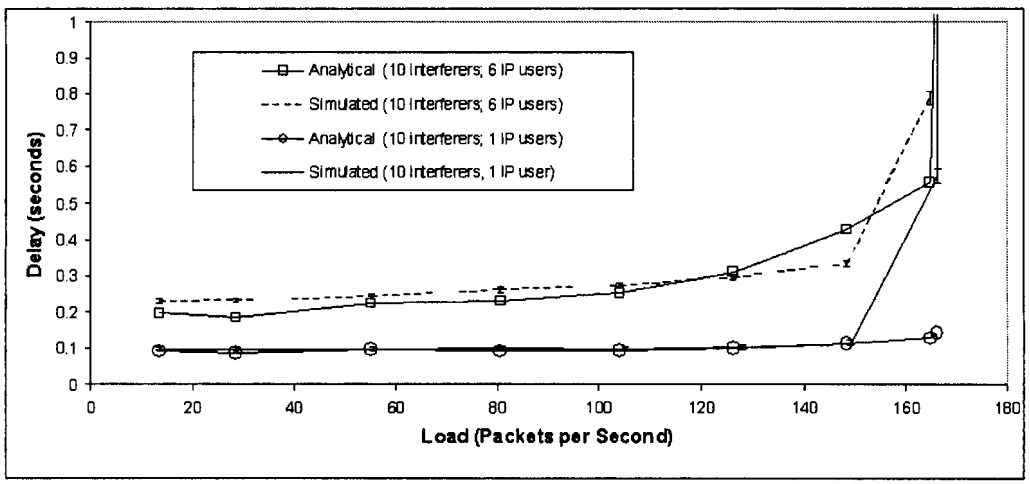

Figure 41: Delay vs. Load for the parameterized $BMAP/D/1$ queue, with simulated results.

reason for this is the fact that the IP-based traffic is naturally bursty, which means that any packet that arrives in the queue will invariably arrive at a very busy period because it is part of a sudden burst of data. The Poisson traffic, however, is far more evenly distributed, and arrivals have a greater chance of arriving at the queue with relatively few other packets, thus decreasing the average delay on the packet dramatically. More precisely, the Poisson process exaggerates the performance of the system by a factor of three, which is far too inaccurate to be of any practical use. The benefit of modelling with the Markovian arrival process is therefore evident.

Another factor to consider is that the number of blocks that can be placed onto the DSCH per user decreases as the number of simultaneous DSCH users increases, because the DSCH is multiplexed on a frame by frame basis. This means that the DSCH is does not perform well when there are a large number of packet network users. We recall that the simulation uses a First Come First Served service pattern, which means that the user to whom the packet at the front of the queue belongs is served first. All of the packets belonging to this particular user are transmitted, and the packets belonging to the next user on the queue are transmitted. If there are

