Performance Evaluation of AAL2 over IP in the UMTS Access Network Iub interface

Prepared by: Bongani R. Chabalala

Supervised by: Neco Ventura

Department of Electrical Engineering
University of Cape Town
31 January 2005

This dissertation is submitted to the University of Cape Town in fulfillment of the academic requirements for the Degree of Master of Science in Engineering.

Declaration

I declare that this thesis is my own work. Where information from other sources has been used herein the relevant material has been referred to in the acknowledgements or references as appropriate.

This work is being submitted for the Master of Science Degree in Electrical Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

Signed by candidate

Bongani R. Chabalala

19/05/05

Date

Acknowledgements

I wish to express my **sincerest** gratitude to the following organisations and individuals for their invaluable support and encouragement during this research:

- Telkom, for providing financial support to me during this project.
- Mr Neco Ventura, for giving me freedom to choose a project of my interest and supervising me.
- Sven Shepstone, for sharing invaluable experience in many aspects of my project and for reviewing the whole of this thesis document.
- David Waiting, for helping in coding, ***sniffing*** the IP packets and proof-reading this document. *Dude, your superb assistance is and will ever be much appreciated*
- Albert Hasson, my cubicle mate, for all your superb help during my project.
- Vitalis, Marshalleno and Tyrell for all your assistance in this thesis.
- Marathon Ntusi, for proof-reading the first two chapters of this document.
- My family, i.e. Mom, Masingita, Langu, Tsundzu na Nhluri, for your everlasting love. ***Dear Mom***, *your daily prayers and encouragement are much appreciated!*
- Uncle Mosa, for supporting me throughout until today. *Inkomu swinene Malume!*
- Jomo Makhubele and Kenneth Mashila, without whom my career would not have been possible, for advising me to re-consider taking **Science subjects in std 9**.
- Kenneth Mashila, for your splendid advises since my early days in varsity until today. "*I know you want to see me doing PhD, but not now. Ndza khensa Hodela*"
- Chileshe "Chichi" Kapapula and Mafanelo "Mafa" Sibuyi, for being the batteries of my flashlight when my days were very dark. *"You, you thaaank Mafa"*
- To all my **other friends** and colleagues in the CRG lab, for motivating me.
- **But above all**, I am eternally grateful to God for sending His angels who keep watching over my soul daily and who kept my coffee mug full whenever I badly[1] needed it. To You Father God belong honour and thanks **forever**.

[1] Proverbs 3: 5-6, Jermiah 29: 11, Isaiah 41: 13, Psalm 113: 3, Psalm 23, Psalm 145: 18, Philippians 4: 13

Synopsis

Third generation (3G) networks based on the Universal Mobile Telecommunication Service (UMTS) are currently being deployed world-wide. 3G combines today's mobile system services such as voice and the Internet. With 3G, mobile users will access a wide range of services at higher data rates. A mobile accesses UMTS through the Iub interface. In the Iub interface, the Base Station (Node B) is connected to the Radio Network Controller (RNC) through 2 Mbps ATM links. ATM was adopted with AAL2 to transport low bit-rate traffic. However as the number of mobile users increases, ATM does not scale well in terms of bandwidth utilisation. For this reason, the 3GPP specified IP as an alternative transport layer in the Iub interface. After IP was specified, research begun looking at ways of replacing the ATM infrastructure with IP.

In this study, we proposed to retain AAL2 and lay it over IP (AAL2/IP). The IP-based Iub interface is therefore designed to tunnel AAL2 channels from the Node B to the RNC. Currently IP routes packets based on best-effort which does not guarantee QoS. To provide QoS, MPLS integrated with DiffServ is proposed to support different QoS levels to different classes of service and fast forward the IP packets within the Iub interface. To evaluate the performance of AAL2/IP in the Iub interface, a test-bed was created. Multiple AAL2 sources generated traffic to maximise the utilisation of the 2 Mbps IP link from the Node B to the RNC. A maximum number of sources that can be supported by this link while meeting the 5 ms delay limit was investigated. The 5 ms delay is specified by the 3GPP for quality voice from the Node B to the RNC.

A performance trade-off was investigated between the IP packet size that carries AAL2 packets, the packetisation delay of this IP packet and link utilisation from the Node B to RNC. A series of tests was conducted and the results obtained are presented. AAL2/IP proved to use bandwidth more efficiently as the number of AAL2 sources increased as compared to AAL2/ATM.

Table of Contents

Declaration i

Acknowledgements ii

Synopsis iii

Table of Contents iv

List of Figures vii

List of Tables viii

Glossary ix

1 Introduction 1

1.1 Background Information 1

1.2 Problem Description 6

1.3 Thesis Objectives 6

1.4 Scope and Limitations 7

1.5 Thesis Outline 8

2 Background Theory and Literature Review 11

2.1 Introduction 11

2.2 Characteristics of Voice 12

2.3 AAL2 Tunnelling in the Iub Interface 14

2.4 QoS in the Iub Interface based on IP 19

2.5 Transport Layers: AAL2 vs UDP 24

3 Design for tunnelling AAL2/IP in the Iub Interface 25

3.1 Introduction 25

3.2 Motivation for AAL2/IP in the Iub Interface 26

3.3 Performance Evaluation Metrics 28

3.4 The UTRAN Topology 30

3.5 The UTRAN Functional Requirements 31

3.6 Definition of the Node B Design 34

3.7 Design for the UTRAN 37
4 A Framework to Evaluate AAL2/IP 44
4.1 Introduction 44
4.2 UTRAN Test-bed Architecture 44
4.3 Traffic Generator and Receiver 47
4.4 Iub Interface Emulator Choice 51
4.5 Dummynet Operation 54
4.6 Traffic Collection 56
4.7 Project Development 59
5 Framework Evaluation and Results 61
5.1 Introduction 61
5.2. Link Efficiency – without QoS 61
5.2.1 Results 62
5.2.2 Analysis of Results 62
5.3 Link efficiency – with QoS 64
5.3.1 Results 64
5.3.2 Analysis of Results 65
5.4 Number of Users – without QoS 67
5.4.1 Results 67
5.4.2 Analysis of Results 68
5.5 Number of Users – with QoS 69
5.5.1 Results 69
5.5.2 Analysis of Results 70
5.6 End-to-end (Node B-to-RNC) Delay 71
5.6.1 Results 72
5.6.2 Analysis of Results 73
5.7 Jitter 73
5.7.1 Results 74
5.7.2 Analysis of Results 75
5.8 Packet Losses 75
5.8.1 Results 75

5.8.2 Analysis of Results 76
5.9 Summary 76
6 Conclusions and Recommendations 79
6.1 Conclusions 79
6.2 Recommendations 81
References 84
Appendices 87
Appendix A: AAL2 and UDP Headers Description 87
Appendix B: The CPS transmitter and receiver 88
Appendix C: DiffServ and MPLS technologies 90
C1: Differentiated Services (DiffServ). 90
C2: Multi-protocol Label Switching (MPLS) 91
C3: Support for DiffServ in MPLS domains 92
Appendix D: SDL Specification 94
Appendix E: Dummynet and NTP 95
Appendix E1: Dummynet pipe and queue configuration 95
Appendix E2: Network Timing Protocol (NTP) 96

List of Figures

Figure 1-1: UMTS network architecture 3
Figure 1-2: UTRAN Iub interface infrastructure 5
Figure 2-1: Multiplexing Users onto an ATM Link using AAL2 15
Figure 2-2: AAL2 Multiplexer at the Node B 16
Figure 2-3: The access network protocol stack based IP transport 19
Figure 3-1: UTRAN network topology 31
Figure 3-2: SDL Representation of AAL2 Tunnelling at the Node B 35
Figure 3-3: Node B Design for Tunnelling AAL2 over IP 40
Figure 4-1: Emulated UTRAN network test-bed, with dummynet router 45
Figure 4-2: Physical network layer of UTRAN test-bed (uplink) 46
Figure 4-3: Test-bed components used to implement AAL2 tunnelling over IP 48
Figure 4-4: CPS Packets Multiplexer at the Node B of figure 4.3 50
Figure 4-5: Bidirectional dummynet pipe architecture 55
Figure 4-6: The results of the NTPQ command output. 57
Figure 4-7: (a) PDU logfile showing 10 CPS packets, (b) the CPS packet contents 58
Figure 5-1: Efficiency vs Payload (for AAL2/ATM and AAL2/IP) 63
Figure 5-2: Link efficiency when QoS have been introduced 66
Figure 5-3: Timer-CU effects on the link efficiency 66
Figure 5-4: Comparison of AAL2/ATM and AAL2/IP Efficiency 68
Figure 5-5: Number of users with QoS, 116 users were supported 71
Figure 5-6: Logfile showing PDUs received at the RNC 72
Figure 5-7: Timer-CU effects on the Node B-to-RNC delay 73
Figure 5-8: IP packets jitter (delay variation) in the dummynet pipe 74
Figure 5-9: Number of packets lost in the pipe vs packets sent from the Node B 76
Figure 5-10: Timer-CU effects on the link efficiency and 5 ms Node B-to-RNC delay. . 78

List of Tables

Table 3-1: Mobile ID to Channel ID and Tunnel Mapping at the Node B........................38
Table 5-1: Number of CPS packets supported given different 5 ms delay budget values 65
Table 5-2: Number of users supported by 2 Mbps link on different 5 ms delay budgets .71
Table 5-3: Framework set-up for a series of tests performed...77
Table 5-4: Timer-CU effects on both link efficiency and end-to-end delay 78

Glossary

For the convenience of the reader, a glossary of the terms which occur through-out this document is provided.

2G, 3G, 4G Second, third, fourth generation networks

3GPP 3G Generation Partnership Project - a 3G standardisation body, established in December 1998 to co-operate in the production of 3G based on GSM.

ATM Asynchronous Transfer Mode – a cell-based technology that provides multiplexing gain and guarantees services to end users.

AAL2 ATM Adaptation Layer Type 2. It was standardised in 1997 to multiplex short and variable-length packets from various low bit-rate applications onto ATM cells. In this study, it multiplexes packets into IP packets.

AAL2 Channel A unique AAL2 channel, characterised by a CID. Multiple AAL2 channels are multiplexed onto a single AAL2 tunnel

AAL2 Tunnel Similar to an ATM virtual connection when carrying AAL2-type traffic. An AAL2 tunnel may contain up to 248 AAL2 channels.

AMR codec Adaptive Multi-Rate codec – a codec recommended by 3GPP for UTRAN, generates a 31-byte frame every 20 ms at 12.2 kbps.

CID Channel Identifier – identifies channel inside an AAL2 tunnel

CODEC Compressor or decompressor algorithm for voice or video traffic

CPS Common Part Sub-layer – It is an AAL2 sub-layer responsible for converting between CPS packet format and AAL2 PDU format.

DCH Dedicated Channels - a logical channel that is used to carry packet data such as IP packets.

FIFO First In First Out – implies that data is released from the buffer in the order of its arrival

FP Frame Protocol - It is responsible for the relaying of transport channels between the mobile device and the RNC via the Node B.

IP Internet Protocol - It is a connectionless, packet-switched protocol where data is transported in discrete packets independently from each other. It is a standard network infrastructure for global communication in the world.

Iub interface A communication link or network that connects the Node B to the RNC.

ITU International Telecommunication Unit – It is a leading publisher of telecommunication technology, regulatory and standards information.

LSP Label switched paths – a path created for the IP packets to flow from the Node B to the RNC.

MID Mobile Identifier – identifies a mobile that uses a channel inside an AAL2 tunnel

MPLS Multi-Protocol Label Switching - It creates paths, like ATM, that IP packets follow on a network, instead of IP packets using undetermined routes. It improves the forwarding speed of the routers by using simple labels.

MTU Maximum Transfer Unit - Ethernet MTU is 1500 bytes (maximum IP packet).

Node B The terminology used in UMTS to refer to the Base Station. It receives traffic from the mobile device and forwards it to the RNC within the UTRAN.

NTP Network Timing Protocol - an open source tool widely used in the Internet to synchronise clocks for computers.

PDU Protocol Data Unit - an IP packet payload that carries CPS packets.

PSTN Public Switched Telephone Network - It is an international telephone system based on copper wires carrying analog voice. It is in contrast to new digital technologies such as Integrated Services Digital Network (ISDN).

QoS Quality of Service – It is an assurance of end-to-end delivery service perceived on the application level, for applications like voice and video

RED Random Early Detection - drops packets from the competing channels according to their bandwidth usage.

RNC Radio Network Controller - the RNC is the terminology used in UMTS. In GSM, it is called the Base Station Controller (BSC).

RRM Radio Resource Management - Manages and ensures that the radio network resources, e.g. bandwidth, are available within the radio access network.

RTP Real Time Protocol - supports end-to-end real-time traffic.

SDL Specification and Description Language – the language used to specify and describe the protocol control for circuit-switched basic calls.

TTI Transmission Time Interval – the inter-arrival time of the frames at the Node B from the mobile.

Timer-CU Time unit to ensure that the outgoing PDUs are not kept too at the Node B.

UCT Universal Coordinated Time - formerly and still widely called Greenwich Mean Time (GMT), is the standard time common to every place in the world.

UDP User Data Protocol – An IP transport protocol for real-time (voice) traffic.

UMTS Universal Mobile Telecommunications System – It is one of the 3G mobile systems standardised by the 3GPP. It uses WCMDA as an air interface.

UTRAN UMTS Terrestrial Radio Access Network – It is a key interface for the mobile device to the UMTS core network.

WCMDA Wideband Code Division Multiple Access - It is a high-speed 3G wireless technology with the capacity to offer higher data speeds than CDMA. It can reach speeds of up to 2 Mbps for voice, video and data transmission.

WFQ Weighted Fair Queueing - smoothes out the flow of data in packet-switched networks by sorting packets to minimize the average latency and prevent exaggerated differences between the transmission efficiency afforded to narrowband versus broadband signals.

Chapter 1

Introduction

1.1 Background Information

In most cases, when people use the word "wireless", they almost always mean a portable wireless device such as cell phone and not necessarily the wireless network. In recent years, there has been a tremendous increase of mobile users in the mobile communication markets around the world. Over these years, the most popular service in the second generation (2G) mobile systems, such as global systems for mobile communication (GSM), has been real-time voice. Today the mobile users demand to access both voice and Internet through their cell phones. They want to surf the Internet, check emails, download files, make video-conferencing calls and perform a variety of other tasks. They expect to access these services whether they are at home, shopping malls, airports, the office, walking in town, or driving on a highway. However, currently the GSM systems and the Internet network have been considered as separate technologies due to the different types of traffic they are intended for. The need to access both voice and Internet through cell phones imposed a great demand on the transmission resources in GSM wireless systems, which pushed a need for a new mobile system.

In 1999, the Universal Mobile Telecommunications System (UMTS) was released and the International Telecommunication Unit (ITU) adopted it as a member of its family of IMT-2000 standards. UMTS, the third generation (3G) mobile system, will provide mobile users access to unlimited applications that will be available 'anywhere, anytime'. It is designed to allow users to access both voice and Internet applications through cell phones. It will also allow users to access the Internet through their laptops and personal data assistant (PDA) which is not supported by the GSM systems. These applications will require the provision of user data rates that are higher than those provided by today's GSM systems. The first deployment of UMTS will be built upon GSM systems [3]. GSM, deployed worldwide in the early 1990s, was designed primarily for mobile telephony with a limited data rate of only 9.6 kbps [1]. Despite this limit, GSM was successfully deployed in 118 countries with 320 operational networks by the end of 1998.

UMTS will offer higher data rates of up to 2 Mbps to transmit mobile traffic. Its architecture is illustrated in figure 1.1 and consists of the mobile devices, UMTS Terrestrial Radio Access Network (UTRAN), and the core network. The mobile device accesses the UTRAN through the Wideband Code Division Multiple Access (WCMDA). The WCDMA was specified as an appropriate air interface in Release 99 [4]. It provides mobile users with data rates of up to 144 kbps in macro cells, up to 384 kbps in micro cells and up to 2 Mbps in indoor or pico cells. The Base Station (referred to as the Node B) receives traffic frames from the air interface and forwards them to the Radio Network Controller (RNC). The RNC[2] is responsible for the radio resource management (RRM) and controls all the radio resources (e.g. bandwidth) within the UTRAN.

The first world deployment of UMTS took place in Japan in October 2001 and was based on WCDMA technology [2]. Japan has been at the forefront of research and development of 3G with a particular focus on WCDMA air interface. With UMTS launch, millions of Japanese mobile users had access to video telephony and Internet at faster speeds through their 3G mobiles. Today UMTS is being deployed worldwide and in the access networks, the Node Bs and RNCs are connected by point-to-point E1/T1 (2 Mbps) links as in figure 1.1 [7][14][36]. The links are expensive and hence they should be used efficiently. These links are Asynchronous Transfer Mode (ATM) links adopted by the 3GPP and currently used in the UMTS (Iub interface) deployed in Japan [2]. ATM has proved to guarantee QoS required by the mobile users. In a point-to-point link, bandwidth is dedicated between the Node B and RNC. Traffic from various mobiles is aggregated at the Node B and transported through this link to the RNC. If the Node B aggregates mobile channels with a maximum of 1.2 Mbps traffic, then the rest of the 2 Mbps bandwidth is wasted.

[2] The RNC is the terminology used in UMTS. In GSM, the node that operates like the RNC is the Base Station Controller (BSC).

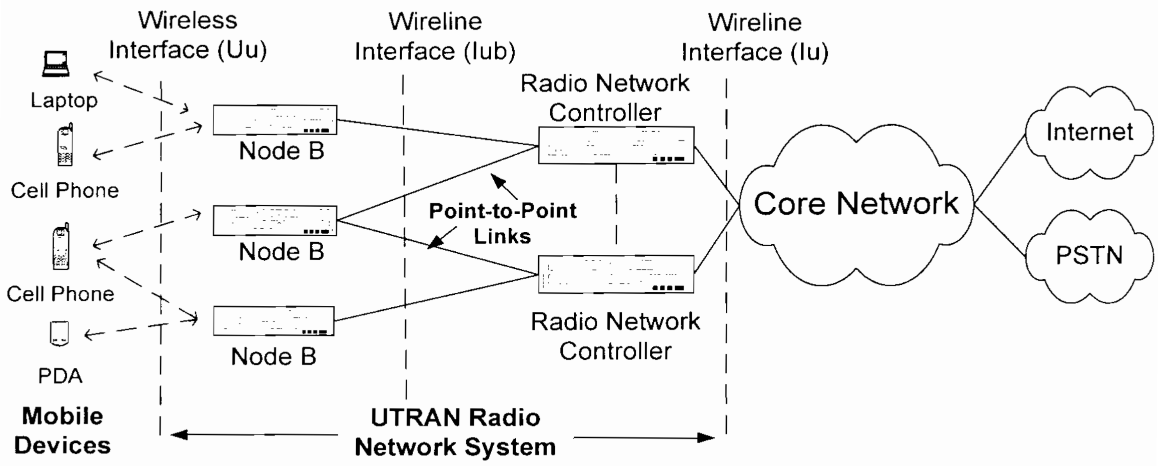


Figure 1-1: UMTS network architecture

The E1/T1 link can support up to X number of mobile users per period when fully utilised. If it happens that X+1 users request for channels, then another E1/T1 link will be needed to cater for the additional mobile user. The bandwidth between the Node B and RNC is now 4 Mbps. If there are no more users requesting for channels, then the rest of the bandwidth in the second link is wasted. This clearly shows that the point-to-point link has scalability problem, where scalability is defined shortly. Furthermore, if the RNC fails, then there will be a great loss of traffic from mobiles since the Node Bs may not be able to connect to other RNCs. For these reasons, the point-to-point link in the Iub interface is replaced by Internet Protocol (IP) network in order to reduce the wastage of bandwidth as shown in figure 1.2 [14][36]. IP has become the basis for packetisation of voice and data in networks at low cost and its benefits in the Iub[3] interface include:

- **Scalability**: Refers to how efficient bandwidth is utilised as more mobiles connect to the access network. Furthermore, how well a transport layer performs as traffic increases in a network. For instance, in a low loaded system, partial filled ATM cells are sent off since the cell payload is fixed and in a high loaded system, there is a high repetition of cell headers which shows lack of scalability and flexibility in ATM. The IP protocol showed to be scalable in high loaded system [14].

[3] The wireline interface between the Base Station and Radio Access Network as shown in figure 1.1

- **Reliability**: If a hundred Node Bs are connected through a network to three RNCs and one RNC fails, then the Node Bs can connect to another RNC through the IP network, which ensures reliability in the access network.

- **Flexibility**: The point-to-point link is not flexible since it dedicates bandwidth amongst each Node B and RNC. The IP-based Iub interface will introduce statistical multiplexing gains so that bandwidth can be shared amongst the Node Bs and RNCs. Multiplexing of traffic saves bandwidth on the links and bandwidth savings increase with increasing traffic variability. Furthermore, the ATM cell size is fixed which lacks flexibility compared to the variable IP packet size.

The Iub interface is the most critical one since it is the access network of the UMTS and bandwidth must be used efficiently. Before IP was specified in the Iub interface extensive research, which will be reviewed in chapter 2, had investigated the performance of ATM and showed that ATM guarantees QoS [8][13][23]. ATM was adopted with its adaptation layer, AAL2 [22], to compress, suppress silence and multiplex multiple voice channels into an ATM cell which ensures low packetisation delay of cells at the Node B. Since AAL2 removes silence periods in the traffic, bandwidth is used efficiently when transmitting low bit-rate traffic in delay sensitive applications. In fact the mobile traffic arrives at the Node B in the form of Frame Protocol (FP) frames from the WCDMA interface and adapted to AAL2. Alongside WCDMA, AAL2 is a good transport protocol of variable bit rate (VBR) traffic that the mobile users generate. AAL2 is currently used in the first UMTS deployed in Japan and is implemented particularly in the Iub interface.

Although ATM guarantees QoS, it does not scale well as the amount of traffic increases in a network [18][19]. Due to this, research compared ATM and IP in the Iub interface and found that IP uses bandwidth efficiently when traffic increases because of its flexible packet payload [7][13]. Due to IP becoming popular in networks, the 3GPP specified IP as an alternative transport in the Iub interface [15]. Extensive research investigated IP as the transport protocol and found it to be viable in terms of efficiency, scalability and flexibility [14][36]. IP uses Real Time Protocol (RTP) to carry real-time traffic such as

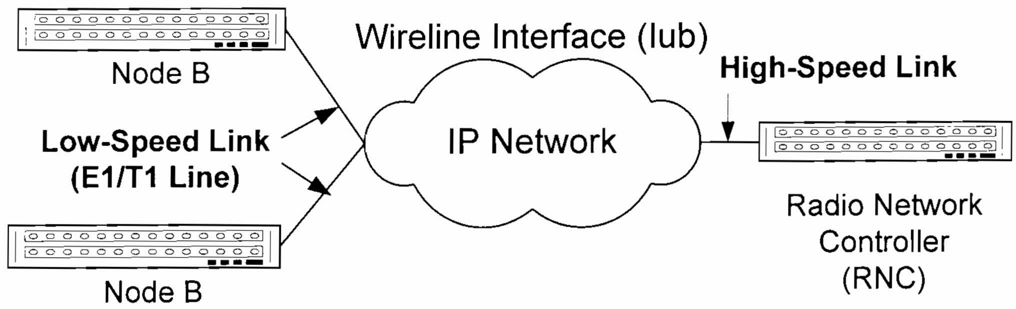


Figure 1-2: UTRAN Iub interface infrastructure

voice. RTP depends on User Data Protocol (UDP) for multiplexing channels. The specification of IP in the Iub interface meant another step forward towards all-IP based UMTS since the UMTS core network is already based on IP [10][16][17].

Due to the efficiency of AAL2 and the benefits of IP in transporting traffic, AAL2 was proposed to be inserted into IP instead of ATM in the Iub interface [7]. At the Node B, voice frames are adapted to AAL2 packets and inserted into IP packets and sent to the RNC through the IP-based Iub interface. By packing several AAL2 packets into IP packets, the bandwidth efficiency improves since all the AAL2 packets are to be delivered to the RNC. When investigating AAL2/IP[4], they were only concerned with using bandwidth efficiently and hence ignored the Iub interface QoS constraints such as delay. However, the air interface imposes rather stringent delay requirement in the Iub interface transport network for real-time traffic. For this reason, the 3GPP specifies 5 ms delay for voice in the Iub interface to guarantee QoS to the mobile users [12][14][46].

In IP networks, QoS is mostly provided with Differentiated Services (DiffServ) [11]. DiffServ is integrated with Multi-protocol label switching (MPLS) to optimise QoS in the IP-based Iub interface [35]. A brief discussion of this integration is provided in appendix C3. Furthermore, although fourth generation (4G) is not yet as well defined as 3G, it is almost clear that 4G will use IP as an end-to-end transport layer. The idea is to have a common transport layer for future "Internet mobile" network. 4G, termed all IP-based, is expected to provide massive benefits for both mobile users and network providers [21].

---

[4] AAL2/IP refers to AAL2 over IP in this study and similarly AAL2/ATM refers to AAL2 over ATM

1.2 Problem Description

ATM was adopted in the Iub interface since it guarantees QoS. It guarantees QoS such as delay requirements by using AAL2 to multiplex multiple channels into the small fixed cell size. Although it achieved the Iub interface delay requirement, it uses bandwidth inefficiently [7][13][19]. This is because when the number of mobile users is low, partially filled cells are sent through the Iub interface. As the number of users increases, there is a high repetition of cell headers as the cells do not take long to be full. ATM uses bandwidth inefficiently in both low and high loaded UTRAN. Due to this problem, AAL2 was encapsulated over IP which showed to improve the Iub interface bandwidth efficiency particularly on a high loaded UTRAN [7]. However, it was assumed that there are no QoS constraints in the Iub interface when showing that AAL2/IP improves bandwidth efficiency. This assumption is invalid in reality since voice is very sensitive to delay and the Iub interface has a 5 ms delay bound to guarantee voice quality.

