

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

A Flexible Medium Access Control Framework for Multimedia Application Support in Wireless ATM

Prepared by:

Dale Graham How

Supervised by:

Mr. Neco Ventura

Department of Electrical Engineering

University of Cape Town

2002

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

Signed by
Candidate

Signature
Removed

Acknowledgements

I wish to thank the following people for their support, both direct and indirect, in this research.

My partner in this research, Patrick Atkins, for his wealth of general knowledge and his ability to track down information.

My other colleagues in the Communications Research Group (CRG) at UCT, for many entertaining and informative discussions and activities. Especially to André van Zyl and Sven Shepstone for proof reading various chapters of this document.

Mr. M.J. Ventura, my supervisor at the University of Cape Town (UCT), for his guidance while I was working on this project and for organising the funding for this work.

My parents, for opening so many doors in my life and for the support that you have always provided.

Finally, my girlfriend, Lisa, who has been such a source of love and support for the duration of this research.

Psa 5:11 But let all who put their trust in You rejoice; let them always shout for joy, because You defend them. And let those who love Your name be joyful in You.

Psa 5:12 For You, O Jehovah, will bless the righteous; with favor You will surround him as with a shield.

Synopsis

The field of wireless communications has seen phenomenal development over the last decade. With the current abundance of applications that use broadband multimedia over wired networks it is logical that users will want to have access to these same multimedia streams from a mobile terminal. Wireless solutions for connectivity to networks such as Ethernet networks already exist, however, a method of supporting access to an ATM network from a mobile terminal has not yet been standardised.

Transporting ATM data over the wireless medium poses a number of problems. The Medium Access Control (MAC) layer of any proposed wireless ATM network would be responsible for resolving many of these problems. Unfortunately, research into MAC layers is hampered by the fact that most existing MAC layers cannot be modified in order to experiment with the effectiveness of the many MAC protocol techniques that exist.

The focus of this project is therefore on designing and implementing a highly flexible MAC framework that is capable of extending the ATM classes of service and QoS guarantees over the wireless medium. This system allows experimentation with MAC protocols as well as reconfiguration of MAC algorithms on the fly. The MAC framework is also capable of processing data at speeds necessary to support multiple multimedia streams.

This MAC framework is implemented on a Texas Instruments Evaluation Module based on the TMS320C6201 Digital Signal Processor. Texas Instruments' software, Code Composer Studio 2, is used to programme, debug and interpret the outputs of the MAC framework.

A simple, weighted round robin scheduler is included in the MAC framework to test the functionality of the framework. LLC packets are generated and processed by the

framework and measurements of the data rate and efficiency of the framework are made. Modifications or fine-tuning of the framework algorithms are then made in order to optimise the speed and efficiency of the MAC framework.

University of Cape Town

Table of Contents

DECLARATION OF COMPLETENESS OF THESIS	II
ACKNOWLEDGEMENTS	III
SYNOPSIS	IV
TABLE OF CONTENTS	VI
LIST OF FIGURES	X
GLOSSARY	XI
INTRODUCTION	1
1.1 BACKGROUND INFORMATION.....	1
1.2 MULTIMEDIA OVER WIRELESS TECHNOLOGIES.....	3
1.2 OBJECTIVES OF THE THESIS.....	5
1.3 SCOPE AND LIMITATIONS.....	6
1.4 PLAN OF DEVELOPMENT.....	7
RESEARCH REVIEW	9
2.1 THE NEED FOR WATM.....	9
2.2 INCOMPATIBILITIES BETWEEN ATM AND WIRELESS TRANSMISSION.....	12
2.3 A PROTOCOL STACK FOR WATM.....	14
2.4 THE PHYSICAL LAYER.....	16
2.4.1 Spectrum and Bandwidth Issues.....	16
2.4.2 Modulation Methods.....	18
2.5 THE MAC LAYER.....	21
2.5.1 Classification of MAC Protocols.....	22
a) <i>Random Access MAC Protocols</i>	22
b) <i>Fixed Assignment MAC Protocols</i>	22
c) <i>Polling MAC Protocols</i>	23

d) Demand Assignment MAC Protocols.....	23
e) Hybrid MAC Protocols.....	24
f) Duplexing Techniques.....	24
g) Network Layout.....	25
2.5.2 MAC Layers Proposed in Literature.....	26
a) The ALOHA MAC Protocol.....	27
b) The Carrier Sense Multiple Access (CSMA) Protocol.....	27
c) Packet Reservation Multiple Access (PRMA).....	28
d) Distributed-Queuing Request Update Multiple Access (DQRUMA).....	29
e) Mobile Access Scheme based on Contention and Reservation for ATM (MASCARA).....	30
2.6 THE LOGICAL LINK CONTROL (LLC) LAYER.....	31
2.6.1 Forward Error Correction (FEC).....	31
2.6.2 Automatic Repeat Request (ARQ).....	32
SYSTEM DESIGN OF THE MAC FRAMEWORK.....	34
3.2 HARDWARE SYSTEM DESIGN.....	35
3.2.1 Network Topology.....	35
3.2.2 Work done by the Technical University of Berlin.....	35
3.2.3 Hardware Requirements.....	37
3.3 SOFTWARE DESIGN.....	38
3.3.1 Primary Software Design Issues.....	38
3.3.2 Data and Control Flow between Modules.....	38
3.3.3 Inter-module Synchronisation and Messaging.....	40
3.3.5 MAC Algorithm Design.....	41
a) The Wireless Dynamic Weighted Earliest Deadline First Approach.....	41
b) Frame Structure of WDWEDF.....	42
3.3.4 Data Structures and Memory Access.....	44
IMPLEMENTATION OF THE MAC FRAMEWORK WITH WEIGHTED ROUND ROBIN SCHEDULING.....	46
4.1 THE CHOICE OF AN EMBEDDED SYSTEM.....	46
4.2 DEVELOPMENT OF A REAL-TIME ENVIRONMENT FOR THE MAC FRAMEWORK...	48
4.3 MAC FRAMEWORK ARCHITECTURE.....	52

4.3.1 Hardware Configuration.....	53
4.3.2 Packet Source/Sink with Statistics Generation and LLC Abstraction Modules	53
4.3.3 PHY Layer Abstraction and Emulation	54
4.3.4 The Mobile MAC Layer and Base-station MAC Layer Processes	55
<i>a) Communication between Modules Within the Mobile and Base-station MAC Layer Processes.....</i>	<i>55</i>
<i>b) Synchronisation between the Mobile and Base-station MAC Layer Processes </i>	<i>55</i>
4.3.5 MAC Framework Changes when a Wireless PHY is Added.....	57
4.4 CREATION OF CONNECTIONS	57
4.5 BUFFERING SYSTEM.....	58
4.6 PACKING ALGORITHM.....	58
4.7 WEIGHTED ROUND ROBIN SCHEDULER (WRRS).....	59
4.8 STATISTICS GENERATION AND PERFORMANCE LOGGING.....	60
FRAMEWORK TESTS, RESULTS AND MODIFICATIONS.....	61
5.1 FUNCTIONAL TESTING	61
5.1.1 White Box Testing	61
5.1.2 Black Box Testing.....	62
5.2 INSTRUCTION CYCLE COUNT	63
5.3 EXECUTION SPEED ON THE EVM	64
5.4 DATA RATES ACHIEVED ON THE MAC FRAMEWORK.....	66
5.5 FLEXIBILITY TESTS AND RESULTS	67
5.6 VALIDATION OF BUFFER EFFICIENCY.....	69
CONCLUSIONS AND RECOMMENDATIONS	71
6.1 CONCLUSIONS	71
6.2 RECOMMENDATIONS AND FUTURE WORK	76
APPENDIX A: ADDITIONAL ATM INFORMATION	79
A.1 CONNECTION-LESS AND CONNECTION-ORIENTED PROTOCOLS	79
A.2 ATM'S CONNECTION-SPECIFIC PARAMETERS	80
APPENDIX B: ADDITIONAL INFORMATION ON EVM PERIPHERALS ...	82

B.1 OVERVIEW OF THE DIRECT MEMORY ACCESS (DMA) CONTROLLER.....	82
B.2 OVERVIEW OF THE MULTI-CHANNEL BUFFERED SERIAL PORTS (MCBSP)	83
APPENDIX C: DATA STRUCTURES USED BY THE MAC FRAMEWORK	85
C.1 DATA STRUCTURES	85
C.2 BUFFERS USED BY THE MAC FRAMEWORK.....	87
APPENDIX D: OVERVIEW OF THE MAC FRAMEWORK CODE.....	90
C.1 BUFFERS.C	90
C.2 INIT.C	91
C.3 MCBSP.C	91
C.4 ALL.C	92
APPENDIX E: THE ACCOMPANYING CD-ROM.....	95
REFERENCES	96

University of Cape Town

List of Figures

FIGURE 1: A WIRELESS ACCESS TO A FIXED ATM NETWORK	11
FIGURE 2: PROPOSED WATM PROTOCOL STACK.....	15
FIGURE 3: THEORETICAL MULTI-PATH FADING WHEN PATH LENGTHS DIFFER BY HALF A WAVELENGTH	18
FIGURE 4: ARRANGEMENT OF OFDM CARRIERS IN THE FREQUENCY DOMAIN, ADAPTED FROM [SPETH01].....	20
FIGURE 5: PHYSICAL LAYOUT OF THE TEST-BED AT THE TECHNICAL UNIVERSITY OF BERLIN	36
FIGURE 6: DATA FLOW DIAGRAM	39
FIGURE 7: WDWEDF MAC FRAME STRUCTURE	42
FIGURE 8: DEBUG ENVIRONMENT OF CCS	48
FIGURE 9: GRAPHICAL CONFIGURATION TOOL IN CCS	49
FIGURE 10: KERNEL/OBJECT VIEW IN CCS	50
FIGURE 11: LOGGED DATA FROM THE MAC FRAMEWORK	51
FIGURE 12: ARCHITECTURE OF THE MAC FRAMEWORK.....	52
FIGURE 13: PROCESS SYNCHRONISATION DIAGRAM	56
FIGURE 14: INSTRUCTION COUNT COMPARISON.....	64
FIGURE 15: TRANSMIT/RECEIVE DATA RATE VS. TOTAL DATA RATE FOR A DEFAULT FRAME	67
FIGURE 16: DATA RATES FOR DIFFERENT FRAME SIZES	68
FIGURE 17: THE BENEFIT OF ZERO-COPYING IN BUFFERS	70
FIGURE 18: BUFFER STATE BEFORE ENQUEUE.....	87
FIGURE 19: BUFFER STATE AFTER A SINGLE ENQUEUE.....	88

Glossary

2G, 3G, 4G - Second, third and fourth Generation

ABR - Available Bit Rate

ARQ - Automatic Repeat Request

ATM - Asynchronous Transfer Mode

BER - Bit Error Rate

CBR - Constant Bit Rate

CCS - Code Composer Studio 2

CDMA - Code Division Multiple Access

CDV - Cell Delay Variation

CER - Cell Error Rate

CLR - Cell Loss Ratio

C-PRMA - Centralised Packet Reservation Multiple Access

CRC - Cyclic Redundancy Check

CRG - Communications Research Group

CSMA - Carrier Sense Multiple Access

CTD - Cell Transfer Delay

DLB - Digital Loop Back

DMA - Direct Memory Access

DQPSK - Differential Quadrature Phase Shift Keying

DQRUMA - Distributed Queuing Request Update Multiple Access

DSP - Digital Signal Processor

EDD - Earliest Due Date

EDGE - Enhanced Data Rates for Global Evolution

EVM - Evaluation Module

FDD - Frequency Division Duplexing

FDMA - Frequency Division Multiple Access

FEC - Forward Error Correction

GBN-ARQ - Go-Back-N ARQ

GMSK - Gaussian Minimum Shift Keying
GPRS - General Packet Radio Service
GSM – Global System for Mobile telecommunications
HEC - Header Error Control/Check
HPI - Host Port Interface
IDE - Integrated Development Environment
ISM - Industrial, Scientific and Medical
IWF - Inter-Working Function
LLC - Logical Link Control
MAC – Medium Access Control
MASCARA - Mobile Access Scheme base on Contention and Reservation for ATM
MBS - Maximum Burst Size
McBSP - Multi-channel Buffered Serial Port
MCR - Minimum Cell Rate
MES - Mobility Enhanced Switch
MPDU - MAC Protocol Data Unit
nrt-VBR - Non Real Time Variable Bit Rate
OFDM - Orthogonal Frequency Division Multiplexing
OS - Operating System
PCR - Peak Cell Rate
PHY - Physical Layer
PRADOS - Priority Regulated Allocation Delay-Oriented Scheduling
PRIME - Partial Selective Repeat superimposed on GBN-ARQ
PRMA - Packet Reservation Multiple Access
QPSK - Quadrature Phase Shift Keying
RAL - Radio Access Layer
rt-VBR - Real Time Variable Bit Rate
S-ALOHA - Slotted ALOHA
SCR - Sustainable Cell Rate
SR-ARQ - Selective Repeat ARQ
TDD - Time Division Duplexing
TDMA - Time Division Multiple Access
TI - Texas Instruments
TU - Technical University of Berlin

UBR - Unspecified Bit Rate

UCT - University of Cape Town

UMTS - Universal Mobile Telecommunications System

VC - Virtual Circuit

VCI - Virtual Channel Identifier

VPI - Virtual Path Identifier

WAND - Wireless ATM Network Demonstrator

WDWEDF - Wireless Dynamic Weighted Earliest Deadline First

WUGS - Washington University Gigabit Switch

University of Cape Town

Chapter 1

Introduction

1.1 Background Information

Initially, when telecommunications networks began, the focus was on creating a network that people could communicate through by using the most natural form of human communication, namely speech. For a network to provide an acceptable voice connection between two users it was necessary to ensure that the natural characteristics of speech were not violated as speech was transported through the network. In this way, both users could hold a natural conversation without caring about how far apart they were. If the network preserved the characteristics of their speech conversation adequately it would seem as if the person they were talking to was right next to them. If there was a high delay between a spoken word and when that word was heard, or if there was any delay variation or jitter in the speech received, then communication between the two people would be impeded. Minimising the end-to-end delay as well as the jitter was therefore a priority in voice networks.

Data networks were the next evolution in telecommunications. These networks were designed with data traffic in mind. The Internet, for example, evolved from a network initially built to carry data for the U.S military, and to allow communications to continue in the event of a nuclear attack. Educational institutions then saw the usefulness of this network as a means to exchange ideas and data and began using it

for non-military purposes. It is still, however, a best effort network and is not suitable for carrying real-time data. It is not, for example, well suited to carrying voice traffic because it does not place limits on the end-to-end delay or the delay variation of traffic.

Additional traffic types, such as video traffic, soon made an appearance. People wanted to communicate not only by using voice but also by sending and receiving video images and by transmitting data to one another (e.g. e-mail.) However, due to the high cost of building and maintaining a network, it became imperative to combine different types of traffic onto one network. This network would need to be able to transport all traffic types adequately, ensuring that all of their needs were met.

The user requirements for an acceptable voice connection and an acceptable data connection are very different. Video traffic also has a different set of requirements to both voice and data traffic. These differences had to be reconciled and addressed within the network if the network was to carry all traffic types simultaneously. It was not feasible to treat all traffic with best effort service, as this would result in unsatisfactory performance with real-time traffic types. Traffic types can also interfere with each other in a best effort network. For example, a flood of data traffic could seriously affect the time taken for the delivery of voice traffic.

Communications networks have therefore evolved from a simple, single traffic type paradigm, into networks that attempt to cater for many different traffic types. Combined streams of data, voice and video flowing from one user to another are now known as multimedia traffic. Networks have developed a number of ways of differentiating between traffic types so that their unique requirements are always taken into account. One network that provides hard guarantees in terms of the service that it will provide to each connection is an Asynchronous Transfer Mode (ATM) network.

1.2 Multimedia over Wireless Technologies

The field of wireless communications has seen phenomenal development over the last decade. This development combined with the freedom that wireless communication gives users to communicate from wherever they wish has led to a massive desire for personal mobile communications devices. Devices such as cell phones have become necessities rather than novelties in a very short space of time. The subscriber base for wireless communications services is growing 15 times faster than that of the wired services [Bing01], clearly illustrating the desire of users to have mobile access to services. It is very likely that this rate will increase and that an environment will soon exist where mobile access is dominant in a network that was previously only accessible via a wired terminal.

With the current abundance of applications that use broadband multimedia over wired networks it is logical that users will want to have access to these same multimedia streams from a mobile terminal. This means that wireless networks will have to evolve to support all of the services (e.g. voice, video, data) that users have come to expect over a wired network. To this end it is essential that wireless networks develop a way to interface seamlessly with the wired network in order to provide the quality of access that users are accustomed to. This need presents researchers with many problems. Protocols designed for use on wired systems seldom function well or at all over the wireless medium, primarily because of the increased error rate caused by the wireless medium [Chan01]. This means that these protocols have to be modified to allow them to achieve reasonable performance over the new medium. Many researchers have been investigating ATM over the wireless medium. It is possible to convert ATM to some wireless protocol before transmission, using what is known as an Inter-Working Function (IWF) where the ATM network ends and the wireless network begins. This, however, results in the loss of all of the connection specific information (class of service, traffic and QoS parameters) that is stored and monitored by ATM networks. The separate priorities of the different classes of traffic are therefore also lost. This can result in one or more connection's requirements being violated. It is for this reason that this research will focus on transmitting native ATM over the wireless medium, without converting to another protocol.

In wireless networks the medium is a broadcast medium. This means that multiple transmitters share the same physical link. Transmitters can interfere with each other's transmissions, resulting in collisions between transmissions and a subsequent loss of throughput. A large amount of research is therefore being dedicated to finding an optimal, wireless Medium Access Control (MAC) layer for any specific application. The MAC layer, as its name implies, governs transmissions over the wireless medium by dictating how each transmitter may access the shared medium. The MAC layer can therefore ensure that the wireless medium is used efficiently by minimising the number of collisions between competing transmitters. Efficient use of wireless resources is critical due to the shortage of bandwidth available to wireless systems [Kubb01].

Access to the medium can be provided in the form of:

- ❑ A physical time slot, known as Time Division Multiple Access (TDMA).
- ❑ A physical frequency slot, known as Frequency Division Multiple Access (FDMA).
- ❑ The entire channel, but with the transmission encoded in such a way so as to distinguish it from other transmissions and to minimise the interference with other traffic streams. An example of this would be Code Division Multiple Access (CDMA).
- ❑ Some combination of the above methods. GSM, for example, uses a combination of TDMA and FDMA to separate different channels from each other [Scou01].

If the wireless system must support ATM, or any protocol that requires QoS or traffic differentiation (as is required when multimedia traffic is carried by the network), then it is also essential that the MAC layer assign bandwidth in a manner that takes the specific requirements of each traffic flow into account. A mechanism to distinguish between different types of traffic is therefore needed, as well as a scheduling algorithm that takes each traffic flow's individual requirements into account. Further details on the MAC layer will be discussed in Chapter 2 and Chapter 3.

1.2 Objectives of the Thesis

The Communications Research Group (CRG) at the University of Cape Town (UCT) wishes to develop a WATM test-bed. This test-bed should be as flexible as possible, allowing researchers to modify any aspect of its functionality in order to evaluate the performance of various techniques. Due to the importance of the MAC layer it is vital that this layer in particular can be modified or enhanced to suit future research.

Unfortunately, most proprietary wireless systems that have been researched (such as Intersil's PRISM chipset for wireless LAN applications, [Inte01]) contain MAC layers that cannot be modified for experimental purposes due to high levels of integration onto single chip technology. Also, most of the current research into MAC layers is only done in simulation, e.g. [Sung01, Petr01], leaving the actual implementation and testing to other researchers. The focus of this project is therefore on building a system that allows the implementation and testing of any MAC layer. This system must be as flexible as possible, allowing experimentation with MAC protocols and methods as well as reconfiguration of various MAC algorithms on the fly. It is also very important that this MAC framework be capable of communicating with other the layers above and below it in the final WATM test-bed.