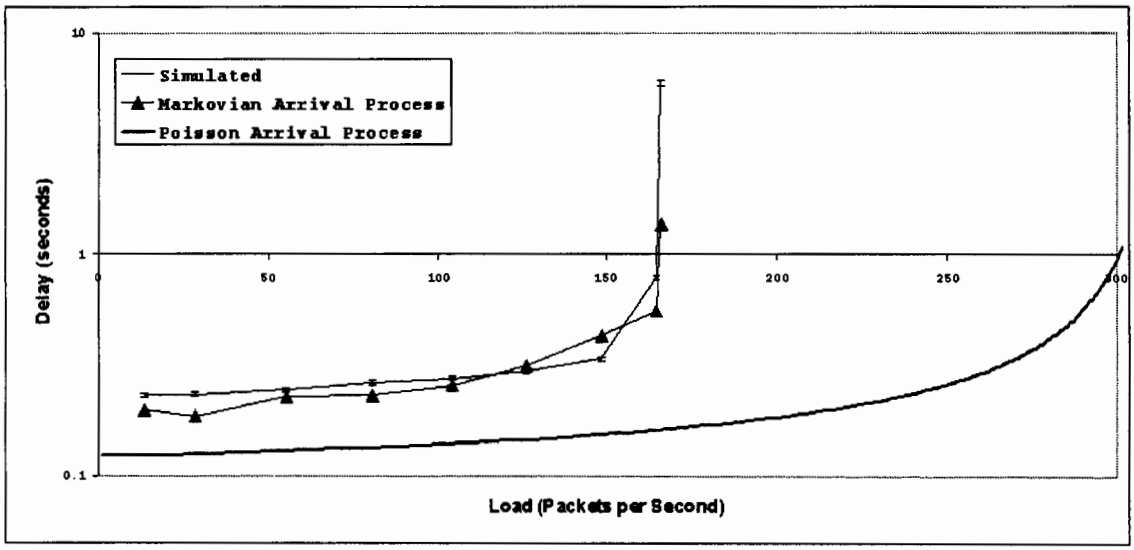

Figure 42: Delay vs. Load for the parameterized $BMAP/D/1$ and $M^{[X]}/D/1$ models, with simulated results.

many users using the DSCH, there is a much lower chance that any one user will have the use of the DSCH during consecutive time slots, and will thus have long waits between transmissions. This is evident in Figure 41 when comparing the results for 1 IP user with the results of 6 IP users.

6.4 Performance Analysis

6.4.1 Radio Channel Performance

The BER and BLER results for the 384Kb/s UMTS DSCH have been discussed and shown in Figure 37. The BER results are particularly interesting, especially when compared to the non-turbo coded simulation in Figure 43. The value of Turbo codes is obvious from these results, despite the drawback of increased processing at both the receiver and the transmitter.

The BLER results in Figure 37 are far more severe, as the BLER is equal to 1 at

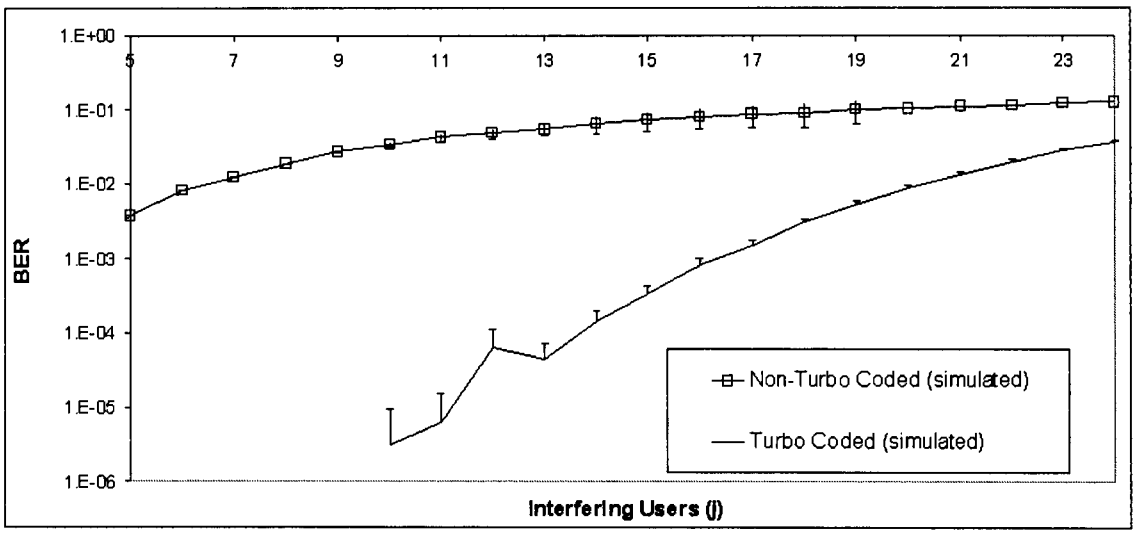

Figure 43: BER results with and without Turbo coding.

high interference levels. Any real system would have to manage the number of users per cell to avoid this drop in performance.