Furthermore, IP still faces QoS challenges to meet the stringent transport requirements for voice. QoS constraints such as delay, jitter and packet loss are still hard to guarantee in IP. These constraints can degrade the quality of voice if they are ignored. When encapsulating AAL2 packets into IP packets, a problem arises when the first AAL2 packet inside an IP packet waits for other AAL2 packets to arrive. Within the Iub interface, IP packets may take different routes since IP is connectionless which causes them to get lost or arrive out of order at the RNC. The IP packets may incur delay variation when traversing the interface. To arrange the IP packets in order and to cancel jitter at the RNC, buffers will need to be inserted in the RNC. However, buffering packets at the RNC also incurs additional delay which makes it hard to guarantee voice quality.

1.3 Thesis Objectives

As outlined in section 1.1, the research conducted in the literature to evaluate the performance of AAL2/IP in the Iub interface only focused on efficiency and ignored the QoS constraints. However in reality, QoS constraints must not be ignored since they can degrade the quality of voice. In our study we aim to enhance AAL2/IP not only to use bandwidth efficiently but also meet the Iub interface QoS constraints in order to provide QoS to the mobile users. Our ultimate goals are to:

- Design the UTRAN Iub interface to transport AAL2 traffic from the Node B through the IP-based Iub interface to the RNC.
- Investigate a trade-off performance between bandwidth efficiency and meeting the Iub interface QoS constraints. Particularly, to find out how efficiently bandwidth will be used by AAL2/IP while the delay from the Node B to the RNC is introduced but kept less than 5 ms.
- Determine the number of users that can be supported by a 2 Mbps bandwidth link while meeting their QoS (delay, jitter and packet loss) constraints. In our study, the 2 Mbps link connects one Node B to the RNC through the Iub interface.
- Observe how the real traffic will co-operate on an experimental test-bed compared to the simulation results reviewed in chapter 2.
- Develop and implement an AAL2/IP framework (test-bed) for the Iub interface emulating the UTRAN to evaluate the above.

1.4 Scope and Limitations

This study is limited to developing an experimental test-bed to evaluate the performance of AAL2/IP in the Iub interface, an interface long occupied by AAL2/ATM. As AAL2/IP is still in its infant stages, no AAL2/IP architecture is available to our knowledge. Hence it is not reasonable to implement a fully functional AAL2/IP test-bed in a single project. For this reason, certain parts of the test-bed will be left out for future research. In this study, we will focus on the development of an uplink user plane for the test-bed, only from the Node B to the RNC and not vice-versa. We will not focus on the control plane functions such as the signalling and admission control on the test-bed. However, discussions with regard to the control plane for the AAL2/IP will be made. AAL2 was encapsulated directly over IP and also over UDP/IP [7]. But both AAL2 and UDP are transport layers with respect to their underlying network layers. A discussion as to whether AAL2 should be encapsulated directly over IP and over UDP/IP will be made.

Furthermore, we will not adapt the FP frames carrying mobiles' traffic to AAL2 packets and classify the packets at the Node B since these do not form the focus of this study. The mobile traffic will be generated in the form of AAL2 packets within the Node B of the test-bed and multiple mobiles' traffic will be generated simultaneously at the Node B. The Adaptive Multi-Rate (AMR) codec, adopted by the 3GPP [15] is used to generate mobile traffic and has the packet inter-arrival time of 20 ms. The AAL2 packets will contain randomly generated and prioritised data. These packets will be encapsulated into IP packets of up to 1500 bytes, the Ethernet maximum transfer unit (MTU) size. A timer will be used to keep record of time elapsed during the packetisation of the IP packets. Scheduling and multiplexing of the AAL2 packets will be done on the AAL2 layer. The benefits of using AAL2/IP rather than AAL2/ATM will be discussed.

AAL2/IP is developed to transport real-time voice traffic, hence it would be feasible to implement it on a real-time environment. However implementing AAL2/IP in an embedded real-time system would require extensive further research and draw the attention away from the primary aim of our study. For this reason we decided to use non real-time Linux operating system (OS) for implementation. The minimum timing resolution offered by Linux is 10 ms. This resolution does not affect the number of mobile users that may be supported in our design.

1.5 Thesis Outline

The remainder of the document is organised as follows:

- Chapter 2 begins with an overview of the characteristics of voice since it is an application considered in our study. It will discuss research performed on how voice is transported using AAL2/ATM, particularly in the Iub interface, to provide a background of our study. An overview of how AAL2 voice will be transported over IP is then discussed in detail. This is essential in understanding tunnelling of AAL2/IP as presented in chapter 3. A motivation of the proposal to tunnel AAL2/IP will be provided. A discussion on both AAL2 and UDP transport layers will be provided. Mechanisms to provide QoS in the Iub interface will be discussed as well. The AAL2/IP performance evaluation metrics in the Iub interface are also covered in this chapter.

- Once issues concerning AAL2/IP have been discussed, chapter 3 will focus on the system design of an AAL2/IP. The access network topology will be presented. A definition of how both signalling and data packets are treated as they enter the Node B will be presented. A procedure as to how voice frames from mobiles are adapted to AAL2 packets will be shown. The Iub interface requirements to support tunnelling of AAL2/IP will be taken into consideration. A full design system will be presented, showing how the overall tunnelling system may be broken into a series of modules that operate independently from one another, and how each module is responsible for its own function. Finally, the operations of the modules will be discussed in this chapter.

- Chapter 4 discusses the hardware and software architecture used to implement the UTRAN test-bed which is used to evaluate the performance of AAL2/IP. The operating system (OS) chosen on the test-bed will be justified. The test-bed components will be discussed. The test-bed will consist of the traffic sources to generate AAL2 packets in the Node B, a dummynet router to emulate the IP-based Iub interface and a traffic receiver in the RNC to receive the AAL2 packets. A series of tests will be conducted on tunnelling of AAL2 packets into variable IP packets. These tests will be aimed at finding the trade-off between performance in terms of delay and bandwidth efficiency. It will be shown how various performance parameters are linked to one another.

- In chapter 5, the results of the tests performed to evaluate the AAL2/IP test-bed will be presented and analysed. The performance metrics to be considered are link efficiency, number of users, CPS and IP packet sizes and packetisation delay of IP packets in the Node B and the Iub interface delay, jitter and packet loss.

- Chapter 6 will present a set of conclusions that were drawn from the research findings and results obtained in our study. This chapter will also provide recommendations for future work on the AAL2/IP test-bed.

- Finally, a set of appendices will follow chapter 6. These appendices will provide additional background information as well as information on certain aspects of the implementation of the AAL2/IP test-bed.

Chapter 2

Background Theory and Literature Review

2.1 Introduction

This chapter will present information about the AAL2/IP transport model in the Iub interface. Note that the Iub interface is also referred to as the UTRAN in this study. In this model, traffic from mobiles will be multiplexed using AAL2 at the Node B and transported through an IP-based Iub interface to the RNC. An important issue is to develop a model that will use bandwidth efficiently while meeting the Iub interface constraints and guaranteeing QoS to the mobile users. This chapter will summarise the issues concerning the performance of AAL2/ATM which is the current transport method in the Iub interface. Then the research work on AAL2/IP which is in its infant stages in the literature will be reviewed to determine which areas of AAL2/IP still need attention.

When investigating AAL2/IP, it is very important to understand the applications that the mobiles will generate. In this study we will primarily address low bit-rate voice and the QoS constraints imposed in the Iub interface. This is to ensure that the Iub interface is dimensioned to use bandwidth efficiently while meeting the QoS, such as end-to-end delay[5], required by voice. Voice is chosen since it is expected to still dominate the UMTS as it dominates today's GSM networks. To understand voice characteristics and the Iub interface to meet QoS for voice requirements, an overview is given in the following section. A motivation to transport AAL2/IP will be provided as well. Furthermore, mechanisms to provide QoS in the Iub based IP interface will be discussed.

[5] End-to-end delay refers to the delay from the Node B to the RNC in this study

2.2 Characteristics of Voice

Voice is a real-time application sensitive to factors such as delays, jitter and packet loss. It is particularly very sensitive to end-to-end delay incurred when traversing the network. Much research has been conducted on transmitting voice and some researchers showed that mean delay is not a significant QoS factor for end-to-end voice transport [14][18] [19]. For this reason, it has been suggested that one should rather examine the tail of the voice delay distribution and quantify the overall fixed delay at the Kth percentile. At the destination, when analysing the effects suffered by voice in networks, what is often more important is the ratio of the packets that successfully arrived over the total number of packets sent. Since delay, jitter and packet losses parameters can really degrade voice quality, it is very important to ensure that these parameters do not exceed the limit specified in networks to provide voice quality. For instance, the 3GPP specifies 5 ms delay in the Iub interface when IP packets carrying voice are transmitted from the Node B to the RNC. To get a good performance, one will need to vary the packetisation delay, the IP packet size and the Iub interface delay until the trade-off required is met.

As the IP packets traverse the Iub interface, they may incur jitter (delay variation) and to cancel this jitter, buffers are required at the receiver. These buffers are used to ensure clock-recovery and to smooth voice play-out at the application level. If jitter is high, then the IP packets will be buffered long enough to cancel it, which once again introduces an additional delay at the receiver. The rate at which the data is played out at the receiver will depend on the voice codec chosen to decode the data. To determine voice quality at the receiver, a subjective metric known as Mean Opinion Score (MOS), has been developed [19]. This metric measures the voice quality at the receiver. A variety of subjects are specified to rate the quality of voice in a session and give a score from 1 to 5, where 1 is considered to be unacceptable quality and 5 to be excellent quality. A lower score indicates that more emphasis is required by the subject to interpret what is being said. A MOS of 4 is considered as a toll-quality voice. The MOS scale is usually used to indicate the quality of voice sustained in the network.

When compressed voice is introduced in a network, the rate at which voice enters the network now varies and is referred to as variable bit rate (VBR) traffic as compared to uncompressed constant bit-rate (CBR) voice traffic. The VBR compression schemes are designed to compress voice and remove silence periods in voice streams. This means that during a silence period, there are no data to transmit and hence the voice source will remain idle during that period. During this period, the network can make use of statistical multiplexing gain to transmit voice traffic from other mobiles. However when a mobile that has been using the channel suddenly transmits traffic, the network will be expected to respond much faster and have enough bandwidth to handle this new talk-spurt. Furthermore, for compressed voice the network should be more intolerant of packet losses. In uncompressed voice, the loss of a few bits of data will have a minimum effect on the perceived quality at the destination. However, if a compression scheme that compresses voice streams by a magnitude of four is used, then losing one bit of data will correspond to losing four times the amount of uncompressed data at the receiver. This implies that the higher the compression ratio being achieved, the higher the network should be able to prioritise voice channels and tolerate very low packet losses.

When a mobile device is transmitting voice in UMTS, on the air interface a dedicated channel (DCH) stream is allocated between that mobile and the Node B by the RNC. The voice traffic is received from this mobile at the Node B in the form of Frame Protocol (FP) frames every transmission time interval (TTI) period. The FP frames are adapted to AAL2 Common Part Sublayer (CPS) packets, multiplexed into IP packets and sent to the RNC through the Iub interface. Voice traffic generated by the mobile user consists of a succession of ON and OFF periods. In our study, voice will be generated by the Adaptive Multi-Rate (AMR) codec every 20 ms at 12.2 kbps recommended by the 3GPP for the UTRAN and used by many researchers [8][15]. When a mobile is ON, the codec of that mobile will generate a 31-byte voice frame every TTI, this frame is encapsulated into a FP frame by adding a 5-byte header. Therefore, in an ON period, the Node B receives a 36-byte frame from the corresponding DCH stream. During the OFF period, no frames are received. The transfer delays for the IP packets through the Iub interface should be kept as minimum as possible, for both real-time and non-real time traffic. In the Iub interface, the 3GPP has specified the delay bound of 5 ms for real-time traffic and a delay bound of 10 ms for data traffic such as file transfer and web browsing [12][14].

2.3 AAL2 Tunnelling in the Iub Interface

This section reviews the performance of AAL2 to transport voice in the Iub interface. To transmit voice over ATM networks, AAL1 was used as a suitable method to emulate circuit-switched services. AAL1 packets are placed into ATM cells and when full, the cell is sent off. Each cell is only filled with packets from one source. This incurs high delays when packetising low bit-rate, compressed voice traffic into cells since the cell is filled with packets from one source. This is due to the fact that compressed traffic is bursty; hence time periods exist where no traffic is generated since voice is modeled as ON/OFF periods. Due to this, AAL1 is not considered an optimum solution, in terms of bandwidth efficiency, to transport voice since it does not multiplex, compress, suppress silence and has high packetisation delay for low bit-rate voice [18][19].

To satisfy the need for voice in ATM, AAL2 was designed to compress, suppress silence and multiplex multiple voice channels into a single ATM cell. This implies that the time taken to fill up ATM cells would become much shorter. The bandwidth would be used efficiently when transmitting low bit-rate, short and variable-length packets in delay sensitive applications since silence is suppressed and multiple channels share the transmission medium. Since multiple channels are multiplexed into an ATM cell, further information is required to identify these channels from each other on an end-to-end basis. For this purpose, each channel is identified by a separate channel identifier (CID), which is stamped onto the header of each packet carrying traffic for that channel. The principle of carrying multiple AAL2 voice channels over a single ATM cell is shown in figure 2.1.

Many studies have investigated the performance of AAL2 in ATM networks. AAL2 performance in terms of bandwidth efficiency was compared with AAL1 and AAL5 through simulations [18]. Their aim was to find the number of voice channels that may be supported on a single AAL2 path, while guaranteeing each channel's QoS. They showed

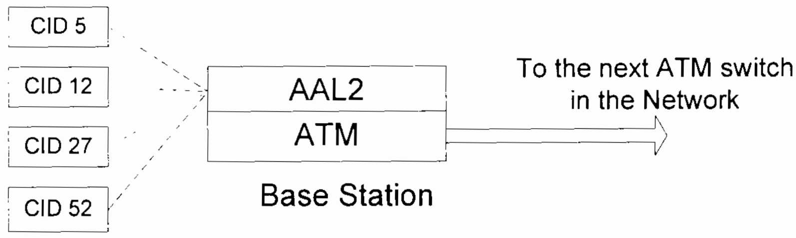


Figure 2-1: Multiplexing Users onto an ATM Link using AAL2

that for a low bit-rate voice of 8 kbps, AAL2 is about five times more efficient than both AAL1 and AAL5. It is for this reason that AAL2 was chosen in our study to transport voice traffic. The AAL2 performance was very dependent on the choice of the packet size. Their results showed that the efficiency increases with the CPS payload size until 44 bytes is reached. Increasing it to 45 byte resulted in a second ATM cell needed to transport the 45th byte. This cell would carry only 1 byte of data, the rest of the cell would be padded which results in bandwidth being used inefficiently by the ATM layer [7][18]. In addition, as the mobile users increase, there is a high repetition of ATM cell headers since cells do not take long to be full which also results in bandwidth being used inefficiently [13][20].

AAL2/ATM was adopted in the Iub interface [4] and extensive research was conducted on its performance in terms of bandwidth utilisation while meeting delays required by the mobile users [6][8][23]. Most of this research work was conducted by analysis and simulations. This research focused on multiplexing several voice channels using AAL2/ATM through the Iub interface to the RNC while meeting delay of each channel. An analytical model to multiplex voice onto AAL2/ATM was proposed and investigated the sojourn[6] time in the Node B in figure 2.2 [8]. This time comprised of the time to multiplex CPS packets into ATM cells and the time the cells spent in the transmission queue. This is to guarantee the delay specified between the Node B and the RNC. They developed expressions to evaluate this time and showed that this time is dependent on both the number of simultaneous active voice channels and cells transmitted per TTI.

[6] Sojourn time refers to the time packets spend between Node B and RNC.

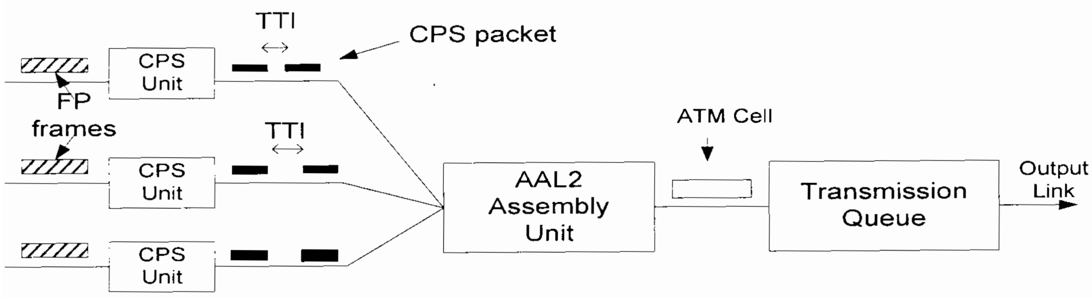


Figure 2-2: AAL2 Multiplexer at the Node B

Their results showed that when the number of active channels is large, the mean sojourn time does not exceed 5 ms, which is the delay bound specified from the Node B to the RNC. In other research, an analysis to evaluate and optimise the performance of the multiplexer in figure 2.2 for voice channels is provided [9]. Their focus was to determine the number of channels that can fit into a fixed capacity link for various assembly unit timeouts and queue sizes. The results showed that an increase in transmission queue increases the statistical multiplexing gain and decreases the cell loss. They concluded that the packetisation delay value has a significant impact on the link utilisation.

AAL2/ATM was adopted to transport aggregated mobile traffic in the Iub interface since it guarantees QoS such as the end-to-end delay. For instance, at the Node B traffic is multiplexed into cells based on the multiplexer time-out to packetise the cell. However, when there are more than 47 active mobile channels, the cells are sent out frequently without depending on the time-out [19]. This results in high repetition of ATM cell headers in the Iub interface which wastes bandwidth. As mentioned above, many researchers used the AMR codec to generate voice traffic which sends a 36 byte frame to the Node B when the mobile is active. If two frames are received at the Node B from two mobiles, then two cells would be needed to carry these two frames since traffic from the second frame would not fit into the cell of 48 byte payload and the second cell may be padded as discussed above. One may conclude that although ATM guarantees the Iub interface QoS constraints, it is clear that it uses bandwidth inefficiently as the mobile channels increase in UMTS. It is for this reason, and the ATM drawbacks discussed in sections 1.1 and 1.2, that the 3GPP specified IP as an alternative transport to ATM in the

Iub interface. The Mobile Wireless Internet Forum (MWIF) concluded in its technical report that IP is a viable option to transport real-time traffic in the Iub interface [5]. Due to this, extensive research investigated ways to deploy IP while providing mobile channels' QoS constraints in the Iub interface [10][11][14].

Other papers compared the performance of ATM and IP through analysis to find a viable transport option for voice in the Iub interface, where delay bounds were to be met to transport both real-time and non real-time traffic [13]. The analytical results showed that ATM performs better than IP in terms of delay for low loaded UTRAN. They also showed that IP is a viable option because it reaches a higher performance in terms of both delay and link utilisation, for high loaded UTRAN. They used RTP protocol which was specified with IP to carry voice in the Iub interface. However, RTP uses bandwidth inefficiently because of the overhead it introduces when transporting traffic. For this reason, in further research they developed a 3-byte header called Multiplexed Header (MH) which has a 1-byte termed User Identifier (UID) unique to each user to multiplex FP frames [14]. This header is equivalent to the AAL2 header of 3 bytes which includes 1-byte of Channel Identifier (CID) unique to each user within an AAL2 path or tunnel.

The researchers multiplexed FP frames and inserted them into an IP packet using the approach shown in figure 2.2 [14]. Their design receives FP frames, appends the MH header instead of AAL2 header and multiplexes these frames into IP packets. Their multiplexer has an IP assembly, instead of AAL2 assembly as in figure 2.2, to multiplex FP frames directly into IP packets and the transmission queue. Their aim was to evaluate the delay of FP frames at the multiplexer. The delay in this multiplexer is incurred in the assembly unit when filling the IP packets and transmission queue when queuing outgoing IP packets to the Iub interface. Their analytical results showed that a time-out of 2 to 3 ms and an IP packet of 390-byte payload would provide a good trade-off performance between delay and link utilisation. They showed that a time-out of 2 ms and a 312-byte payload would provide a good trade-off if a direct link exists between the Node B and the RNC. However, if the Node B and the RNC are connected through a multi-node IP network, they recommended a time-out value less than 2 ms and a 234-byte payload.

Due to the overhead incurred by RTP/UDP to multiplex voice channels and the inefficient bandwidth utilisation in the ATM layer, AAL2/IP was proposed in the Iub interface [7]. This is due to the efficiency of AAL2 to multiplex multiple voice channels and the benefits introduced by IP in networks (as discussed in chapter 1). They proposed to adapt the incoming FP frames to AAL2 packets, as in figure 2.3 (b), by appending an AAL2 header on the frames and multiplex these packets into IP packets at the Node B. This is a similar approach used in the research reviewed above except that MH header was appended on the FP frames. They only focused on the AAL2/IP model to use the Iub interface bandwidth efficiently. However for the IP protocol to be successful in UMTS, it should use bandwidth efficiently while meeting the Iub interface QoS constraints. Furthermore, they transported AAL2 over UDP/IP. But both AAL2 and UDP are transport layers with respect to their underlying layers and only one transport protocol is needed as illustrated in figure 2.3 (a). These reasons motivated our research in this field.

The research reviewed above is valuable to us in enhancing and evaluating the effectiveness of AAL2/IP on a test-bed to optimise QoS in the Iub interface. AAL2/IP showed to improve bandwidth efficiency when multiple AAL2 packets are inserted into IP packets at the Node B and delivered to the RNC. It was showed through simulations that AAL2/IP improves the link by 10 % compared to AAL2/ATM [7]. Since the Iub interface transports aggregated traffic from various mobiles with different QoS, it is very important that IP uses bandwidth efficiently while meeting delay as stated above. In this study we are interested in how AAL2/IP implemented in a test-bed will respond to the trade-off performance between bandwidth utilisation and delay obtained in the simulations reviewed above. Particularly, how efficiently bandwidth will be used by AAL2/IP when the delay bound of 5 ms for voice is introduced in the Iub interface. Furthermore, whether the number of mobile users and the amount of traffic sustained in the AAL2/IP simulations above can be validated by means of the test-bed. Traffic was multiplexed into IP packet payload[7] while keeping track of the packetisation delay.

[7] This study refers to the IP packet payload as the protocol data unit (PDU)

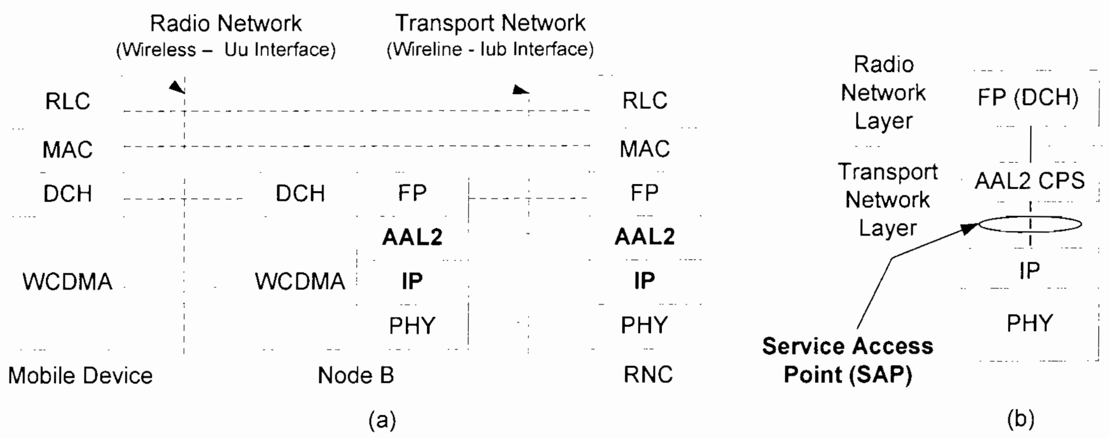


Figure 2-3: The access network protocol stack based IP transport

A Protocol Data Unit (PDU) is a variable structure that holds multiplexed AAL2 Common Part Sublayer (CPS) packets. Each PDU with multiplexed packets is passed to the IP layer as illustrated in figure 2.3 (b) and the IP header is appended to the front of this PDU before transmitted to the RNC. The AAL2 layer will pass the multiplexed CPS packets to the IP layer through a service access point (SAP). It will pass these packets as a service data unit (SDU) to the IP. To multiplex CPS packets into a PDU, the use of a state-machine based CPS transmitter is recommended [22]. On the receiver, a mechanism is required to extract individual CPS packets out of the PDU. For this purpose, a state-driven CPS receiver is recommended as well. These two state-machines, i.e. the transmitter and receiver, have already been successfully implemented on the AAL2 layer on the Linux operating system [19]. The detailed operation of these state-machines can be found in the literature [22], however a brief overview of each is provided in appendix B.

2.4 QoS in the Iub Interface based on IP

Although IP is specified in the Iub interface, it still faces QoS challenges in order to meet the stringent transport requirements of the Iub interface. QoS may be defined as, from the user's point of view, the assurance of end-to-end service perceived on the application level such as delay, jitter, packet loss and delivery order, for applications like voice and video. From the service provider's point of view, it is the efficient usage of bandwidth while guaranteeing the above QoS factors to the mobile user. Furthermore QoS involves treating packets differently according to each mobile user's specification. The packet differentiation is very important since the Iub interface transports aggregated voice and data traffic. The ATM guarantees QoS through connection set-up prior to traffic transfer. For IP to succeed in UMTS, voice quality needs to be close to the one provided by today's PSTN service [25][28]. Therefore, it is very important to enhance AAL2/IP to guarantee QoS. Currently the Internet is dominated by delay insensitive traffic such as email; however it has become the basis to transport delay sensitive traffic such as voice.

IP is a connectionless, best-effort service which implies that packets may be lost in the network or arrive out of order at the destination. This makes it difficult to guarantee QoS in IP networks. However, with the demand to use IP to carry delay sensitive traffic, IP is being investigated to guarantee QoS. As the Internet gained in popularity, there was growing concern over its ability to scale with a large increase in user traffic. This is because in IP, bandwidth is shared amongst users as discussed above and possibilities may arise where a large amount of traffic could be imposed to the network than the available resources can handle. These possibilities may arise since IP lacks the resource management schemes, such as admission control, to limit the amount of traffic. Due to the lack of resource management schemes, the scalability concern could be real. However to prevent this problem, mechanisms such as DiffServ [32] and MPLS [34] were developed by the IETF to manage traffic and provide QoS in IP networks. A brief discussion of DiffServ and MPLS is provided in appendices D1 and D2 respectively.