The MAC framework must support ATM classes of service and traffic/QoS parameters and must be capable of processing data at rates of approximately 10Mbps in order to support multiple multimedia traffic streams. A further aim of this research is to find MAC techniques that are most suited to handling multimedia traffic (multiple connections of voice, video and data). These techniques can then be used in the MAC framework. Research into various microcontrollers and Digital Signal Processors (DSPs) will also be conducted in order to determine which system is best suited to being used in the MAC framework. Further research into real-time Operating Systems (OS) must also be conducted. A real-time OS will be used in the implementation of the MAC framework to allow the MAC framework to react in real-time to events.

Once the MAC framework has been implemented its performance can be tested. A simple, weighted round robin scheduler will be included in the MAC framework to

facilitate this testing. LLC packets will be generated and processed by the framework and measurements of the data rate and efficiency of the framework made. Modifications or fine-tuning of the framework can then take place in order to optimise the speed and efficiency of the MAC framework.

1.3 Scope and Limitations

This project is the first project with the aim of developing a WATM test-bed for the CRG lab. It is not reasonable to implement a fully functional WATM system in a single project. It is for this reason that certain parts of the final WATM test-bed have been singled out for research and development in this project. More specifically, this project will focus on the development of a MAC layer for the final WATM test-bed.

This project does not concern itself with functionality such as:

- Error detection or correction, as this is handled by the LLC layer, which does not form part of the MAC layer. This project will, however, ensure that adequate communication of data can take place between the LLC and the MAC layers. Recommendations on LLC techniques that should be considered in future projects will also be made.
- Physical Layer (PHY) functionality, however, recommendations with regard to which PHY layer future projects should consider as possibilities for the WATM test-bed will be made.
- The connections between the mobile terminal and the base-station will be created and initialised prior to the running of MAC algorithms. There will be no support for signalling to establish a connection.
- Handoff mechanisms and their effects on multimedia traffic have been addressed elsewhere and are not in the scope of this thesis [Elber01].

As the higher layers of the WATM test-bed do not yet exist the MAC layer cannot receive packets from the higher layers. In this project the packets that would be received from the LLC layer (above the MAC layer) will be generated within the MAC framework. These packets will not contain data relevant to any specific application; only information used by the MAC layer will be valid. Functionality allowing the MAC layer to receive valid packets from the LLC layer will, however,

be built into the MAC framework. The framework will also assume that, in the future, data from the LLC layer will not contain errors.

The scheduler used in the MAC framework will be discussed in this thesis but it is not within the scope of this project to fully develop it. A simple scheduling algorithm will, however, be used to test the functionality of the MAC framework. More detailed aspects of MAC layer scheduling to support multimedia and QoS parameters will be addressed elsewhere [How01].

1.4 Plan of Development

The following chapters are organised as follows:

- Chapter 2 will delve deeper into the research performed in WATM to provide background information for this project. This chapter will motivate the need for WATM as an access technology. A protocol stack for a generic WATM system will be presented. This chapter will also discuss the incompatibilities between ATM and the wireless medium. The functionality of the radio access layers of WATM will be discussed and issues specific to those layers highlighted.
- Once the issues concerning WATM networks have been discussed, Chapter 3 will focus on the system design of a MAC framework capable of offering a seamless interface to the wired ATM network and of supporting multimedia connections. An abstract view of the system under development in this project will be presented. The MAC layer chosen for implementation as well as the specifications that the embedded system will have to meet will be discussed.
- Chapter 4 will move on to the architecture and implementation of the system. The embedded system chosen will be justified based on the requirements of the MAC framework. Further discussion of the chosen embedded system and its capabilities will follow. The software tools that will be used to implement the MAC layer on the embedded system will then be discussed. The software architecture on the embedded system will also be discussed. This includes mention of the embedded real-time OS used as well as how separate modules of the framework were implemented and how they communicate with each other.

- Chapter 5 will first describe the functional testing performed on the MAC framework. This testing verifies that the framework functions according to its design. The chapter will then discuss the various performance-related tests performed on the MAC framework and the results obtained by these tests. These tests concern the speed at which the framework is capable of processing data. Modifications made to the MAC framework in order to improve performance are also discussed. The flexibility of the MAC framework was also tested and the test and its results are presented.
- Chapter 6 will discuss conclusions that can be drawn from the research and development of the MAC framework. Conclusions concerning integration with future layers of the WATM test-bed as well as the results obtained in this research will also be presented.
Chapter 6 will also provide recommendations for future work on the WATM test-bed. These recommendations are mainly concerned with the development of advanced schedulers for the MAC framework, as well as the development of other layers of the WATM test-bed.
- A set of appendices follow chapter 6. These appendices provide additional background information as well as more detailed information regarding certain aspects of the implementation of the MAC framework.

Chapter 2

Research Review

This chapter will present information about WATM systems. Many different sources of information on WATM were consulted to learn as much as possible about this wide topic. This chapter will focus on summarising the key issues concerning WATM discovered during this research. This chapter will also highlight the incompatibilities between ATM and the wireless medium and other problems that are faced by WATM researchers. The proposed protocol stack for a WATM system will be presented and the different layers discussed.

2.1 The Need for WATM

As mentioned in chapter one, the use of wireless communications has grown beyond expectations in recent years. Mobile access to fixed networks has become a valuable commodity, one that generates large amounts of revenue for companies offering such access. However, the increasing desire for high bit rate communications has found wireless networks lacking.

Traditional second generation (2G) wireless networks (e.g. GSM) only offer a 9.6Kbps basic connection [Holle01]. This has prompted the development of wireless networks capable of transmitting information at higher rates. GPRS, a service using the GSM standard, offers faster data rates and lowered access times [Bett01]. Enhanced Data Rates for Global Evolution (EDGE) is also based on the GSM

standard, and offers a maximum bit rate of 384 Kbps in good conditions [Siem01]. Both GPRS and EDGE are sometimes referred to as 2.5G systems. The Universal Mobile Telecommunications System (UMTS) forms the basis for third-generation (3G) systems. UMTS aims at offering users up to 2Mbps over short ranges and in ideal conditions [Siem01].

The next logical development after 3G networks is fourth generation (4G) wireless networks, which are currently being researched. WATM aims at being a 4G network, capable of delivering high-speed access along with the guaranteed QoS that users are used to receiving when using ATM over an optical link.

WATM is mainly considered to be an access technology [Ayan01]. This means that it would be used to provide mobile end-users with wireless access to the wired ATM network, as opposed to being used as a distributed, entirely wireless network. Figure 1 illustrates wireless access to the wired ATM network. The Mobility Enhanced Switches (MES) are responsible for handling mobility functions such as handoff and path rerouting.

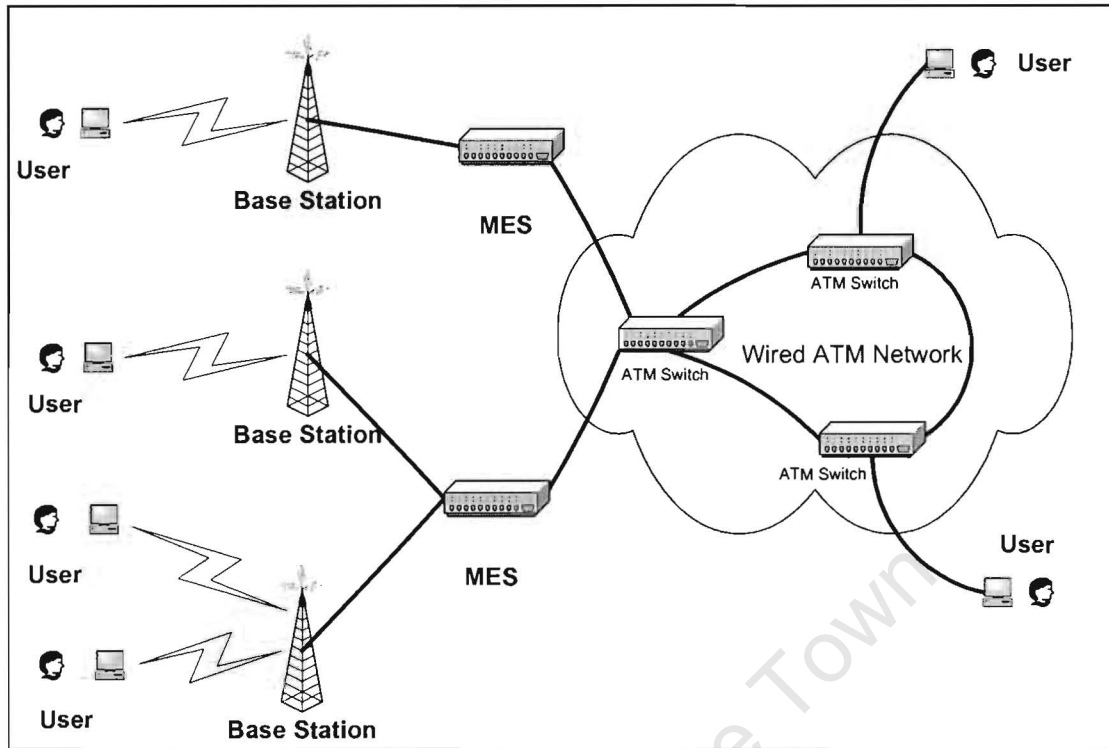


Figure 1: A Wireless Access to a Fixed ATM Network

There are other wireless protocols that could be used in place of WATM in such an access network (e.g. 802.11) [IEEE01]. However, the use of these protocols would most often result in the homogenisation of all of ATM's classes of service into one, best effort class of service. WATM, on the other hand, preserves all of ATM's defined classes of service and the QoS parameters associated with each connection. This means that the requirements of each connection will still be considered when they are transmitted across the wireless medium and at the mobile terminal using the WATM network. This factor is essential in an environment where different connections demand different types of service from the network. The transmission of multimedia, for example, is one application where each communication stream has its own requirements in terms of bandwidth, Cell Loss Ratio (CLR), Cell Delay Variation (CDV) and Cell Transfer Delay (CTD). It is essential to the integrity of the multimedia streams that these requirements are satisfied. WATM offers a way of satisfying the requirements of each stream in the simplest way without having to convert ATM cells into packets of some other, intermediate, wireless protocol.

ATM makes use of small, fixed size packets (53 bytes) known as cells. The fixed nature of the size of these cells increases the speed at which they can be processed. The small size of these cells reduces packetisation delay, thereby minimising the cell delay variation that a connection will experience. Using smaller cells in the wireless medium also has its benefits. Any cells with errors in them may need to be retransmitted (depending on the class of service to which they belong.) It is highly desirable that the bandwidth consumed by these retransmissions is minimised. Retransmitting a small, 53-byte cell constitutes a far lower overhead than retransmitting an entire Ethernet packet for example. (Ethernet packets are of variable size and can be as large as several thousand bytes.) Due to its smaller size, an ATM cell transmitted over the wireless medium is also less likely to contain errors when compared to an Ethernet packet (for example) transmitted over the same medium. Using ATM cells over the wireless medium will therefore result in a higher throughput than using another protocol with a greater packet size.

2.2 Incompatibilities between ATM and Wireless Transmission

Transmitting information over the wireless medium presents many problems that need to be resolved if a useful system is to be constructed. Presented in this section are some of the fundamental problems with using ATM in the wireless environment. Subsequent sections of this chapter will deal with some of the more specific issues that each layer of the WATM stack has to deal with and some of the solutions that have been implemented to deal with these problems.

The wireless medium is characterised by a high Bit Error Rate (BER). The BER also varies in time and depends on the physical locations of the receiver/transmitter pair. The wireless medium may have a BER of 10^{-3} or higher [Chan01]. This means that there is a much higher chance of a packet error when using the wireless medium as opposed to a wired medium. ATM is designed to use a medium with a BER of about 10^{-10} and a packet error rate of about 10^{-6} [Ayan01]. This discrepancy between the error rate of the optical and the wired mediums brings about the need for an error correction and/or packet retransmission scheme in order to obtain a useful throughput. These schemes require bandwidth and therefore affect the efficiency of the

connection. Usually this error detection/correction or retransmission is controlled by the Logical Link Control (LLC) layer, which will be further discussed later in this chapter.

ATM was designed to operate in a bandwidth-rich environment. ATM sacrifices bandwidth for switching speed. Each ATM cell has a header that constitutes approximately 10% of the cell. The overhead of the header is acceptable in networks where bandwidth abounds (such as in an optical ATM network). The wireless environment, however, does not have an abundance of bandwidth. The overhead of the header is considered to be too great for the wireless environment. Add to this the fact that it will be necessary to sacrifice additional bandwidth in the physical layer for tasks such as channel equalisation and synchronisation, and that methods for compensating for the high BER of the medium will also require additional bandwidth, and there exists a situation where any possible increase in efficiency must be investigated and exploited to the full. Some researchers propose compressing the header to cut down on the bandwidth wasted by using the full ATM header. The Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) fields of the ATM header are particularly well suited to compression. It is hardly likely that the full address resolution provided by these VPI and VCI fields will be required in a wireless network where each base-station only needs to resolve between mobile terminals within its area of coverage. The Header Error Check (HEC) field of the ATM header could also be discarded if adequate error checking of the whole cell was added at the LLC layer (as mentioned earlier in this section.) Discarding these and other fields of the ATM header would result in an increase in efficiency over the wireless medium, but would also require that the full ATM header be restored in the base-station so that the ATM cells could continue into the wired ATM network as if they had never travelled over the wireless ATM network. It should be possible to add this functionality to the MES mentioned earlier, along with the other wireless support functions that will need to be added.

The wireless medium is a broadcast medium. This has many ramifications on security and access control. The fact that all users within range of a transmitter can intercept its transmissions means that users wishing to keep their communications secret will need to employ some sort of encryption. It is also possible for a user to impersonate

another user and to transmit false information. A form of user identification can also become essential in this case in order to verify the source of information. The 802.11 wireless LAN networks have experienced security problems due to the broadcast nature of the medium [Stubb01]. It is a trivial matter for a malicious user to connect to a wireless network without actually acquiring a physical plug in point. Wireless networks therefore need mechanisms to ensure privacy of information.

The manner in which each transmitter transmits over the medium needs to be strictly controlled due to the fact that the wireless medium is a broadcast medium. Transmissions that could interfere with one another must be isolated from each other in time or in frequency, or must have a way of being distinguished from other transmissions occurring at the same time and frequency. This control is implemented in the Medium Access Control (MAC) layer. The MAC layer will be discussed in detail later on in this thesis. Some different MAC layers will also be presented in this thesis with a view to examining the capabilities of each one and determining which one is most suitable for supporting multimedia traffic.

By definition, the mobile user will be moving from one location to another. This means that they will be moving from areas serviced by one base-station to another. This introduces the need for the network to support handoff of the user between base-stations. ATM signalling does not have the functionality built into it to support handoff. Thus existing signalling within the wired ATM network needs to be augmented to include support for mobile end users. Another challenge that connection handoff introduces is that of continuous QoS support. As previously mentioned, ATM is built on the guarantees that it can supply to the user. These guarantees need to be maintained seamlessly during a handoff operation. The MESs within the wired ATM network need to implement this functionality.

2.3 A Protocol Stack for WATM

WATM is a technology that is not yet defined by any specifications. There is no standardised protocol stack for WATM. The standardisation of the WATM stack is under investigation by the ATM Forum. Shown below, in Figure 2, is the ATM forum WATM stack, adapted from [ATMF01].

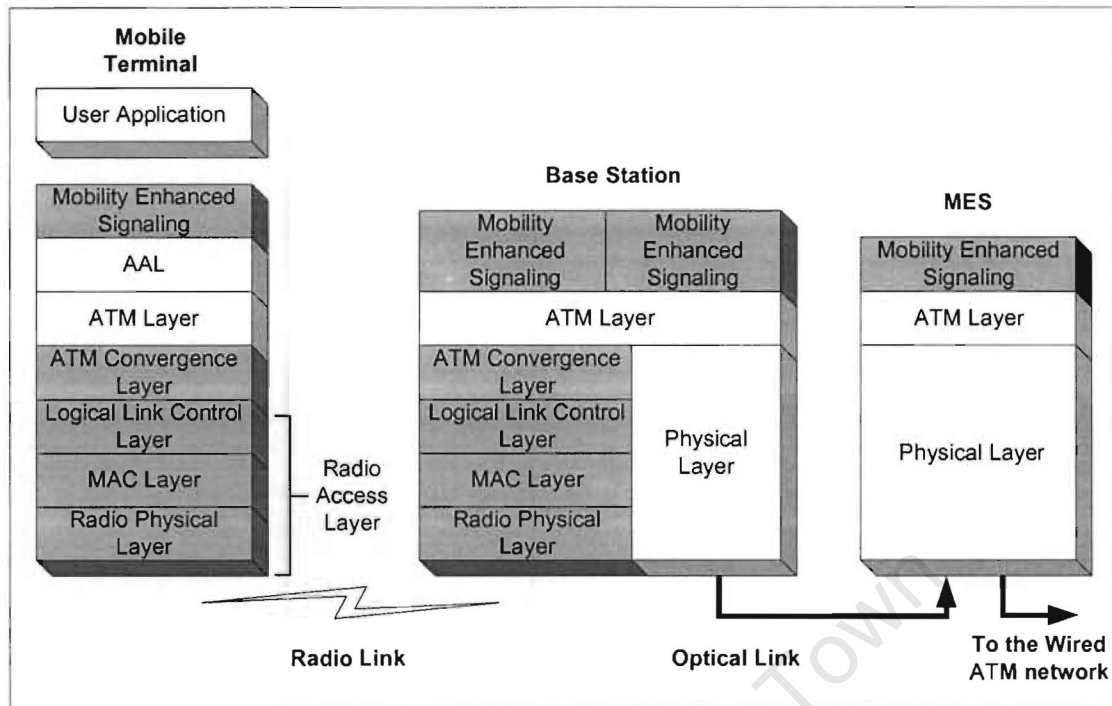


Figure 2: Proposed WATM Protocol Stack

In the above figure the shaded regions indicate layers that have been added to or adapted from the standard ATM stack in order to support wireless access.

The standard ATM signalling has been augmented in this stack to include mobility functionality such as handoff and the subsequent re-routing of data. The Mobility Enhanced Switch (MES), which is situated on the edge of the wired ATM network, can perform mobility related functions, and can use the enhanced mobility signalling to communicate mobility related information with other MESs in the fixed ATM network.

The ATM and AAL layers remain unchanged, but the ATM forum decided to add an ATM convergence layer below the ATM layer. This layer will provide a standard interface to a generic Radio Access Layer (RAL), which is situated immediately below the ATM convergence layer.

The RAL consists of those layers in the protocol stack that are concerned with actions specific to communication over the wireless medium. It is this section of the protocol

stack that is largely responsible for dealing with the incompatibilities between ATM and the wireless medium. The RAL contains three sub-layers, namely:

- The Physical Layer (PHY)
- The Medium Access Control layer (MAC)
- The Logical Link Control layer (LLC)

2.4 The Physical Layer

This layer is responsible for the actual transmission of data over the wireless medium using radio transmission techniques. Although the physical layer will only be emulated in this project, its discussion here is important as it is very closely related to the MAC layer. How the physical layer is implemented also directly affects the implementation of the MAC layer and can limit the type of MAC layer that can be used. For this reason, recommendations on a PHY layer approach for future projects will also be made in this project.