6.4.2 Packet Network Performance

Queue Buffer Sizes

Figure 41 shows the BMAP and simulated results which are generated using measured IP trace data. The mean delay per IP packet is between 0.1 ms and 0.5 ms when the load on the system is less than 160 packets per second. At approximately 170 packets per second, the mean delay experienced by each IP packet tends to infinity. To obtain a realistic estimate for buffer sizes, we will assume that the minimum performance level of the system would be a 1 second delay per IP packet, or at a load of 165 packets per second. If we assume that the average IP packet is $P = 1000$ bits long, then we can use Little's Law to obtain the minimum buffer size L required at each UTRAN network element:

$$\begin{aligned} L &= (\lambda \cdot W) \cdot P \\ &= 165 \cdot 1 \cdot 1000 \\ &= 165000\, bits \\ &\geq 20000\, bytes \\ &= 20\, Kb \end{aligned}$$

Therefore, buffer sizes must be larger than 20Kb to be able to provide the minimum performance requirements.

Throughput

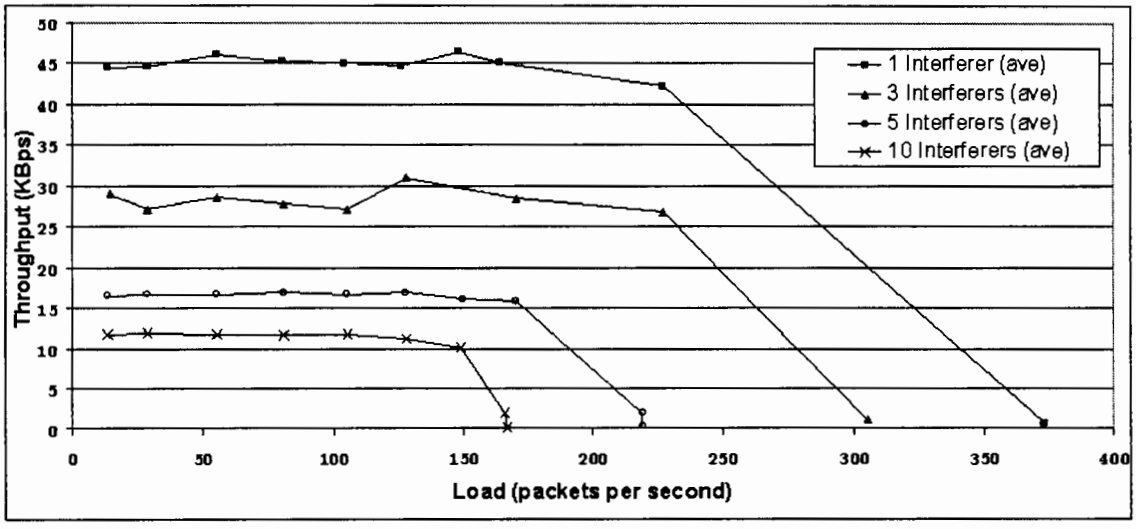

Figure 44: Channel throughput for 1 and 6 simultaneous packet network users.

The data rate for the 384Kb/s DSCH channel is calculated in the following way: if there are 15 slots available per frame, and if there are 100 frames transmitted per second, then, given that there are 256 bits of raw usable (i.e. non-header) data per

slot, the data rate is $15 \cdot 100 \cdot 256 = 384$ Kb/s. This data rate, however, assumes that all 15 slots of the DSCH are utilised. Figure 44 shows how the throughput of the channel differs for varying sizes of mean interfering users for an 6 concurrent channel users. The behaviour with respect to increasing interference is as we would assume: the performance drops significantly as the mean interference increases, and the maximum load also decreases as the interference increases. It is interesting to note that with our model, the DSCH never reaches its maximum throughput of 384 KBps: the best performance noticed is roughly 46 Kbps. This can be understood be noting that at a low packet load, there are too few packets arriving within the 10ms frame boundary to fill the entire frame; at a high packet load, the frames are filled, but the average delay experienced by each packet due to queueing is much larger. Therefore, though the maximum channel throughput is 384Kbps, the actual experienced throughput is much lower. Note that we have only investigated the use of one DSCH per cell. It is reasonable to assume that by increasing the number of available shared channels, the perceived throughput will also increase.