The DiffServ marks and classifies packets into small number of aggregated flows[8] with a specific forwarding treatment or Per Hop Behaviour (PHB). The classifying and marking functions are only performed at the edge routers of the DiffServ domain while the routers inside the domain perform classification of packets based on PHB, which provides scalability in the domain. It puts a reduced set of rules (PHB) in the edge routers to mark the incoming packets about their PHB inside the domain. DiffServ provides unacceptable QoS because it specifies only PHBs and does not have resource management scheme

[8] A series of packets grouped together that belong to the same class of service

which has been investigated in [11][16][26]. In DiffServ, high priority can be provided to voice by means of mapping voice to the Expedited Forwarding PHB (EP PHB) [33]. Furthermore, MPLS was developed to improve the forwarding speed of the routers by using simple labels and is now emerging as a crucial underlying technology for IP transport to offer QoS requirements for large-scale IP networks.
In UMTS, the 3GPP specified DiffServ to manage traffic since it is scalable [15] and research has been conducted in the Iub interface based DiffServ. To meet the Iub interface QoS constraints, resource management in DiffServ (RMD) was proposed [11] [26]. RMD scheme probes the DiffServ domain to check if resources such as bandwidth are available. It checks this by measuring the aggregate throughput at the edge router of the domain in order to find the actual bandwidth usage on the outgoing link from the Node B to the Iub interface. Resources can be reserved in RMD using either the admission control (AC) or measurement based admission control (MBAC) algorithm [31]. MBAC helps to achieve high bandwidth utilisation in real-time traffic and provides acceptable QoS. MBAC has been investigated for RMD to achieve better link performance [11][26]. With MBAC, the traffic load in the Iub interface can be estimated, since the interface edge router can mark the incoming packets with the resource state information. This load estimation can allow one to admit or reject new channels.

Other research has investigated the admission control and resource reservation schemes in the IP-based Iub interface [16]. These schemes are required at the edge routers to control the amount of traffic entering the network, control delay and limit packet loss. As the DiffServ gained popularity, EF PHB has been recommended to handle voice [28]. Specific to the Iub interface is the stringent delay bound for the transport of aggregated applications with various QoS requirements imposed by the WCDMA radio control functions on the wireless interface. As mentioned above, voice and video require a delay less than 5 ms in the Iub interface to guarantee QoS to the mobile users. File transfer and web browsing has a delay limit of 10 ms. It is important to minimise the delays experienced by the voice packets in the Iub interface in a cost-effective way in terms of maximal utilisation of bandwidth limited to 2 Mbps (E1/T1) link.

In the IP networks a number of factors are taken into consideration when specifying QoS requirements. These factors may degrade the voice quality if not considered carefully. As a result, it is crucial to understand these factors, how they impact on the interactive voice traffic, as well as obtaining the tools to measure and optimise them. The most common factors are described as follows:

- **IP Packet size**: PDU to carry digitised voice from the mobile device. The PDU choice can affect voice quality in terms of losses and delay. If a bigger PDU is dropped on the network, then voice quality will be highly degraded. A large PDU (IP payload) uses the bandwidth efficiently but it increases delay to the packetisation process, as the sender needs to wait more to fill up the PDU. A smaller PDU requires higher bandwidth per channel bandwidth because the IP header remains the same. In fact, research discovered that the packet loss rate is relatively low when the PDU is small [30]. The PDU choice is a compromise between the bandwidth efficiency and the voice quality requirements.

- **Packet loss:** Number of packets lost in the Iub interface during transmission. Since voice is transmitted over IP, it is important to consider the packet loss. The loss does not severely affect voice quality if it is less than 5 % of the total packets transmitted and the packet sizes are small [27]. However, the packet loss makes the conversation difficult when the loss is grouped together in large packet bursts due to a consistently congested network. In this case, even the best codecs are unable to hide the packet loss from the mobile user.

- **Packetisation delay**: Time taken to packetise voice packets into PDUs. It has a significant impact on the link bandwidth efficiency. An appropriate choice of its value depends on the number of channels to be multiplexed. It was concluded that the voice packet size choice is very sensitive to this delay [19].

- **Queuing delay**: Time spent by the IP packets in the storage of the transmission queue of figure 2.2 due to the output link being busy processing other packets, referred to as head of line blocking.

- **Jitter**: Delay variation experienced by the IP packets when transversing the IP-based Iub interface. The packets transmitted at equal intervals from the Node B may arrive at irregular intervals at the RNC because of delays not being guaranteed and undetermined route used. A buffer for incoming packets at the destination is needed to compensate for jitter. This buffer holds packets for a specified amount of time before forwarding them to be decompressed. However, this buffer may also incur additional delay. Late packets are discarded.

Voice is classified as a real-time service because it cannot tolerate high delays and packet losses. A time-out was associated with the multiplexer when multiplexing FP frames into IP packet to avoid packetisation delay (as discussed in section 2.3). They achieved a trade-off between time-out and utilisation for a 2 Mbps link. Other research work developed a method to configure delay, jitter and packet loss for an end-to-end (mobile-to-mobile) UMTS system through analysis and simulations [25]. In this paper, a jitter buffer at the receiver was investigated and configured in such a way that no frames were discarded due to late arrival or buffer overflow. Buffering of frames plays a very important role in the overall voice quality since it ensures constant data flow (smooth) play-out of voice on the application level. But the buffer may incur additional delay (large size) to voice and drop packets (small size) if it is not configured carefully.

The research reviewed above is also valuable to enhance AAL2/IP to provide QoS since QoS was ignored in the literature [7]. This is because they were interested in investigating a model to use bandwidth efficiently. In reality, the QoS constraints for voice must not be ignored. We will investigate ways to use bandwidth efficiently while meeting delay, jitter and packet loss constraints in the Iub interface as it was emphasised earlier that these parameters can degrade voice quality. In the IP networks, the larger the number of hops a packet traverses will affect the end-to-end delay. Furthermore, the non-deterministic route that packets follow in the network may lose the IP packets or cause them to arrive out of order at the destination, which makes it hard to guarantee QoS for voice. If IP packets arrive out of order at the destination, an additional delay is incurred since these packets have to be buffered and arranged in order.

2.5 Transport Layers: AAL2 vs UDP

When ATM was specified in the UTRAN, AAL2 was adopted to transport real-time traffic, mainly voice. As reviewed above, AAL2 showed to be a good candidate to transport UMTS traffic. An AAL2 tunnel supports a maximum of 248 channels. In the literature, AAL2 is proposed to be placed over UDP/IP in the Iub interface if more than 248 channels are required [7]. The protocol stack becomes AAL2/UDP/IP which consumes a lot of bandwidth. But both AAL2 and UDP are transport layers with respect to their underlying network layers and only one transport layer is needed. As discussed in section 2.4, RTP/UDP is used to multiplex channels with an overhead of 20-byte. On the other hand AAL2 multiplexes several channels into a single cell using a low header of a 3-byte header compared to 20-byte overhead. The AAL2 packets for each ATM connection are uniquely identified by the Channel Identifier (CID – 8 bits) field. The CID identifies the AAL2 channel which is bidirectional. The CID values from 0 to 7 have been reserved for future use and channel 8 is reserved for signalling purposes [22].

If more than 248 channels are required, then another AAL2 tunnel will be used instead of UDP as proposed in the literature. Each tunnel will be allocated its own IP connection in the outgoing link from the Node B connecting to the Iub interface. In the UDP header, the port number uniquely identifies the RTP session user. The RTP header contains the payload field, which identifies the encoding type data used to carry a session. While in AAL2, Length Indicator (LI) defines the payload and the CID uniquely identifies the channel. The RTP header contains the timestamp field which tells the RTP packet time. In AAL2, sequence number interval and UUI range indicate the AAL2 packet time. For a full description of the AAL2 and UDP headers, refer to appendix A.

Chapter 3

Design for tunnelling AAL2/IP in the Iub Interface

3.1 Introduction

In the preceding chapters we presented information regarding efficient transport for voice traffic from multiple mobiles while meeting their delay constraints in the Iub interface. The voice frames from these mobiles are adapted to AAL2 Common Part Sublayer (CPS) packets and multiplexed into IP packets at the Node B. Multiplexing multiple CPS packets into IP packets at the Node B increases the link efficiency since all the CPS packets are transported to the RNC. This chapter focuses on the design of an AAL2 tunnelling over IP model for the UTRAN (Iub interface) with emphasis on the user and control planes uplink[9]. There is extensive research on the performance of AAL2/ATM and IP in the Iub interface. However none of this research focused on the design of AAL2/IP for the Iub interface, which is the primary focus of our study.

This chapter will first discuss the UMTS Terrestrial Radio Access Network (UTRAN) topology. It will then discuss the functional requirements of this network and define how the CPS packets which belong to one AAL2 tunnel are treated as they enter the Node B. Then the design for both the Node B and the RNC will follow. The most important design considerations are:

- The signalling packets, to set-up channels at the Node B or tunnels at the RNC, should always be granted higher priority than user data during processing.

[9] The uplink refers to sending mobile traffic from the Node B to the RNC and not vice versa.

- The Node B should admit or reject a new mobile channel based on measuring the current utilisation on the outgoing link from the Node B to the Iub interface. It should be able to adapt Frame Protocol (FP) frames from the WCDMA dedicated channel (DCH) traffic streams to CPS packets.

- Each AAL2 tunnel supports up to 248 mobile channels. Multiple tunnels should be supported in case more than 248 channels are requested. Each tunnel should be allocated its own outgoing IP connection at the Node B.

- The Node B should have separate input queues for each channel as mobiles may have different QoS requirements. Multiplexing of multiple channels into PDUs should be timed to avoid prolonging packetisation delay at the Node B.

- The Node B should support buffering of the IP packets in case the outgoing link to the Iub interface is still busy with other IP packets. The Iub interface should support MPLS paths for fast switching and for avoiding to have IP packets arriving out of order to the RNC which provides QoS for voice.

3.2 Motivation for AAL2/IP in the Iub Interface

At this stage, the reason for proposing voice over AAL2/IP in the Iub interface may still need to be clarified to the reader. The designers of ATM had envisioned it to be a ubiquitous transport protocol that all existing and future protocols will utilise. Later, it also became necessary to transport IP over ATM and protocols were therefore established for this purpose. The inverse, carrying ATM over IP, was not considered. However, over the decade that ATM has been in existence, it has not achieved the global penetration that its designers had envisioned it for. Within the Iub interface, there have been serious considerations to replace ATM with IP. This is due to the fact that IP has recently become very popular in communication networks worldwide since IP, unlike ATM, has succeeded in dominating the end-to-end connections. The main advantage of IP compared to ATM is that it resides in networks such as the Internet and in mobile devices such as cell phones, laptops and PDAs, which makes it a truly end-to-end protocol. IP is a network protocol based on packet-switching used widely on the Internet.

The universal presence of IP in mobiles has motivated development for voice over IP (VoIP). For this reason, the Internet has become the basis for carrying voice at low cost [5]. IP integrates both voice and data in the same transmission medium which allows bandwidth to be shared in the network and fills up the channels more efficiently than in the traditional PSTN based circuit-switched network. It allows for a telephone call where voice is transmitted in digital form in discrete packets over IP. In PSTN, bandwidth is allocated statically[10]. IP allows bandwidth to be shared amongst users and not dedicated to one user. This means that the user only uses bandwidth when there is voice to send. This eliminates the telephone call time-based charging scheme associated with PSTN.

VoIP has been used in some cases to replace the traditional long-distance telephone systems to reduce costs to the end user [25]. However, a major challenge faced by VoIP is to guarantee voice quality. Perceived voice quality factors include delay, jitter and packet loss, which are discussed in the following section. Although IP still does not guarantee QoS, the 3GPP and MWIF have specified that IP is a viable transport in the Iub interface. This is due to its flexibility and scalability (as discussed in chapter 1), its bandwidth savings as the traffic increases in the interface and its universal presence in both mobile and network equipment.

In IP, voice is encapsulated into RTP, which supports end-to-end real-time traffic. The 3GPP specified IP together with RTP to carry voice [15] and research has investigated RTP as the voice carrier in the Iub interface. RTP depends on UDP for multiplexing which incurs a large 40-byte RTP/UDP/IP overhead to transport a packet. This overhead results in poor bandwidth utilisation in networks. RTP is not a good voice carrier in the Iub interface because of its 20-byte overhead, RTP/UDP, for multiplexing [25][27]. The 40-byte overhead may be compressed to 4-6 bytes using schemes like Robust Header Compression, Compressed RTP, and others defined in RFCs or Internet drafts. However

[10] The user is given bandwidth even when the user is not talking and this wastes bandwidth in the network

multiplexing schemes are still in an infant stage in RTP/UDP and are discussed in IETF Internet draft [29].

Furthermore, compression and silence suppression schemes in RTP are not as popular as AAL2. AAL2 has matured techniques to compress, suppress silence, multiplex and remove idle channels using its 3-byte header and a variable size payload of up 64 bytes. These reasons, the efficiency of AAL2 to transport multiplexed voice channels and the bandwidth savings introduced by IP, motivated the research towards AAL2/IP where AAL2 packets are multiplexed into IP packet at the Node B and extracted at the RNC. By packing multiple AAL2 packets into an IP packet at the Node B and sending them over the IP-based Iub interface, the total efficiency can be increased since all the AAL2 packets are transmitted to the RNC.

3.3 Performance Evaluation Metrics

The following metrics will be used to evaluate the performance of the AAL2/IP framework under various conditions and compared to the metrics used in the research reviewed above. These metrics are defined as follows:

- **Number of mobile (AAL2) users**: For a given 2 Mbps Iub interface link, the number of users that can be supported by AAL2/IP will be investigated. For each user, a separate CID is allocated to distinguish the users from each other in the multiplexing process. Since an AAL2 tunnel can support a maximum number of 248 channels, if more than 248 channels are required, then another tunnel will be used. It has been shown in the literature that the number of users that can be sustained by the tunnel can be very sensitive to the maximum allowed packet size. A maximum of 230 users were supported on a 2 Mbps link in the Iub interface when delays were ignored [7] and 120 users were supported when delays were considered [14]. Furthermore, 125 users were supported on a 1.536 Mbps link when the delays were considered [45]. Our interest is to determine the maximum number of users that can be supported by the 2 Mbps link while meeting the 5 ms Node B-to-RNC delay limit.

- **Link efficiency**: A measure of how well a given link bandwidth is utilised, in terms of PDU payload compared to total IP packet or ATM cell size. AAL2 was used to multiplex the CPS packets into PDUs. The multiplexing gain of AAL2 increases with the number of mobile users supported, as well as the CPS packet size. The efficiency of the link depends on a number of factors, namely: the PDU size, CPS packet-size, time-CU value and the number of mobile users on the tunnel. In the simulations carried out in the literature, the efficiency of AAL2/IP was evaluated where the timer-CU and Iub interface delay were ignored [7]. Here we determine the maximum link efficiency while ensuring the 5 ms delay limit.

- **Timer-CU**: Various values for the timer-CU to packetise variable PDUs (234 to 390-byte) range from 2 to 3 ms in the research reviewed above. When ATM was used, it was found that for 47 users or less, the timer-CU was critical [19]. The sole function of the timer-CU is to ensure that the packetisation delay of voice in the Node B is met. It maintains an upper threshold when packetising the PDUs. The timer-CU impacts both the link efficiency, in terms of number of CPS packets inserted into each PDU, and the 5 ms delay limit. We use the timer-CU to optimise a trade-off between bandwidth efficiency and the 5ms delay limit.

- **CPS packet size**: The AAL2 packets size that are inserted into the PDUs. It has been shown that AAL2 performance, in terms of efficiency and the number of users to support, is very much dependent on the choice of CPS packet size [18][20]. Low CPS packet sizes such as 20 bytes are recommended since they have low effects on voice quality in case they get lost within the network. In our study, the CPS packets were generated by the AMR codec (as outlined in section 2.2). Multiple sources generated 39-byte (36-byte FP frame plus 3-byte AAL2 header) CPS packet at the Node B. The sources also generated 20-byte CPS packets in order to compare our results with the results in the literature [7].

- **PDU size**: Various PDUs will be investigated to find out the suitable IP packets that will use the link efficiently. It will depend on the number of mobile users supported and the timer-CU at the Node B. The PDU sizes from 234- to 390-byte and time-CU values of 2 to 3 ms as reviewed above provided a good trade-off.

- **End-to-end delay**: The sum of the packetisation delay at the Node B and Iub interface delay (time spent by the IP packets to traverse the interface). For high quality voice, we ensure that the 5 ms delay limit is not exceeded while striving to use bandwidth efficiently. For instance, 2 ms may be allocated to the PDU packetisation delay at the Node B and 3 ms to the Iub interface.

- **Jitter**: The delay variation experienced by the IP packets traversing the unreliable IP-based Iub interface, emulated by the dummynet pipe. To cancel jitter, a buffer is inserted at the receiver. Jitter needs to be measured in order to set the buffer size properly at the receiver to avoid adding more delay.

- **Packet loss**: An amount of the IP packets lost when traversing the unreliable IP-based Iub interface as discussed in section 2.5. For good quality voice, the packet loss ratio should not exceed 5 %.

3.4 The UTRAN Topology

The UTRAN consists of several Node Bs connected to the RNC through the Iub interface and each RNC may control about one hundred Node Bs. Each Node B may support a few hundred users that fall within its area of coverage. Within the UTRAN today, the Node B connects to the RNC through a point-to-point[11] link based on AAL2/ATM [36]. The RNC controls the radio resources and is the key interface for the mobile user towards the core network. It is responsible for admitting or rejecting mobile users since it controls the resources which results in many signalling packets carrying channel requests being transported from the Node B through the Iub interface to the RNC and back to the Node B. If all the Node Bs that are connected to the RNC connect through a point-to-point link, then there will be a lack of scalability and reliability (as discussed in section 1.1). In the UTRAN design, each Node B connects to the RNC through an IP-based Iub interface as

[11] A point-to-point link refers to a statically created pipe between the Node B and the RNC.

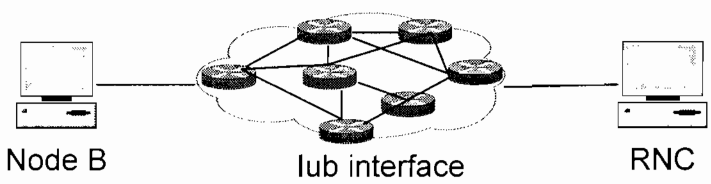


Figure 3-1: UTRAN network topology

illustrated in figure 3.1. This interface allows traffic from multiple Node Bs to share bandwidth, which provides scalability within the interface. Furthermore, if one RNC fails then a Node B can connect to another RNC (not indicated in figure 3.1) through this interface, which ensures reliability through redundancy.

In our study, the UTRAN topology consists of one Node B, the Iub interface and one RNC as illustrated in figure 3.1. Since our primary focus is on the uplink only, the Node B will be responsible for admitting or rejecting mobile channels to avoid the channel requests being sent to the RNC. This allows for scalability and simplicity in the Iub interface. This topology is adequate to perform the initial developments and testing of an AAL2/IP framework in the Iub interface. The protocol stack for our UTRAN design, for both user and control planes, is shown in figure 2.3 (a) and the SAP from AAL2 layer to IP layer is in figure 2.3 (b). Note that AAL2 (user and control planes) only resides in the Node B and the RNC, not in the Iub interface. The experimental test-bed based on PCs running the Linux operating system (OS) to emulate the UTRAN in figure 3.1 is provided in figure 4.1. The ultimate goal of the test-bed is to use bandwidth efficiently while providing QoS to each mobile user in the Iub interface. The test-bed should support logging for accounting purposes, and traffic flow information for evaluation purposes.

3.5 The UTRAN Functional Requirements

In the preceding chapters, we examined AAL2 as a good carrier of compressed voice, IP as a viable transport and the need for tunnelling AAL2/IP in the Iub interface, but with emphasis on feasibility, efficiency and QoS assurance. This section presents the technical overview of the requirements of the Node B, the Iub interface and the RNC with more emphasis on the Node B since we primarily focus on the uplink. These requirements have been placed into groups for clarity to the reader where the following requirements relate to both the Node B and the RNC (like in [19]):

- The Node B and RNC should have the AAL2 layer to multiplex multiple channels and IP layer to transport these channels to the RNC. As UMTS grows in popularity, it is likely that more than 248 channels will connect to the Node B. For this reason, multiple AAL2 tunnels should be supported by the AAL2 layer. Each tunnel will accommodate a limit of 248 channels and channel 8 on each tunnel is reserved for signalling purposes imposed by ITU specifications [22].

- To have a dynamic Iub interface, signalling is required in the Node B to create, manage and tear down idle channels within a tunnel. If a channel request arrives at the Node B, it is forwarded to the entity which admits channels within the Node B. If this channel is admitted, the signalling entity will be informed to create this channel. Once 248 channels are fully occupied, the admission entity in the Node B will request the admission entity in the RNC, which controls the radio network resources, to allocate resources for another tunnel. Like user data, this request for a new tunnel is transported on the AAL2 layer and not on the IP layer. Furthermore, the signalling is responsible for the maintenance of the mapping of internal mobile ID to Channel ID and tunnel tables in the Node B (the mapping table is illustrated in table 3.1 in section 3.5).

- In the telecoms networks, voice compression imposes jitter which is usually dominated by the codec used for a particular channel. Furthermore, the more powerful the compression scheme used for a channel, the more sensitive the application will be to jitter and packet loss. Each channel in the tunnel should be treated with a special QoS level that must match its delay requirements. This implies that scheduling is needed at the Node B to support various QoS levels per channel during packetisation. The compressed voice will incur delays when waiting to be scheduled and packetised at the Node B and also when traversing the Iub interface to the RNC.

The following requirements relate to that of the Node B to support AAL2 over IP:

- For each tunnel created, the Node B admits channels based on the measured outgoing IP link utilisation associated to that tunnel. When a mobile is active, it sends FP frames to the Node B. These frames are adapted to the CPS packets for packetisation purposes at the assembly unit of the Node B. The CID of each CPS packet is checked to find out if the packet belongs to an already created channel. If it does not belong to any of the existing channels and it is not classified as a channel request, then it is simply discarded. Otherwise it is forwarded to the buffers where it will wait to be scheduled and multiplexed into a PDU of its tunnel for transmission to the RNC.

- To forward the mobile frames, the Node B uses three labels which are kept in its lookup table. These labels are derived from the mobile IDs associated with the DCH traffic streams of that mobile, the CID and the tunnel to which this CID belongs as illustrated in table 3.1. By using these labels, the functional AAL2/IP model may be achieved. An important requirement is that the entries in this table be accessible to the entities that will provide dynamic update of the entries.

- At the Node B, input queues are needed to hold the pending CPS packets. These queues hold packets as they accumulate to ensure that packet loss is kept within a reasonable limit under heavy loaded UMTS system. When packetising the CPS packets into PDUs, the Node B should conform to the packetisation delay to support voice quality. It is important to keep track of each CPS packet as it enters the PDU for a tunnel. This is to maintain the correct state for each input tunnel so that when more packets arrive, processing may continue from where it left off.

In the IP-based Iub interface, the larger the number of hops the IP packet traverses will affect the end-to-end delay. Furthermore, the undetermined route that the IP packets follow can lose them or cause them to arrive out of order, which makes it hard to guarantee QoS for voice. In our study, to provide QoS in the Iub interface and to avoid having IP packets arriving out of order at the RNC, MPLS will be used [36]. MPLS creates label switched paths (LSP) that the IP packets follow to the RNC, instead of the IP packets using undetermined routes. At the MPLS domain edge router, a simple label is tagged on the IP packets header. The label forwards the IP packets faster than performing the IP address lookup at each node which minimises delay within the Iub interface. It identifies uniquely the path that the IP packet will use to traverse the domain to the RNC.

The IP packets' QoS requirements can be mapped from the IP packet's header onto this label. The label operates only within the domain, not up to the receiver. Another advantage of MPLS is that traffic can be aggregated into streams with similar QoS requirements and forwarded to the same destination [33]. Furthermore, in a real UTRAN network, it is very important that the Node B as the traffic aggregator is able to respond to the feedback from the Iub interface. For instance, the Iub interface may inform the Node B to reduce the sending rate since the IP packets are being dropped on the paths as a result of congestion.

3.6 Definition of the Node B Design

This section defines the actions that are taken to process the CPS packets as they enter the Node B. It covers the CPS packet treatment only at the Node B since we focus on the uplink. With the design requirements in section 3.3, one may now look at the actions that are taken to multiplex the CPS packets into a PDU. A single tunnel is considered in figure 3.2, adapted from [19][22], since packetisation of the PDUs in other tunnels is the same although each tunnel maintains its own state for packetisation. The description of this SDL repertoire is presented in appendix D. It is not possible to show the current state of the packetisation of the PDU in the SDL diagram [19]. For demonstration purpose, we assume that the incoming CPS packets are mapped directly to the outgoing tunnel's PDU according to the CID table for that tunnel. Figure 3.2 covers both data and signalling packets since it is possible that the incoming packet may be either of the two packets.

When the tunnel is in an idle state, it has completed processing all the CPS packets in the input buffer and is waiting for new CPS packets to arrive from mobiles that belong to it.