2.4.1 Spectrum and Bandwidth Issues

The limited available radio spectrum is a major factor in the implementation of any wireless communications system. The frequencies that are available vary from one geographical location to another [Ayan01], further complicating any decisions on a standard operating band for a global system. Licenses to operate exclusively in a specific frequency range are extremely expensive and are not easy to obtain. Various unlicensed bands do exist, but their use is subject to certain restrictions. These restrictions are that the transmitters use spread-spectrum techniques at very low power. It is these unlicensed bands that are used for product research and testing, as it is only when a product is earning large amounts of revenue that it can pay for its own, licensed, spectrum.

Very little bandwidth is free at the lower end of the frequency spectrum. High bandwidth applications have therefore prompted research into the use of the higher, less cluttered, frequency ranges. Various unlicensed, Industrial, Scientific and Medical (ISM) bands exist at higher frequencies such as the 2.4GHz or the 5GHz ranges. The 2.4GHz band can theoretically support transmission speeds of the order of 10Mbps, but suffers from being cluttered due to use by the 802.11 wireless LANs

[IEEE01] and interference from microwaves [Bing01]. The 5 GHz spectrum is less cluttered and provides more bandwidth. Frequencies as high as 60GHz are also being investigated due to the large amount of unused bandwidth available in that frequency range.

Unfortunately, the use of higher frequencies is not without its problems. The formula for the attenuation of a signal travelling through free space is as follows:

$$\text{Free space power loss} = 32.4 + 20 \times \text{Log } F + 20 \times \text{Log } R$$

F is frequency in MHz.

R is the distance between the transmitter and receiver in km.

The resulting free space power loss value is in dB.

[Bree01]

Higher frequency signals do not propagate as well as lower frequency signals for this reason, resulting in a shorter effective transmission range. Also, as the frequency of the transmission increases so does the ease with which it is absorbed by walls or other obstacles between the transmitter and receiver. Higher frequencies therefore increase the reliance of the connection on line of sight between the transmitter and receiver. Higher frequencies also introduce more errors due to multi-path fading. Multi-path fading occurs when a signal takes more than one path from the transmitter to the receiver. If the path lengths differ by a significant percentage of the wavelength of the signal the two different versions of the signal can interfere with each other. At higher frequencies the wavelength is much shorter, meaning that two path lengths differing by only a small amount could still create interference. Figure 3 shows theoretically how this can happen for the case where the two different path lengths differ by half a wavelength (Exactly what the signals will look like depends on the modulation scheme.)

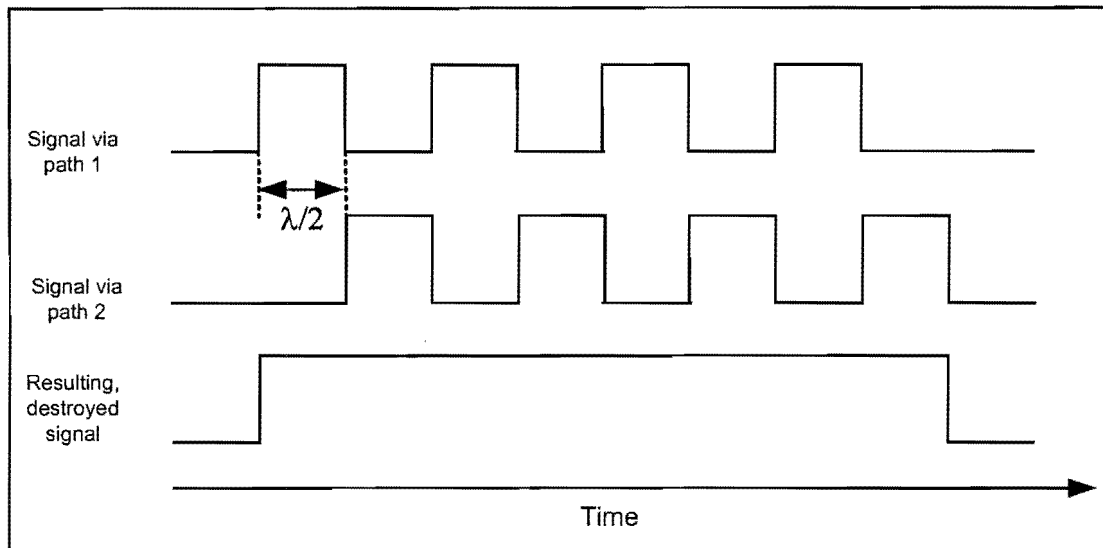


Figure 3: Theoretical multi-path fading when path lengths differ by half a wavelength

2.4.2 Modulation Methods

The high BER of the wireless medium forces a wireless system to use a modulation method that can reduce the effect of this BER on the quality of the signal. Due to the limited frequency range available for the transmission of data the modulation method must also be efficient in its use of the spectrum. The spectrum used may place additional requirements on the modulation method used. If, for example, the wireless system makes use of one of the ISM frequency bands it will have to ensure that the modulation scheme chosen will limit the power of the transmission so that interference with other users is limited.

Schemes such as Quadrature Phase Shift Keying (QPSK), Differential QPSK (DQPSK) and Gaussian Minimum Shift Keying (GMSK) have all been considered for use in a WATM system [Bing01], [Ayan01]. Orthogonal Frequency Division Multiplexing (OFDM) was initially ruled out in some earlier publications [Ayan01] as it required further research before being a feasible option. However, due to the recent development of OFDM and its use in high-speed (up to 54Mbps) 802.11a networks and HIPERLAN 2 [BRAN01], it can now be viewed as a reasonably mature and viable alternative [Geie01].

OFDM is a multi-carrier scheme, using multiple carriers at different frequencies to modulate data. Older multi-channel schemes using Frequency Division Multiple Access (FDMA) to divide the available bandwidth up into N non-overlapping bands. While this results in very little inter-carrier interference it also results in an inefficient use of bandwidth. Guard bands are required in the frequency domain on either side of each carrier. These guard bands are a waste of spectrum, as they are not used for communication purposes. OFDM makes very efficient use of the spectrum allocated to it by placing different carriers very close to each other in the frequency domain. In any other modulation scheme this would result in the carriers interfering with each other and preventing communication of data. OFDM, however, uses carriers spaced at very specific intervals in the frequency domain so that the carriers are orthogonal to one another. This enables the demodulator at the receiver to distinguish the different carriers from each other. Figure 4 shows how the carriers are arranged in the frequency domain so that inter-carrier interference is minimised. OFDM is also very resilient against interference and multi-path distortion, [WAVE01] making it a promising prospect for high bandwidth, high frequency based systems such as WATM. This benefit is due to the increased symbol length due to the distribution of the signal over multiple carriers. The symbol length is increased N times if N is the number of carriers. This increases the required difference in path length for any interference to occur. The increased symbol length also increases the resilience of an OFDM system to burst errors in the wireless environment. The error burst will affect fewer bits due to the increased symbol length [Rama01].

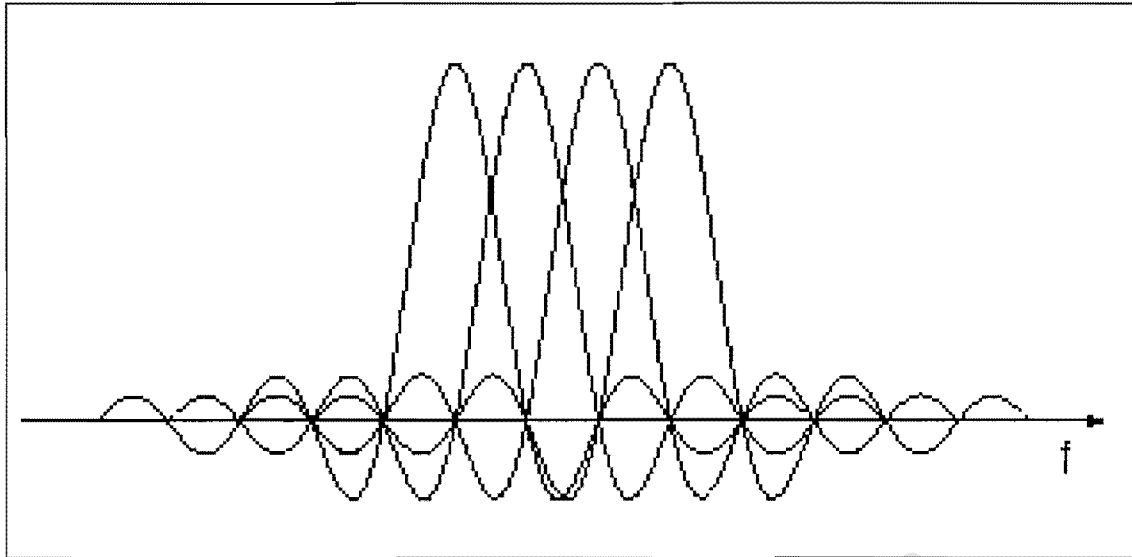


Figure 4: Arrangement of OFDM carriers in the frequency domain, adapted from [Speth01]

OFDM, as with all modulation methods, has its drawbacks. One of the more serious of these drawbacks is that OFDM exhibits a high peak to average signal ratio. This means that the amplitude of the output signal varies over a very wide range. The amplifiers used by the system must therefore be linear over a large range. The high peak amplitude of the signal can also cause clipping of the signal. This clipping would result in a distorted signal and the radiation of frequencies not within the allocated bandwidth.

Synchronising the OFDM transmitter and receiver can also prove problematic. The frequencies used for the separate carriers must be almost identical between the transmitter and the receiver. Inter-carrier interference will result if the carrier frequencies of the receiver and the transmitter differ by even a small amount. The receiver must therefore attempt to compensate for any frequency offsets that it experiences. The receiver must also estimate where one symbol ends and another begins. From this it must decide on the optimal time to measure the value of a symbol with the minimal amount of inter-symbol and inter-carrier interference [Rama01].

OFDM can be used in two scenarios. In the first scenario each carrier is used to modulate a separate channel. In other words, each channel is assigned a fixed bandwidth. When used in this way OFDM can also be considered to be performing MAC functionality, as it regulates how each channel/user accesses the medium. In the

second scenario, the OFDM PHY layer splits a continuous stream of data into multiple channels, each of which is then modulated by a separate carrier. The continuous stream of data would be fed from the MAC layer. The OFDM PHY layer would then only be responsible for transmitting the data given to it. No MAC-like functionality would be exhibited in this case. It is this second mode of operation that should be used if OFDM is used as the PHY layer in projects subsequent to this research.

2.5 The MAC Layer

When two or more users transmit information via a shared medium in the same transmission slot there will be a collision. When this occurs it is probable that neither transmission will be correctly received. The data must then be retransmitted if the quality of the wireless connection is to be maintained. The task of the MAC is therefore to ensure that each user is given access to the medium in a manner that is fair and minimises collisions, thus minimising the need to discard/retransmit packets and improving throughput.

Apart from ensuring that the medium is used efficiently, it is also essential that the MAC layer in a WATM system assign bandwidth in a manner that takes the specific requirements of each traffic flow into account. A method of distinguishing between traffic flows and their needs is therefore required. These needs must then be taken into account when allocating resources to each traffic flow.

The MAC layer is situated directly above the PHY layer. Once traffic has been scheduled it is passed down to the PHY layer for modulation and transmission. In some cases the distinction between MAC and PHY layer can be difficult. An example of this is when OFDM is used as a PHY layer. OFDM, as previously discussed, can perform some MAC functionality as well as performing the modulation and transmission tasks.

2.5.1 Classification of MAC Protocols

There are three main types of MAC. These are random access, fixed access and demand assignment. Then there are hybrid MAC protocols that combine the above types in a specific way to achieve their goals. MAC protocols can also be classified according to the duplexing technique that they employ as well as the network layout that they are designed for.

a) Random Access MAC Protocols

When using random access protocols users transmit at will, using the entire channel for short periods of time [Ayan01]. These protocols do not require any scheduling overhead and therefore work well for networks with small traffic volumes. As soon as the network traffic increases, however, the likelihood of a collision between packets increases, resulting in decreased throughput. Eventually the system becomes little more than useless as collisions become very frequent. The ALOHA MAC is a random access MAC [Ayan01]. Carrier Sense Multiple Access (CSMA) is also classified as a type of random access MAC. These MACs will be elaborated on in later sections of this chapter.

Random Access MAC protocols are not deemed suitable for transporting multimedia traffic due to their instability at high traffic loads as well as their inability to distinguish between connections that require different treatment. They are, however, often used as part of a more complex MAC protocol, usually controlling access to a frame slot in which new or existing connections contend for bandwidth.

b) Fixed Assignment MAC Protocols

Fixed Assignment techniques allocate a permanent channel to each user or traffic flow. Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) can be used to implement a fixed assignment scheme. Each channel would consist of a fixed time-slot per frame or a frequency that would be constantly available to the user. Code Division Multiple Access (CDMA) can also be considered to be a fixed MAC protocol. In CDMA each user is assigned a specific code with which to encode their data, also known as a channelisation code [UMTS01]. All users transmit their encoded information at the same time and using the same spectrum. In

any other system this would result in users interfering with each other to such an extent that communication would be impossible. The codes allocated to users must therefore be chosen in such a way so as to correlate as little as possible with other codes or with time shifted versions of themselves. This allows the receiver to correctly decode the transmission that it is interested in.

The fixed assignment MAC method is wasteful of bandwidth when the user has nothing to transmit, but works well when there is an abundance of bandwidth or when users are constantly transmitting data. It is therefore suited to constant bit rate applications, such as voice, but does not function efficiently when used for applications with a rapidly varying bit rate such as video on demand. Fixed assignment MACs are therefore not suited to carrying multimedia traffic due to their lack of flexibility.

c) Polling MAC Protocols

A base-station using this scheme polls each mobile for its bandwidth requirements and then allocates the available bandwidth based on the replies from the mobiles. This method can cause delays in transmission, as the mobile has to wait to be polled. This delay has negative effects on time critical traffic, such as a video stream in a multimedia connection.

d) Demand Assignment MAC Protocols

This method requires users to declare their requirements before they can transmit information. Bandwidth is then reserved for them accordingly. This process introduces some overhead into the system, as a section of the channel must be used for the transmission of reservation requests and acknowledgements. This section of the channel is known as the reservation channel. Access to the reservation channel is often governed by a random access protocol such as ALOHA. Once bandwidth is reserved the user can transmit using the allocated resources without any collisions with other users. Collisions do occur in the reservation channel. Methods that can be used to minimise these collisions are implemented in certain MAC protocols and will be discussed later in this thesis.

Demand assignment MAC protocols increase throughput in high-load systems but they also introduce a delay into packet transmission. Unlike random or fixed assignment schemes, packet transmission is not immediate. Users must wait for bandwidth to be allocated to them before they can begin transmission. This is a drawback when the load on the network is light and scheduling each packet is not really needed to ensure high throughput.

Demand assignment MAC protocols work well when carrying traffic of differing types because each connection can specify its requirements at connection set-up time. Demand assignment MACs also support traffic types with varying bit rates as they can allow a source to re-negotiate the bandwidth allocated to them. Demand assignment MAC protocols are best suited to carrying multimedia traffic because of their flexibility, support for different traffic types and because they maintain the service required by high priority traffic under heavy load.

e) Hybrid MAC Protocols

Each of the types of protocols already mentioned has both advantages and disadvantages. Some applications demand a combination of more than one of the three major types of MAC protocol to service their diverse requirements. The channel can either be split into separate channels, each using a different type of MAC, or the system could switch from using one MAC to another based on a decision influenced by factors such as network load or the type of traffic being carried [Ayan01].

The line between a hybrid MAC protocol and a demand assignment MAC protocol is a thin one, as many demand assignment MACs use more than one type of access. Some papers refer to demand assignment MACs as a subset of hybrid MAC protocols [Chan01]. Both are mentioned here for completeness.

f) Duplexing Techniques

Wireless systems are often referred to as half-duplex systems. This is because they cannot transmit and receive data in the same channel at the same time. The rapid rate at which the power of a signal decreases in the wireless medium means that the power of the transmitted signal is usually orders of magnitude greater than the power of a received signal. The local receiver will receive a large portion of the locally

transmitted signal energy, preventing it from correctly receiving an incoming signal. This phenomenon is known as self-interference [Chan01]. Self-interference eliminates the option of using collision detection in a wireless MAC. While collision detection schemes work well in wired networks, where the transmitted signal power and the received signal power are of the same order of magnitude, they often fail in wireless networks due to the vast difference in power between transmitted and received signals.

This self-interference forces wireless systems to incorporate a method to separate incoming and outgoing transmissions. This is done using Time Division Duplexing (TDD) or Frequency Division Duplexing (FDD). TDD, as its name implies, separates the incoming and outgoing transmissions by assigning them different time slots. FDD, as implied, separates incoming and outgoing transmissions by assigning them different frequency slots. FDD carries the advantage of being able to send and receive information at the same time, and therefore being able to receive immediate confirmation of packet reception. This is helpful when a packet is errored and needs to be scheduled for retransmission as soon as possible. The receiver can inform the transmitter immediately of any errors that it encounters via the separate channel. It does however add complexity to the system by requiring it make use of two distinct frequency bands for the incoming and outgoing traffic.

MAC protocols incorporate a duplexing technique, either FDD or TDD. Different situations call for the use of different duplexing techniques and hence different MACs. MAC layers can also therefore be classified according to the duplexing technique that they use.

g) Network Layout

MAC protocols can be further grouped by the type of network that they are designed to operate in. Networks that have a central base-station which communicates with many mobile terminals are known as centralised. The master scheduling in a centrally controlled network is usually performed in the base-station. The master scheduler assigns slots to mobile terminals. The mobile terminals can then perform further scheduling within the slots allocated to them. This is referred to as slave scheduling. Slave scheduling differentiates between connections from one mobile, deciding which

of the resources allocated to a particular mobile by the master scheduler will be used by which connection from that mobile.

Distributed networks, such as Ad-hoc networks, have no central control point and no pre-existing infrastructure. All members of the network perform scheduling and consult one another for permission to transmit their packets. There is no master scheduler who dictates the actions of other mobiles. This makes distributed networks more robust than centrally controlled networks, as failure of one terminal does not necessarily affect the rest of the network. Distributed networks do, however, suffer from problems that centralised networks are exempt from. Distributed networks often rely on all terminals being able to “hear” all other terminals within their area of interest. This leads to problems known as hidden nodes, exposed nodes and capture [Chan01]. This also means that some scheduling information transmitted by one member of the distributed network may not reach another member of the same network. The loss of signalling data in this manner can cause collisions when packets are transmitted.

The layout of the network therefore affects the choice of MAC protocol. We can therefore further classify MAC protocols by the network layout that they were designed for.

2.5.2 MAC Layers Proposed in Literature

Many different MAC protocols are proposed in literature, each usually claiming to have features or algorithms superior to other similar protocols. Many of these have been researched in order to evaluate their suitability for use in this project. A short summary of some of these protocols and their features will be presented in this section. This is to provide the reader with a background into MAC protocols before chapter 3, which will focus on the design of the system to be implemented as well as the proposed MAC layer.

As discussed in previous sections, there are many different criteria for classifying MAC protocols. This also means that for each possible type of MAC there exist many different implementations/simulations of MAC layers; far too many to discuss each

one on its own [Chan01, Kubb01, Sanc01, Akyi01, Petr01, Hang01, Sobi01, Akyo01, Geor01, Suh001, Frat01, Yoon01, Khan01, Sing01, Sung01, Iera01, Hoss01, Davo01, Kim001, Chen01, Zhou01]. For this reason the discussion of MAC protocols here will be limited to a few, well-known MACs from which many variations have spawned. This discussion will also move from the initial, less complicated, random access MACs to the more complicated demand assignment MACs, which could be used in a WATM network.

a) The ALOHA MAC Protocol

This protocol is the classic random access MAC as mentioned in previous sections. The ALOHA MAC was proposed in [Abra01], and is still used in some form in many MAC protocols today.