6.5 Summary

In this chapter we have presented the results of both analytical models. The results for the Bit Error Rate (BER) and the Block Error Rate (BLER), derived via the Hidden Markov Model (HMM) from Chapter 4, were presented first, along with simulated values. The correlation between the analytical and simulated BLER is sufficient for us to have a high level of confidence in the validity of the model. The HMM is parameterized with $\theta = 35$. Values for the Turbo coding approximation is also given.

The queueing models have also been presented, both as separate modules, as well as integrated with the HMM results. Both queues were found to be best parameterized as single batch arrivals, as opposed to multiple batches, which we assumed would be the case (Chapter 3). The Poisson arrival process in the $M^{[X]}/D/1$ model is shown to be drastically inferior to the Markovian arrival process in the $BMAP/D/1$ model in terms of its ability to model IP traffic effectively. The BMAP model correlated well with the simulated values.

The performance of the DSCH was also analysed. The minimum buffer sizes were found to be realistic. Furthermore, the utilisation of the channel was also found to be quite low as the number of packet network users increased.

Chapter 7

Conclusions

7.1 Physical Channel Models

The HMM proved to be effective for modelling highly complex phenomena such as Ricean Fading and Multiple Access Interference, despite its approximations and simplifying assumptions. However, the combination with Turbo Coding is not elegant and the method provided is not a very general solution, despite the fact that it provides the necessary functionality that we require.

The HMM was well suited to modelling the conditions that were of interest to us as it is a narrowband model, which accurately reflects the conditions in an indoor environment. However, this fact means that the model is not well suited to wideband conditions.

The HMM model of the DSCH can very easily be extended to the High Speed DSCH (HS-DSCH) and similar UMTS high-speed enhancements. Overall, the HMM approach provided us with adequate results that were derived in a fairly simple manner.

7.2 Packet Network Models

The packet network described in Chapter 2 is a small sub-system within the UMTS network. Despite this, it is still fairly complex. By using queueing theory to model this system (Chapter 3), we were able to simplify the model to a sufficient level so as to make the system easier to understand and analyse, without losing any significant details.

The challenge of modelling this particular packet network lay in finding a good approximation for IP traffic. Our results have clearly shown the advantage of using a Batch Markovian arrival process (BMAP) as opposed to the Poisson arrival process. The complexity of the BMAP is balanced by the greater accuracy that it provides. In fact, by using the `ip2bmap` package provided by Lindemann *et al* [37], the complexity of the model is greatly reduced, while the accuracy is greatly increased. This is because there is no need to derive a closed solution for the waiting time distribution, which eliminates the need to know complex statistical mathematics, while the

accuracy of the model is improved by parameterizing it with measured IP trace data.

To summarise, the queueing models were an effective abstraction of the UMTS IP network. The $BMAP/D/1$ model proved to be fairly accurate, and the first choice for our model.

7.3 UMTS Packet Services

The Third Generation UMTS IP services must compete with existing WLANs such as IEEE 802.11 to provide users with mobile Internet connectivity. UMTS has the advantage of greater coverage, as well as a single, integrated network that has an efficient and effective multiplexing technology, namely CDMA. The DSCH option can provide fairly high data rates (384 kbps, up to 2048 kbps) under low interference, and is capable of handling high traffic loads, although the performance deteriorates during high interference. Other options, such as the High Speed DSCH (HS-DSCH) should be used for time-sensitive data, but the DSCH performs sufficiently well to provide non-real time packet services across the UMTS network. The HS-DSCH provides channel frames at 2ms intervals, which would increase throughput for the reasons described in Chapter 6, namely that the IP packets waste a large amount of time waiting for transmission. A smaller transmission window would greatly improve throughput.

7.4 Conclusions And Future Work

7.4.1 Summary

this work aimed to provide a means of modelling IP traffic in UMTS networks. In this chapter we have concluded that the models we have discussed and developed throughout the course of this work provide sufficiently accurate results in order to provide meaningful insights into the performance of the system in question. However, the approach used to model the physical channel needs to be refined and generalised, in particular the Turbo coding analysis.