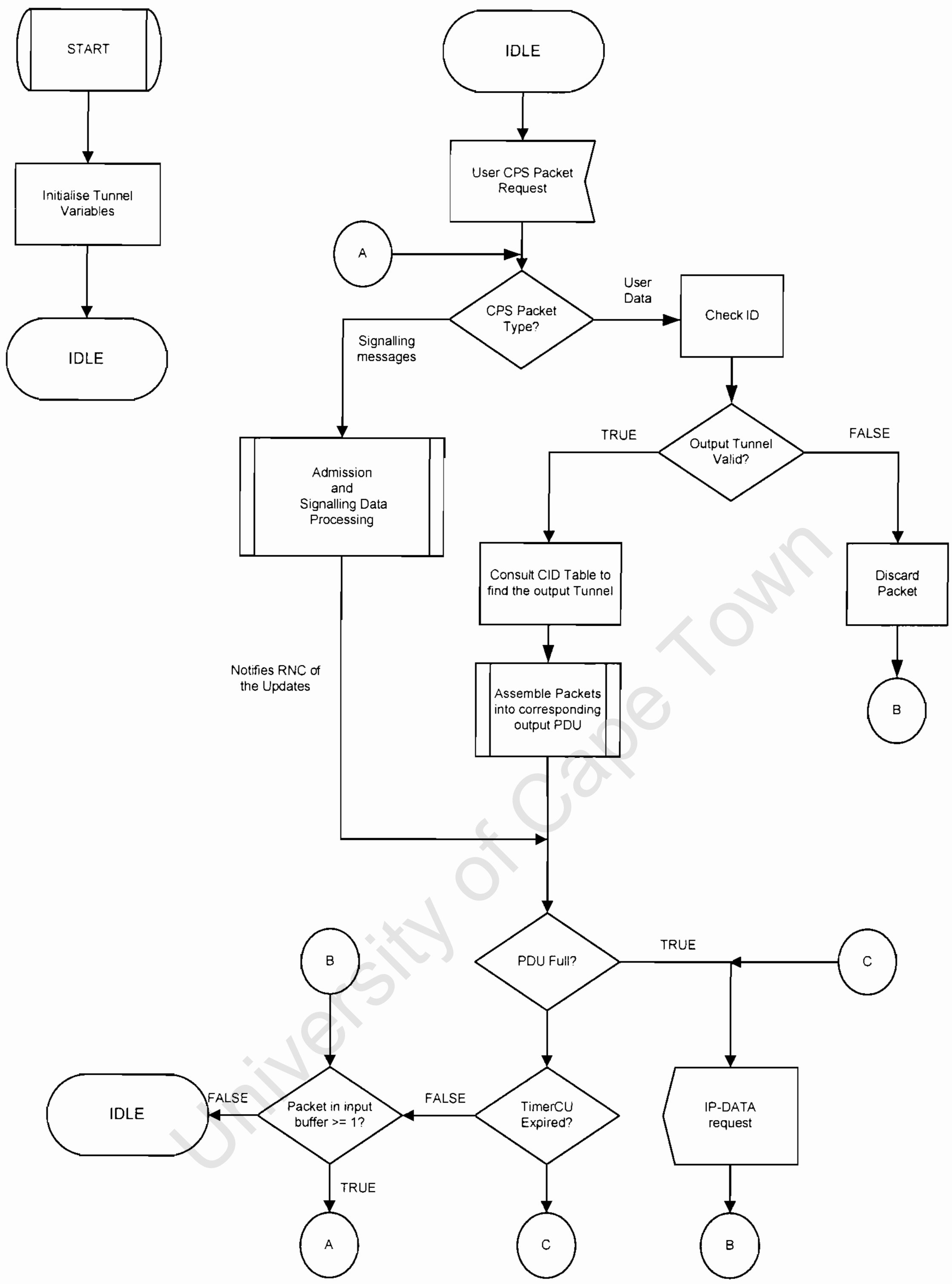


Figure 3-2: SDL Representation of AAL2 Tunnelling at the Node B

When the CPS packet arrives, a decision about this packet is made before it is processed any further. If the packet contains signalling, it will be forwarded to the admission and signalling entities within the Node B for processing and the requested updates will be performed. This update is more likely to be a channel request in the tunnel. If the request is indeed a channel request, before a channel can be created, bandwidth utilisation on the outgoing link that connects the Node B to the Iub interface will be measured by the admission entity. This is to find out if the new channel can be admitted and whether this channel will not interfere with the existing channels. If the tunnel is full, then the admission entity will inform the RNC to allocate resources for another tunnel. In essence, channel requests are processed at the Node B and tunnel requests are initiated by the Node B and sent to the RNC which controls resources in the UTRAN network.

If the packet contains user data, then the CID of that packet will be checked to see if it belongs to any of the existing channels. If the CID is invalid, then the CPS packet will be discarded. Otherwise the CPS packet is placed into the input buffer. Each channel created is allocated a separate buffer, including channels reserved for signalling. In this study we will focus on a single tunnel; however in case multiple tunnels exist, these tunnels may have duplicates of the CIDs which results in multiple buffers being created for these channels. Although tunnels may have channels with the same CIDs, these channels are not the same and are treated according to their QoS requirements at the Node B (as discussed in section 3.3). Note that each tunnel retains its own state information.

If the mobiles keep sending traffic, more CPS packets will be stored in the input buffers. This means that a loop will be entered at C in figure 3.2 where the packets are taken from the buffers and multiplexed. The number of packets processed before the loop expires will depend on the QoS requirements of those packets and the scheduling policy adopted to provide QoS for that tunnel. The QoS requirements are negotiated between the mobile user and the Node B during channel set-up. The channels with similar QoS requirements are grouped together. In the scheduling process, the packets with similar QoS are copied from the input buffers and placed in the output tunnel buffers. Once scheduling of the packets with similar QoS is finished, then the following group of packets with similar

QoS will be processed. This process will go on until all the input buffers have been accessed at least once. This is to avoid starving the channels with the lower priority.

The packets' CIDs are only examined before they enter the input buffer, however as they leave the output buffer they are assembled into PDUs without examining their CIDs again. This is due to all the CPS packets, which belong to one tunnel, going to one RNC. When the PDU gets full, a new PDU is started to carry incoming packets. In case the PDU gets full while a packet is being packetised, the remaining data of that packet will go into the new PDU. Once there are no more CPS packets to be multiplexed in the input buffers, the system will return to the idle state, to wait for more traffic to arrive. For each outgoing tunnel, a separate timer is associated with this tunnel in order to ensure that each PDU is forwarded to the IP layer on time to avoid packetisation delays. The value of this timer is part of the state of that tunnel. When the timer expires, the PDU being packetised is forwarded to the IP layer for transmission. Unlike in ATM, the PDU is not padded but it is adjusted according to the amount of traffic it is carrying and sent to the IP layer.

3.7 Design for the UTRAN

This section presents the system design for the UTRAN[12] to support tunnelling of AAL2/IP. Currently the UTRAN is based on the ATM transport protocol, where each Node B connects to the RNC via a virtual circuit [18][36]. In our study, each Node B connects to the RNC through an IP-based Iub interface as in figure 3.1. To tunnel AAL2 over IP, the Node B and the RNC should have AAL2 and IP layers in their protocol stacks as shown in figure 2.3. The Node B and RNC are broken up into several modules which operate sequentially and depend on each other for operation in terms of passing traffic. Particularly, the Node B will have modules to adapt FP frames to CPS packets and multiplex these packets into a PDU. Of more importance in our design is that the Node B and the RNC operate dynamically in terms of channel or tunnel creation and buffer management. This requires modules such as signalling entities to update information dynamically and queue managers to dynamically allocate buffers on demand.

[12] The words UTRAN and Iub interface are used interchangeably to refer to the Iub interface here

MID	CID	Tunnel
1148	9	1
2164	10	1
:	:	:
1309	255	1
2926	9	2
:	:	:
2367	255	2

Table 3-1: Mobile ID to Channel ID and Tunnel Mapping at the Node B

For each channel created in the tunnel, it is assigned an appropriate QoS in the scheduler. Note that in GSM and Ethernet networks, each mobile device or PC has a unique ID. For instance, a cell phone's SIM card or a PC's Ethernet card is developed with a unique ID which is appended to each frame sent to identify frames from each other in the network. When a mobile sends a frame to the Node B to request for a channel, the mobile ID (MID) will be associated with a CID in the tunnel and this will be kept in the lookup table in the Node B. As outlined before, each Node B can support a few hundreds mobiles within its coverage and each tunnel can support up to 248 channels. If there are 300 mobiles within its coverage, then the first 248 mobiles may use channels 9 to 255 as illustrated in table 3.1. On the second tunnel the 249th MID may use CID 9, (note that the CID 9 in tunnel 1 is not the same as the CID 9 in tunnel 2, similarly for the other CIDs). The number of channels that can be allocated in a tunnel depends on each mobile's sending rate. If all the channels send at a low bit-rate, then more channels will be allocated into one tunnel. Furthermore, each tunnel is allocated its own outgoing IP connection to the Iub interface over the 2 Mbps (E1/T1) link.

In our design, the signalling packets are treated with high priority during processing. For this reason, user data will have to wait until the signalling data are processed. As the number of tunnels increases, the number of state machines in the Node B will increase. That is, the state machine that adapts the FP frames to CPS packets and multiplexes these packets will increase. These will affect the operation of the Node B in terms of the time it takes to process each tunnel. Furthermore there are three stages where the CPS packet may be queued which incurs delay. First stage, in the input buffer as the CPS packet waits to be scheduled. Second stage, as the CPS packet waits to be inserted into a PDU. Third stage, in the IP buffer before the IP packet leaves the Node B to the Iub interface.

In a real UTRAN, the admission control is situated in the RNC. This means that when a channel request arrives at the Node B, it is forwarded to the RNC. If the admission control at the RNC accepts this channel, it will send the message to the Node B to create the channel and notify the mobile device that the channel has been accepted. This results in too many signalling packets being transported across the Iub interface. To reduce these packets in our study, the Node B will have the admission control authority. This is due to our primary focus being to transport the mobiles traffic from the Node B to the RNC referred to as the uplink and not from the RNC to the Node B referred to as the downlink. This means that the Node B will have full control of the Iub interface utilisation since we do not consider the downlink. The admission control unit will admit channels in the tunnel based on the measured bandwidth in the link that connects the Node B to the Iub interface. This unit should ensure that the new channel does not interfere with the existing channels. The complete Node B design is given in figure 3.3 and consists of several modules. Since we only focus on the uplink, only the Node B design is shown in figure 3.3. The RNC design is the inverse of the Node B which adapts the CPS packets to FP frames [24]. The modules are described as follows:

- **CPS Unit:** Receives FP frames through the DCH traffic streams from the mobiles generating voice and adapts these frames to CPS packets and passes them to the Classifier. It has the FP frames to CPS packets translation table (MID to CID table) to map frames to CPS packets and an AAL2 tunnel.

- **Classifier:** Scans all the incoming CPS packets to determine whether the packet may be classified as signalling or user data based on the CID, which can take on certain values[13]. It also checks whether the CPS packet, in case it is carrying user traffic, belongs to one of the existing channels and places that packet into its associated buffer for processing. The CPS packets with any value from 0 to 7 on the CID field are automatically discarded.

[13] CID values from 0 – 7 are not allowed and have been reserved for future use [22].

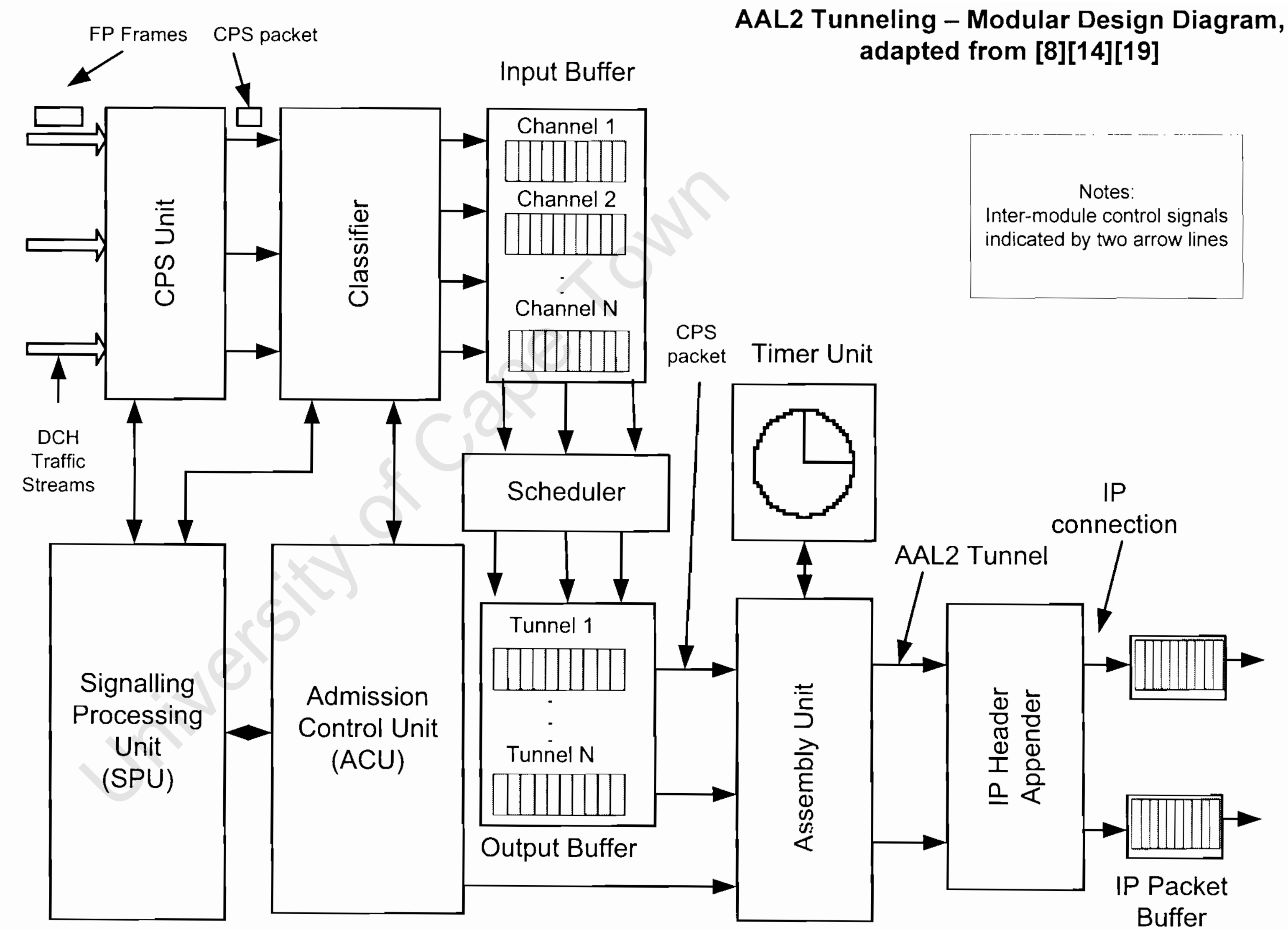


Figure 3-3: Node B Design for Tunnelling AAL2 over IP

- **Admission Control Unit (ACU):** Admits new channels in the tunnel based on measurement based admission control (MBAC - as discussed in section 2.5). The MBAC admits or rejects channels based on measured bandwidth usage of the link from the Node B to the Iub interface. If a new channel is admitted, the ACU notifies the SPU to create the channel. However, if the tunnel is full, the ACU will send a packet to the ACU in the RNC to request for another tunnel. If the resources are available in the UTRAN, the ACU will inform the SPU in the RNC to create the tunnel, which will then inform the ACU in the Node B about the new tunnel. The communication links from the Node B to the RNC and vice versa are not indicated in figure 3.3. Note that the Node B allocates mobile channels, while the RNC allocates tunnels in demand (based on the Node B).

- **Input Buffer:** Stores the CPS packets that are waiting to be scheduled. It is accessible to the SPU as it creates and updates these buffers dynamically and to the Classifier as it inserts packets in the buffers. Due to the stringent delay and packet losses requirements for voice traffic, it is important to separate voice channels queues since they have different quality of service specifications [28].

- **Scheduling Unit (SU):** Copies the CPS packets from the Input Buffer to the Output Buffer. It differentiates QoS requirements between different channels and groups channels with equal or similar requirements into a single type of service and performs suitable scheduling algorithm on that group.

- **Output Buffer:** Stores the CPS packets that are waiting to be multiplexed into the PDUs. It is accessible to the Assembly Unit (AU) as this unit removes packets and places them into the PDUs. In this buffer, the packets from all the channels that belong to one tunnel are stored in one buffer since they are all going to the RNC and the CPS packets from different tunnels are not mixed in this buffer.

- **Assembly Unit (AU):** Reads in the CPS packets from the Output Buffer and multiplexes them into the PDUs. If more than one tunnel exists, it reads packets into separate PDUs for each tunnel. It has access to the CID tables in the Node B to perform lookup for each packet to determine the packet's outgoing tunnel. Once the tunnel is found, the packet will be placed into the PDU for that tunnel. Furthermore, the AU may be interrupted at any stage by the TU (discussed below), in which case the AU will have the PDU being packetised immediately sent off. This happens when the PDU packetisation time (TU) has expired.

- **Timer Unit (TU):** Interrupts the AU when the packetisation time has expired. Due to this interrupt, the AU has to immediately forward the PDU being constructed to the IP Header Appender (discussed shortly). The TU has the highest priority of execution in the whole design system. It maintains the packetisation times for each active outgoing AAL2 tunnel and ensures that the relevant transmitter entity is informed when its timer expires.

- **Signalling Processing Unit (SPU):** Creates, maintains and releases channels dynamically within the Node B. It communicates with the ACU before creating a channel. After releasing a channel, it notifies the ACU of this release so that the ACU becomes aware of the available bandwidth to admit other channels. Note that the SPU in the Node B creates channels based on the permission granted by the ACU. When the channel is created, the SPU in the Node B informs the CPS unit and the Classifier about the newly admitted channels so that these units can enter the MID to CID translation into their tables for this new channel.

- **IP Header Appender and IP Packet Buffer:** Appends the IP headers to the outgoing PDUs and holds the outgoing IP packets in case the outgoing link is still processing other IP packets, respectively.

The design for the RNC is an inverse of the Node B design, however the RNC receives IP packets and disassembles these packets to deal with the CPS packets inside. It classifies the CPS packets as to whether they are user traffic or signalling messages (requesting for a tunnel) and stores them into an Input Buffer. If it is a signalling message, it will be passed on to the ACU unit in the RNC. If it is user traffic it will be passed on to the CPS unit which will then convert them back to the FP frames [24]. The following modules in the RNC differ as to the modules that appear in the Node B:

- **CPS Unit:** Receives CPS packets and converts them to FP frames to be passed to the FP layer as shown in protocol stack in figure 2.3 (a).

- **Signalling Processing Unit (SPU):** Creates tunnels as the signalling requests for tunnels arrive from the ACU.

- **Admission Control Unit (ACU):** Allocates new tunnels in the Iub interface. It uses MBAC algorithm to admit tunnels based on measured bandwidth usage of the existing tunnels in the incoming link between the Iub interface and the RNC. After the new tunnel has been created, it is the responsibility of the ACU and the SPU in the Node B to admit and create channels respectively as discussed above.

- **Radio Resource Management (RRM):** Manages and ensures that the radio network resources, e.g. bandwidth, are available within the UTRAN. The IP based UTRAN will transport aggregated voice channels that require guaranteed end-to-end QoS as well as extremely high data rates. The Iub interface bandwidth is a limited resource (as discussed in chapter 1) and in order to use it efficiently while guaranteeing QoS, an efficient RRM algorithm is required. This algorithm deals with the allocation and maintenance of the radio network resources. It is needed to guarantee QoS to the mobile users, maintain the planned coverage area and offer high capacity. This algorithm can be divided into power control, handover control, admission control (discussed above), load control and packet scheduling functionalities. The RRM family except the admission control unit does not form the focus of our study and is covered in detail in the literature [11][37].

Chapter 4

A Framework to Evaluate AAL2/IP

4.1 Introduction

The previous chapter presented a complete design of the UTRAN based on AAL2/IP with emphasis on the user and control planes in the uplink direction. This chapter focuses on the UTRAN test-bed used to implement the AAL2/IP framework. The test-bed provides a platform on which to evaluate the performance of AAL2/IP in the Iub interface. The implementation was based on chapter 3. The control plane, where the signalling and admission control units reside, was not implemented in our study. This is due to our primary focus, being to achieve a functional user plane of AAL2/IP that will use bandwidth efficiently while providing QoS to each mobile user in the Iub interface.

The chapter begins by describing the test-bed set-up to emulate the UTRAN. It covers hardware and software architecture including the operating system (OS) chosen for implementations. It is very important that this OS meets the design requirements. The chapter further discusses the test-bed components which are used to generate multiple mobiles traffic at the Node B and transport it through the Iub interface to the RNC. It discusses the emulator used to emulate the IP-based Iub interface and the method used to collect traffic during run-time. It concludes with notes on how the project was developed.

4.2 UTRAN Test-bed Architecture

The effectiveness and performance of the network protocols could well be evaluated in the real-world networks with real voice. However, there are very limited real-world networks with the desired architectures such as AAL2/IP for research. Furthermore, the real-world network parameters such as bandwidth, queue sizes and delays are often not

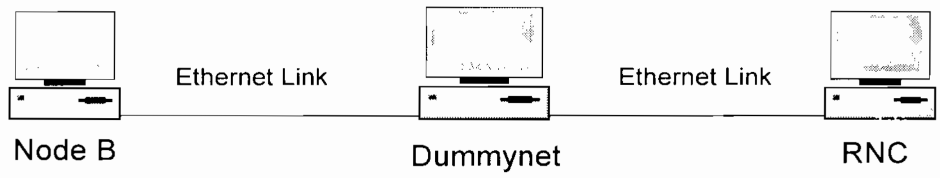


Figure 4-1: Emulated UTRAN network test-bed, with dummynet router

easily controllable. Hence using a real-world network with the real-time operating system would require extensive further research and draw the attention away from the primary aim of our study. Due to this, network emulation was performed to evaluate the performance of AAL2/IP in the UTRAN. Emulation allows network architectures to be modeled realistically where the network parameters are much more controllable as opposed to performing experiments on the real networks. An emulated network uses the real network protocols to transport generated traffic from one end station to another and the processing overhead is real. A motivation for our choice to emulate instead of simulate the UTRAN topology is provided in section 4.5.

The emulated UTRAN test-bed[14] is shown in figure 4.1 and consists of hardware and software components. The test-bed could have been set-up on a standalone workstation with the Iub interface emulator installed on it. This workstation would act as the traffic generator and multiplexer, Iub interface emulator and traffic receiver. However, due to the processing CPU resources needed to perform these tasks, the test-bed performs various tasks on separate workstations as illustrated in figure 4.1. The test-bed generates and multiplexes traffic from N mobiles at the Node B. It emulates bandwidth limitation and the unreliable effects of IP in the Iub interface using dummynet (discussed in section 4.5) and receives traffic at the RNC.

The hardware used to implement the Node B and RNC is as follows: Pentium 4 Celeron, 2.4 GHz, 256 MB RAM. The operating system (OS) installed on the Node B and RNC is Linux kernel version 2.4.26 which is a fairly updated version. The IP-based Iub interface workstation is a Pentium 2, 400 MHz, 64 MB RAM. The OS installed in it is FreeBSD

[14] In this study, the words test-bed and framework are used interchangeably to refer to our experimental test-bed

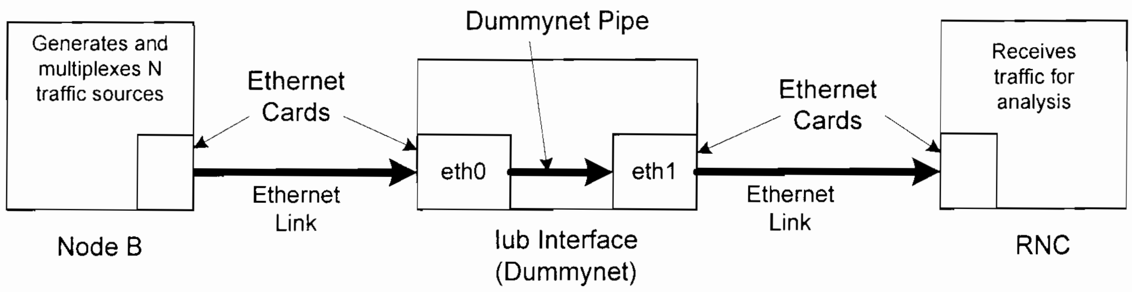


Figure 4-2: Physical network layer of UTRAN test-bed (uplink)

version 3.4-STABLE and dummynet runs on this workstation. Dummynet is an IP network emulator developed on FreeBSD kernel. It emulates IP networks based on IP firewall rules (*ipfw*) in the kernel. All the three workstations have Ethernet cards installed on them and they communicate through Ethernet links as shown in figure 4.2. Ethernet easily accommodates the necessary bandwidth required for our study. The Node B sends IP packets to the dummynet workstation where a dummynet pipe is created as illustrated in figure 4.3. The pipe treats the IP packets based on the specified user-configurable parameters such as the Iub interface delay and bandwidth limit. In order to measure the end-to-end delay, the clocks of the Node B and the RNC are synchronised through dummynet using the Network Timing Protocol (NTP) (as discussed in section 4.6).

As stated above, Linux was chosen to implement both the Node B and RNC. In its native form, it is a non real-time operating system (OS) supported by most of the GNU applications. A non real-time OS was chosen because it was realised that implementing a real-time OS would consume more time and draw the attention away from the aims of our study. Linux was chosen because of its well-integrated packet filtering tools. It is a free Unix-type OS, developed using the C programming language (the language used to develop the Node B and RNC software components). It generates interrupts every 10 ms minimum. This means that the timer at the Node B cannot expire before 10 ms to send the PDUs. This was an obstacle experienced with Linux since the UTRAN end-to-end delay (including packetisation delay) is 5 ms. This required the packet generating process to be slowed down so that they could keep up with the OS (as discussed in section 4.6). The software components of the test-bed are discussed in detail in the following sections.

4.3 Traffic Generator and Receiver

AAL2 is proposed to transport mobile traffic in the IP-based Iub interface. It is selected due to its matured technique to multiplex several low bit-rate channels and its high bandwidth efficiency on a given link. The low bit-rate data is generated randomly according to the AAL2 specifications [22]. In order to perform an analysis on the capabilities of AAL2 in the implementation, we require traffic generators to emulate the CPS packets which carry the encoded voice. The Node B software design should be flexible enough to support multiple traffic generators and multiplex their data into PDUs (IP payload). The mobile traffic generator should meet the following requirements:

- A generator should not generate CPS packets with random sizes. The sizes must be set beforehand. In fact, all generators generate traffic that models the AMR codec at the Node B. However, each generator should be flexible enough to generate CPS packet sizes that range from 1 byte up to 64 bytes as specified by the ITU 363.2 [22]. CPS payloads larger than 64 bytes must not be generated.