Any wireless node using this MAC will attempt to transmit over the wireless channel whenever it has data. If a collision between transmissions occurs the node waits for a random time period before trying again. This very simple approach only yields an 18 percent throughput [Chan01]. However, subsequent research on this protocol has been performed and it has been discovered that the throughput can be doubled if the available transmission time is divided into time-slots and each node only attempts transmission at the start of a time-slot. This new version of ALOHA became known as Slotted ALOHA, or S-ALOHA. S-ALOHA is often used as the access protocol for reservation contention periods in demand assignment MACs, hence its mention here [Akyi01], [Sanch01].

The ALOHA MAC does not have any support for distinction between traffic classes and can therefore only offer best effort service. Add to this ALOHA's low throughput and instability at high channel utilisation and it cannot be considered for a multimedia carrying WATM system.

b) The Carrier Sense Multiple Access (CSMA) Protocol

CSMA based systems wait until they sense that the medium is not being used before transmitting information. This reduces the amount of collisions that occur. This method is also considered to be of random access type. If one incorporates collision detection into the system (CSMA with collision detection or CSMA/CD) it is possible

to further improve performance [Ayan01]. CSMA/CD is widely used in wired LANs and performs very well in that environment. However, using CSMA/CD in the wireless environment is not as feasible. The received signal power level will vary greatly in the wireless medium, depending on factors such as the distance from the transmitter, interference, fading, the hidden terminal¹ phenomenon etc. It is therefore difficult to either:

- a) sense whether the channel is busy or not before transmitting or
- b) decide whether there has been a collision based on received signal power.

The CSMA MAC does not have any support for different traffic classes and performs badly in the wireless medium. It cannot be considered for use in a wireless multimedia system.

c) Packet Reservation Multiple Access (PRMA)

This protocol was designed to have better support for voice services than the traditional best effort [Chan01]. The uplink medium is slotted and mobiles attempt to transmit at the start of an empty slot with a certain probability. This probability is greater for voice channels than for data channels. If a voice packet transmission is successful the slot used is then reserved in future frames for subsequent packets in the voice burst. No future reservation is made for data packets. In this way, as well as by using different access probabilities, a distinction is made between voice and data, giving voice channels an advantage. PRMA can be seen as an attempt to implement different classes of service, although the division between the two classes is not definite.

PRMA suffers from variable channel access delay, which can adversely affect voice channels [Kubb01]. In PRMA the full channel is used for both requests and for data. This means that when the majority of slots have been reserved, very few slots are available for requests.

¹ This phenomenon occurs when the transmitter is out of range of another transmitter that may or may not be using the same channel. The hidden terminal phenomenon makes it difficult for any one terminal to be confident that the channel is free by simply monitoring the channel.

Modifications to PRMA include PRMA++, where the data and request channels are separated, making the protocol more stable, and Centralised PRMA (C-PRMA) [Akyi01], which uses Earliest Due Date¹ (EDD) scheduling to provide some sort of QoS guarantee [Chan01]. C-PRMA makes the transition from a random access protocol to a demand access protocol.

The separation of traffic into different classes, while still a simple distinction between voice and data in PRMA, is vital to the support of multimedia applications over the wireless interface. The concept of different traffic classes is well supported by ATM, and must be supported in the MAC framework developed in this thesis.

d) Distributed-Queuing Request Update Multiple Access (DQRUMA)

The uplink channel in DQRUMA consists of a request channel and a data channel, while the downlink channel consists of a request acknowledge, or ACK channel, transmit permission channel and a data channel.

DQRUMA uses FDD to separate the uplink and downlink, allowing for very rapid acknowledgement of requests for access and transmission of data very soon after permission to transmit is granted. The uplink data channel also caters for the “piggybacking” of requests to transmit onto data packets [Sanc01], [Chan01]. If a mobile is transmitting data in the uplink channel and it still has additional packets in its buffers to transmit it will set the piggyback bit field of the uplink packet. This informs the base-station, in a contention free manner, that the mobile needs to be assigned further uplink data slots. This reduces the number of requests made in the random access contention channel, thereby reducing the number of collisions that occur, improving throughput and reducing time to transmit delay.

DQRUMA also uses the concept of mini-slots for contention and ACKs [Kubb01]. A full packet slot is not needed for an access request or ACK. The slot is therefore split into a number of mini-slots, each capable of carrying a request for access or an ACK.

¹ Each connection’s traffic rates and delay requirements are used to calculate which connection needs service most urgently. The scheduler will allocate bandwidth to this connection.

This reduces the amount of bandwidth wasted on requests/ACKs and also reduces the chances of two requests for access colliding and requiring retransmission.

The concept of “piggybacking” and the use of mini-slots for contention both improve the performance of a MAC protocol and are hence seen as useful techniques to experiment with in the WATM test-bed.

e) Mobile Access Scheme based on Contention and Reservation for ATM (MASCARA)

This MAC layer was proposed for use in the Wireless-ATM Network Demonstrator (WAND) project. MASCARA uses variable length frames [Sanc01]. Each frame is divided into uplink and downlink sub-frames. The downlink frame carries the frame header and the downlink data. The uplink frame carries the uplink data and the requests for access. The base-station decides on the format of each frame based on the needs of the users. It transmits information regarding the frame size, structure and uplink slot assignments in the frame header. Contention for bandwidth is performed in an S-ALOHA contention channel.

MASCARA introduces the concept of a cell train [Hac001]. Multiple cells from a single connection are transmitted as a single, continuous block of data. This is done to reduce the overhead associated with physical layer overhead for each transmission.

MASCARA, like DQRUMA, uses piggybacking of reservation requests to reduce contention [Akyi01]. MASCARA uses the service class of a connection, its negotiated QoS and Priority Regulated Allocation Delay-Oriented Scheduling (PRADOS) to provide ATM level service to its connections [Sanc01], [Chan01], [Akyi01].

MASCARA, to its detriment, uses large contention packets, resulting in more collisions in the contention channel. Variable length frames also create difficulty in assigning slots to CBR sources, as the number of slots required per connection per frame will change as the frame size changes.

The use of the cell train concept to reduce physical layer overhead is certainly worth investigating in the WATM test-bed. Also, the difficulty with variable sized frames

and large contention packets encourages experimentation with fixed size frames and the use of mini-slots (as in DQRUMA) for contention in the WATM test-bed.

2.6 The Logical Link Control (LLC) Layer

The LLC layer resides above the MAC layer in the protocol stack. Its responsibility is to detect and, if possible, correct or request retransmission of errored data. Although the implementation of this layer is not within the scope of this project it is discussed here for completeness, and so that recommendations on LLC techniques can be made.

In a wired ATM network no payload error checking is done on cells at the ATM layer. The only error checking done is on the header of each cell, using the Header Error Check (HEC) field of the header. This is acceptable as there are generally very few errors that need correcting. However, if this approach was taken in a wireless network the resulting throughput would be very low. The LLC in any wireless network must therefore implement additional functionality to improve the throughput of the system.

The LLC layer often uses a technique called Cyclic Redundancy Check (CRC) to detect errors [Sheu01]. In some cases, a single method for handling detected errors can be applied to all connections. However, as with the MAC layer, if different traffic types are to be supported the LLC layer should handle each traffic type in an intelligent manner that corresponds to that traffic type's requirements. Two main approaches exist to handle errors in the wireless medium. These approaches are discussed in the following sections.

2.6.1 Forward Error Correction (FEC)

This error correction approach relies on the transmission of redundant data across the wireless medium to allow the receiver to correct any errors that are found in the transmission [Scha01]. FEC is generally used as the error correction method of choice when transmitting time critical traffic, i.e. traffic whose usefulness expires after a period of time, e.g. voice. This is because it often takes too long to retransmit the errored packet; hence correcting it at the receiver is the most feasible option. If the LLC fails to correct the cell it will be dropped. This is usually acceptable as traffic types using FEC are often resilient to occasional lost cells. If an errored cell is

correctly fixed using FEC it does not need to be dropped, hence improving the quality of the connection.

The main disadvantage of FEC is that the additional, redundant data appended to each transmission consumes additional resources. A partial solution to this problem is to use a variable amount of FEC, depending on the state of the wireless medium, so as to waste as little bandwidth as possible. Were this solution to be implemented it would require a flexible LLC and MAC layer to support the varying packet sizes that would result.

2.6.2 Automatic Repeat Request (ARQ)

ARQ, as the name implies, improves throughput by automatically requesting a retransmission of cells containing errors. This retransmission takes time, and it is for this reason that ARQ is not often used for time sensitive traffic unless the retransmission can be performed before the usefulness of the cell at the receiver expires. ARQ is more often used for loss-sensitive traffic e.g. data.

There are two classic ARQ schemes, the first is Go-Back-N-ARQ (GBN-ARQ) and the other is Selective Repeat-ARQ (SR-ARQ) [Ohta01]. GBN-ARQ retransmits the first cell in a frame containing errors as well as all following cells in that frame, whether they have errors or not. This is a very simple scheme to implement but it suffers from rapid throughput degradation when there are many errors in the arriving cells. SR-ARQ only retransmits the cells containing errors. While this is far more efficient in terms of the number of cells retransmitted, it requires additional signalling capacity to inform the transmitter which cells contained errors as well as added complexity in the transmitter to find and re-order the cells to be retransmitted.

These two ends of the spectrum in terms of retransmission schemes (low complexity, inefficient and high complexity, efficient) have been combined into a hybrid scheme called Partial selective Repeat superIMposed on GBN ARQ (PRIME) in [Ohta01].

It is generally accepted that most wireless protocols wishing to support multiple classes of service will implement a combination of FEC and ARQ in order to cater to each connection's specific requirements.

University of Cape Town

Chapter 3

System Design of the MAC Framework

Since the general information and issues regarding WATM have been discussed in chapter 2, this chapter will move on to the design of the MAC framework to be implemented. This chapter will first discuss the main goals of the MAC framework. It will then discuss the network topology in which the framework will operate, making comparisons with an existing wireless MAC test-bed at the Technical University of Berlin [Hoen01]. A hardware design adapted from that used at the Technical University of Berlin will be presented. A design of the software modules, including the MAC layer chosen for implementation, will also be discussed.

The main goals of the MAC framework are as follows:

- Support for multimedia through the extension of the ATM classes of service and QoS guarantees over the wireless medium.
- The MAC framework must be capable of processing data at rates of approximately 10Mbps in order to support multiple multimedia traffic streams.
- The MAC framework must be as flexible as possible, allowing experimentation with MAC protocols and methods as well as reconfiguration of MAC algorithms on the fly. This implies that the MAC framework must be flexible at both compile time and run time.
- To facilitate the high bandwidth required, the framework will need to have high-speed interfaces to the LLC and PHY layers, as both of these layers will

be external to the MAC framework. The MAC framework must also facilitate easy integration with future LLC and PHY layers through these high-speed interfaces.

- The MAC framework must be reasonably simple to program and debug, thus increasing the ease of implementation and testing of various MAC algorithms.

The above goals have further implications on the hardware and software design of the system. These implications will be discussed in the following sections.

3.2 Hardware System Design

3.2.1 Network Topology

A centralised system, in which a base-station allocates bandwidth to mobile terminals that fall within its area of coverage, is being considered as the eventual topology for the WATM test-bed (as illustrated in Figure 1). This would allow the system to be used as an extension to the existing ATM network. Mobility support would be required in the ATM switches at the edge of the wired network in order to support handoff of a mobile from one base-station to another. The CRG lab currently has two, fully programmable Gigabit Switches from Washington University (Washington University Gigabit Switch, or WUGS) that could be programmed to provide support for handoff and mobility in future projects. Supporting mobility functionality is not, however, in the scope of this project. For this reason this project will use a single base-station, single mobile terminal topology. This is adequate to perform initial development and testing of the MAC framework and is similar to the topology of the system developed at the Technical University of Berlin, which will be discussed in the next section.

3.2.2 Work done by the Technical University of Berlin

A visit to the Technical University of Berlin (TU) was made during the course of this project to investigate the research into wireless MACs underway there. The hardware topology of the first stage of the WATM test-bed will be similar to the hardware topology of the test-bed at TU shown in Figure 5. However, as far as purpose and functionality is concerned this project is very different from the one underway at TU.

The test-bed built at TU was not designed to support ATM and its various classes of service, nor was it built to cater for high bit rate traffic. Additionally, the research at TU is focused on investigating the qualities of the wireless medium (such as the bit error rate) and is not concerned with testing MAC layer algorithms and approaches. The MAC that is used at TU is therefore designed to simply dispatch packets from a FIFO buffer as fast as their system is capable of transmitting them.

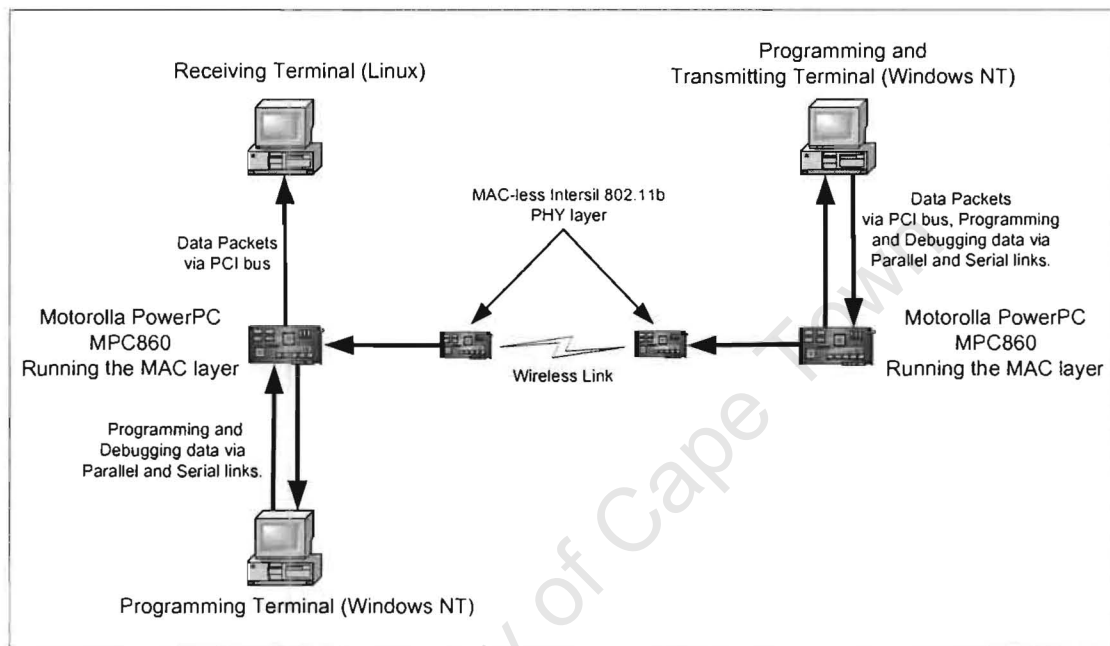


Figure 5: Physical Layout of the Test-bed at the Technical University of Berlin

The requirement of separate PC's to program and to send data to the embedded system is seen as a drawback as it makes system management more cumbersome. The TU system can also only transmit data in one direction, due to problems with receiving interrupts from the microprocessor (MPC860) on a PC running Windows NT.

Furthermore, the research group at TU has experienced problems with the speed of the microprocessor used to implement the MAC layer in their system, as it is not capable of continuously streaming packets to the PHY layer. A delay needs to be inserted between data packets to allow time for interrupts from the PHY layer to be processed. Researchers at TU advised that the MPC860 might not be fast enough to be use in the implementation of a more complex MAC than the best effort, FIFO MAC used in their test-bed.

The Intersil 802.11b hardware that the Technical University of Berlin used in their design, which does not incorporate a MAC layer, is also no longer manufactured. Subsequent versions of Intersil's 802.11 PRISM chipsets [Inte01] incorporate the MAC layer and the base band processor onto the same chip, making them impossible to separate.

An effort was made to find a similar PHY layer (which did not incorporate a MAC layer) for use in this project, but unfortunately none could be found. Developing a PHY layer is not within the scope of this project, and so the physical link will have to be emulated initially. For future projects, the physical layer could be implemented using Orthogonal Frequency Division Multiplexing (OFDM), as mentioned in chapter 2. High-speed systems such as 802.11a [IEEE01] and HIPERLAN 2 [BRAN01] use OFDM instead of CDMA, as it provides better support for high bit rates. This makes OFDM an attractive option for the physical layer of the WATM test-bed [Rama01].

3.2.3 Hardware Requirements

Due to the factors mentioned in the previous section, very little of this project is similar to the research performed at the TU. Modifications to their hardware design have been made based on the goals of the MAC framework mentioned earlier in this chapter. The specific hardware requirements for the MAC framework are therefore as follows:

- Use of an embedded processor capable of processing data/interrupts at rates necessary to support a total data rate of approximately 10Mbps.
- The memory capacity of the MAC framework should be expandable to cater for the possibility of increased memory requirements in the future, thereby ensuring that the MAC framework is scaleable in terms of the number of multimedia flows that can be buffered in the framework.
- A high-speed hardware interface between the embedded processor and PHY layer as well as between the embedded processor and the LLC layer.
- All control, management and user plane communications to a single MAC layer should be possible from a single PC.

- Bi-directional transmission of data between the mobile terminal and base-station must be possible.
- The implementation should also include an emulated PHY layer, as no MAC-less PHY layer could be found.

3.3 Software Design

3.3.1 Primary Software Design Issues

The primary concerns of the software design were:

- The MAC framework must support the implementation of an intelligent scheduling algorithm, not merely best effort or FIFO, in order to fully support ATM, its various classes of service and QoS parameters at the MAC layer.
- The MAC framework must be event driven, enabling synchronisation by means of interrupts or events with the PHY and LLC layers.
- Abstraction from the PHY and LLC layers, which will exist on different hardware platforms to the MAC layer, must exist. In other words, the MAC layer should not need to be changed if the PHY or LLC layer or the interface to these layers changes.
- Flexibility in terms of the data structure sizes and the algorithms used must exist in order to facilitate experimentation with MAC protocols and methods.
- An efficient buffering system that is capable of enqueueing and dequeueing different data types with minimum overhead should be developed.
- Logging management data should be simple and should interfere as little as possible with the real-time data processing being performed by the MAC framework.
- Conversion from a base-station MAC to a mobile terminal MAC should be simple. This implies that the code for the base-station MAC should be almost identical to that used for the mobile terminal MAC. This will simplify code maintenance and any updates in future projects.

3.3.2 Data and Control Flow between Modules

Figure 6 is an image showing the software modules in the system software design and illustrating the interaction between the various modules. It clearly shows the modules

that will be implemented in the MAC layer. Within this figure, thick arrows represent data paths while thin arrows represent control/signalling paths.