7.4.2 Future Work

Future work should focus on comparing both the DSCH and the HS-DSCH options with existing wireless technologies to determine the benefits of each. An HMM that more accurately considers Turbo Coding should also be investigated and developed, keeping in mind that simplicity is the key feature of such a model. As a final point, a single, standalone, general tool should be developed that provides a simple means of modelling IP traffic with BMAP queues, as the theory involved is highly complex for non-experts.

Bibliography

[1] ETSI 3GPP TS 22.060. GPRS Service Description: Stage 1. Available online: ftp://ftp.3gpp.org. Accessed: January 2004.

[2] ETSI 3GPP TS 23.060. GPRS Service Description: Stage 2. Available online: ftp://ftp.3gpp.org. Accessed: January 2004.

[3] ETSI 3GPP TS 25.211. Physical channels and mapping of transport channels onto physical channels (FDD). Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[4] ETSI 3GPP TS 25.212. Multiplexing and channel coding (FDD). Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[5] ETSI 3GPP TS 25.213. Spreading and Modulation. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[6] ETSI 3GPP TS 25.301. Radio Interface Protocol Architecture. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[7] ETSI 3GPP TS 25.302. Services provided by the Physical Layer. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[8] ETSI 3GPP TS 25.321. UMTS: MAC protocol specification. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[9] ETSI 3GPP TS 25.323. PDCP Protocol Specification. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[10] ETSI 3GPP TS 25.401. UTRAN Overall Description. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[11] ETSI 3GPP TS 26.937. Transparent end-to-end packet switched streaming service (PSS); RTP usage model. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[12] ETSI 3GPP TS 34.108. Common test environments for User Equipment (UE) conformance testing. Available online: ftp://ftp.3gpp.org. Accessed: April 2003.

[13] A. Klemm, C. Lindemann and M. Lohmann. Traffic Modeling and Characterization for UMTS Networks. In *Globecom 2001 Internet Performance Symposium*, pages 1741 – 1746, 2001.

[14] A. Klemm, C. Lindemann and M. Lohmann. Modeling IP Traffic Using the Batch Markovian Arrival Process. *Performance Evaluation* , 54:149–173, 2003.

[15] C. Caldera A. Mate and M. Rinne. Performance of the Packet Traffic on the Downlink Shared Channel. Nokia Research Group, Finland.

[16] Anders Andersson. Capacity Study of Statistical Multiplexing for IP Telephony. Technical Report T2000:03, SICS - Swedish Institute of Computer Science, January 2000. MSc thesis.

[17] Anders Furuskär and Stefan Parkvall and Magnus Persson and Maria Samuelsson. Performance of WCDMA High Speed Packet Data.

[18] Cecilio Pimentel and Ian F. Blake. Modelling Burst Channels Using Partitioned Fritchman's Markov Models. *IEEE Transactions on Vehicular Technology*, 47:885–899, 1998.

[19] J Cheng and N Beaulieu. Accurate DS-CDMA Bit-Error Probability Calculation in Rayleigh Fading. *IEEE Transactions on Communications*, pages 3 – 15, January 2002.

[20] David M. Lucantoni. New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7(1):1–46, 1991.

[21] David M. Lucantoni. The BMAP/G/1 queue: A tutorial. *Models and techniques for Performance Evaluation of Computer and Communications Systems*, pages 330–358, 1993.

[22] Dr Wolfgang Granzow. *3rd Generation Mobile Communications Systems.* University of Erlangen Press, 2003.

[23] E Dinan, A Kurochkin and S Kettani. UMTS Radio Interface System Planning and Optimization. *Bechtel Telecommunications Technical Journal*, 2002.

[24] R. Garello. The Turbo Code Minimum Distance Algorithm. Available online: http://www1.tlc.polito.it/ garello/turbodistance/turbodistance.html.

[25] E Geraniotis. Direct-Sequence Spread-Spectrum Multiple-Access Communications Over Nonselective and Frequency-Selective Rician Fading Channels. *IEEE Transactions on Communications*, pages 756 – 764, August 1986.

[26] E Geraniotis and M Pursley. Performance of Noncoherent Direct-Sequence Spread-Spectrum Communications over Specular Multipath Fading Channels. *IEEE Transactions on Communications*, pages 219 – 226, March 1986.