- Each CPS packet header structure must conform to the ITU 363.2 specifications.

- Each CPS packet payload must be randomly generated as if it is real encoded voice traffic from a mobile device. This is a simplest method to generate traffic. A generator must support a variable number of CPS packets to be generated in order to provide a satisfactory environment for achieving results.

- Multiple generators should be multiplexed into PDUs. The PDU sizes should vary in order to investigate different sizes until a trade-off performance between packetisation delay and link utilisation is obtained.

- During run-time, each CPS packet packetised into a PDU at the Node B should include the time at which it was packetised and the sequence number. At the RNC, upon arrival the IP packets should be stamped with the arrival time. This is to determine the end-to-end delay, jitter and the number of lost packets.

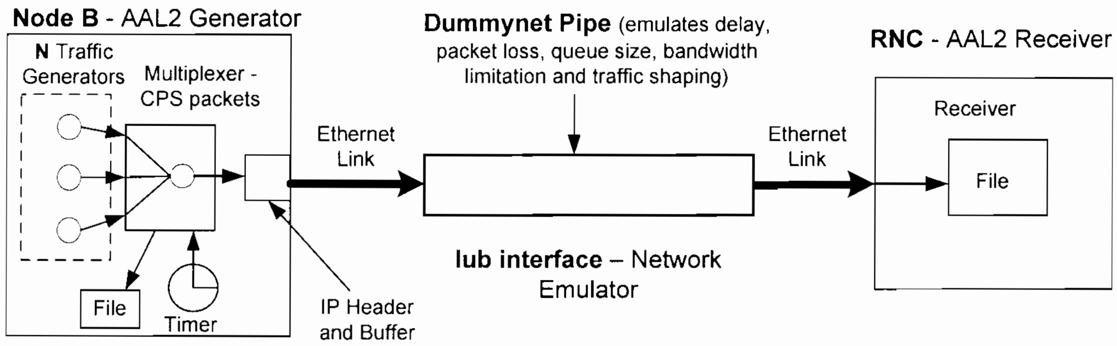


Figure 4-3: Test-bed components used to implement AAL2 tunnelling over IP

- Both the generator and receiver should have files to log the CPS packets and IP packets generated for analysis during offline mode.

To evaluate the performance of AAL2/IP, not all the modules that comprise figure 3.3 were implemented in our test-bed. In order to optimise the trade-off performance between end-to-end delay and bandwidth utilisation, the modules that were fully implemented are the Multiplexer (Assembly Unit), Timer, IP Header Appender and Buffer as in figure 4.3. The modules such as the SPU Unit, Classifier, Input Buffer, Scheduler and Output Buffer were not implemented and are represented by the Traffic Generators in figure 4.3. We assumed that each generator always has a CPS packet to submit to the multiplexer. Furthermore, all the generators have the same priority to submit a CPS packet to the multiplexer to ensure fairness amongst them. The first generator will submit a CPS packet to the multiplexer. It will wait for the other generators to submit at least one CPS packet before it can submit another packet. No module pertaining to the control plane (ACU or SPU) was implemented. The implemented modules are described as follows:

- **Traffic generator**: Represents an active mobile device sending voice frames. Generates CPS packets inside the Node B as if these CPS packets are coming from a mobile device. The CPS packets are generated by the AMR voice codec recommended for the UMTS. Each generator generates a 39-byte CPS packet every 20 ms and submits it to the multiplexer. N number of traffic generators may be modeled simultaneously.

- **Multiplexer:** Inserts CPS packets from multiple generators into PDUs as they arrive to it. To multiplex CPS packets from several mobiles, each generator submits a CPS packet to the multiplexer and it will be inserted into the PDU being constructed. The multiplexer operates in a Round Robin fashion to schedule the CPS packets. A Round Robin is an algorithm in which processes are activated in a fixed cyclic order. The multiplexer liaises with the Timer to send PDUs on time.

- **Timer:** Keeps the elapsed packetisation time of each PDU. The timer liaises with the multiplexer to ensure that the packetisation time for each PDU is not violated. It has the highest priority of execution over the Multiplexer.

- **IP Header Appender and IP Buffer:** Appends IP headers to the outgoing PDUs and stores the IP packets in case the Ethernet link is still busy processing other IP packets, respectively. These two modules, illustrated as one module in figure 4.3, reside in the IP layer in our study.

In UMTS, when a mobile is active, its AMR codec generates a 31-byte voice packet every 20 ms and sends it to the FP layer on the mobile device. The FP layer appends a 5-byte header and sends the FP frame to the Node B. The CPS layer at the Node B receives a 36-byte FP frame and appends a 3-byte CPS header to form a 39-byte CPS packet. In our study, we generate traffic on the AAL2 layer at the Node B as mentioned above. This means that each generator should generate a 39-byte CPS packet every 20 ms. All the generators generate CPS packets and submit them to the multiplexer at the same AMR codec rate. This implies that the multiplexer must be able to work at N times the rate at which a single generator is submitting a CPS packet. N is the number of generators that are being modeled per period. The multiplexer inserts the CPS packets into a Protocol Data Unit (PDU) as they arrive as illustrated in figure 4.4. It is assumed that no more than one PDU can be packetised at the same time.

Three generators are illustrated in figure 4.3. If they all generate CPS packets every 20 ms, the multiplexer must be able to accept at least three CPS packets every 20 ms, i.e. 1 packet every 20/3 or 7 ms period. If twenty mobiles are being multiplexed, then the

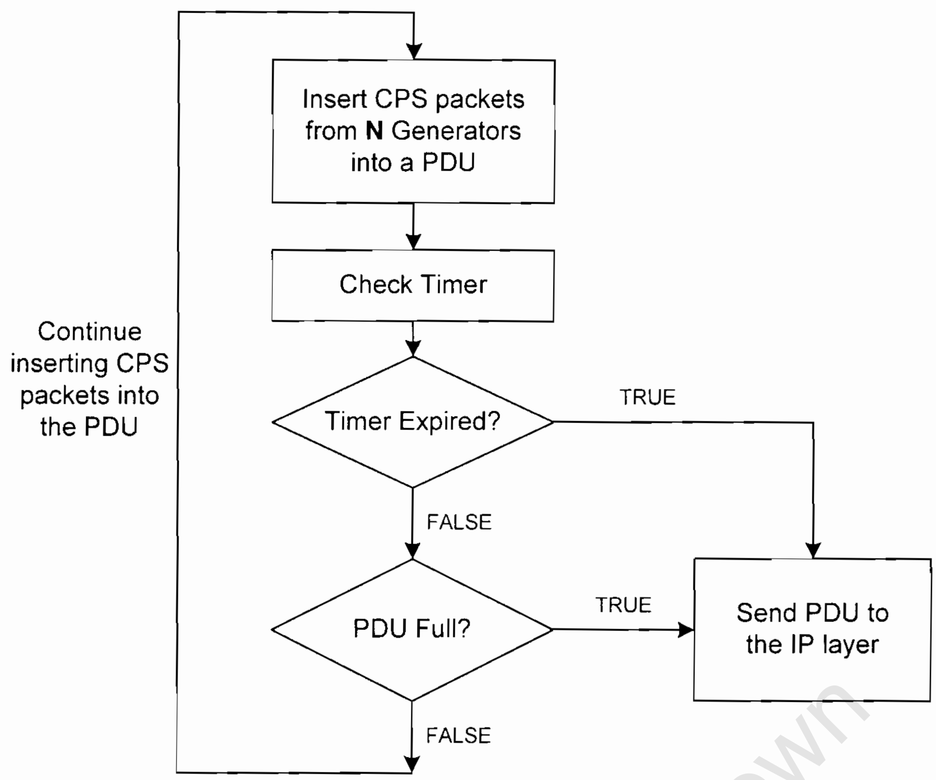


Figure 4-4: CPS Packets Multiplexer at the Node B of figure 4.3

multiplexer must be able to accept 20 CPS packets within 20 ms period, i.e. 1 packet every 1 ms. This implies that the higher the number of generators being modeled, the faster and harder the multiplexer should work. The multiplexer will receive a packet every 1 ms and insert it into a PDU until the multiplexer is interrupted by the timer when the packetisation time has elapsed. All CPS packets include the time in which they are inserted into a PDU and sequence numbers. Various PDU sizes and their packetisation times will be investigated to optimise a trade-off performance.

Once the packetisation delay has expired, the timer notifies the multiplexer to immediately send the PDU under construction to the IP layer. The IP Header Appender module in the IP layer will append the IP header on the PDU. The IP packets may be stored inside the Node B in case the Ethernet link is busy with other IP packets. The IP packets will be transmitted over the Ethernet link to the dummynet pipe that emulates the 2 Mbps E1/T1 link bandwidth limitation, traffic shaping, delay, jitter and packet losses.

After these parameters are emulated, the IP packets will be forwarded to the receiver over another Ethernet link. At the receiver, the IP header is stripped off at the IP layer and the PDU will be logged into a file for analysis offline.

An end-to-end delay bound of 5 ms was specified in the UTRAN Iub interface as discussed in the previous chapters. In our test-bed, this delay bound was broken into packetisation delay at the Node B and the delay to traverse the Iub interface to the RNC. That is, we allocated 2 ms to the timer-CU and 3 ms to Iub interface delay. The Iub interface was allocated 3 ms since 3 ms is a good worst-case approximation for a multi-hop IP-based Iub interface. This delay budget is the same as in the literature reviewed [23]. However we decided upon this budget prior to coming across this referenced paper.

The receiver, the RNC, receives the IP packets and logs them to a file to compare with those logged at the generator. The CPS packets inside each IP packet were analysed to evaluate the end-to-end delay (packetisation delay included), jitter and the packet losses. These were evaluated as follows: Upon arrival, each IP packet was stamped with the time it was received, referred to as time-now. The time-now and previous-time (the time stamped on the first CPS packetised in the PDU at the Node B) determine the end-to-end delay. The reason for using time stamped on the first CPS packet at the Node B is because this CPS packet stayed the longest time in the PDU while waiting for other CPS packets to be inserted. Furthermore, the time-now on the IP packet received now and the time-now on the previous IP packet received determine jitter. The sequence numbers were used to determine packet losses in the Iub interface.

4.4 Iub Interface Emulator Choice

Traditionally, there are three environments in which to test and evaluate network performance. These are network simulation, using simulators such as ns-2 [42], network emulation using emulators such as dummynet [40], and real-world networks. With real-world networks, it is difficult to set-up and control the desired operational parameters such as bandwidth, delays, queue sizes and packet losses [40][43]. It is difficult because queues and bandwidth are changing all the time due to users using at different times. In addition, there is a limited number of real networks with desired parameters to carry out research with architectures such as AAL2/IP, as stated before. It is this challenge that stimulated the development of network simulators and emulators as tools for research purposes. These tools allow researchers to create network topologies with desired parameters. They provide a much more controllable and reproducible environment for testing network performance. Most of these tools are open source on the Internet.

A simulator does not generate real traffic and does not run on real protocols. As a synthetic network environment, it easily simulates complex network topologies. This is because it is based on simplified assumptions about the network conditions and traffic dynamics which may poorly mimic the real network. The advantages of a simulator are that it simplifies a study of complex network topologies and not limited by the speed of the hardware that makes up a network. Emulators, also referred to as real-time simulators, are developed and used to validate the results obtained from simulators and analyse environments. Emulators run protocols in real time. In our study, we chose to perform emulation to evaluate AAL2/IP (as stated in section 4.2). We used an IP network emulator to emulate the bandwidth limitation and unreliability of the IP-based Iub interface. Emulation was chosen as opposed to simulation due to the following reasons:

- Our aim, as outlined in section 1.3, is to validate the simulations and analytical results reviewed in chapter 2.
- Emulation models network architectures and traffic close to reality. That is, validate the simulation results. It allows researchers to move network algorithms from an emulated to a real-world network. A simulator usually runs a time-independent network algorithm rather than the real network algorithms.
- It emulates network traffic between nodes in a physical network to model various network configurations. It uses real network protocols and real processing headers to transmit traffic from one node to another in a physical network.

Although it allows users to specify their desired parameters to emulate, an emulator appears as a black box to the user. The challenge is how to make this black box a good emulator of a real network. The disadvantage of an emulator as compared to the simulator is that the speed used to emulate traffic cannot be greater than the speed of the underlying physical network. Furthermore, complex network topologies are difficult to set-up since they must be constructed physically.

The emulator chosen for our implementation is dummynet [40]. However, a number of distributed IP network emulators, such as ONE [43] and NIST Net [83], are available for research in the Internet. Some of them are developed to emulate bandwidth limitation, traffic shaping and the unreliability of real IP networks. ONE is developed on Solaris OS and uses only three effects specified by the user to emulate IP networks, namely transmission delay, propagation delay and packet queues. ONE does not emulate bandwidth limitation and shaping traffic effects. NIST Net is implemented as a kernel module on Linux and on an X-Window system. NIST Net emulates effects such as delay, losses and bandwidth limitation. NIST Net does not apply drops and delays to outgoing packets, but instead only to incoming packets. This means that to apply delays and losses symmetrically, a workstation installed NIST Net must be used as an intermediate router.

Dummynet, a similar tool to NIST Net, applies the specified effects to both incoming and outgoing IP packets. Unlike other emulators, dummynet has a very little overhead as all processing is done within the FreeBSD kernel. It was developed with minimal modifications to the existing protocol stack. It is an open source tool which is available from its author, Luigi Rizzo [41]. To use dummynet, one may install FreeBSD on the hard disk or use a one-floppy version of FreeBSD called PicoBSD which is easy to compile. It is well documented, used by many researchers and supports a helpful mailing list [41]. In addition, its author keeps updating it and is available to answer questions concerning dummynet bugs. It is for these reasons that dummynet was chosen to emulate the IP-based Iub interface effects. In fact, the University of Cape Town used dummynet to manage bandwidth and shape traffic for its approximately 15 000 users [44].

4.5 Dummynet Operation

Dummynet is a simple, flexible and accurate network emulator. It manages bandwidth, shapes traffic and emulates network delay, jitter, finite queues, packet losses and multi-path effects. It is a fully configurable IP firewall (*ipfw*). It can be run on a standalone workstation with the traffic generators and receivers where it intercepts incoming or outgoing IP packets. The drawback of this set-up is the CPU processing resources as outlined above. Alternatively it can be run on a separate workstation where it acts as a router between two end stations as illustrated in figure 4.1. Although it is limited to one operating system for modifications, it provides a useful environment for testing the same code in both emulated and real-world networks [43]. It runs in a fully operational system, hence allows the use of real traffic generators and protocol stack implementations.

Dummynet manipulates the IP queue inside the kernel. It intercepts IP packets on their way through the protocol stack and passes these IP packets through a pipe. It uses the *ipfw* rules to apply the desired effects of bandwidth limitation, delay, etc on the IP packets. These desired effects are specified when the pipe is created. Due to its tight integration with the kernel and firewall system, the selection of traffic to shape or drop is very flexible [44]. Inside the pipe, the IP packets are passed through one or more queues which emulate the desired effects. Dummynet uses the *iptables* tool to forward the IP packets to the RNC. This tool is configured such that all the IP packets are forwarded to the RNC after the desired effects have been applied on the packets by the pipe.

In our study, we installed dummynet on a standalone workstation that acts as a router as shown in figure 4.1. We created a pipe with queues and configured this pipe with the desired bandwidth limit, queue size, delay and packet loss effects. The structure of this pipe is illustrated in figure 4.5. The pipe is a 2 Mbps bandwidth link from the Node B to the RNC. Dummynet associates the queues with a particular weight. The pipe was added to a list of filtering rules which are used to treat the IP packets passing through this pipe. The IP packets can be passed through multiple pipes depending on the *ipfw* rules configuration. Dummynet supports various types of IP filters for network interfaces.

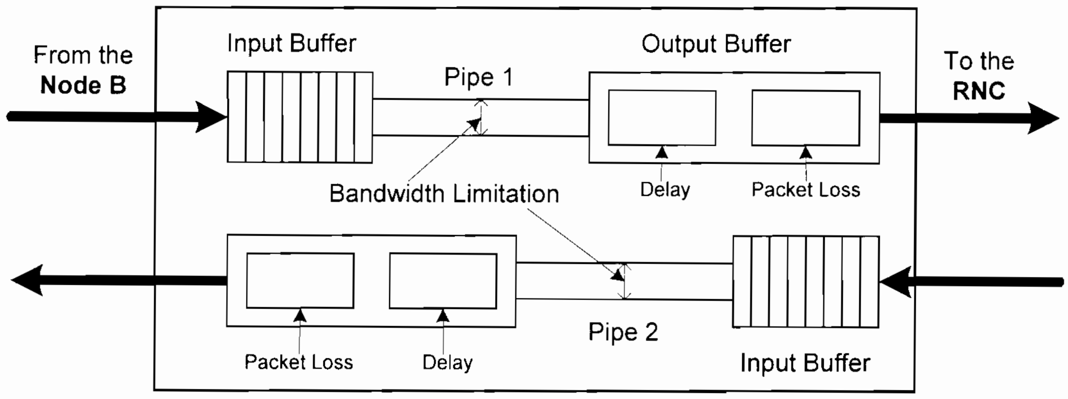


Figure 4-5: Bidirectional dummynet pipe architecture

The pipe illustrated in figure 4.5 is a bidirectional pipe, however in our study we only configured it to send traffic from the Node B to the RNC. Our goal is to create a pipe and configure it precisely so that this pipe emulates our desired effects that are used to treat the IP packets inside the pipe. For instance, the end-to-end delays and packet losses at the receiver should fall into the expected range.

The following information, on dummynet implementation, has been included for the interested reader, but may be glossed over in a first reading. More information on implementation may be found in appendix E1. The pipe was created using the command: *ipfw add pipe NN* and configured as follows: *ipfw pipe NN config bw B delay D queue Q plr P*[15]. Where NN is the pipe number, from 1 to 65534. The bandwidth B was expressed in bit/s, delay D in ms, queue size Q in packets, packet loss ratio (plr) P is the fraction, between 0 and 1, of the packets randomly dropped. B was a 2 Mbps bandwidth link. The general mechanism in which the pipe processes IP packets is as follows:

- When a packet arrives at the dummynet pipe, it is inserted in the input buffer. The number of IP packets that can be kept in this buffer depends on the queue size specified. In the queue, the IP packets are processed according to the queuing policy of choice. Typically queues use First In First Out (FIFO) with drop-tail, but other policies such as Random Early Detection (RED) and a variant of Weighted Fair Queueing (WFQ) are possible.

[15] An example of a pipe configuration: *ipfw pipe 1 config bw 2 Mbit/s queue 30 delay 300 ms plr 0.05*

- The IP packets are moved from the output buffer through a pipe. This pipe emulates the specified bandwidth limitation and shapes traffic if need be. The output buffer also uses FIFO policy.

- In the output buffer, the IP packets are kept for the specified Iub interface delay. Random packet losses, if specified, are also applied in this buffer. The losses due to the congested Iub interface are emulated by the queue bound-size. Furthermore, an IP packet re-ordering effect can be introduced to emulate the unreliability of IP networks. This effect is very important when the real network has redundant paths or noisy links. Then the IP packets are de-queued and sent to the RNC.

4.6 Traffic Collection

This section describes the way data was collected during a series of tests performed. To determine the end-to-end delay, the Node B and the RNC were synchronised through dummynet using the Network Timing Protocol (NTP) [38]. NTP is an open source tool widely used in the Internet to synchronise clocks for computers. It synchronises the client or server clock to another server or reference time source. It uses the Universal Coordinated Time (UCT) to synchronise time. To synchronise the RNC clock, a time offset of the RNC clock relative to the Node B clock was determined by the Node B running NTP program. NTP keeps clocks within a millisecond on Local Area Networks (LANs). The NTP query (NTPQ) command was run to inspect the offset between the Node B and RNC. The results of the command, run at 19h10 on the 02nd of November 2004 confirmed an offset of 0.104 ms as in figure 4.6. This offset performance is acceptable for our test-bed.

As discussed in section 4.3, at the Node B, every CPS packet multiplexed into a PDU included a sequence number and the time at which the CPS packet was inserted into a PDU. The time included in the CPS packets was the current time returned by the *gettimeofday* function. The contents of the PDUs were logged into a file at the Node B. This information was very important to trace the packet losses and to ensure that the

```
bongani@starsky:~$ ntpq -p
   remote           refid      st  t  when  poll  reach  delay   offset  jitter
==============================================================================
*hutch.ee.uct.ac LOCAL(0)      6   u   36   512   377   0.159   0.104  0.048
```

Figure 4-6: The results of the NTPQ command output

delay for each CPS packet did not exceed 5 ms from the Node B to the RNC. It was also important to ensure that jitter is kept as low as possible and to ensure that the IP packets arrived in order. Variable PDU sizes were used to carry CPS packets to find out the suitable size based on the packetisation delay and link utilisation.

The Iub interface delay was fixed (e.g. to 3 ms) and the timer-CU was varied to different values (e.g. 1.6 to 2.4 ms) to find out how the timer-CU affects the 2 Mbps link. Once timer-CU has elapsed, the timer informed the multiplexer to immediately send the PDU under construction. However as stated above, the obstacle experienced with Linux is that it does not generate interrupts with a time granularity of less than 10 ms. The solution was to slow down the sending rate of the traffic generators. This required the generating process to be calibrated by normalising the time-scale to base units, where the time at which an interrupt is generated, i.e. 10 ms, was declared to be 1 base unit. This implied that the 20 ms timer-CU was represented by 2 base units and the inter-arrival time of the packets from the mobiles was 200 ms (20 base units). The performance evaluations were conducted and the results collected were scaled down to real time units (i.e. ms).

To determine the correct operation of the Node B and RNC, the following requirements had to be met. Every PDU received at the RNC had to exactly match with corresponding PDU logged at the Node B. A typical logfile of a PDU is shown in figure 4.7. PDUs had to arrive in the sequence in which they were sent. PDU sizes were adjusted to send exactly the amount of traffic that was available when timer expired. There was no padding of PDU since the IP packet size is variable. Tests were performed with various PDU sizes to determine the PDU size that will provide a trade-off performance between bandwidth utilisation and packetisation delay. The logfiles for these tests could not be shown due to their large sizes.

1238001101481636210080100012500000000000**523800**11014816362300921000125000000000**16**
38001101481636250094100012500000000000**283800**1101481636270097100012500000000000**4238**
001101481636290103100012500000000000**343800**1101481636310106100012500000000000**393800**
1101481636330119100012500000000000**133800**1101481636350122100012500000000000**2238001**1
0148163637012610001250000000000 **203800** *1101481636* **390131** *1000125*0000000000

(a)

203800 *1101481636* **390131** *1000125*0000000000

This CPS packet consists of both the payload and header which are broken down as follows:

CPS (AAL2) header: **203800**
CID = **20**, LI = **39 - 1 = 38**, UUI = **0** & HEC = **0**

The CPS payload: consists of the following info that represents encoded voice traffic:

Seconds since the **epoch**:
1101481636

Micro seconds since the **epoch**:
390131

Sequence number (offset by 1000000)
1000125

The rest of CPS payload is filled with zeros which is considered to be part of the encoded voice.

(b)

Figure 4-7: (a) PDU logfile showing 10 CPS packets, (b) the CPS packet contents

The number of IP packets sent in each case was set to 10000. In all the tests performed, each IP packet was logged into a logfile both at the Node B and RNC. The logfile was analysed offline and the IP packets were found to match one another unless otherwise dummynet dropped the IP packet to emulate losses. The sequence number included in the CPS packets shows the sequence number for the IP packet not for the CPS packet. This is for tracing lost IP packets in the pipe. The seconds and micro-seconds values are the values returned by the *gettimeofday* function. This is a C function which returns the current calendar time as the elapsed time since the *epoch*. The *epoch* is the number of seconds elapsed since 00:00:00 on 1 January 1970, Universal Coordinated Time (UCT).

To evaluate the number of mobile users that can be supported by AAL2/IP, the timerCU was not used as in the literature [7]. The PDU (IP packet payload) size was set to Ethernet maximum transfer unit (MTU), i.e. 1500 bytes. Ethernet protocol appends a header of 18 bytes to the MTU. PDUs were packetised until they were full at the Node B.

To enhance AAL2/IP to support QoS, a packetisation time (timer-CU) constraint was introduced at the Node B. When timer-CU constraints were introduced, the PDU size dropped to a range of 250 to 500 bytes. This timer-CU was varied to different time-out values to determine the impact it introduced on the link bandwidth utilisation. To analyse the effect of the timerCU at the Node B, the content of the logged files was analysed. To analyse the QoS constraints such as delay and losses introduced in the Iub interface, the log files logged at the RNC were compared with those logged at the Node B (as discussed above). Figure 4.7 illustrates also different times in which the CPS packets were inserted into the PDU and the sequence numbers of these CPS packets.

Upon receipt of each IP packet, the receiver stamped each IP packet with the arrival time. The RNC subtracted the arrival time of that IP packet from the time its first CPS packet was inserted into a PDU at the Node B to get the end-to-end delay of the IP packet. The time was inserted to the CPS payload since there was no space on the IP header. If the difference between the two times does not exceed 50 units (5 ms), this would mean that all the other CPS packets also do not exceed 50 units since they were inserted after the first AAL2 packet. To determine jitter in the dummynet pipe, the receiver calculated the time difference between the current IP packet received and the time for previous received IP packet. To determine packet losses in the pipe, the sequence numbers were examined at the receiver. All these results were logged into a logfile which was analysed to determine the experimental findings.

4.7 Project Development

In order to compare the performance of AAL2/ATM and AAL2/IP, we modified the code used in the implementation of AAL2 Switch [19]. During the implementation of the AAL2 Switch, AAL2 packets were transported over ATM. We successfully modified the code written to transport AAL2/ATM to support AAL2/IP. However, due to the differences in specifications when programming for ATM (fixed size cells) and IP (variable packets sizes) layers, the modifications we could make were not enough to meet the design requirements. Due to this, we looked at network traffic sources or generators.