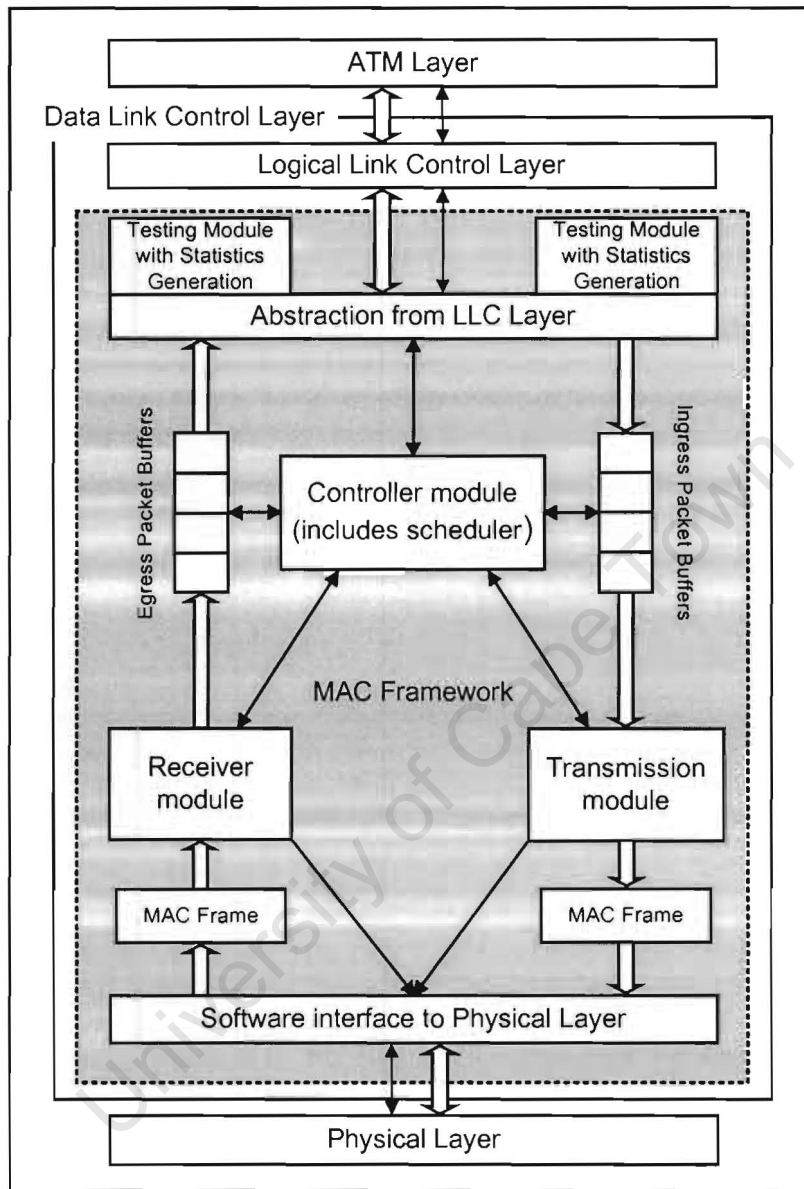


Figure 6: Data Flow Diagram

The transmission module is responsible for the assembly of MAC frames from packets that are passed down from the Logical Link Control (LLC) layer. The controller module makes the scheduling and frame structure decisions, based on information received from the transmission and receiver modules, as well as from higher layers. The receiver module is responsible for disassembly of the received MAC frames and transferring the data into the correct information streams. Data packets are sent to the LLC layer for error checking, error correction and forwarding

on to layers above the LLC layer, while signalling packets are sent to the controller module to aid it in making scheduling decisions.

LLC packets flowing to and from the LLC layer should be buffered at the interface with the LLC layer. An abstraction layer should be used to make the MAC layer independent of the LLC layer. The testing module will emulate some of the actions of the LLC layer in the absence of the LLC layer. This testing module will generate LLC packets and pass them into the ingress buffers if the LLC does not exist. In future, when an LLC layer is included into the project, only the abstraction layer would need to be modified to read packets from the LLC layer instead of from the testing module. The MAC layer itself should remain unchanged. The testing module should also sink packets buffered in the MAC's egress buffers to prevent overflow of these buffers. The abstraction layer should be able to pass these packets onto the LLC layer with a minimum amount of modification. This testing module will also be responsible for logging statistics about the number of packets generated, as well as statistics about the packets received from the MAC layer.

Abstraction from the PHY layer should also be present. It is possible that a standard hardware interface to the PHY layer could provide the necessary abstraction. This hardware interface to the PHY layer has been mentioned in the hardware design.

3.3.3 Inter-module Synchronisation and Messaging

Because of the need for the MAC framework to be an event driven system and to react in real-time to events, it is logical to assume that the MAC layer must be implemented on a processor running a real-time Operating System (OS). The OS chosen depends on the platform and the OSs that the platform supports. Different software modules could be run as separate tasks or processes on the OS. A method of synchronising access to shared resources as well as a means of communicating between modules must be part of the implementation of the MAC framework. The use of semaphores and mailboxes should be possible on the framework, using the OS, to accomplish this synchronisation. Mailboxes and semaphores will also play a vital role in the synchronisation of the MAC layer with future PHY and LLC layers.

A priority system should exist to ensure that critical tasks are given priority access to the CPU and shared data resources. For example, statistics and performance logging should only take place when all critical processes are waiting on semaphores/mailboxes or are idle. (It should, however, be possible to force an immediate log of data when it is important that certain information, such as information about a critical failure, gets logged.)

High-resolution timers should also be included in the implementation. These timers would also have a synchronisation function as well as allowing the performance of the MAC framework to be monitored.

3.3.5 MAC Algorithm Design

A discussion of some MAC algorithms proposed in literature was presented in chapter 2. Many MAC approaches have been studied and a great degree of similarity was found between MACs of a similar type (random, fixed and demand access). It was decided to base the MAC layer on the Wireless Dynamic Weighted Earliest Deadline First (WDWEDF) with two-phase scheduling MAC [Sung01].

a) The Wireless Dynamic Weighted Earliest Deadline First Approach

The WDWEDF MAC layer uses TDD to separate the uplink and downlink data streams and uses different methods to schedule traffic in the uplink and downlink streams. In the downlink stream, it uses Dynamic Weighted Earliest Deadline First (DWEDF) scheduling, which schedules the traffic using a two-stage technique. This algorithm first gives a priority to each connection based on that connection's traffic class (CBR, VBR, etc.). Then it prioritises each connection within each traffic class according to that connection's state. The DWEDF algorithm periodically calculates each connection's state based on the service class of the connection, the required Sustainable Cell Rate (SCR) and Peak Cell Rate (PCR) values for each connection and the current status of the SCR and PCR during the calculation period.

In the uplink stream, the base-station does not have complete control over the connections to make the necessary calculations for DWEDF scheduling. Therefore, the wireless terminal has to schedule its own connections by using the DWEDF algorithm and then provide information to the base-station about its requirements. The

base-station then schedules uplink time to each mobile depending on the information that it receives from the mobile terminals. This uplink scheduling is referred to as “two-phase scheduling,” a term which will be explained further in the next section.

b) Frame Structure of WDWEDF

The structure of each frame is important primarily because it is possible to provide service differentiation by dividing the frame into different sections for each service class. This can ensure that service classes do not interfere with each other. The boundaries between transmission periods are also moveable, allowing the time allocated to a certain service class to be dynamically adjusted on a frame-by-frame basis if needed. This is only possible because WDWEDF uses TDD. In FDD systems it is often impossible to shift bandwidth between classes of service in this manner¹. The frame structure of the MAC also indicates how uplink and downlink data will be separated. Figure 7 shows the frame structure of the WDWEDF MAC. The figure has been adapted to include additional fields and information not depicted in a similar figure in [Sung01].

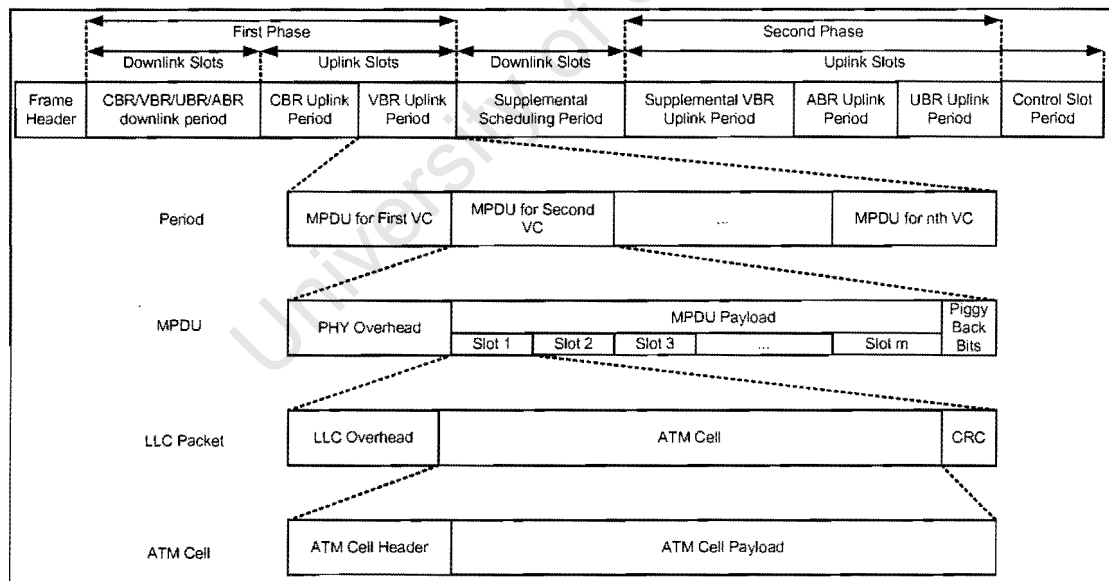


Figure 7: WDWEDF MAC Frame Structure

The WDWEDF scheduler is run on the base-station and on each mobile terminal before each new frame begins. The scheduler prioritises connections within their

¹ Changing the frequency range of a transmission is difficult to do on the fly. It is therefore difficult to change the spectrum allocation of one or more users/streams.

service class. The base-station scheduler also allocates uplink MAC Protocol Data Unit (MPDU) slots to mobile terminals according to the requests and requirements of the mobile's connections. The base-station scheduler then generates the frame header to be transmitted to the mobile terminals. The frame header is responsible for carrying information regarding slot assignment and frame structure to the mobile terminals. Once scheduling is completed, a packing algorithm (operating on both the base-station and mobile terminals) is used to pack LLC packets that have been scheduled for transmission into MPDUs. Each MPDU is made up exclusively of LLC packets from a single ATM Virtual Circuit (VC). In this way all packets going to the same destination are put into the same MPDU. This packing of packets eliminates the need for physical layer overhead before the transmission of each packet. Minimising physical layer overhead is crucial to wireless systems, where bandwidth is scarce. This approach is similar to the "cell-train" concept used in MASCARA [Hac001].

The base-station transmits its packed MPDUs in the downlink period following the frame header. Mobile terminals then transmit their CBR and VBR MPDUs in the relevant uplink period, according to the time slots assigned to them by the base-station in the frame header.

Supplemental (or second phase) scheduling is then performed at the base-station by using information carried in the piggyback field of each VBR uplink MPDU to allocate the supplemental VBR uplink slots. Information about the allocation of these supplemental slots is transmitted to the mobile terminals. Mobile terminals with connections that are allocated supplemental slots then transmit MPDUs in the supplemental uplink VBR slots.

The supplemental VBR period is followed by a period in which any mobiles with uplink ABR or UBR connections that have been allocated slots can transmit the relevant MPDUs in the uplink ABR and UBR periods respectively.

The control slot is a random access channel, (S-ALOHA,) which is used for ABR, UBR or new connections to contend for bandwidth. Collisions often occur in this channel, causing requests to be lost. Subsequent delays in packet transmission could violate the maximum cell delay of a VBR source if it were required to contend for

bandwidth in the same way as the ABR, UBR or new sources. The two-phase scheduling algorithm, along with the piggyback field in the MPDUs effectively eliminates the need for VBR sources to contend for bandwidth. This method also reduces the amount of traffic in the contention channel and therefore reduces the chance of a collision occurring, improving the throughput of requests in the contention channel. The piggyback approach to scheduling additional data slots can also be observed in MASCARA [Akyi01], DQRUMA [Sanc01], [Chan01], and many other MAC protocols.

3.3.4 Data Structures and Memory Access

Because the MAC framework will be expected to process data at approximately 10Mbps, it is important that both data structure usage and memory accesses be carefully controlled. The most important data structures (the ones that will be accessed most) in the MAC framework will probably be data buffers. For this reason a generic buffer type should be defined that is capable of handling any size of element and any number of elements, memory permitting. These values will be passed to the buffer's initialisation function. The buffers will be implemented as fixed size circular queues. This type of buffer will require very little dynamic memory allocation¹ and traversing, when compared to linked lists, and very little re-organisation, compared to an array where data must be re-shuffled frequently.

Buffers should also use zero copying² in their enqueue and dequeue procedures. Any globally accessible buffers should also use semaphores to limit simultaneous access to global data. Buffers should also allocate all memory resources that will be needed by them during initialisation.

Other data structures such as packets, frames or MPDUs should be simple to modify. More specifically, it should be possible to change the packet size or MPDU size by changing a single line of code. This change must then propagate through to all other data structures affected by the change. The size of a frame should also be modifiable

¹ Dynamic memory allocation causes memory fragmentation and involves significant delay.

² No buffer elements should be copied, as this involves a high CPU overhead. Instead pointers to buffer elements should be used to enqueue and dequeue elements.

at compile time. Divisions within the frame, mentioned in the previous section, should be moveable at run time, allowing dynamic structuring of each frame according to the desires of the scheduler.

The arrangement of data in memory is also crucial to the performance of the MAC framework. In most embedded systems there is more than one type of memory. Usually there is at least a fast, onboard cache and a slower, external memory block. The implementation of the MAC framework must use these different memory blocks a way that makes the most use of the faster memory types. Less critical data should be stored in the slower memory, making space for critical data structures in the faster memory.

University of Cape Town

Chapter 4

Implementation of the MAC Framework with Weighted Round Robin Scheduling

4.1 The Choice of an Embedded System

The choice of an embedded system on which to implement the MAC framework was very important, as the embedded system chosen could limit the framework's capabilities. Research into various available embedded processors has therefore been conducted. It is essential that the embedded system chosen meets the hardware design requirements addressed in chapter 3.

The MPC8260 embedded processor from Motorola was initially considered, as it is the next version of the processor that TU used in their system. However, it could not be directly connected to a host computer via the PCI bus. This denies the framework a high bandwidth connection to the LLC layer, which is to be implemented on the host computer.

The Texas Instruments (TI) TMS320C6xxx range of DSPs was also considered. This range consists of the 'C62xx, 'C67xx and 'C64xx subsets of DSPs. Of these, only the 'C62xx and 'C67xx have Evaluation Modules¹ (EVMs) that can be directly connected to the PCI bus of a host computer. The host then has access to all on-chip RAM, peripherals, and any external memory present via the PCI bus [TI190d]. All programming, debugging and transmission of data between the host and the EVM can be performed using the PCI bus, meaning that only one host computer will be required per EVM. This arrangement also allows for bi-directional data transfers, as mandated in the design. The burst data rate of the PCI bus is 132M bytes/second [TI305], meaning that the speed of the PCI bus would not be a limitation on the system. The onboard Direct Memory Access (DMA) of the 'C62xx and 'C67xx processors is capable of servicing the PCI bus directly [TI190d], thus eliminating the need for the CPU to handle PCI transfers. More information on the onboard DMA controller is provided in the appendices.

The 'C6xxx DSP family also has extension ports (Multi-channel Buffered Serial Ports or McBSPs) that would allow communication with the physical layer at speeds of up to 100Mbps [TI455a].

The EVM/DSP combination has 128KB of internal memory along with 256KB of external SBSRAM and 8MB of external SDRAM. Additional SDRAM could be added, up to a total of 512MB [TI190d]. The system memory is therefore expandable and can cater for possible increases in the memory requirements of the MAC framework.

The primary difference between the 'C62xx and 'C67xx is that the 'C62xx family of processors uses fixed-point arithmetic, making it slightly faster than the 'C67xx family. The 'C62xx family of processors is capable of running at 200MHz, (up to 1600 MIPS) allowing it to process data at high speed.

The 'C62xx fixed point DSP family was hence chosen as the family suitable for use in the MAC framework. Evaluation modules (EVM) using the 'C6201 DSP are therefore

¹ Boards holding the DSP that support rapid development and testing with the DSP

used in this project. The EVMs use a 160MHz version of the 'C6201 processor. The EVM also provides onboard JTAG emulation. The TI SN74ACT8990 JTAG test bus controller (TBC) provides control of the 'C6201 JTAG interface and can be accessed via the PCI bus. This means that a debugger can be used on the EVM without the need for an external JTAG emulator [TI305].

4.2 Development of a Real-time Environment for the MAC Framework

Simple programming and debugging is vital if the test-bed is to be easy to modify. TI's Code Composer Studio 2 (CCS) was investigated and found to be an ideal Integrated Development Environment (IDE). CCS performs all of the essential operations (compiling, assembling, linking and downloading of code to the target board) in a seamless fashion [TI301c].

CCS also provides the programmer with a very comprehensive debugging environment. CCS is capable of interfacing with the DSP through the PCI bus while the DSP is running. The DSP's current state as well as the state of any relevant variables, processes or memory blocks can be remotely monitored by CCS at any point during the execution of the program, providing the programmer with useful debugging information. Figure 8 is a screenshot taken while debugging the MAC framework.

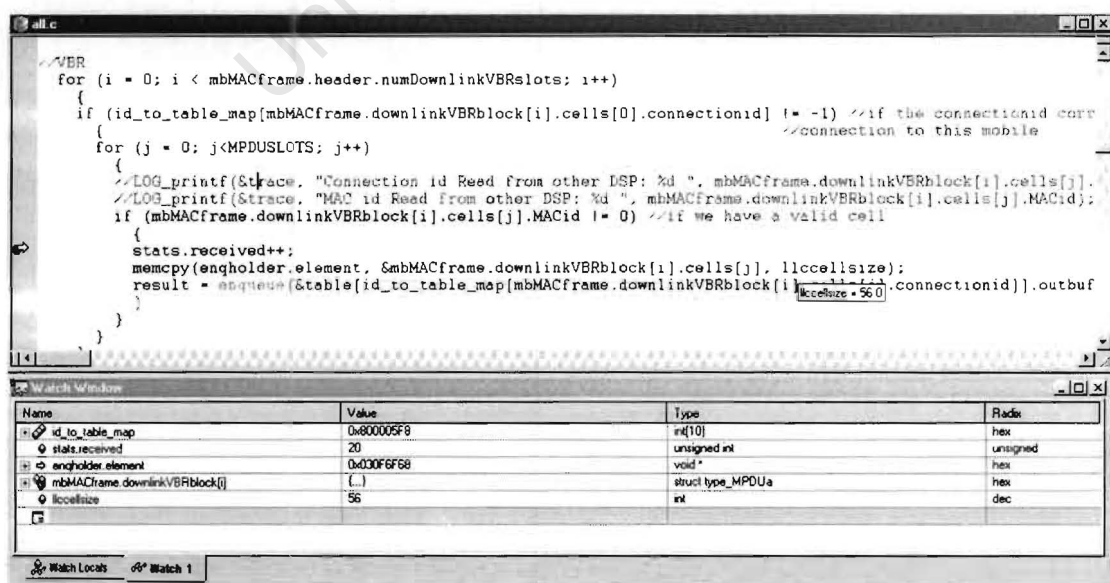


Figure 8: Debug Environment of CCS

As mentioned in the design chapter, the MAC framework needs to run on a real-time system. CCS includes its own real-time kernel, DSP/BIOS [TI423a], as well as a number a graphical kernel configuration tools and various methods of monitoring the processes running on the kernel. DSP/BIOS is fully scalable, allowing the programmer to include and configure objects supported by DSP/BIOS before compile time (These objects can also be created and configured at run time if the programmer desires). This graphical configuration tool is illustrated in Figure 9.