[27] C. Berrou A. Glavieux and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo codes. In *Proceedings of ICC 93*, pages 1064–1070, 1993.

[28] D. Gross and CM. Harris. *Fundamentals of Queueing Theory, Third Edition.* John Wiley and Sons, Inc., 1998.

[29] S. Gruhl. Opportunistic QoS Scheduling for Cellular Packet Data. PhD Thesis. Accessed: April 2003.

[30] H Bourlard and N Morgan. Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions. *Adaptive Processing of Sequences and*

Data Structures, Volume 1387 of Lecture Notes in Artificial Intelligence, pages 389–417, 1998.

[31] H. Tijms. *Stochastic Models: an Algorithmic Approach.* John Wiley and Sons, Inc., 1994.

[32] Simon Haykin. *Communications Systems.* John Wiley and Sons, Inc., 2001.

[33] J Abate and W Whitt. The Fourier-Series Method for Inverting Transforms of Probability Ditributions. *Queueing Systems*, 1991.

[34] Garth Judge and Fambirai Takawira. A Simple Hidden Markov Model for a CDMA Channel with Correlated Rayleigh Fading. *The Transactions of the SAIEE*, March:17–26, 2002.

[35] L. Kleinrock. *Queueing Systems, Volume 1: Theory.* John Wiley and Sons, Inc., 1975.

[36] J. W. Lee and R. E. Blahut. Bit error rate estimate of finite length turbo codes. In *IEEE 2003 International Conference on Communications (ICC 2003), Anchorage, AK*, May 2003.

[37] C. Lindemann and M. Lohmann. ip2bmap software package. Available online: www.ip2bmap.de.

[38] Lourens Walters. A Web Browsing Workload Model For Simulation. Master's thesis, 2004.

[39] B. Nordgren M. Degermark and S. Pink. RFC 2507 - IP Header Compression. Available online: http://www.faqs.org/rfcs/rfc2507.html.

[40] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach.* The John Hopkins University Press, 1981.

[41] Laurence B. Milstein Michele Zorzi, Ramesh R. Rao. Error Statistics in Data Transmission over Fading Channels. *IEEE Transactions on Communications*, 46:1468–1477, 1998.

[42] Erwin Mondre. Complex and Envelope Covariance for Rician Fading Communication Channels. *IEEE Transactions on Communications*, pages 80 – 84, February 1971.

[43] M.B. Pursley. Performance Evaluation for phase-coded spread-spectrum multiple access communication - Part I: System Analysis. *IEEE Transactions on Communications*, COM-25(8):795 – 799, August 1977.

[44] R. Garello and P. Pierleoni and S. Benedetto. Computing the Free Distance of Turbo Codes and Serially Concatenated Codes with Interleavers: Algorithms and Applications. *IEEE Journal on Selected Areas in Communications*, 19:800–812, May 2001.

[45] T.S. Rappaport. *Wireless Communications: Principles & Practices.* Prentice Hall, 1996.

[46] W. Ryan. A turbo code tutorial, 1997. submitted to Globecom 1997.

[47] S. Malik and D. Zeghlache. Improving Throughput and Fairness on the Downlink Shared Channel in UMTS WCDMA Networks. *European Wireless Conference*, February 2002.

[48] William Turin and Robert van Nobelen. Hidden Markov Modeling of Flat Fading Channels. *IEEE Journal on Selected Areas in Communications*, 16:1809–1817, 1998.

[49] H.S. Wang and P Chang. On Verifying the First-Order Markovian Assumption for a Rayleigh Fading Channel Model. *IEEE Transactions on Vehicular Technology*, 45(2):353 – 357, May 1996.

[50] M. Zorzi, R. Rao, and L. Milstein. the accuracy of a first-order markov model for data transmission on fading channels. In *Proc. IEEE ICUPC'95*, pages 211–215, November 1995.

[51] Michele Zorzi. Capture Probabilities in Random Access Mobile Communications in the Presence of Ricean Fading. *IEEE Transactions on Vehicular Technology*, Feb, 1997.