There are a number of software packages to emulate network traffic sources that exist on the Internet [39]. For instance, we considered using the *mgen*, the multiple generator toolset. However it was unable to meet our design specifications to emulate AAL2/IP traffic. This meant that we had to write our own code to incorporate with this software. Amongst all, this code would emulate AAL2 sources and multiplex multiple sources into PDUs. Furthermore, one had to modify this traffic generator to emulate AAL2 packets according to AAL2 specifications. For these reasons, we decided to write our own programs to generate and multiplex traffic at the Node B through dummynet and receive this traffic at the RNC. The programming language used for the implementation of this project is C, as it is fast, efficient and constant in the results it produces [19]. Furthermore, FreeBSD OS was implemented using C language. The following external resources were required to make the development of the project possible:

- Socket programming guide for standard IP socket operation.
- GNU Library Manual.
- Dummynet IP network emulator.
- The time library to implement the timer of the system.
- The signal library to allow a handler function to be attached to the SIGALRM system signal on timer expiry.

Chapter 5

Framework Evaluation and Results

5.1 Introduction

This chapter discusses the evaluations that were performed on the AAL2/IP framework, presents the results obtained and analyses them. The framework evaluated here is one where multiple mobile users generate voice traffic and send it to the Node B which forwards this traffic to the mobile users on the other side of the UMTS system. The Node B forwards the traffic through the Iub interface and RNC. The framework is emulated as shown in figure 4.3. The framework was evaluated before and after it was enhanced to support QoS requirements. The evaluation metrics are as outlined in section 2.7.

To analyse how these metrics affect one another in the framework, tests were performed. In the tests performed before it was enhanced, a 20-byte CPS packet (17-byte payload + 3-byte header) was used as in the simulations carried out in the literature [7]. This was to compare our results with their results. After the framework was enhanced to support QoS, a 39-byte[16] CPS packet (36-byte payload + 3-byte header) was used. For each test performed, the results are presented and then an analysis of the results follows. The results are presented in milliseconds instead of base units (as discussed in section 4.6).

5.2. Link Efficiency – without QoS

Link efficiency is defined as the ratio of the payload size compared to the IP packet size or ATM cell. Tests were performed to evaluate the AAL2/IP performance in terms of link efficiency and compare with AAL2/ATM. During the tests, the IP payload (PDU) only carried the amount of CPS packets that were available to be sent. The fixed size ATM

[16] CPS packet size generated by the AMR codec recommended for the UTRAN as discussed in section 2.2

PDU was filled with CPS packets and padding was applied when a PDU was not full. The Node B and RNC were directly connected by the 2 Mbps link, where the delay and jitter were very small and no packets were lost. During the tests, multiple AAL2 traffic sources generated 20-byte CPS packets every 20 ms and PDUs were packetised at the Node B. The timer-CU was set to 100 ms so that it does not affect the outgoing PDUs. Efficiency was measured for each CPS payload inserted into ATM and IP PDUs. Our results were compared with the results in the simulations carried out in the literature [7].

5.2.1 Results

Figure 5.1 shows the results of the efficiency for ATM and IP when 153 bytes (9 CPS payloads) had to be sent. The figure shows that IP uses bandwidth more efficiently than ATM except in one region, when 2 CPS packets were inserted into a PDU. Figure 5.1 is almost in line with the simulations results of the efficiency as the payload increases [7]. In figure 5.1, the efficiency of ATM begins at 32.1 % when a 17-byte CPS payload was inserted into an ATM cell and IP begins at 42.5 % when the CPS payload was inserted into a 40-byte IP packet. The 40-byte is made of a 20-byte CPS packet and a 20-byte IP header. The results were analysed for different payloads to compare the ATM and IP efficiencies. Three ATM headers were used to send 6 CPS packets as compared to the IP header which was used once. The repetition of the ATM headers and padding of cells consumed a large portion of bandwidth which resulted in the link used inefficiently.

5.2.2 Analysis of Results

Due to IP being more efficient than ATM as the payload increases, IP is a more suitable layer to transport the Iub interface traffic. But ATM is more efficient than IP when 2 CPS payloads (34 bytes) have to be sent as in figure 5.1. Efficiency for ATM is 34/53 = 64.2 % and for IP is 34/60 = 56.7 %, 60 is a total IP packet which is made of two 20-byte CPS packets and a 20-byte IP header. In the literature [7], two regions (at 2 and 6 CPS payloads) were found where ATM had higher efficiency than IP. Note that 2 CPS payloads are equal to 34 bytes and 6 CPS payloads are equal to 102 bytes in figure 5.1.

The first one region, at 2 CPS packets, complies with our results. However, the second region does not and is analysed as follows: When 6 CPS packets are inserted into an IP

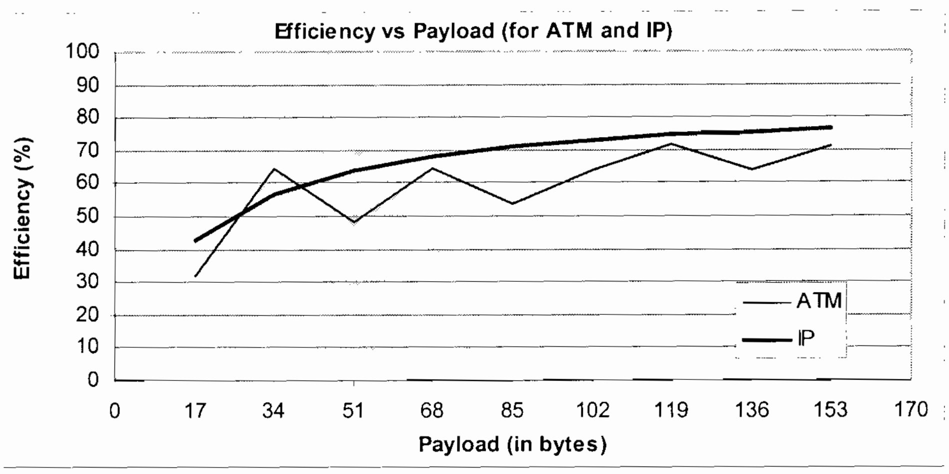


Figure 5-1: Efficiency vs Payload (for AAL2/ATM and AAL2/IP)

PDU, the efficiency is 72.9 % (102/140), 140 is made of 6 CPS packets and an IP header. ATM segments 102 bytes into 47-byte PDUs where a full PDU has 71.7 % (38/53), 38 is made of the first two CPS payloads and 4 bytes from the third CPS payload. With 102 bytes, the overall ATM efficiency dropped to 64.1 % due to the last PDU not being full. It is for this reason that we have only one region where ATM is more efficient than IP. We suspect that the last cell efficiency is the culprit in the literature since they found that at 6 CPS packets, ATM is more efficient than IP.

As the number of CPS packets increases, the ATM graph oscillates due to the 47-byte fixed PDU size. The graph increases until the third CPS packet was inserted which dropped the efficiency since it did not fit into the ATM PDU. A second PDU was needed to transmit the remaining content of the third packet. The efficiency of the first cell is 71.7 % and the second cell is 22.6 % (12/53). The overall efficiency dropped to 47.2 %. This drop was because of the padding on the second cell and the two ATM headers which consumed the bandwidth. The IP header is larger but it is used once and there is no padding applied in IP since the packet size is not fixed, it sends the available data traffic.

Although regions appear where ATM is more efficient than IP, IP remains an efficient transport layer for the mixture of traffic (more users) expected in UMTS. This is because IP uses the link more efficiently as the number of users increases than ATM which has a high repetition of cell headers. When mobile users are accessing traffic such as email and ftp, IP will transport this traffic much more efficiently since it can take bigger chunks per time as compared to voice traffic. This is because data are not sensitive to delay.

5.3 Link efficiency – with QoS

Tests were performed to evaluate the link efficiency after QoS was supported and compare our results with the analytical results in the literature [14]. The results were also compared with those obtained before QoS was supported to observe the effects of QoS on the link. The Node B and RNC were connected through the dummynet pipe. Only IP was evaluated since it has already been shown to be more efficient than ATM. Also our interest is to create a framework that uses the link efficiently while meeting QoS.

During the tests, AAL2 traffic sources were started and increased gradually to fill the PDUs with CPS packets before the timer-CU expired. From the 5 ms Node B-to-RNC delay, the timer-CU was allocated 2 ms and Iub interface was allocated 3 ms fixed delay. Although the timer-CU was decided to be 2 ms, it was varied from 1.6 to 2.4 ms to observe how it affects the efficiency in terms of numbers of CPS packets per PDU. The timer-CU resolution was observed to be 0.2 ms, i.e. it expired only at 1.6, 1.8 ms, etc. Furthermore, a timer-CU value of 3 ms and Iub interface delay of 2 ms were tested to observe how the efficiency will compare with efficiency when timer-CU was 2ms.

5.3.1 Results

Table 5.1 shows the results achieved when the 5 ms Node B-to-RNC delay budget was allocated differently. That is when the timer-CU value was 2 ms and Iub interface delay was 3 ms and also when the timer-CU value was 3 ms and Iub interface delay was 2 ms. Figure 5.2 shows that the link efficiency increases with the number of CPS packets inserted into a PDU as the timer-CU value increases. Only ten CPS packets were inserted into an IP PDU where the link efficiency was 87.8% at 2.2 ms. Increasing the timer-CU value to insert more than ten CPS packets resulted in a violation of the 5 ms delay limit. The link efficiency increased with an increase in the number of users as well as the timer-CU value. Figure 5.3 shows that the number of CPS packets inserted into an IP PDU increased with an increase in the timer-CU value.

Timer-CU (in ms)	Iub interface delay (in ms)	Max. number of CPS packets per PDU
1.6	3	7
1.8	3	8
2.0	3	9
2.2	3	10
2.4	3	11
3.0	**2**	**14**

Table 5-1: Number of CPS packets supported given different 5 ms delay budget values

5.3.2 Analysis of Results

Due to the introduction of QoS, particularly the timer-CU, in our test-bed a maximum of 10 CPS packets were inserted into a PDU giving the link an efficiency of 87.8 %. In the results in the literature [14], when the timer-CU was set to 2 ms, 10 39-byte CPS packets were inserted into a PDU per time. In our tests, a timer-CU value of 2.2 ms inserted 10 CPS packets per PDU and provided a 5.05 ms delay. This is because of dummynet's internal delay constraint and the packets were delayed for 3 ms with 6.6 % variation. When the timer-CU value was more than 2.2 ms, more than 10 CPS packets were inserted which increased the link efficiency further. However, a timer-CU value of more than 2.2 ms resulted in drastic Node B-to-RNC delay violations. Violating the Node B-to-RNC delay can violate the overall mobile-to-mobile delay and degrade the voice quality.

As shown in table 5.1, with a timer-CU of 3 ms, 14 CPS packets (566-byte IP packet) were inserted into a PDU. A 566-byte IP packet gives a higher efficiency than a 410-byte IP packet and dummynet delayed it by 2 ms. However, larger IP packets take longer to traverse a real multi-hop IP-based Iub interface. A 566-byte IP packet has a higher risk to be delayed than a 410-byte IP packet. This incurs additional mobile-to-mobile delay and hence degrades voice quality. Larger IP packets can highly degrade voice quality when dropped in the network. A drop or delay of a larger IP packet means that a number of mobiles will be affected as compared to smaller packet. A maximum IP PDU size of 351 bytes (9 CPS payloads) will use the link efficiently while ensuring the 5 ms delay limit.

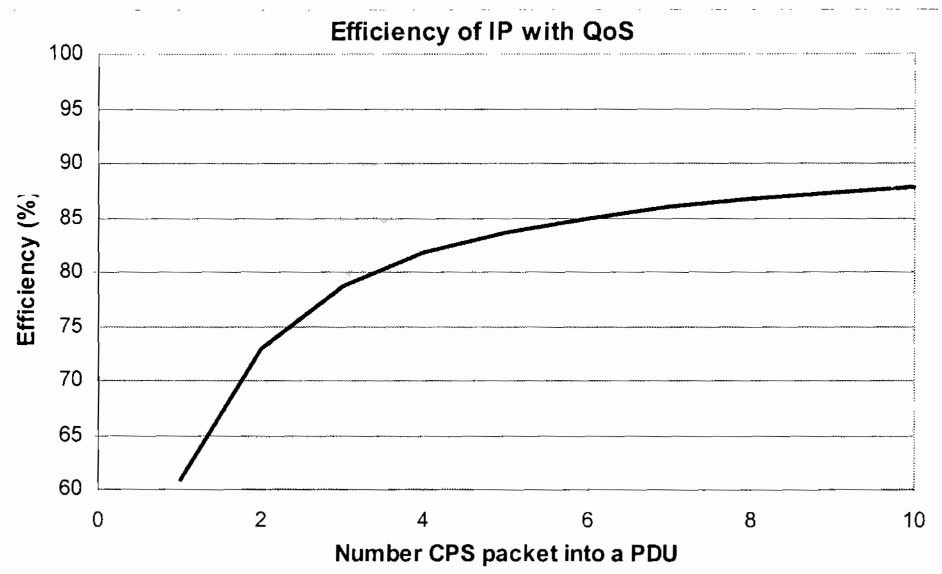


Figure 5-2: Link efficiency when QoS have been introduced

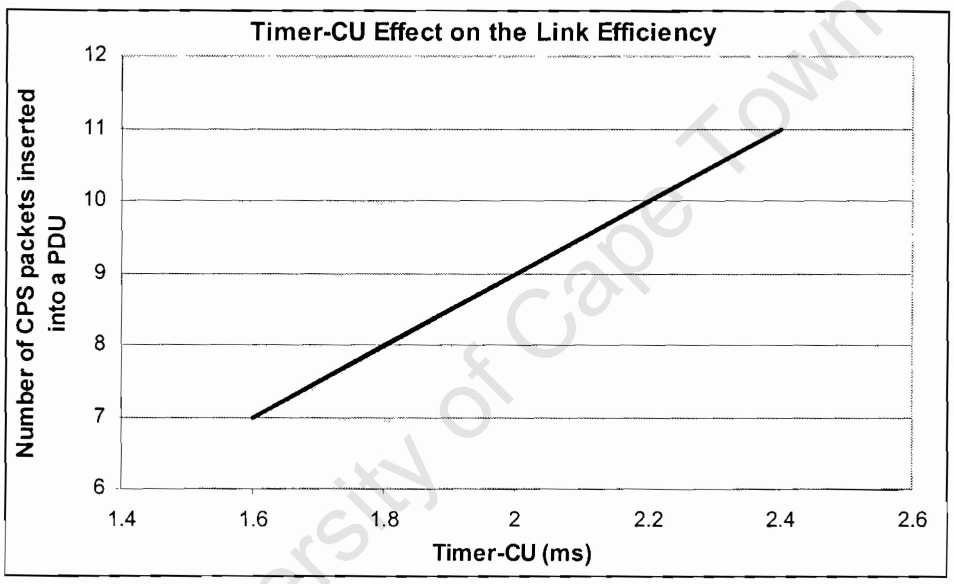


Figure 5-3: Timer-CU effects on the link efficiency

Our framework was only evaluated for voice traffic. But the UTRAN transports a mixture of voice and data (http, ftp, email, etc). AAL2 was proposed for voice, but it is also used for data in the Iub interface [6][47]. Voice and data traffic have different QoS requirements. Voice is very sensitive to both delay and packet loss while data is sensitive to packet loss. At the Node B, voice channels should be scheduled and multiplexed with higher priority than data channels. To assure QoS, it is not only important to allocate a specific bandwidth to a class of service with higher priority but it is advantageous to share bandwidth amongst classes of service. With these classes, the link efficiency can be higher because the PDU size for data can be set high and more CPS packets can be inserted into a PDU, since the Node B-to-RNC delay is 10 ms as discussed in section 2.2.

In an IP-based Iub interface, QoS for different classes of service is mostly provided by DiffServ. DiffServ aggregates classes of service with priority. DiffServ has been proposed in the literature with a new resource reservation scheme to meet the Iub interface QoS requirements [11]. To provide connection-oriented paths to minimise delay, packet loss and ensure packets routing order in the Iub interface, MPLS can be used as discussed in section 3.3. An MPLS edge router tags a label on the IP packet header for fast forwarding within the MPLS domain. MPLS integrated with DiffServ can provide determined paths for the IP packets to traverse within the Iub interface [34][35].

5.4 Number of Users – without QoS

This section determines the maximum number of users that can be supported by the 2 Mbps link before the framework was enhanced to support QoS. Tests were performed to determine the maximum number of users that can be supported by the 2 Mbps link when ATM and IP transport layers were used. During the tests, multiple AAL2 traffic sources generated 20-byte CPS packets every 20 ms and sent to the multiplexer. The CPS packets were packetised into both the IP and ATM PDUs at the Node B and sent to the RNC. The timer-CU value was set to 100 ms so that it does not affect the outgoing PDUs. Our results were compared with the results in the simulations carried out in the literature [7].

5.4.1 Results

Figure 5.4 shows that 248 mobile users were supported by the 2 Mbps link before QoS was introduced. The figure shows that at 25 users, ATM and IP performance in terms of link efficiency were almost the same. But as the number of users increases, IP uses the link more efficiently (about 10 %) than ATM. Increasing the number of users resulted in a faster multiplexing process which increased the efficiency. After 215 users were

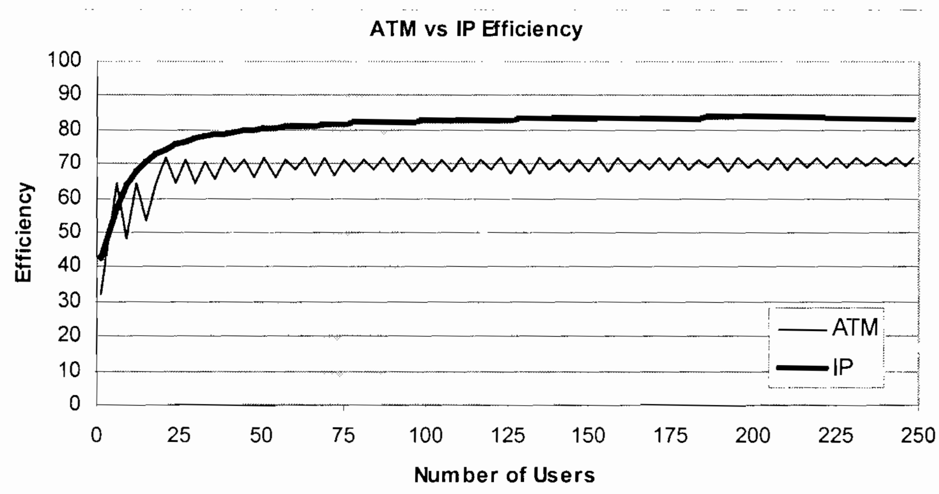


Figure 5-4: Comparison of AAL2/ATM and AAL2/IP Efficiency

supported by the 2 Mbps link, it was observed that CPS packets were dropped before being inserted into a PDU. This was observed when the CPS packet count was analysed from the PDUs logged at the RNC. The loss increased with an increase in the number of users. When 248 users were reached, the multiplexer was accepting the CPS packets every 80.6 (20 ms/248) microseconds, 20 ms is the inter-arrival time of the CPS packet.

5.4.2 Analysis of Results

Figure 5.4 shows that when the number of users is low, a region appears where ATM has higher efficiency than IP. However as the number of users increases, IP uses the link more efficiently than ATM. This is because ATM sent more cells due to the PDUs that were filled quicker while IP sent a number of CPS packets at once. For instance to send ten 20-byte CPS packets, the larger header of IP was only used once as compared to ATM header which was used five times. A bigger portion of the link was used by the headers when ATM was used which resulted in low link efficiency. The loss of the CPS packets, after 215 users were supported, was due to the multiplexer which could not keep up with accepting and inserting CPS packets into PDUs fast enough.

Our test-bed supported 248 users as compared to the simulation results in the literature where 230 users were supported [7]. Our results are not suprising because our AAL2 traffic sources generated a fixed CPS packet (20 bytes) and in the literature, a random

CPS packet size was generated between 15 bytes and 25 bytes. Nevertheless, their results also showed that IP is more efficient than ATM. The ATM graph in figure 5.4 oscillates as the number of users increases. The calculations to explain the oscillations were analysed in section 5.2.2 when 6 CPS packets needed to be sent off.

The performance comparison between ATM and IP may be summarised as follows: When the number of users is low, the efficiencies of ATM and IP are almost the same. But as the number of users increases, IP uses bandwidth more efficiently than ATM. Furthermore, IP can use the link much more efficiently if its header is compressed.

5.5 Number of Users – with QoS

This section determines the maximum number of users that can be supported by the 2 Mbps link after the test-bed was enhanced to support QoS. Our interest is to find the maximum number of users that can be supported by the 2 Mbps link while meeting QoS. Only AAL2/IP was evaluated since our primary aim is to create a framework that uses the link efficiently while meeting QoS. Tests were performed to determine the maximum number of users. We observed how the number of users is affected by the timer-CU at the Node B. Our results were compared with the analytical results in the literature [14].

During the tests, multiple AAL2 traffic sources generated 39-byte CPS packets every 20 ms and IP PDUs were packetised at the Node B. The PDUs were packetised until the timer-CU expired since we were interested in the maximum number of users while meeting 5ms Node B-to-RNC delay. As in section 5.3, the timer-CU values of 2 ms and 3 ms and the Iub interface delays of 3 ms and 2 ms respectively, were investigated.

5.5.1 Results

Figure 5.5 shows that the number of users increases linearly with an increase in the timer-CU value. The maximum users supported were limited by the timer-CU which ensures that the IP PDU being packetised does not wait too long at the Node B. Figure 5.5 shows that a maximum of 116 users were supported by the 2 Mbps link while meeting the 5 ms delay. The figure shows the number of users only up to a maximum of 2 ms timer-CU value where the Iub interface delay was fixed to 3 ms which is a good worst-case approximation for the IP packets to traverse the multi-hop IP interface. The efficiency of 87.8 % was achieved. Table 5.2 shows the number of users that were supported when the 5 ms delay budget was partitioned differently between the Iub interface delay and timer-CU. In the literature, when the timer-CU was set to 2 ms, 120 users were supported [14]. In other research, 125 users were supported on a 1.53 Mbps link [45]. Our results are not in line with the results in the literature possibly because of the following reasons:

- Analytical results rely on assumptions. Simulations are time-independent, i.e. the resource allocation and packet processing time does not depend on the hardware system time constraints such as on Network Interface Cards (NICs).
- Real protocols like Ethernet add overhead for transporting the packets and they are time-dependent. Hardware buffers can add unforeseen delays.

5.5.2 Analysis of Results

When the Node B-to-RNC delay limit was introduced in the test-bed, the number of users supported by the link decreased from 248 to 116. This is because the timer-CU ensures that the IP PDU does not wait too long at the Node B in order to meet the delay limit. Also the CPS packet of 39 bytes was used compared to the CPS packet of 20 bytes. As analysed in sub-section 5.3.2, increasing the timer-CU value supported more than 116 users and hence increased the link efficiency, however this violated the delay limit.

Table 5.2 shows that with a timer-CU value of 3 ms (Iub interface delay of 2ms), 158 users are supported. The link efficiency is higher because 14 CPS packets (566 bytes IP packet) are inserted into a PDU than with 116 users (10 CPS packets per PDU). In real multi-hop IP networks, larger IP packets travel slower than smaller IP packets. Larger IP packets can degrade voice quality when delayed or dropped when traversing the network. The timer-CU has a significant impact on both the link efficiency and 5ms delay limit. To get maximum number of users and link efficiency, the 2 ms timer-CU value should be maximised while ensuring an acceptable 5ms delay limit. With a mixture of traffic, the maximum number of users to be supported will depend on how many users are generating voice, how many users are generating data and their QoS requirements. The

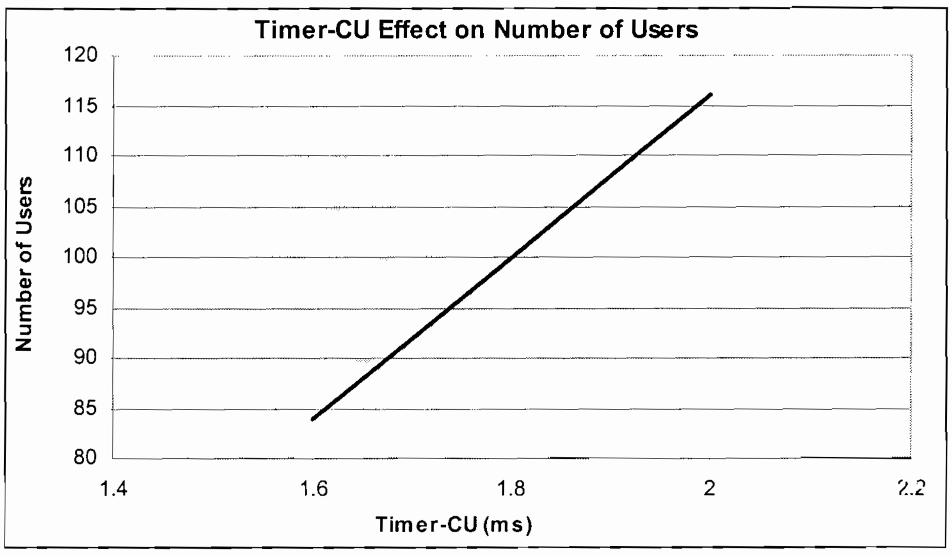


Figure 5-5: Number of users with QoS, 116 users were supported

Timer-CU (in ms)	Iub interface delay (ms)	Max users supported by 2 Mbps link
2.0	3	116
3.0	2	158

Table 5-2: Number of users supported by 2 Mbps link on different 5 ms delay budgets

following sections were evaluated after the framework was enhanced to support the Iub interface QoS constraints. The dummynet pipe was configured to emulate the delay and packet loss as shown in figure 4.5. Jitter was achieved since the pipe delayed packets for 3 ms with 6.6 % variation. The pipe was set to drop packets with the probability of 5 %.