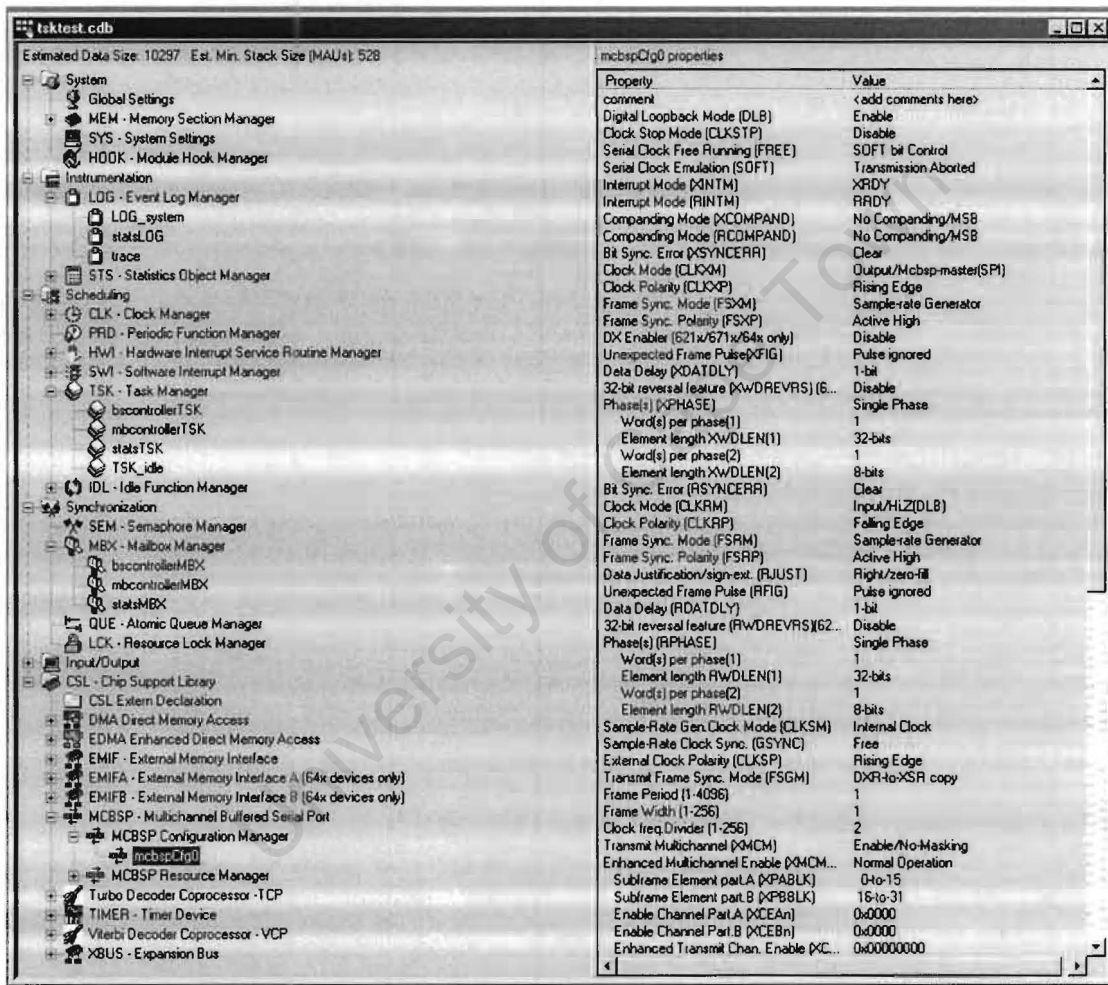


Figure 9: Graphical Configuration Tool in CCS

DSP/BIOS also caters for real-time task scheduling, synchronisation, analysis and instrumentation. DSP/BIOS allows prioritisation of the different functions of the MAC framework according to their importance in the system through the use of hardware interrupt handlers, software interrupt handlers and tasks (sometimes called processes) with different priorities. All of these objects can be set up and configured using the graphical configuration tool before compile time, minimising the amount of

code that must be downloaded to the board as well as the time spent initialising objects at run time. Furthermore, DSP/BIOS provides various methods of inter-process communication, such as semaphores and mailboxes. This allows the various processes that are used in the MAC framework to be synchronised with each other as well as with the data transfers to and from the LLC and PHY layers. These structures are also used to prevent simultaneous access by two processes to shared, global data. A kernel/object view, shown in Figure 10 provides detailed information about the kernel status, tasks, mailboxes, semaphores, software interrupts and memory that DSP/BIOS is using.

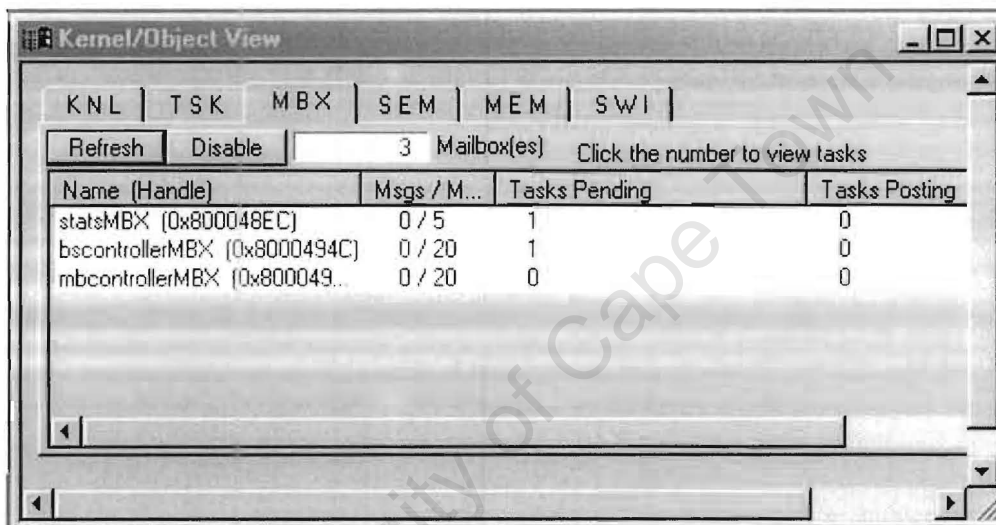


Figure 10: Kernel/Object view in CCS

The time taken and/or instruction cycles used for any block of code to complete execution can be monitored using the tools embedded in Code Composer. This in turn allows the time that any stage of MAC framework processing takes to execute to be monitored, which is vital when attempting to adhere to real-time deadlines and guarantees.

Performance data collected by the MAC framework and any other outputs from the framework can be logged to the host computer, using the host-DSP communications that Code Composer Studio provides. This information can then be viewed in real-time (in the form of a windowed display within Code Composer Studio) or stored in a file on the host computer for analysis after the DSP has been stopped. Some real-time data logged from the MAC framework is shown in Figure 11.

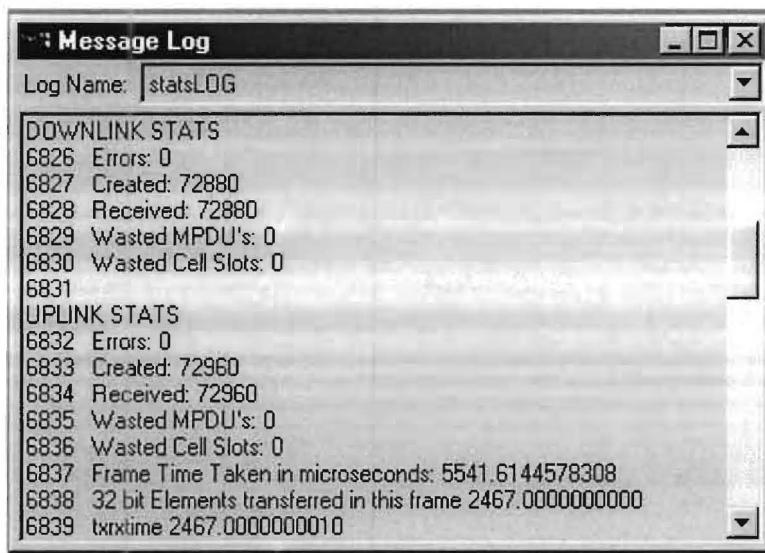


Figure 11: Logged Data from the MAC Framework

CCS's graphical configuration tools also allow the strategic placement of objects in memory. When an object is created using the graphical configuration tools the programmer can decide which block of memory the object should reside in. This allows for the placement of frequently used objects or variables into the faster, onboard SBSRAM memory of the DSP, while less important data can be moved out of the SBSRAM, freeing more SBSRAM for more important data. This flexibility increases the performance of the MAC framework.

All memory allocated for MPDUs, buffers etc. is based on the packet size value. It is possible for the framework to be used to transport any type or size of packet (within limits placed by available memory on the buffer and frame sizes) simply by changing this packet size value. All data structures used by the MAC framework will automatically resize at initialisation time to accommodate the new packet size. Any changes to the format of the LLC packets will automatically cause the packet size value to be modified accordingly, making the MAC framework capable of supporting a wide range of packet sizes and formats.

4.3 MAC Framework Architecture

Figure 12 illustrates the architecture of the system that has been implemented. This figure combines hardware and software architecture components to show how the hardware and software components of the architecture fit together.

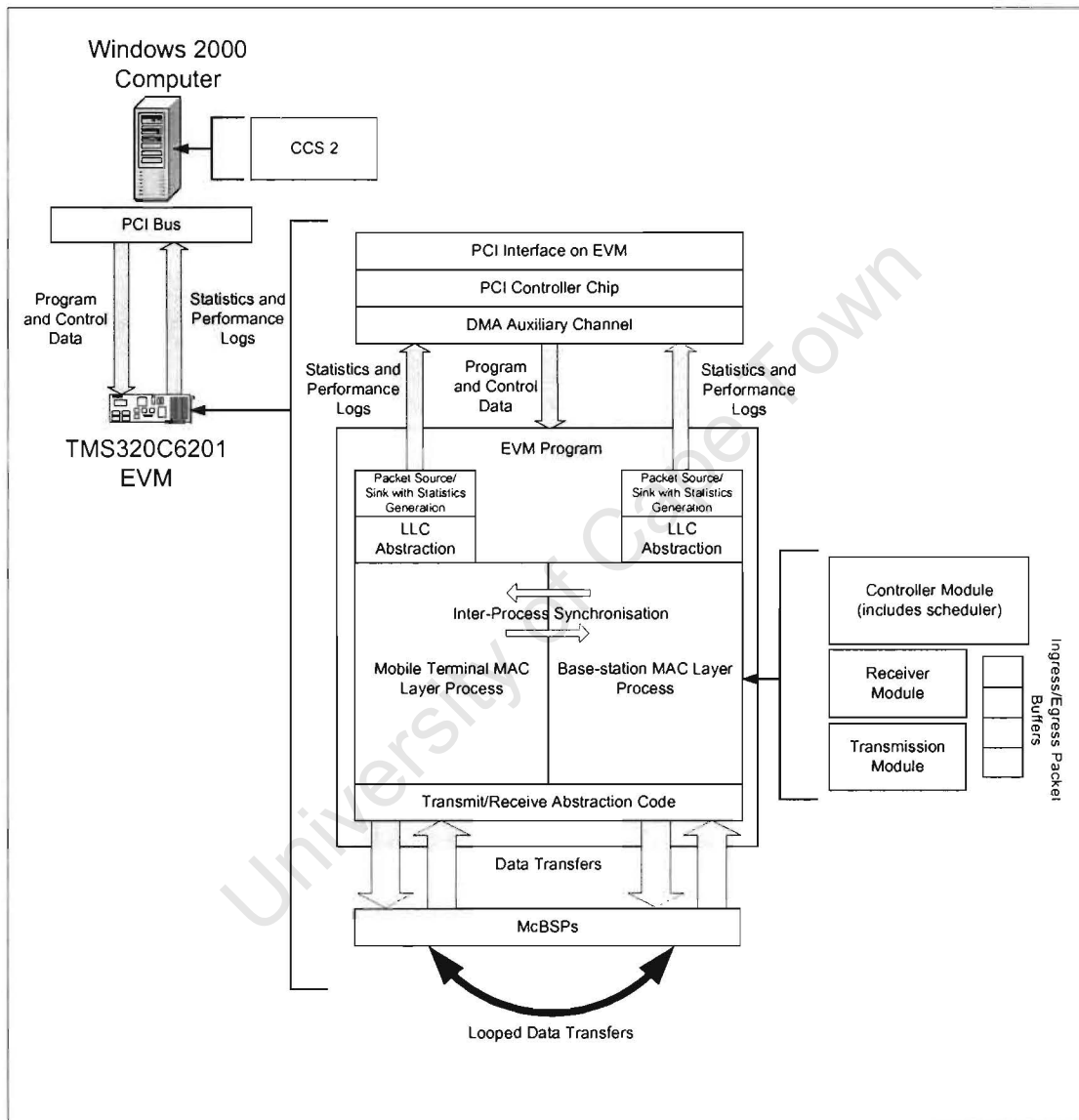


Figure 12: Architecture of the MAC Framework

4.3.1 Hardware Configuration

As previously mentioned, the TMS320C6201 EVM board is used to implement the MAC framework. The EVM is connected to a host computer running CCS via the PCI bus. This host performs the programming and debugging of the MAC framework on the EVM and receives the statistics and performance log information. In future projects this computer could also be used to implement the LLC layer. The EVM sends and receives data to the PCI bus via its PCI controller chip [TI305]. In the EVM, all data transfers from the PCI chip are handled by the auxiliary channel of the DMA controller [TI190d]. The DMA auxiliary channel and the PCI controller chip used in PCI transfers are transparent to the program running on the DSP. DSP/BIOS objects handle the transfer of DSP state and debug information and performance logs to CCS on the host.

4.3.2 Packet Source/Sink with Statistics Generation and LLC Abstraction Modules

The packet source generates ingress LLC packets and passes them to the LLC abstraction layer. The packet source also updates statistics concerning the number of packets generated. The packet sink receives packets from the LLC abstraction layer and generates statistics regarding the packets received.

The LLC abstraction layer provides an interface to the MAC layer. It receives packets from the packet source or LLC layer, depending on whether the LLC has been implemented, and enqueues the packets in the relevant connection's ingress buffer. This layer is also responsible for removing packets from a connection's egress buffer and passing them on to the user of the MAC layer i.e. the packet source/sink or the LLC layer.

If the LLC layer were implemented on the host the LLC abstraction layer could easily be modified to use the DMA controller to transmit egress packets, via the PCI bus, to a LLC layer on the host. This layer could also respond to interrupts from the DMA controller, signalling the arrival of an LLC packet from the host, and place the packet

in the ingress buffers of the MAC layer. The design for the LLC abstraction layer is therefore met.

The packet generation and removal mentioned above takes place once per MAC frame. LLC packets are generated at the maximum rate that the MAC framework can process them. All available packet slots and MPDUs are used. This allows testing of the MAC framework's maximum throughput.

The only part of the LLC packet that contains data relevant to the MAC framework is the LLC header. Each packet is given a unique LLC header so that it can be easily tracked as it moves through the MAC framework.

4.3.3 PHY Layer Abstraction and Emulation

Abstraction from the PHY layer is provided by means of a software layer (called the "Transmit/Receive Abstraction code" in Figure 12) that handles all transmissions and receptions of data to and from the PHY layer. PHY layer emulation is performed by the McBSP output available on the EVM. In the MAC framework, the McBSP is operated in Digital Loop Back (DLB) mode, i.e. the output of the McBSP is connected to the input of the same McBSP. This arrangement allows the mobile terminal MAC layer and the base-station MAC layer to reside on the same EVM during development of the MAC framework. Transmissions between the MAC layers still travel via the McBSP, as they would if they were being transmitted over a wireless PHY. This approach has the following advantages:

- All MAC framework development and testing can be done on a single EVM. Only one processor needs to be re-programmed each time a change is made to the code. As separate researchers are working on the MAC framework at the same time this arrangement allows each EVM to be used to develop and test separate aspects of the framework, thus speeding up development by removing the need for researchers to share resources.
- A physical connection between two EVMs does not need to exist for meaningful testing of the MAC framework to take place. Building a temporary, wired, physical connection between two EVMs (which would be replaced by the wireless PHY in the future) would be unproductive. Having a wired PHY between EVMs

would also affect the results obtained while testing the MAC framework, possibly obscuring issues in the MAC framework itself that would need to be resolved.

4.3.4 The Mobile MAC Layer and Base-station MAC Layer Processes

a) Communication between Modules Within the Mobile and Base-station MAC Layer Processes

The mobile MAC layer and Base-station MAC layer processes execute code for the various modules of each MAC layer. These modules and their functions were discussed in chapter 3 and illustrated in Figure 6.

Communication between these modules within a MAC layer is performed using function calls. The parameters of the function call are used to pass any necessary information between modules. It is possible to create processes to run each module, making use of mailboxes to communicate between modules; however, this approach was not used for the following reasons:

- Running all modules on a single process is faster than having a process for each module. This is because no context switching is necessary to switch between modules.
- The processing performed in the MAC layers is of a serial nature i.e. algorithms must be run one after the other. There is therefore no advantage to using processes for each module as none of the algorithms used in the MAC layers can be run in parallel with each other.

b) Synchronisation between the Mobile and Base-station MAC Layer Processes

The base-station MAC layer and the mobile terminal MAC layer are run on separate processes on the DSP/BIOS kernel. Synchronisation between the mobile MAC layer process and the base-station MAC layer process must exist in order for data transfer to take place, both in the case where they reside on the same EVM and in the case where they are on separate EVMs. This synchronisation is performed by using mailboxes. A message is posted to these mailboxes when a block of data is ready to be read in by the relevant MAC layer process.

Figure 13 shows how synchronisation between the mobile terminal and base-station MAC layer processes was achieved. In this figure the terms “First Phase Of MAC Frame”, and “Second Phase Of MAC Frame” refer to the phases of the frame shown in Figure 7.

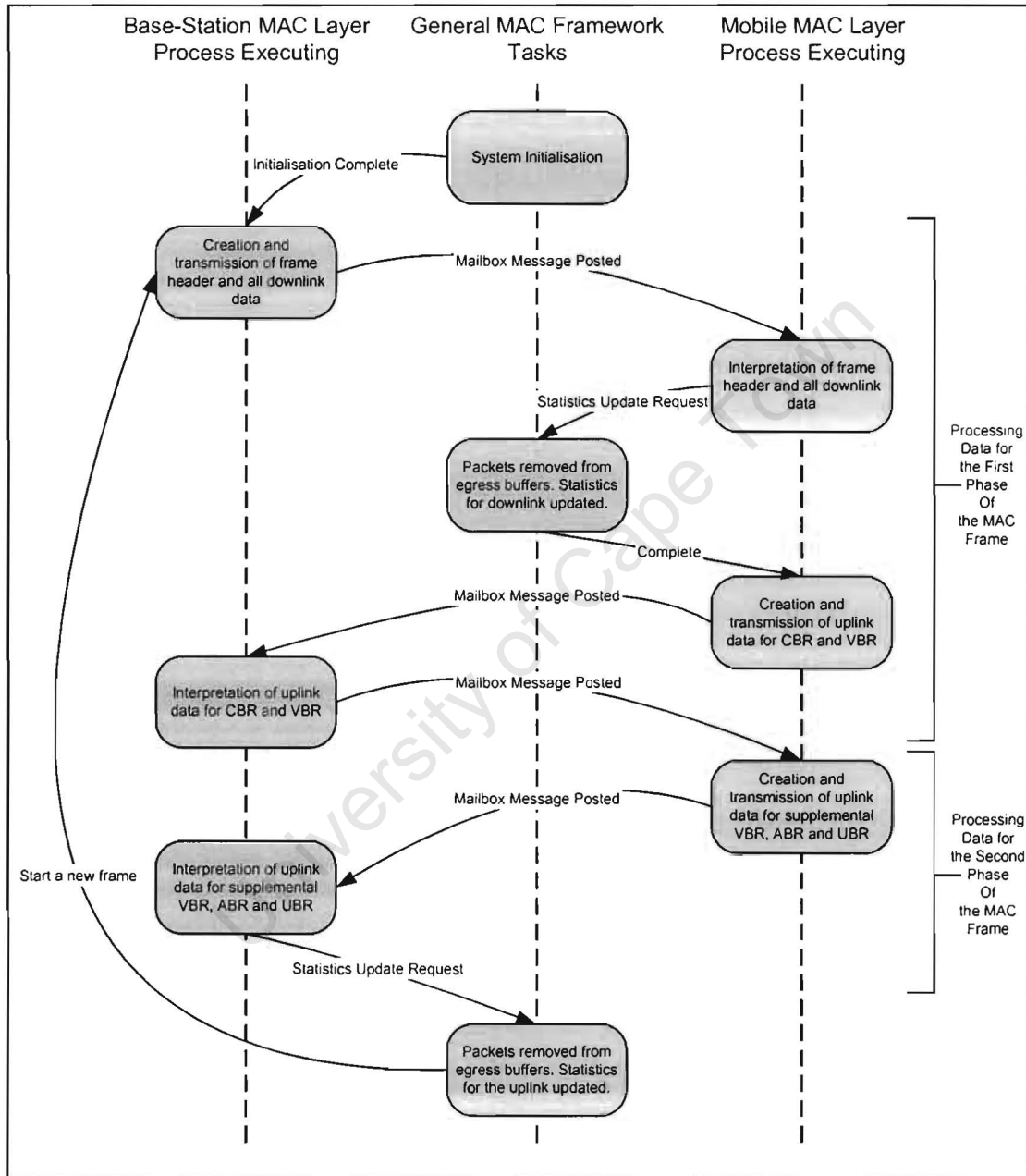


Figure 13: Process Synchronisation Diagram

Conversion from a mobile MAC to a base-station MAC is performed by changing the functions called by each process at various stages. The rest of the code base is identical, making code maintenance simpler.

4.3.5 MAC Framework Changes when a Wireless PHY is Added

The only changes that will need to be made to the MAC framework when a wireless PHY is added to the system are:

- The McBSP would no longer operate in DLB mode, but would be connected directly to the wireless PHY layer. The wireless PHY must be designed to be able to interface to the McBSP of the EVM.
- Hardware Interrupt Service Routines (ISR), called when a block of data arrives from the wireless PHY via the McBSP, would post messages to the mailboxes controlling the MAC layer processes. This would be necessary because the MAC layer processes could no longer post messages to each other's mailboxes, as they would reside on different EVMs.

4.4 Creation of Connections

A connection in the context of the MAC framework refers to the structures in EVM memory that need to be created and initialised to facilitate data flow from the packet source to the relevant packet sink. Creating a connection involves, among other things, creating ingress, egress and MPDU buffers for the connection. Connections are classified as CBR, VBR, ABR or UBR so that they can be treated according to the priorities associated with these ATM traffic types.

In the MAC framework the initialisation of connections is done once and is not repeated unless the framework is reset. No dynamic creation or destruction of connections is performed. The number of connections initialised per traffic class is determined at compile time. Dynamically creating or destroying connections is definitely possible on the framework, but, as no signalling for the creation or destruction of connections is currently received by the test-bed, this functionality has not been implemented.

All connections are arranged in a table of connection entries. Each connection is identified by a connection ID. When a connection is created this ID is mapped to an entry in the table of connections. When a MAC frame arrives at a terminal that terminal checks to see which MPDUs belong to connections that exist in that terminal. This is done by checking the connection ID of the first packet in the MPDU against

the connection ID to connection table entry map. Only relevant MPDUs are extracted from the MAC frame and unpacked. Processing time is not wasted processing unwanted MPDUs.

4.5 Buffering System

A buffer class has been defined for use in this thesis. Buffers belonging to this class can be initialised to hold any type of element. One set of enqueue and dequeue buffer functions cater for buffers of all different data types. This is achieved by using void pointers as well as the size of the element that the buffer stores in the buffer functions.

These buffers pre-allocate all memory that they will need when initialised and perform pointer copying (also referred to as “zero-copying”) instead of copying an entire element on enqueue or dequeue. This minimises time spent enqueueing and dequeueing elements in the MAC framework.

The enqueue and dequeue functions are not atomic (they can be pre-empted by the kernel) and hence the buffers need to be protected from simultaneous access by two different processes. Each buffer is protected from simultaneous access by a semaphore. Additional semaphores are used to protect the holding elements that enable pointer copying. All buffers containing the same type of element share these holding elements, which is why the holding elements themselves need to be protected from simultaneous access. Further details regarding the buffers used in the MAC framework can be found in Appendix C.

4.6 Packing Algorithm

A packing algorithm is included in the WDWEDF system [Sung01]. However, it is not actually part of the scheduler mechanism. The packing algorithm and the scheduler can operate independently from one another. The packing algorithm is therefore seen as a support function for the scheduler, and hence forms part of this implementation.

The packing algorithm used in this implementation differs slightly from that discussed in [Sung01]. The packing algorithm used in the MAC framework uses multiple, fixed

size MPDUs for each connection instead of the dynamically sized MPDUs suggested in [Sung01]. This change reduces the amount of dynamic memory allocation that takes place in the MAC framework. Dynamic memory allocation is costly in terms of the time that it takes to perform and will also result in increased memory fragmentation. If too much memory fragmentation occurs it may become impossible to allocate a large enough contiguous block of memory for an MPDU, in which case the MAC framework would fail.

Unfortunately, there is also a disadvantage to using fixed size MPDUs. If there are not enough LLC packets in the ingress buffer of a specific connection to fill an entire MPDU then the remaining packet slots in that MPDU must be packed with dummy packets. The packing algorithm performs this dummy packet packing and modifies the “wasted packets slots” statistic accordingly. Dummy packets are not unpacked at the receiver, limiting the time wasted processing these dummy packets.

4.7 Weighted Round Robin Scheduler (WRRS)

A simple, weighted, round robin scheduler is used to test the functionality of the MAC framework. Implementation of the WDWEDF scheduler and investigating its effectiveness in dealing with multimedia traffic streams is underway in other research and is therefore not performed in this thesis [How01]. No WDWEDF scheduling is therefore performed before the packets are packed into MPDUs.

The WRRS allocates MPDU transmission slots to each connection. As shown in Figure 7, the transmission frame is divided up into flexibly sized sections, each section transporting traffic belonging to a certain class of service. The MPDU slots in each section are assigned to connections of that section’s class of service type in a round robin manner. Each service class is hereby kept separate from the other service classes. UBR traffic, for instance, cannot occupy MPDU slots assigned for CBR traffic.

If an MPDU slot has been allocated to a connection and that connection has no MPDUs to transmit then that MPDU slot is marked as unused in the frame and the “wasted MPDUs” statistic is increased accordingly. An unused MPDU is not

extracted from the MAC frame at the receiver, as this would be a waste of processing time.

4.8 Statistics Generation and Performance Logging

It is important to be able to track quantities such as packets transported, packets errored and the time taken to complete a certain phase of the MAC framework's processing. Packet statistics are updated in the packet source/sink module, as mentioned in section 4.3.2.

The efficiency of the MAC framework in terms of bandwidth wastage is also monitored at various points in the framework's algorithm. The number of packet slots in MPDUs that are not used as well as the number of MPDU slots that are reserved for a connection but never used are monitored and logged to the host. These values will be useful when implementing the WDWEDF scheduler, as they will help the WDWEDF scheduler to make decisions on issues such as frame size, frame section size and how many MPDU slots are allocated to each connection.

A method of timing the execution of sections of the MAC framework algorithm has been implemented. A hardware clock on the EVM boards increments a counter variable once every 4 CPU clock cycles. Assembly functions are used to retrieve the value of this counter. This value is used to measure the passage of time. This method allows time to be measured with a granularity of approximately 6.25ns [TI269d]. Since the MAC framework is only concerned with time intervals of the order of microseconds, this granularity is adequate.

Chapter 5

Framework Tests, Results and Modifications

5.1 Functional Testing

Functional testing consisted of tests designed to verify that the MAC framework operated according to its design. This testing was performed in order to discover flaws in the flow of control signals or data through the MAC framework. Functional testing did not concern itself with the speed or efficiency of the MAC framework. Instead, it focused on verifying the integrity of the MAC framework over a wide range of possible scenarios.

5.1.1 White Box Testing

White box testing requires intricate knowledge of the code and the intended function thereof. White box test strategies ensure that all lines of code are executed at least once or that all functions are called [Dobb01]. This often requires modification of the program to force unlikely conditions to occur and close monitoring of variables and resources used.

White box testing of the MAC framework was performed by extensively stepping through the code and monitoring its actions as well as by outputting the value of important variables to a log at various stages of code execution. In this manner, the different modules of the MAC framework were tested to ensure that they performed as desired internally. All functions were called and all lines of code within each function were run, complying with the definition of white box testing.

5.1.2 Black Box Testing

In black box testing, one is oblivious to the inner workings of the object being tested. Wide ranges of possible inputs are given to the object and its response to these inputs is compared to the ideal response [Dobb01].

Black box testing of each functional block of the MAC framework was performed. The block being tested was placed into a test harness designed to generate the necessary inputs and to measure the outputs of the functional block. This testing was performed to gain confidence in each functional block before it was integrated with the other functional blocks in the MAC framework. Specific focus was placed on supplying edge/corner¹ case and pathological² inputs to the functional blocks, to test how well they would respond to extreme test cases.

Further black box testing of the entire MAC framework was performed once all of the functional blocks had been tested and integrated. This testing involved creating a varying number of connections (connections in the same context used in section 4.4) belonging to the different service classes. LLC packets were generated at varying rates for each connection. The LLC packets then flowed through the MAC framework and into the egress buffers of the relevant receiver. After the packets were dequeued by the LLC abstraction layer they were checked to ensure that they corresponded with the packets generated and that they were in the correct connection's egress buffer.

¹ Inputs on the "edge" of all of the possible inputs

² Inputs judged likely to cause the functional block being tested to fail

White box and black box testing of the MAC framework revealed some errors, which were subsequently corrected. The MAC framework was then tested for speed and efficiency, which will be discussed further in the following sections.

5.2 Instruction Cycle Count

Initially, when the MAC framework code was implemented, it was being run on CCS's DSP simulator. Testing was performed in order to discover whether it would be possible to perform accurate speed and efficiency tests on the simulated DSP. This would negate the need for an EVM board to perform speed and efficiency testing in future projects.

CCS provides code-profiling tools. These tools can be used to calculate the number of assembly instructions needed to run certain functions or blocks of code on either the DSP or the simulated DSP. These profiling tools were used to calculate the number of assembly instructions needed to execute the major steps in creating and copying a MAC frame to a specific destination. The frame was copied and not actually transmitted via the McBSP, as the DSP simulator used did not simulate the McBSP. If the simulator was accurate, the number of instructions taken could be used, along with the MIPS rating of the DSP being simulated, to calculate an approximate speed at which the frames were being processed.

The test case took *119819* assembly instructions to execute on the simulated DSP environment according to the profiling tool CCS provides. However, when the same code was downloaded to the DSP and the same profiling tests were performed, the test case took *658578* instructions to execute, as shown in Figure 14. The reason for this discrepancy is the fact that the simulator uses a greatly simplified model of the DSP in order to reduce the time taken to run code on the simulated DSP.

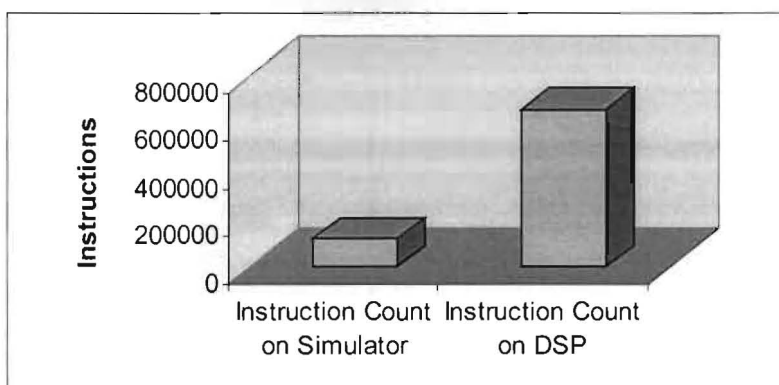


Figure 14: Instruction Count Comparison

Having shown that the simulator is not an accurate tool on which to perform speed and efficiency measurements, all further speed and efficiency tests were conducted on the DSP itself.

5.3 Execution Speed on the EVM

The main concern when testing the MAC framework was the maximum speed at which it could process data from the packet sources to the packet sinks (illustrated in Figure 12.) Further measurements with regard to the QoS offered to multimedia traffic can only be conducted when the WDWEDF scheduler is in place, and will therefore be evaluated in other research [How01]. For this thesis, a major concern was whether the MAC framework could schedule and process data at a speed that would support multiple multimedia streams.

The timer functionality built into the MAC framework and discussed earlier was used to measure the rate at which frames of data could be processed and transmitted between the packet sources and the packet sinks. Some important factors that influenced these results are:

- Since both the mobile and base-station MAC layers reside on the same EVM, the McBSPs have to handle all transmission and reception of data for *both* MAC layers. This also ties up the device servicing the McBSP, be it the CPU or the DMA controller, for twice as long, as both transmission and reception of the same data must be handled. This factor increases the time taken to transmit data between the MAC layer processes.

- All processing that needs to be performed (packing, round robin scheduling, unpacking etc.) for both MAC layers is performed on a single DSP. This increases the time taken to transmit data from the packet sources to the packet sinks.

Both of the above factors will no longer exist when the wireless PHY is developed and the different MAC layers are on separate EVMs. A significant increase in the speed at which the MAC framework can process data can therefore be expected when the MAC layers are migrated to separate EVMs.

Initially, all processing, transmitting and receiving for a single, default¹ frame took approximately 19000 microseconds. Modifications to the MAC framework reduced this time to about 6822 microseconds. These modifications included:

- Using two McBSPs in tandem, transmitting sequential elements via a different McBSP. This reduces transmit and receive time drastically, as the McBSPs operate in parallel with each other. The possibility of using more than two McBSPs was not possible because only two McBSPs exist on the EVMs.
- Initially, the CPU serviced the McBSP. The DMA controller was then used to service the McBSP and the frame time compared with the frame time when the CPU serviced the McBSP. No significant difference was found between the two times. This could either be because the time taken to service the McBSP is insignificant when compared to the total time taken to process an entire frame, or because the time taken for the DMA to service the McBSP is almost identical to the time taken for the CPU to perform the same task. It was decided that, since the CPU has to wait for the data transfer to complete before continuing processing, the CPU should be used to service the McBSP.
- Configuration of the McBSP ports was fine-tuned² by using the graphical configuration tool in CCS until the optimal transfer rate was achieved.
- As there is already an overhead associated with the transmission via the McBSP, it was decided that the PHY overhead associated with each MPDU in the WDWEDF simulation [Sung01] could be discarded. Although the PHY overhead

¹ LLC payload of 48 bytes, four connections, four MPDUs per connection, five packets per MPDU

² The element length for the frame and the frame period, frame width and clock frequency divider of the sample-rate generator were varied.

will change when using a wireless PHY, it is not possible to predict how much it will change as this is highly dependant on the PHY layer implementation. No assumptions about this change will therefore be made here.

The semaphores used to protect the buffers as well as those protecting the enqueue and dequeue elements were also removed with no adverse affects to the integrity of the buffers. This is because the semaphores synchronising the base-station MAC process with the mobile MAC process prevent simultaneous access of any of the buffers or the enqueue and dequeue elements. This change did not affect performance noticeably.

5.4 Data Rates Achieved on the MAC Framework

The number of 32 bit data elements transmitted per frame was also measured and used with the frame rate to calculate the rate at which data was being transmitted and received from the packet sources to the packet sinks. For the frame speed mentioned in the previous section (for the default frame), i.e. 6822 microseconds per frame, 2291 32-bit elements were transmitted per frame. Note that the amount of time taken per frame includes all transmit and receive time as well as all time needed to process the data. Thus, the MAC framework can process, transmit and receive all data at a rate of 10.75 Mbps under these test conditions and with the default packet size and frame size. (The packet size and frame size were varied and the results of these variations are presented later in this chapter.)

As the frame rate is used to calculate this value, the data rate achievable on the MAC framework will be affected by the same factors affecting the frame rate mentioned in the previous section.

The rate at which information was being transmitted and received only, i.e. the time taken for the McBSP to transmit and receive all information in one frame, was also measured. It was found that the McBSP could transmit and receive elements in DLB mode at an approximate rate of 33 Mbps. This rate should also improve when transmission and reception of a section of the MAC frame are performed by different McBSPs on different EVMs.

The respective data rates for transmission and reception only and for all processing (including transmission and reception) of a default frame are compared in Figure 15.

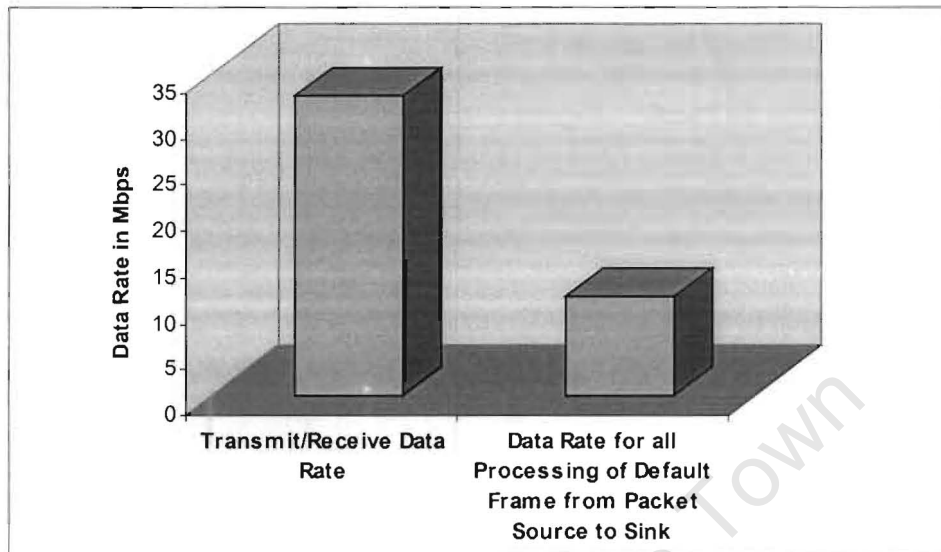


Figure 15: Transmit/Receive Data Rate vs. Total Data Rate for a Default Frame

5.5 Flexibility Tests and Results

The size of the LLC payload was varied over a wide range in order to evaluate the ability of the MAC Framework to handle different LLC packet sizes as well as the rate at which frames of differing sizes are processed between the packet sources and packet sinks. The LLC packet size affects the size of most data structures (buffers, MPDUs, etc.), which in turn affect the frame size.

Figure 16 shows the results from these tests. Unfortunately each test requires the MAC framework to be recompiled and downloaded to the EVM, meaning that this testing was time consuming and could not be performed for all possible frame sizes. The points tested are shown in the figure, joined by lines of approximation. It is important to note that the performance of the MAC framework between test points was not verified.

Figure 16 shows two separate sets of data. During testing, a significant difference in processing speed for frames with odd and even numbers of 32-bit elements was

observed. Results for odd and even sized frames were separated and graphed separately to clearly show this trend. Frames containing an odd number of elements performed significantly and consistently better than frames containing an even number of elements. The only possible explanation for this is that the arrangement of data structures (buffers, MPDUs etc.) in memory is more efficient when the LLC payload size chosen results in a frame containing an odd number of 32 bit elements.