5.6 End-to-end (Node B-to-RNC) Delay

Tests were performed to ensure that the IP packets from the Node B to RNC meet the 5 ms delay limit. This is because voice is very sensitive to the mobile-to-mobile delay. Exceeding the 5 ms delay violates the mobile-to-mobile delay and hence degrades voice quality. Additional delay may be incurred when canceling delay variation at the receiver. The Node B and RNC were synchronised using NTP to measure the delay accurately. In the literature, the timer-CU was rendered redundant [19]. In this study, we use the timer-CU to optimise a trade-off between the link efficiency and 5ms delay.

```
Packet contents:
12380011014816359897231000124000000000005238001101481636009733100012400000000001638001101481636029736100012400000000002838001101481636
0497401000124000000000004238001101481636069747100012400000000003438001101481636089751100012400000000003938001101481636109755100012400000
00000133800110148163612975810001240000000000022380011014816361497631000124000000000002038001101481636169767100012400000000000
End-to-end delay 500944 micro seconds
Packet jitter 35066 micro seconds
Seq Number 124          Lost Packets 10

Packet contents
12380011014816362100801000125000000000005238001101481636230092100012500000000001638001101481636250094100012500000000002838001101481636
2700971000125000000000004238001101481636290103100012500000000003438001101481636310106100012500000000003938001101481636330119100012500000
000001338001101481636350122100012500000000002238001101481636370126100012500000000002038001101481636390131100012500000000000
End-to-end delay 500785 micro seconds      => 500 ms/100 = 5 ms
Packet jitter 25075 micro seconds          => 25 ms/100 = 0.25 ms
Seq Number 125          Lost Packets 10

***************     Note that a series of * was used to indicate that an IP packet was lost (for easily identifying lost in the PDU log-files) at the RNC

Packet contents:
12380011014816366508361000127000000000005238001101481636670847100012700000000001638001101481636690850100012700000000002838001101481636
7108531000127000000000004238001101481636730857100012700000000003438001101481636750860100012700000000003938001101481636770863100012700000
000001338001101481636790867100012700000000002238001101481636810875100012700000000002038001101481636830879100012700000000000
End-to-end 0 micro seconds      -> Note: re-setting end-to-end reference to zero after IP packet number 124 was lost
Packet jitter 0 micro seconds   -> Note: re-setting jitter reference to zero after IP packet number 124 was lost
Seq Number 127          Lost Packets 11

Packet contents.
12380011014816368711921000128000000000005238001101481636891201100012800000000001638001101481636911204100012800000000002838001101481636
9312081000128000000000004238001101481636951212100012800000000003438001101481636971214100012800000000003938001101481636991217100012800000
000001338001101481637011220100012800000000002238001101481637031225100012800000000002038001101481637051233100012800000000000
End-to-end delay 500549 micro seconds
Packet jitter 22647 micro seconds
Seq Number 128          Lost Packets 11
```

Figure 5-6: Logfile showing PDUs received at the RNC

During the tests, 1000 IP packets were sent from the Node B to the RNC. The results were logged into logfiles at the Node B and RNC per run. Figure 5.6 shows a logfile with PDUs received at the RNC. The CPS packet within the dotted line is the received version of the one in figure 4.7 (a). The Node B-to-RNC delay is the difference between the time stamped on the IP packet at the RNC and the time stamped on the first CPS packet inserted into that IP packet at the Node B. If an IP packet was lost in the pipe, the end-to-end delay and jitter references were set to zero as shown in figure 5.6. These were re-determined on the arrival of the second IP packet at the RNC.

5.6.1 Results

Figure 5.7 shows that the Node B-to-RNC delay increases linearly with an increase in the timer-CU value. The figure shows that approximately 5 ms delay was achieved when the timer-CU value was 2.2 ms. It is at this stage when 10 CPS packets were supported. A timer-CU of more than 2.2 ms supported more than 10 CPS packets but the Node B-to-RNC delay was violated (more than 5 ms) as shown in figure 5.7.

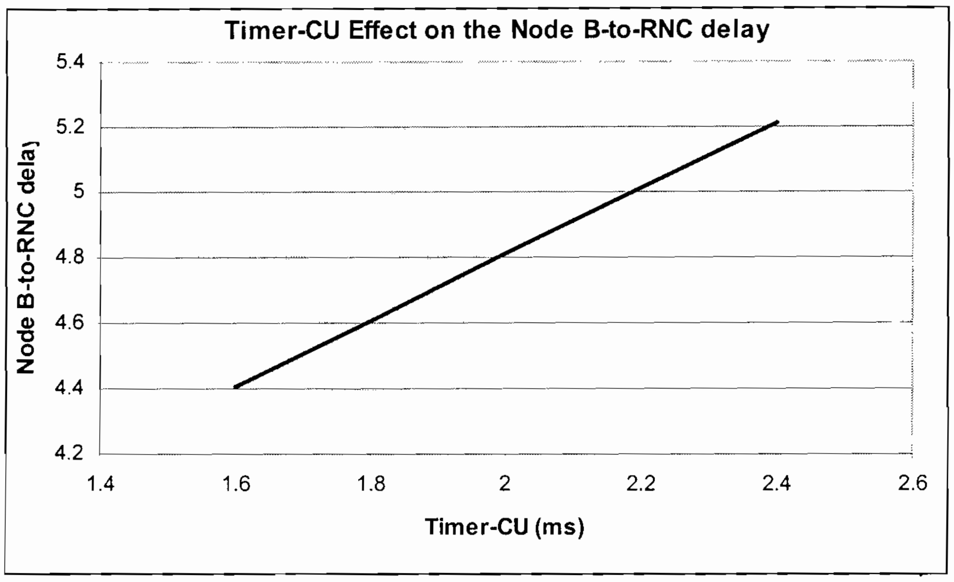


Figure 5-7: Timer-CU effects on the Node B-to-RNC delay

5.6.2 Analysis of Results

Because the timer-CU can violate the 5 ms delay which will violate the mobile-to-mobile delay, it should be handled carefully. Although a timer-CU value of 2.2 ms has provided a delay limit within 5 ms in our test-bed, in the real UTRAN, the timer-CU should not exceed 2 ms to avoid additional delay. To control the 5 ms delay in a real UTRAN, NTP can be used to synchronise the Node B and RNC clocks through the multi-hop IP-based Iub interface. NTP, briefly described in appendix E2, is widely used in the Internet to synchronize computer clocks to national standard time. At the Node B, voice channels are aggregated and sent as a separate class of service from data channels. A mixture of traffic will not affect the 5 ms delay since voice is treated with a higher priority than data.

5.7 Jitter

Tests were performed to determine delay variation (jitter) experienced by the IP packets in the dummynet pipe. It is very important to measure jitter in a network in order to control and configure properly the buffer size to cancel jitter at the receiver. This buffer may incur additional delay or lose packets if it is not configured properly which will degrade voice quality. The length of the buffer is set to the maximum jitter value. The

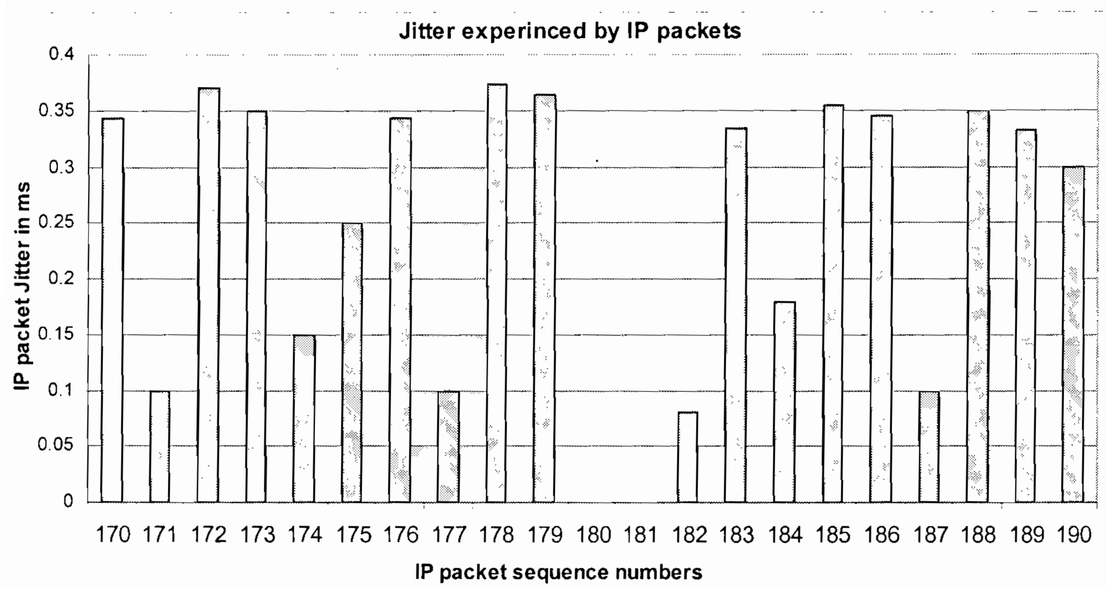


Figure 5-8: IP packets jitter (delay variation) in the dummynet pipe

dummynet pipe was set to 3 ms, but there was a variation of 6.6 %. During the tests, 1000 IP packets were transmitted from the Node B to the RNC. The results were logged into logfiles at the Node B and RNC. Jitter is given by the IP packets delay variation in the pipe. Jitter is the difference between the Iub interface delay of the IP packet received now and the IP packet previously. After an IP packet was lost, jitter was set to zero as shown in figure 5.6 and re-determined after the first two consecutive IP packets were received.

5.7.1 Results

Figure 5.8 shows the delay variations between the IP packets when they were traversing the pipe. The delay variation, in all the logfiles analysed offline, was observed to be less than 0.4 ms. Dummynet delayed the packets randomly between 2.82 and 3.2 ms. It delays packets randomly to emulate the fluctuating IP network conditions. The maximum delay variation can be used to configure the buffer to cancel the jitter at the receiver. Figure 5.7 shows that the jitter value for IP packet number 176 is 0.343 ms. This value is the variation between IP packet number 176 and number 175. Jitter, for the other IP packets, is determined in the same way. The IP packet number 180 was dropped in the pipe to emulate losses. The IP packet number 181 was received successfully, however jitter and end-to-end delay were not determined for 181 since 180 was dropped.

5.7.2 Analysis of Results

Because the buffer to cancel jitter can incur additional delay when canceling jitter at the receiver, it is very important to measure jitter within the multi-hop IP-based Iub interface and configure the buffer. A mixture of voice and data in the Iub interface will not have any special impact since packets are delayed regardless what they are carrying. Enough buffer space must be allocated to cancel jitter to avoid degrading voice quality by either incurring additional delay (larger buffer) or losing packets (smaller buffer).

5.8 Packet Losses

Tests were performed to ensure that the packet loss is within the desired range. The packet loss was set to 5 % on the dummynet pipe which is the maximum loss to get good quality voice as discussed in section 2.5. Dummynet dropped the IP packets randomly to emulate the fluctuating IP network conditions. During the tests, 1 000 IP packets were sent from the Node B to the RNC. The results were logged onto logfiles at the Node B and RNC per run. To determine the packet lost in the pipe, the sequence number of the IP packet received now was subtracted from the sequence number of the previous IP packet. If the difference was greater than one, then an IP packet was (or IP packets were) lost. This difference minus one was added to the packet lost count as shown in figure 5.6.

5.8.1 Results

Figure 5.9 shows the 5 % packet loss obtained after 1000 packets were sent The packet loss for all the logfiles was observed to range between 4.6 % and 5.2 %. The packet loss is independent of the number of users supported since packets were randomly dropped based on the IP packets getting into the pipe. When dummynet was ignored, all the IP packets sent from the Node B were received at the RNC. Figure 5.9 shows that out of 1000 IP packets that were sent from the Node B, 50 IP packets were lost within the pipe. To be exact, after 995 packets sent, 50 packets were lost. This packet loss falls within the expected loss ratio which is 5 % maximum for good quality voice. The packet loss was observed to be lower when smaller IP packet size than when a larger IP packet was used.

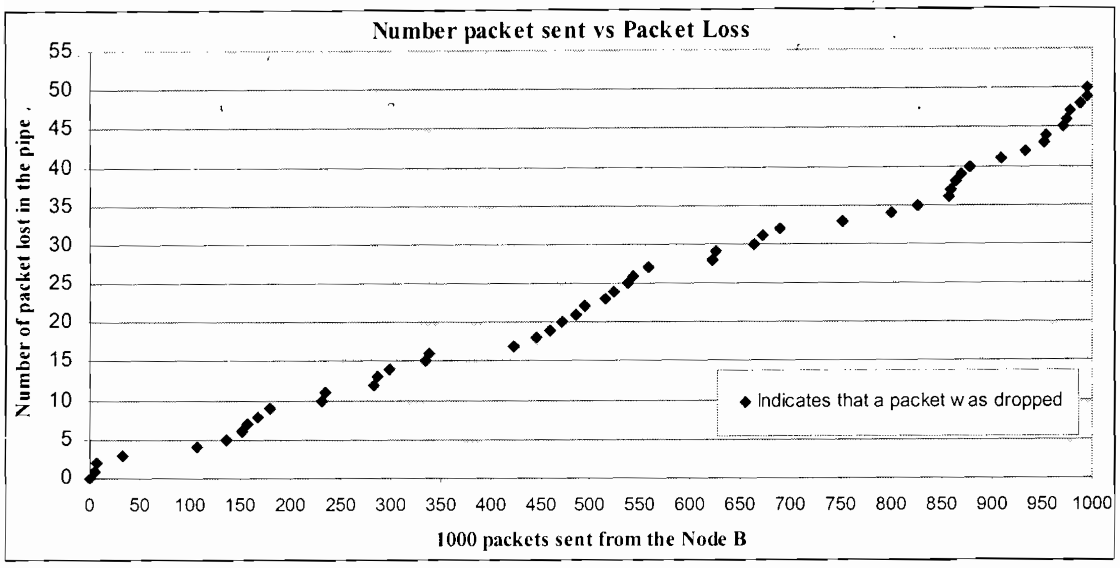


Figure 5-9: Number of packets lost in the pipe vs packets sent from the Node B

5.8.2 Analysis of Results

Because the packet loss can degrade the voice quality, it should be ensured that the mobile-to-mobile packet loss does not exceed 5 %. Within a network, the loss of the IP packets is independent of the number of mobile users supported because the network treats the IP packet, it does not treat the packet content. When a larger IP packet (566 byte) is used, the packet loss is higher than when a smaller size (410 bytes) is used. This is because larger IP packets take longer to traverse a network, where sometimes their time to live expire before they exit the network and hence they are dropped. The packet loss should be kept within an acceptable range to get good quality voice at the receiver.

5.9 Summary

This chapter has presented the experimental results carried out in our study to create an AAL2/IP framework for the Iub interface. The framework was evaluated before and after it was enhanced to support QoS. Table 5.3 summarises the tests performed on the framework. Before it was enhanced, ATM and IP were compared in terms of link efficiency. The results showed that ATM uses the link efficiently when the number of users is low. But as the number of users increases, IP uses the link more efficiently than

Before framework enhanced	After framework enhanced
No QoS supported (No timer-CU)	QoS were supported (timer-CU at Node B)
20-byte CPS packet (248 users)	39-byte CPS packet (see table 5.2 for users)
No dummynet	Dummynet was used
AAL2/ATM and AAL2/IP compared	Only AAL2/IP evaluated

Table 5-3: Framework set-up for a series of tests performed

ATM. Although AAL2/ATM guarantees QoS in the Iub interface and IP still does not, AAL2/IP is a more suitable transport layer for a mixture of traffic in the Iub interface. Also the tremendous growth of the mobile users expected in the UMTS networks calls for a more efficient transport layer in the Iub interface.

After enhancing the framework to support QoS, the timer-CU was found to have a significant impact on both the link efficiency and Node B-to-RNC delay limit. As shown in table 5.4 and figure 5.10, the timer-CU impacts the link efficiency because if it elapses early, then few CPS packets are packetised into a PDU and hence results in low link efficiency. Alternatively, if it elapses too late, the link is used more efficiently but the 5 ms delay limit is violated. This shows that the timer-CU is a very critical Iub interface QoS parameter in optimising the trade-off between the link efficiency and the 5ms delay. Note that number of users and end-to-end delay plotted in figure 5.10 have the same scale

When AAL2/IP is used to transport traffic in the Iub interface, a timer-CU should be used to ensure that the IP payload (PDU) does not wait too long when being packetised at the Node B. This is because IP has a variable PDU size up to 1500 bytes, which can use the link efficiently, but can incur additional delay if the timer-CU is not used. A maximum timer-CU value of 2 ms gives a good trade-off between the efficiency and the 5 ms delay limit since the IP packets traverse a multi-hop IP-based Iub interface. The 2 ms timer-CU should expire rather than the PDU getting full. At 2 ms, a maximum IP PDU size of 351 bytes (9 CPS payloads) was achieved for voice. The maximum number of users that were supported when the timer-CU was set to 2 ms and 3 ms are shown in table 5.2.

Jitter should be controlled to avoid additional delay at the receiver. The packet loss should also be controlled to avoid degrading voice and data quality. In a real IP-based Iub

Timer-CU (in ms)	Number of CPS packets per PDU	Average end-to-end delay (in ms)
1.6	7	4.40811
1.8	8	4.60848
2.0	9	4.81292
2.2	**10**	**5.05109**
2.4	11	5.21121

Table 5-4: Timer-CU effects on both link efficiency and end-to-end delay

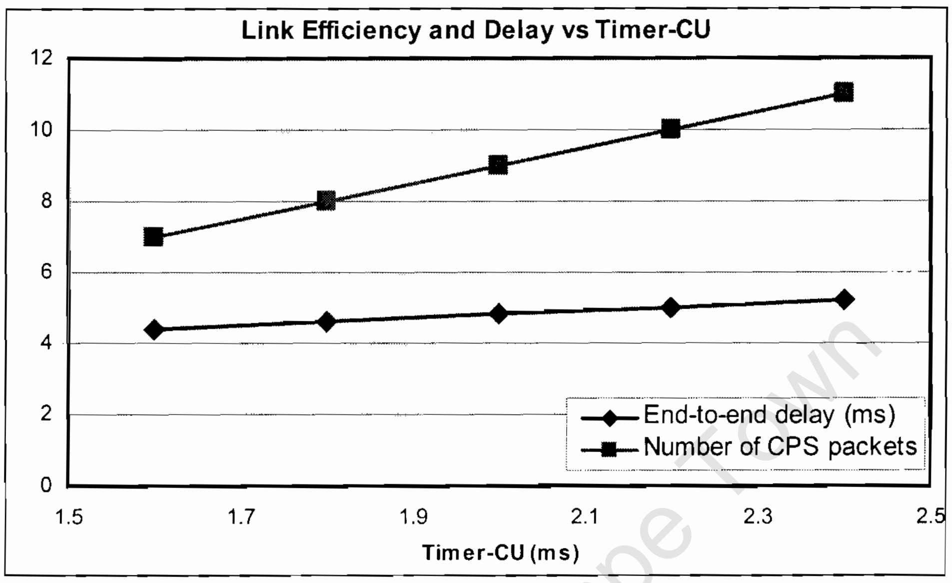


Figure 5-10: Timer-CU effects on the link efficiency and 5 ms Node B-to-RNC delay.

interface, it may not be easy to control jitter and packet loss due to the IP packets taking undetermined paths to the RNC and hence results in IP packets arriving out of order or getting lost. This is also due to the fluctuating IP network conditions (buffering, congestion, etc). Integrating the DiffServ approach into the MPLS system will provide an Iub interface with determined paths to the RNC while offering different service classes. The paths will deliver the packets in order at the RNC and minimise packet losses.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

This study focused on designing the AAL2/IP system (user and control plane) for the Iub interface and implementing the user plane framework as discussed in chapters 3 and 4. The performance of the AAL2/IP framework was evaluated. The framework results, presented in chapter 5, were compared with the results obtained in the literature and conclusively IP transports traffic more efficient than ATM.

In this study, we compared the efficiencies of both ATM and IP when transporting AAL2 voice traffic in the Iub interface. Our results showed that the efficiencies for ATM and IP were almost the same for a low number of users. However as the traffic increased with more users introduced, IP used the link more efficiently than ATM. This is because a bigger portion of the link bandwidth was consumed by the repetition of the ATM cell headers and the padding applied when the cells were not full and hence resulted in low link efficiency.

When the Iub interface QoS constraints were introduced, our results showed that IP uses the link efficiently as the amount of traffic increased while meeting the 5 ms delay, jitter and packet loss limits. With a timer-CU value of 2 ms, a maximum number of 116 voice users was supported by the 2 Mbps link and a maximum of 9 CPS packets per PDU while providing an acceptable delay limit within 5 ms. We conclude that when using IP to transporting voice traffic in the Iub interface, a maximum timer-CU of 2 ms will provide a good trade-off between the link efficiency and 5 ms Node B-to-RNC delay limit. The timer-CU ensures that the IP PDU does not wait too long at the Node B and to achieve a good trade-off, it should expire rather than the IP PDU getting full.

ATM has occupied the Iub interface for a long time because it guarantees QoS requirements while IP does not. Although ATM guarantees QoS, it lacks flexibility and scalability, it is complex to maintain and uses bandwidth inefficiently as the amount of traffic increases. Our research showed that IP is more flexible in terms of packet size and scalable as the amount of traffic to be transported increases in the Iub interface. Furthermore IP, unlike ATM, has become very popular in telecoms networks worldwide since it resides in both the network and the mobile devices such as laptops and PDAs.

This popularity makes IP a truly future end-to-end protocol to transport traffic. Furthermore, the tremendous growth of the mobile users in UMTS pushes for a more efficient transport layer in the Iub interface. Hence we conclude that the 3GPP has made a very good decision to specify IP in the Iub interface. It is a good decision also because of the recent development of protocols such as MPLS and DiffServ to provide QoS in the IP networks. This development has given a high hope in having IP replacing ATM in the Iub interface.

After the 3GPP specified IP in the Iub interface, extensive research has investigated ways to replace ATM with IP. Other research in the literature proposed to retain AAL2 and transport over IP. However it was decided that for more than 248 channels, AAL2 will be transported over UDP/IP in the Iub interface. This introduces substantial overhead which consumes a lot of bandwidth. Also both the UDP and AAL2 are transport layers with respect to their underlying layers. The AAL2 layer carries the traffic and supports a maximum of 248 channels per path and uses bandwidth efficiently by using a small header to multiplex channels. UDP is not a traffic carrier and depends on higher layers such as RTP to carry voice which introduces a lot of overhead to multiplex channels.

AAL2 can support more than 248 users. But if more than 248 users are required, then another AAL2 path can be created instead of using UDP as suggested in the literature. AAL2 uses bandwidth more efficiently when multiplexing low bit-rate, variable size traffic such voice. Hence we conclude that to support more than 248 users, AAL2 should not be laid over UDP/IP but always directly over IP. AAL2 was initially proposed for voice traffic, but it has been used for data traffic as well in the Iub interface.

Since the Iub interface transports a mixture of traffic, AAL2/IP will even be more efficient when transporting data traffic which is not sensitive to delay. Data traffic such as email can be transported much more efficiently since the 1500-byte maximum transfer unit (MTU) can be used once with a 20-byte IP header. This MTU can use the link more efficiently but can degrade the quality of data if it is lost in the interface since data is very sensitive to packet losses. One culprit for losing packets within an IP network is the use of undetermined paths to the destination. But with the use of MPLS in the Iub interface, packet loss can minimised since the packets will follow determined paths. Also, the packets delivery order will be guaranteed.

With the mixture of traffic in the Iub interface, each class of service should be multiplexed separately with its own priority and timer-CU value at the Node B according to the service' QoS requirements. For instance voice channels have a limit of 5 ms and data has a limit of 10 ms within the Iub interface. DiffServ can be used to provide different services by marking and classifying the IP packets according to their QoS requirements at the edge router. To assure QoS, we conclude that a certain bandwidth should be allocated for voice services since it has higher priority than data, but it is a very good practice to use that bandwidth to transport data during the periods that some voice sources are not sending traffic (OFF periods).

Since both MPLS and DiffServ provide QoS in the IP-based Iub interface, they should be integrated to allow fast forwarding of the IP packets on determined paths while ensuring QoS requirements to different prioritised services. We therefore conclude that because of these recent developments to provide QoS in IP networks, the efficient IP protocol can now become a successful transport in the Iub interface.

6.2 Recommendations

As a result of the scope, limitations, findings and conclusions of our study, the following recommendations and future work on the AAL2/IP framework are made:

- During the framework implementation, we did not implement any of the control plane functions, i.e. neither signalling nor admission control units. We assumed that the mobile channels were admitted and created beforehand where admission depended on the link bandwidth utilisation. Further research is therefore required to implement the admission control to admit channels and ensure that these channels do not interfere with the existing once. The signaling unit also needs to be implemented to create, maintain and tear down the channels. This unit (signaling), after tearing down a channel, should notify the admission control so that the resource for the torn down channel can be re-used by new channels.

- When implementing the AAL2/IP user plane framework, we only focused on the uplink, from the Node B to RNC, due to our primary aims outlined in section 1.3. However, when two mobile users are communicating, i.e. sending voice traffic to each other, the packets will traverse the Iub interface in both ways, uplink and downlink. For this reason, further investigation needs to be undertaken to enhance the framework to support the downlink (from RNC to Node B) as well.

- In our framework evaluations, we evaluated only voice as our Iub interface traffic. However, the real IP-based Iub interface does not only transport voice traffic. It transports other applications such as real-time video and data (Internet traffic such as email, hyper text and file transfer). Hence we recommend that the AAL2/IP framework be enhanced to transport a mixture of applications.

- When the framework is enhanced to support a mixture of traffic, a priority scheduling unit is required at the multiplexer unit at the Node B. For instance, voice channels have higher priority than data channels. When a channel is created, its QoS requirements should be placed on the scheduling unit. As discussed in section 3.4, the scheduler should ensure that it does not starve channels with lower priority. It is therefore recommended that a scheduling unit that will handle different classes of service with different QoS be implemented at the Node B. As the interface transports a mixture of traffic, it is strongly recommended that DiffServ be implemented to provide QoS in the Iub interface. DiffServ will classify this traffic into classes of service and forward packets from each class with different priority.

- In a multi-hop IP-based Iub interface, the IP packets use undetermined routes to traverse the interface to get to the RNC. This delays and losses the IP packets, and makes them to arrive out of order at the RNC. Also the more hops the packets traverse, the more delay they incur. It is therefore required that further research be done to implement MPLS in the Iub interface. MPLS will provide QoS since it is a connection-oriented based technology. The IP packets will arrive in order at the RNC. An MPLS edge router will tag a fast forwarding label on the IP packet header as the IP packets enter the MPLS domain. This label identifies the path the IP packet will use to traverse the domain. The IP packets' QoS requirements are mapped from the packet header onto the label. The Node B-to-RNC delay will be minimised since the IP packets are forwarded based on the label instead of performing an IP address lookup at each node within the interface.

- Both MPLS and DiffServ address the same issue within the IP-based Iub interface which is QoS by classifying packets at the interface edge routers. DiffServ marks and classifies packets according to their Per Hop Behavior (PHB). MPLS creates paths and tags a fast forwarding label on the packet headers. MPLS integrated with DiffServ (MPLS/DiffServ) will provide the IP packets with different fast forwarding treatments according to the classification and marking performed at the edge routers. The core routers will only classify and fast forward the IP packets which will provide scalability within the interface. Therefore further research is required to design an Iub interface based MPLS/DiffServ approach. For more details, on the research done in the literature on MPLS/DiffServ integration, refer to appendix C.3.

References

[1] M Mouly and M. B. Pautet, 'The GSM system for mobile communications', Mouly and Pautet, Palaiseau, France, 1992

[2] K.W. Richardson, "UMTS", Electronics and Communication Engineering Journal, vol. 12, no. 3, pp. 93 - 100, June 2000

[3] A. Juttner, A. Orban, Z. Fiala: "Two New Algorithms for UMTS Access Network Topology Design", European Journal of Operational Research, 2001

[4] 3GPP Technical Specification Group RAN, TS 25.401 V3.0.0 (1999-10), Release 99

[5] Mobile Wireless Internet Forum IP in the RAN as a Transport in 3rd Generation Mobile Systems, Technical Report MTR-006, Release v2.0.0, June 2001

[6] M Menth, M Schmid, H Heiß, T Reim, "MEDF - A Simple Scheduling Algorithm for Two Real-Time Transport Service Classes with Application in UTRAN", IEEE INFOCOM 2003, vol. 22 no. 1, March 2003, pp. 1116-1122

[7] W Brunnbauer and G Cichon: "Bringing two worlds together: AAL2 over IP for Radio Access Networks", IEEE GLOBECOM 2001, Nov 2001, pp. 2606 – 2610

[8] A-F. Canton, S. Tohme, D Zeghlache, T Chad, "Performance Analysis of AAL2/ATM in UMTS Radio Access Network", IEEE PIMRC, Lisbonne, 2002

[9] H. Saito, "Performance Evaluation and Dimensioning for AAL2 CLAD", IEEE INFOCOM '99, New York, 1999

[10] K Venken, IG Vinagre and J de Vriendt: "Analysis of the Evolution to an IP-based UMTS Terrestrial Radio Access Network" IEEE Wireless Com, October 2003

[11] G. Heijenk, G. Karagiannis, V. Rexhepi, and L. Westberg, "Diffserv Resource Management in IP-based Radio Access Networks," in Proceedings of 4th International on Wireless Communications (WPMC'01), Denmark, Sept 2001

[12] 3GPP TR 25.853 V3.1.0 (2001-03) and TR 25.853 V4.0.0 (2001-03), Technical Specification Group (TSG) RAN, Delay Budget within the Access Stratum

[13] A. Samhat, T. Chahed, G. Hébuterne, "Transport in UMTS Radio Access Network: IP versus AAL2/ATM", 8th International Conference on CIC, Seoul, October 2003

[14] A. Samhat, T. Chahed, "Transport in UMTS Radio Access Network: Analytical Study and Empirical Validation", IEEE Communications Society 2004

[15] Overview of 3GPP Release 5: Summary of all Release 5 Features, V 0.10, Sep 2003

[16] S Dixit, Y Guo and Z Antoniou: "Resource Management and Quality of Service in Third- Generation Wireless Networks", IEEE Commag, February 2001, pp125-133

[17] L. Bos and S. Leroy, "Toward an All-IP-Based UMTS System Architecture", IEEE Network, Jan./Feb. 2001

[18] D. W. Petr, R. R. Vatte, P. Sampath, and Y. Lu, "Efficiency of AAL2 for voice transport: simulation comparison with AAL1 and AAL5," IEEE ICC, June 1999

[19] S Shepstone, "AAL2 Switching Node to Support Voice Services in 3rd and 4th Generation Networks", MSc Thesis, University of Cape Town, August 2002

[20] C Liu, S Munir, R Jain, "Packing Density of Voice Trunking using AAL2", Proceedings IEEE GLOBECOM December 1999, Brazil, Vol 1 (B), pp611 – 615

[21] Y. Koucheryavy, D. Moltchanov, "On Quality of Service and Performance Evaluation in 4G All-IP Networks", ANWIRE Workshop, April 2003, UK

[22] ITU-T, "B-ISDN ATM Adaptation Layer specification: AAL2" ITU-T, Nov 2000

[23] A. B. García, M. Álvarez-Campana, E. Vázquez, "Simulation of Quality of Service Mechanisms in the UMTS Terrestrial Radio Access Network", IEEE 2002

[24] C Saraydar, S. Abraham, M Chuah and A Sampath: "Impact of rate control on the capacity of an Iub link: Multiple service case", Bell Labs, IEEE 2003

[25] R Cuny, A Lakaniemi, "VoIP in 3G Networks: An End-to-End Quality of Service Analysis", the 57th IEEE Semiannual Vehicular Technology Conference Final Program April 22 – 25, International Convention Center Jeju, Korea - IEEE 2003

[26] H Allali, G Heijenk, "Traffic Characterisation for a UMTS Radio Access Network", Proceedings 4th International Workshop on Comm, Sweden, September, 2002

[27] P Andersen, "Feasibility study of VoIP in 3GPP UMTS Release 5 mobile networks interworking with fixed networks", MSc Thesis, Agder University, May 2003

[28] M.J. Karam, F.A. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic Over the Internet", IEEE INFOCOM 2001

[29] K. El-Khathib, G. Luo, G. Bochmann, and F. Pinjiang, "Multiplexing scheme for RTP flows between access routers", draft-ietf-avt-multiplexing-rtp-00.txt

[30] "Bandwidth requirements for Voice over IP transmission", http://www.erlang.com/bandwidth.html

[31] S Floyd, “Comments on Measurement-based Admissions Control for Controlled-Load Services”, Draft, Lawrence Berkeley Laboratory, July 1996

[32] Blake S, “An Architecture for Differentiated Services”, RFC 2475, Dec 1998

[33] V. Jacobson, K. Nichols, “Expedited Forwarding PHB”, RFC 2598, June 1999

[34] F Le Faucheur: “MPLS Support of Differentiated Services” RFC 3270, May 2002

[35] P Cebrian, “Support for Differentiated Services in Multi-Protocol Label Switching”, MSc Thesis, Kungliga Tekniska Högskolan (KTH) University, October 1999

[36] T Bu, MC Chan, R Ramjee, “Connectivity, Performance and Resiliency of IP-Based CDMA Radio Access Networks” IEEE INFOCOM 2004

[37] Y Chen, “Soft Handover Issues in Radio Resource Management for 3G WCDMA Networks” PhD Thesis, University of London, September 2003

[38] “The Network Timing Protocol (NTP) Distribution”, http:/www.eecis.udel.edu/~ntp/ntp_spool/html/, March 2004

[39] “MGEN, the multiple generator toolset”, http://mgen.pf.itd.nrl.navy.mil/

[40] L. Rizzo, DummyNet: a simple approach to the evaluation of network protocols, ACM Computer Communication Review, Vol 27, No 1, pages 31-41, January 1997

[41] “Dummynet”, http://info.iet.unipi.it/~luigi/ip_dummynet/

[42] “The Network Simulator - ns-2”, http://www.isi.edu/nsnam/ns/

[43] M. Allman, A. Caldwell, and S. Ostermann, “ONE: The Ohio Network Emulator. Technical Report TR-19972”, Ohio University, Aug. 1997.

[44] BVW Irwin, “Bandwidth Management and Monitoring for IP Network Traffic: An Investigation”, MSc Thesis, Rhodes University, January 2001

[45] S Abraham, A Sampath, C Saraydar, M Chuah, “Effect of Timing Adjust Algorithms on Iub Link Capacity for Voice Traffic in W-CDMA Systems”, Bell Labs, IEEE 2003

[46] S Malomsoky, S Rácz and S Nádas: “Connection Admission Control in UMTS Radio Access Networks”, Computer Com, Elsevier Science, 2002

[47] JH Chung, YH Kwon, KH Cho; DK Sung; OH Jang, “Performance Evaluation of an AAL2 Link Transmission Scheme for Voice and Data Packets in BS-BSC Links”, IEEE Vehicular Technology Conference, 2000, Vol.4, pp 1610 – 1614

Appendices

Appendix A: AAL2 and UDP Headers Description

The 3-byte AAL2 header consists of the following fields:

- Channel Identifier (CID – 8 bits): Identifies the AAL2 channel (CPS user), which is bidirectional, i.e. same CID is used for both directions. The value 0 is not used and values 1 to 7 are reserved for signalling purposes.
- Length Indicator (LI – 6 bits): Indicates the length of the CPS payload. The CPS packets have variable lengths; hence the LI field can specify a length of 1 byte to 64 byte. The LI field is always one less than the size of the CPS packet payload.
- User-to-User indication (UUI – 5 bit): Conveys specific information transparently between the two CPS users, sender and receiver through the protocol stack, where it is passed up to the application. Applications may use UUI field as they see fit as the structure and contents of UUI is not specified in this recommendation, for instance UUI may be used to convey the codec to be applied at the receiver to uncompress voice packets.
- Header Error Correction (HEC – 5 bit): Detects any errors in the payload.

The 8-byte UDP header consists of four fields, each with length of 2 bytes:

- **Source Port**: UDP packets from a client use this field, the source port, as service access point (SAP) to indicate the session on the local client that originated the UDP packet. UDP packets from a server use this field to carry the server SAP.
- **Destination Port:** UDP packets from a client use this field, destination field, as a SAP to indicate the service required from the remote server. UDP packets from a server use this field to carry the client SAP.
- **UDP Length:** Indicates that number of bytes comprising the combined UDP header and the payload data.
- **UDP Checksum:** Verifies that the end-to-end has not been corrupted by routers in the network or by other processing in the end system.

Appendix B: The CPS transmitter and receiver

The CPS transmitter contains four states, i.e. IDLE, PART, FULL and SEND, as shown in figure C.1 [19]. If the transmitter has not begun creating a PDU, it will remain in the IDLE state for a CPS packet to arrive. When a CPS packet finally arrives, a timer-CU is started and the transmitter begins to create a PDU. When the transmitter has finished inserting the first CPS packet into a PDU, it will jump to state PART, until either the timer-CU expires or the PDU is full as shown in figure 4.4.

Delay can degrade voice quality, where packetisation delay of voice is mostly the culprit. The packetisation delay is the culprit especially with IP because of the IP packets being larger. Therefore, once the IP PDU creation process starts, the PDU should not wait longer than the timer-CU value for more CPS packets to arrive. Waiting longer will violate the delay limit of the first CPS packet that has already been inserted into the PDU. The timer-CU value can highly affect the bandwidth utilisation. When the value reaches its limit, the state machine jumps to state SEND, where it will wait for permission from management layer before sending the PDU.

The CPS receiver resides at the receiver and contains only one state, IDLE, which is shown in figure C.2 [19]. After processing each PDU and extracting the CPS packets, the receiver will return to IDLE state to wait for the following PDU to arrive. Upon receiving the PDU, the CPS packets are extracted and their sequence numbers are checked. If enough bytes are extracted to match the LI field in the CPS packet header, the CPS packet is considered full and sent to the application layer.

If the end of the PDU is reached before enough bytes are extracted, the remaining data can be expected in the following PDU. There is no timer-CU needed at the receiver, its operation is purely asynchronous [19][22]. A state of this nature has already been used in simulation models [18]. A transmitter and receiver on Linux operating system have already been successfully implemented on AAL2 [19].

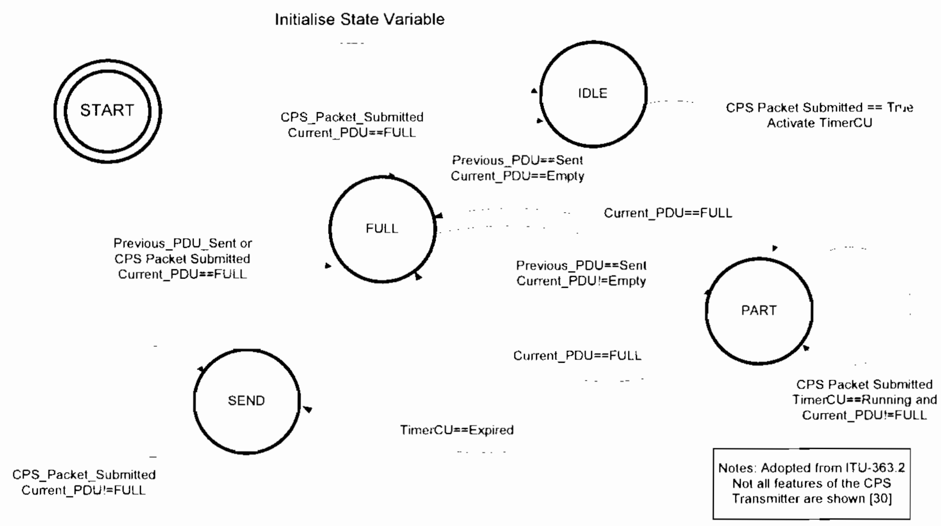


Figure C.1: CPS Transmitter State Machine

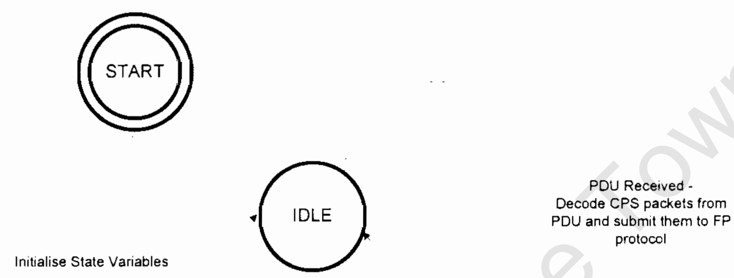


Figure C.2: CPS Receiver State Machine

In the literature it has been shown that the more users, the less critical is the timer-CU operation to assemble a PDU [19]. For less than 47 users, the timer-CU was critical to preserve the traffic quality of each channel on the AAL2/ATM network. But for more than 47 users, the timer-CU was not critical since ATM cells were no longer dependent on the timer-CU to be sent off; hence high repetition of ATM headers as users increase. For this reason, AAL2 was proposed to be placed on top of IP. In this case, the timer-CU value can be maximised, while ensuring Node B-to-RNC, to allow more CPS packets to be sent under one IP header. This will increase the link efficiency. In the IP-based Iub interface, we have investigated the maximum IP payload (PDU) and the timer-CU value to guarantee QoS.

Appendix C: DiffServ and MPLS technologies

C1: Differentiated Services (DiffServ)

One of the today's most pressing challenges in designing IP networks is the providing QoS. The current Internet operates in a best-effort manner, which is considered insufficient for QoS demanding applications such as voice, video conferencing and mission-critical transactions. In addition, as the Internet migrates to commercial enterprise, providing reliable QoS is a crucial factor in influencing the customer's propensity to pay for network services. One approach of providing QoS is DiffServ [32]. DiffServ provides a framework in which QoS mechanisms can be developed such that "service differentiation" can be achieved for different IP service classes.

In practice, the basic idea is to classify IP streams into service classes and forward packets from each class differently. DiffServ is an IETF-driven paradigm for providing scalable QoS across an IP based network. The cornerstone of DiffServ is the usage of the Type of Service (TOS) byte field in the IP (v4) header. The basic concept is relatively straightforward: edge routers mark the TOS field with a particular Per Hop Behaviour (PHB) and aggregate the packets according to their forwarding resources. Resources such as buffer space, bandwidth, and scheduling policy, that are pre-allocated before a channel sends traffic. Classifying, marking and policing are only needed at the edge routers of DiffServ while the routers inside the DiffServ need only to perform PHB classification, hence scalability within the DiffServ domain. The routers inside the domain forward the packets in a manner that provides service differentiation. However the scalable DiffServ is considered to provide an unacceptable level of QoS because it currently specifies only PHBs and does not have resource management scheme yet [18].

The problem of QoS is that it poses a number of functionalities both in the application and network layers. In the application layer, it poses these questions: what application should receive QoS, what kind of QoS, how much QoS. In the network layer: what scheduling algorithm should be used, how will packets be classified. IntServ put all these functionalities in both hosts and routers. DiffServ put a reduced set of QoS functionalities in the edge routers only, hence scalable within the network. After realising the QoS of the DiffServ, research has proposed to extend the support of QoS in DiffServ by developing resource management. The resource management schemes, i.e. the admission control and resource reservation, to provide acceptable QoS in the DiffServ in UTRAN are proposed in [11]. Therefore the Diffserv mechanism is a suitable candidate to provide QoS within the Iub interface since it transports a mixture of traffic, real and non-real time.

C2: Multi-protocol Label Switching (MPLS)

Traditional IP networks are typically based on a connection-less packet-switching technology. That is individual packets are routed separately and independently on a hop-by-hop basis within the network and there is no guarantee of timely delivery of packets and delivery order (best-effort service). However, the rapid growth of Internet recently and the strong interest to migrate real-time and multimedia services towards IP, requires technologies that are able to provide users with QoS and network operators with more dynamic and flexible resource utilisation. In this scenario, traffic management plays a key role and has attracted a lot of interest to design protocols to provide QoS.

A number of traffic management protocols have been designed to improve the performance of IP networks and to provide different levels of QoS to different service users. This support of multiple traffic classes with different QoS requirements poses new challenges in the network design especially in the 3G access network where radio and transmission resources are scarce. MPLS, originally developed to improve the forwarding speed of the IP routers, is now emerging as a crucial underlying technology for IP transport to offer QoS for IP networks. It is a connection-oriented technology with traffic engineering techniques. It has raised expectations that QoS constraints applications such as mobile traffic can be integrated into IP networks of the future cellular networks.

MPLS offers an efficient and robust solution to the problems of over-utilized paths in transit networks by establishing an explicitly-routed label switched path (LSP) to handle a large volume of traffic that takes a particular route. The focus of QoS support in the MPLS domain is scalability, which is achieved by flow aggregation that ensures individual end-to-end QoS guarantees using a simple MPLS label tagged to the IP packet as it enters the MPLS domain. MPLS is a promising IP fast forwarding technique that combines the performance of layer 2 networks while maintaining the wide connectivity of IP (layer 3) addressing. The basic concept is to append a label to each incoming packet at the edge router of the domain. These packets can be forwarded along a LSP at faster rates via the labels instead of performing an IP address lookup at each router.

The label appended to each packet identifies uniquely the path the IP packet will traverse through the MPLS domain. Classification and mapping of packets onto virtual paths can be performed based on the packet's layer-3 and layer-4 information that is accessible to the edge router. Within the domain, a packet is forwarded based exclusively on the contents of the label. With the separation of layer-3 routing, this provides the foundation for the deployment of advanced traffic engineering features (e.g. explicit routing). It also allows enhanced security because of the following:

- The full IP packet can be encrypted since all the information required to forward the packet is in the MPLS label. Thus, the only part of the packet which remains unencrypted is the label.
- The label has only local significance, it does not convey any information on the packet's final destination.
- Other forwarding parameters (e.g. QoS parameters) can be bound to the label, without having to be explicitly coded in the packet's headers and therefore invisible to packet sniffers.

C3: Support for DiffServ in MPLS domains

MPLS and DiffServ provide QoS in IP networks. Integrating the DiffServ approach into the MPLS system will produce an Iub interface able to provide paths to the same destination offering different service classes. MPLS integrated with DiffServ will provide packets with different fast forwarding treatments according to the classification and marking performed by the edge router. But the problem with the integration of MPLS and

DiffServ is that the DiffServ routers look at the TOS field in the IP header (layer-3) to decide which forwarding treatment must be given to a packet. The MPLS core routers do not look at the IP header to forward the packets. Forwarding is only based on the MPLS label. This means that the MPLS core routers cannot make an independent decision on which scheduling treatment must be given to each packet.

Therefore mapping of layer-3 information to the MPLS labels must be done at the MPLS edge router by allocating different paths to different forwarding treatments. With this done at the edge routers, MPLS core routers can be signalled with a specific path that maps to a certain destination and forwarding treatment. When a packet arrives, the core routers will look at the MPLS label and perform a lookup in their routing table. The core routers will determine through which path the packet is to be forwarded and which scheduling treatment it must receive. This information will be recorded in the routing tables when the path is established. The core routers will only perform PHB classification and fast forward the packets which will provide scalability within the Iub interface.

The research in the literature concluded that Diffserv is a good approach to provide QoS within MPLS domains [35]. This is because packets are treated based on a PHB and aggregate forwarding resources, such as buffer space, bandwidth, and scheduling policy. These resources are pre-allocated in the routers for each class of service. The functions such as classification, marking and policing are only needed at the edge routers while the core routers the domain need only to have PHB classification, hence scalability in the domain. This research investigated the issues such as how the preferential paths should be managed in an MPLS integrated with DiffServ. That is, when and why the paths must be established and torn down, and how to co-ordinate this with users' requirements without being wasteful of resources.

Appendix D: SDL Specification

The various symbols that are used in figure 3.2 are explained in table E.1. Note that the symbols used in figure 3.2 form a small subset of the symbols available in SDL system.

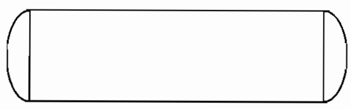
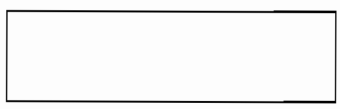
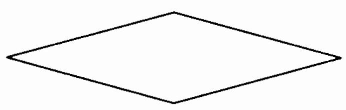
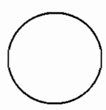
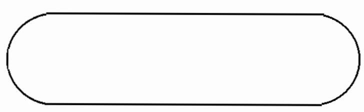
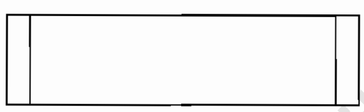
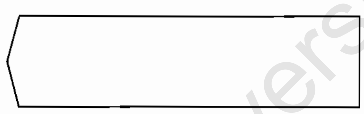
SDL Symbol	Meaning
	Start
	Action
	Decision
	Connector
	State
	Procedure
	User Indication/Output Signal
	User Request/Input Signal

Table E.1: SDL Symbols used in figure 3.2 and their Meaning

Appendix E: Dummynet and NTP

Appendix E1: Dummynet pipe and queue configuration

The following ***ipfw*** commands control dummynet pipes:

- ***ipfw pipe NN config*** - This command is used to create or configure a pipe. i*pfw* is the ip firewall. NN is a numeric identifier (between 1 and 65535) of the pipe.

- ***ipfw [-s field] pipe [NN] show*** - This command shows the parameters of the pipe. If the pipe is a dynamic one, then all the dynamic pipes created from this one are listed. The list can be very long. The -s option allows one to sort the listing on one of the four counters associated to the pipe.

- ***ipfw pipe NN delete*** – This command destroys a single pipe.

- ***ipfw pipe flush*** – This command destroys all pipes.

The following parameters can be configured for a pipe, adding the command ***pipe config***:

- **Bandwidth:** *bw NN unit* - NN is the bandwidth assigned to the pipe, unit (which must follow the number with no intervening spaces) can be any of bit/s Kbit/s Mbit/s Byte/s KByte/s MByte/s. A bandwidth of 0 results in no bandwidth limitations and hence no queues will ever build up.

- **Queue size:** *queue NN [unit]* – This command sets the queue size only if NN is specified. When there is no room in the queue, packets are dropped. The default queue size is 50 slots. The combination of bandwidth and queue size influence the queueing delay. One should be careful when using low bandwidths not to use too large queues because it might end up with several seconds of queueing delay.

- **Delay:** *delay NN ms* – This command sets the propagation delay of the pipe, in milliseconds. Note that the component for the queueing delay is independent of the propagation delay.

- **Random Packet Loss:** *plr X* - X is a floating point number between 0 and 1 which causes packets to be dropped at random. This is done to emulate lossy links. The default is 0 for a loss free link.

- **Dynamic queue creation:** *mask* - A *mask* is associated to a pipe so that bandwidth and queue limitations are enforced separately for packets belonging to different flows. The *mask* command allows one to specify which parts of the following fields contribute to identify a flow:
  [proto N] [src-ip N] [dst-ip N] [src-port N] [dst-port N] - N is a bitmask where significant bits are set to 1. One or more masks can be specified. The default, when no *mask* is specified, is to ignore all fields so that all the packets will be considered to belong to the same flow. Whenever a new flow is encountered, a new queue with the specified bandwidth and queue size is created. Note that the number of dynamic queues that can be created in this way can become very large.

Appendix E2: Network Timing Protocol (NTP)

Today in the Internet, there are 79 public primary servers synchronised directly to the UTC and over 100 public secondary servers synchronised to the primary servers providing synchronisation to more than 100,000 clients and servers [38]. Furthermore, there is an unknown number of private servers that use the NTP protocol. To discover the clock offset, a server sends a message that includes its current clock value to the client, which could be another server or workstation. The client records its own current clock value upon arrival of the message. For accuracy, the client measures the server-client propagation delay. The NTP measures the total roundtrip delay and assumes the propagation times are statistically equal in each direction. The clock errors are due to variation in network delay and latencies in computer hardware and software (jitter), as well as clock oscillator instability. The NTP protocol can keep the clock synchronisation within a millisecond on LANs and up to a few tens of milliseconds on WANs relative to UTC via Global Positioning Service (GPS) receiver, for instance.