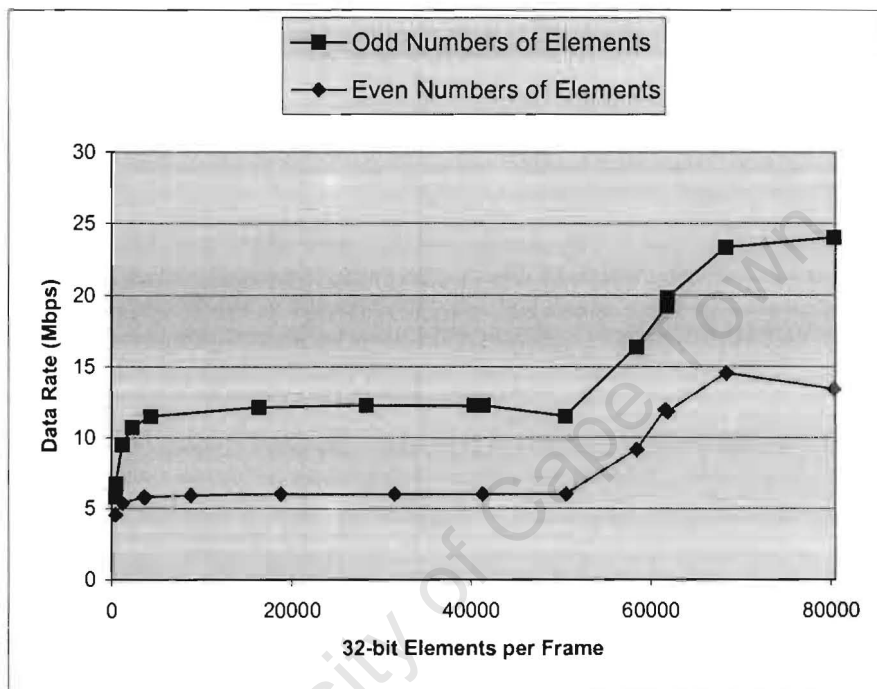


Figure 16: Data Rates for Different Frame Sizes

An LLC payload of 1538 bytes, the maximum size for an Ethernet frame, was also successfully tested, yielding a frame size of 61891 and a corresponding data rate of 19.8 Mbps.

Frame sizes higher than 85000 32-bit elements caused unpredictable results, sometimes causing CCS to encounter problems with reading kernel data from the DSP. CCS also crashed at times when testing with frame sizes greater than 85000 elements.

Performance of the MAC framework has therefore been verified for a large range of frame sizes. It has been shown that the MAC framework can be easily adapted to

carry LLC payloads from 1 byte up to 2000 bytes (resulting in frame sizes of 410 up to 80371 32-bit elements.)

On average, the data rate increases with frame size, up to an approximate maximum of 24 Mbps for a frame containing 80371 elements. This was expected, as an increase in frame size results in a reduced processing overhead per element transmitted.

It is important to note that, while these tests were performed by varying the size of the LLC packets, similar results would be expected if the size of the LLC packet remained the same but the number of packets in a frame was varied. The number of LLC packets per frame section is as easy to modify as the size of an LLC packet.

While it has been shown that the MAC framework is capable of supporting a wide range of frame sizes, it is also necessary to take the nature of the wireless medium into account when choosing a frame size. The larger the frame size used over the wireless medium the greater the number of errors per frame will be. This can severely affect throughput and must therefore be taken into account when a wireless PHY is included in the WATM test-bed.

5.6 Validation of Buffer Efficiency

A further test was performed to evaluate the effect of the zero-copying approach to buffering data that was implemented in the MAC framework. Buffers are used extensively in the MAC framework, for buffering ingress and egress packets, as well as for buffering MPDUs after packing.

The zero-copying method of buffering used in the MAC framework was temporarily replaced with a less complex buffering system that simply copied elements in the enqueue and dequeue procedures. The effect on the data rate from the packet sources to the packet sinks of the MAC framework for a default frame is shown in Figure 17.

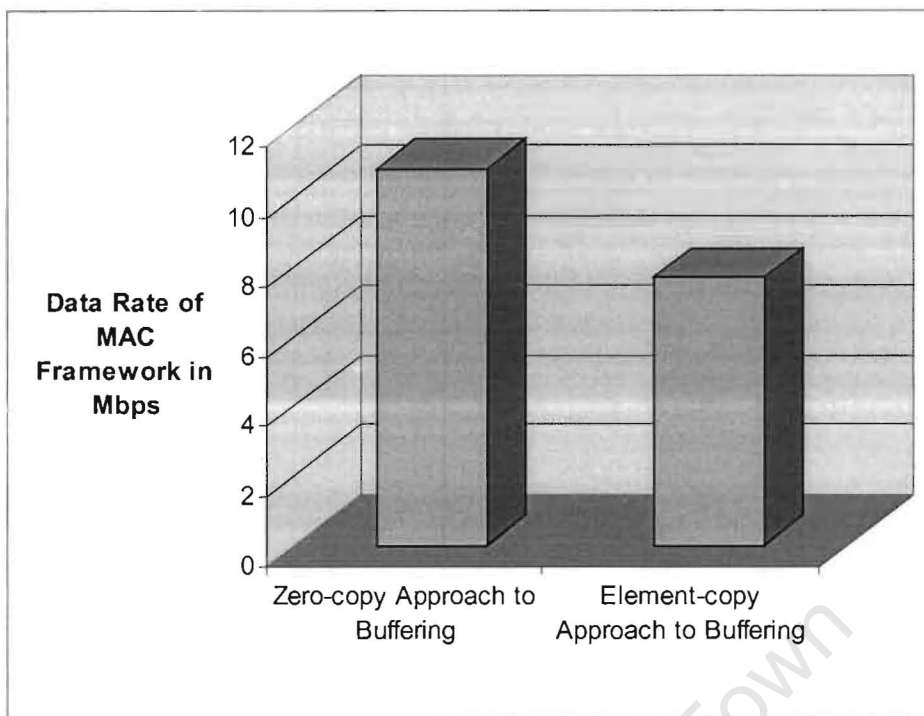


Figure 17: The benefit of zero-copying in buffers

The zero-copying approach is shown to have a significant advantage (about 41%) in terms of the increase in the data rate at which the MAC framework can process data. The implementation of the more complex, zero-copying approach to buffering in the MAC framework can therefore be justified.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

Previous chapters have presented a study into the need for WATM and the issues surrounding WATM, as well as the design, implementation and testing of a fast, flexible framework for MAC development. Conclusions made during this research are presented in this section.

There are many incompatibilities¹ between ATM and the wireless environment. The correct choice of MAC layer is therefore vital if any WATM system is to overcome these incompatibilities and succeed. This is especially true if the WATM system must support multimedia traffic. Demand assignment MAC protocols are particularly well suited to carrying multimedia traffic. This is largely due to their flexibility², support for different traffic types and because they maintain the service required by high priority traffic under heavy load. However, there are many different demand assignment MAC protocols, most

¹ e.g. High BER of the wireless medium, limited bandwidth availability and the broadcast nature of the wireless medium.

² i.e. Their ability to adjust the distribution of bandwidth amongst different connections.

of which have not been implemented and tested in a working WATM environment. A test-bed that could be used to investigate the effectiveness of various approaches mentioned in literature was therefore proposed in this thesis. The Technical University of Berlin (TU) has developed a wireless test-bed in which the MAC layer can be modified. Researchers at the TU have used this test-bed to implement a MAC layer of their choice. It was thought that this test-bed could be used to further experiment with MAC protocols. However, the study conducted on the TU test-bed showed that it does not have the correct infrastructure to support the requirements of this project. More specifically, the wireless test-bed at the TU was not designed to cater for any form of multimedia. The TU system also only supports a uni-directional, FIFO service. The processor used in the TU system is also not capable of handling high bit rates or complex MAC operations.

A new system, referred to as the MAC framework in this project, was therefore designed and implemented. The architecture of the MAC framework supports the development and testing of different traffic schedulers as well as straightforward interfaces to future LLC and PHY layers. The effectiveness of a particular traffic scheduler at scheduling multimedia traffic can therefore be investigated through the use of the MAC framework. The MAC framework was implemented on the Texas Instruments' (TI) TMS320C6201 Evaluation Module (EVM) using TI's Code Composer Studio 2 (CCS) IDE. TI's real-time operating system, DSP/BIOS, was used in the MAC framework to allow the framework to react in real-time to events. This allows real-time exchange of data between processes running on the EVM and between the MAC framework and the emulated PHY and LLC layers. DSP/BIOS also allowed the prioritisation of processes according to their importance to the MAC framework. The combination of CCS2 and DSP/BIOS proved to be invaluable tools in the development of the MAC framework.

The MAC frame structure used in the MAC framework was based on research performed by [Sung01], and provides full support for all of ATM's classes of service. Testing verified that all the classes of service are supported and that traffic flows do not interfere with other traffic flows with different classes of service. Black and white box testing verified the integrity of the MAC framework. The speed at which the MAC framework

could process data was measured by using a timer system with an approximate granularity of 6.25ns.

The flexibility of the MAC framework was then tested by varying the LLC payload size between 1 and 2000 bytes. This range includes the maximum size of an Ethernet frame. The MAC framework was able to correctly process these LLC packets of varying size at rates of between 5 and 24 Mbps. The MAC framework can also easily handle LLC packets containing an ATM cell as their payload. The speed at which this can be done depends on the frame size used. It must be noted that the choice of frame size is not only affected by the speed at which the MAC framework can process that frame size. The choice of frame size is also affected by the nature of the wireless medium. If a large frame size is used the number of errors per frame will become unacceptable. This issue is not investigated in this research but is for future work.

Whether the number of elements in a frame is odd or even also has an effect on the speed at which the MAC framework can process data. The only possible explanation for this is that the arrangement of data structures (buffers, MPDUs etc.) in memory is more efficient when the LLC payload size chosen results in a frame containing an odd number of 32 bit elements.

The following conclusions can also be made regarding the MAC framework:

- Using the DMA controller to service the McBSP does not significantly affect the data transfer rate of the MAC framework. This could either be because the time taken to service the McBSP is insignificant when compared to the total time taken to process an entire frame, or because the time taken for the DMA to service the McBSP is almost identical to the time taken for the CPU to perform the same task.
- The rate at which the MAC framework can process data will be favourably affected when the base-station MAC layer is run on a separate EVM to the mobile

terminal MAC layer. The migration of the MAC layers to separate EVMs will result in a reduced processing load per DSP per frame, as well as a reduced amount of data for each McBSP to handle per frame. This will improve the speed at which the respective MAC layers can process data.

- Using the McBSP in DLB mode and hence having a system with the Mobile MAC and the base-station MAC on the same EVM had significant advantages in the development of the MAC framework. Only one EVM was needed per researcher working on the project, allowing separate issues to be investigated simultaneously. The need for a temporary, wired PHY between two EVMs was also negated.
- Migration to a system with a wireless PHY will require few modifications to the MAC framework. The McBSP would no longer operate in DLB mode, but would be connected directly to the wireless PHY layer. The mailboxes controlling the MAC layer processes would be posted to from hardware Interrupt Service Routines (ISR), called when a block of data arrives from the wireless PHY via the McBSP, instead of being posted to from another MAC layer process. The TMS320C6201 EVM provides sufficient bandwidth (up to 100Mbps, [TI455a]) over the McBSP interfaces to the future PHY layer.
- Migration to a system with a LLC layer will also require few modifications to the MAC framework. This is possible through the abstraction provided by the LLC abstraction layer. This layer could be easily modified to use the DMA controller to transmit packets from the egress buffers, via the PCI bus, to a LLC layer on the host. This layer could also respond to interrupts from the DMA controller, signalling the arrival of an LLC packet from the host, and place the packet in the ingress buffers of the MAC layer. The TMS320C6201 EVM provides sufficient bandwidth over the PCI interface to the future LLC layer.

An investigation into the accuracy of the DSP simulator provided with CCS was also conducted. The same code was profiled¹ on the simulator and on the DSP itself. The number of instructions taken to perform the same tasks was significantly larger on the DSP. The simulator is therefore not seen as a reliable tool when performing speed and efficiency tests.

University of Cape Town

¹ The number of assembly instructions needed to execute the block of code is measured.

6.2 Recommendations and Future Work

Based on the research performed during this thesis certain recommendations for further development of the MAC framework and other layers of the WATM test-bed can be made.

The WDWEDF scheduler should be implemented on the MAC framework as a replacement for the weighted round robin scheduler currently being used. The WDWEDF scheduler should closely monitor the individual requirements of each connection and the QoS that each connection is being offered. The grouping of connections and synchronisation between connections in the same group should also be considered. This would enable connections of different types to be grouped into a single multimedia group, enabling the synchronisation of voice, video and data travelling to the same destination. Other schedulers could also be implemented and the effectiveness of each scheduler at scheduling multimedia traffic can be evaluated and compared.

The efficiency of the MAC framework in terms of the number of reserved MPDU slots wasted as well as the number of packet slots within MPDUs wasted is currently recorded. Schedulers implemented on the MAC framework should use the statistics about wasted MPDUs and wasted packet slots to adjust bandwidth allocated to connections and classes of connections.

The high-resolution timing functionality in the MAC framework should be used to implement a method of time stamping each packet as it travels through the MAC framework. This will aid in monitoring and maintaining the required QoS to each connection. Time stamping each packet will also allow the scheduler to decide which packets are not useful to the end application anymore, and should therefore be dropped.

The following recommendations are made regarding the development of the LLC and PHY layers:

- OFDM should be considered as an option when implementing the wireless PHY layer. OFDM is very resilient against interference and multi-path distortion, [WAVE01] making it a promising prospect for high bandwidth, high frequency based systems. It has been used in 802.11a networks and HIPERLAN 2. OFDM should be used in the mode where a continuous stream of data is split into multiple channels, each of which is then modulated by a separate carrier. This will mean that all MAC functionality will be performed in the MAC layer i.e. the PHY layer will not perform any MAC functions.
- The LLC layer should be implemented, either on the EVM host machine or on the EVM itself. The responsibility of the LLC layer should be to detect and, if possible, correct or request retransmission of errored data. The LLC layer should implement a combination of FEC and ARQ in order to perform this task. Implementing the LLC layer on the EVM will adversely affect the speed at which the MAC framework can process data. If the intention is to implement the LLC layer on the EVM then an effort must be made to limit the impact this has on the MAC framework.

The frame size for the WATM test-bed must be carefully chosen as it affects the speed of data flow through the MAC framework. The choice of a frame size should take into account the speed at which the MAC framework can process that particular frame size, as well as other factors such as how efficiently that frame size could be transmitted over the wireless medium.

Any future work should not rely on the DSP simulator for speed or efficiency tests. The CCS profiler reported two very different instruction counts for the same code running on the simulator and the DSP. The simulator also does not simulate many of the peripheral devices e.g. the McBSP. The additional instructions required on the DSP may be needed

to retrieve objects from the various types of memory in which they reside or to service peripherals.

University of Cape Town

Appendices

Appendix A: Additional ATM Information

In order to justify the need for Wireless ATM (WATM) it is appropriate to explain some of the basic concepts on which ATM is based. Only an introduction to some ATM concepts will be given here, simply to put the concept of WATM in context.

A.1 Connection-Less and Connection-Oriented Protocols

In the conventional telephone system a user has to establish a connection to a remote phone before any communication can take place. Once this connection is established it remains in place for the duration of the call. Once the call is terminated the connection between the two end stations is torn down. This method of operation is known as a connection-oriented approach to communication, with each connection being allocated a certain amount of bandwidth. The IP layer of the Internet (layer 3), on the other hand, utilises what is known as a connection-less approach to data transmission. Each connection transmits packets at will, without an established connection to the destination in place. Each packet has an address field that is used to route it through the network to

the final address. This approach makes no reservation of bandwidth for any individual connection, which means that the available bandwidth is shared amongst all connections.

When one wishes to guarantee a certain bandwidth or service to an end-user it is more appropriate to use a connection-oriented protocol over a connection-less protocol. This is because the user's requirements can be taken into account at call set-up and the necessary reservations of bandwidth can be made before any information transfer takes place. Bandwidth that is reserved will then be exclusively used by a particular connection, meaning that fluctuations in the demands of other users on the same network should not affect the service provided to that connection. This is highly desirable in situations where the users rely on a certain minimum amount of bandwidth being made available to them.

A.2 ATM's Connection-Specific Parameters

ATM is a connection-oriented protocol. It was developed out of the need for a technology that could support the high-speed transmission of voice, video and data using one infrastructure. It achieves this by defining different classes of service (with their corresponding traffic parameters) and various QoS parameters. The different classes of service include:

- ❑ Constant Bit Rate (CBR) is defined for loss and delay sensitive traffic that has a known, constant rate of arrival e.g. voice traffic.
- ❑ Real Time Variable Bit Rate (rt-VBR) is defined for traffic that has a variable bit rate but requires the delay as well as the jitter of a connection to be strictly controlled e.g. video conferencing.
- ❑ Non Real Time Variable Bit Rate (nrt-VBR) is defined for traffic with a variable bit rate but which can allow some delay variation or delay for buffering e.g. viewing a predefined video stream.
- ❑ Available Bit Rate (ABR) is intended for traffic that is sensitive to loss but less sensitive to delay. The rate at which the source transmits is controlled by the network to ensure that the network can support the transmission rate of the source without losing data.

- Unspecified Bit Rate (UBR) is a traffic class that has no loss or delay guarantees. This class of service is equivalent to the best effort service offered to TCP/IP users.

There are three traffic parameters that can be used when setting up a connection (depending on the class of service of the connection) and they are:

- Peak Cell Rate (PCR)
- Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS)
- Minimum Cell Rate (MCR)

The QoS parameters that are monitored for each connection are as follows:

- Cell Error Rate (CER): Percentage of cells that have errors
 - Cell Loss Ratio (CLR): Percentage of lost cells
 - Cell Transfer Delay (CTD): Maximum end to end latency of the cells
 - Cell Delay Variation (CDV): The peak to peak variation of the CTD
- [Scha01], [CRG-01]

The user specifies what class of service they require (along with the traffic and QoS parameters that apply to that class of service) for each connection when the connection is set up. The network (if it can support these requirements) then agrees to provide the connection with the negotiated resources. In this way the user is guaranteed a certain service for each connection. This is in stark contrast to the best effort service that is usually offered to all connections going over other networking protocols such as TCP/IP.

ATM is therefore better suited to transporting multimedia traffic, and it is for this reason that this project wishes to extend the benefits offered by ATM over the wireless medium.

Appendix B: Additional Information on EVM Peripherals

B.1 Overview of the Direct Memory Access (DMA) Controller

The DMA controller can be used to transfer data from any memory-mapped location (from the full 32-bit address range) to any other memory-mapped location [TI529a]. These locations include internal and external memory, peripherals, and any devices connected to a memory mapped extension port. These data transfers take place without any intervention from the CPU. The CPU is only needed to configure the DMA controller for the type of transfer that will take place. The DMA controller can then transfer data in a once-off or continuous manner.

There are four general DMA channels onboard the TMC320C6000 series of DSPs, with a fifth DMA channel allowing the DMA controller to service requests from the Host Port Interface (HPI) [TI190d]. Each of the four general channels can be configured to perform different transfers by initialising the set of control registers associated with that DMA channel. Different DMA channels can have different priorities, allowing more important data transfers to interrupt transfers that are less time-critical to the system.

The size of each element transferred can be configured to be 8, 16 or 32 bits. The number of elements per transfer can be changed by changing the number of elements per frame, as well as the number of frames per block of the transfer.

Most importantly, in the context of this project, the DMA can be synchronised to transfer data based on an event external to the DMA controller. More specifically, it is possible to synchronise the DMA transfers on events from the McBSP, allowing DMA channels to be configured to service the McBSP. It is possible to configure a DMA channel to read or write a single element or an entire frame per external event upon which it is synchronised. The graphical configuration tools in CCS2 can be used to perform all configuration of the DMA controller and DMA channels.

B.2 Overview of the Multi-channel Buffered Serial Ports (McBSP)

Two McBSPs exist on the TMS320C6201 Evaluation Module (EVM). Each McBSP provides a fast, full duplex communications port to external devices. It makes use of up to 128 channels and supports 8, 12, 16, 20, 24 and 32 bit element sizes [TI190d]. Furthermore, the McBSP is capable of generating events that can be used to synchronise the McBSP with any DMA channel that is being used to service the McBSP. It also provides two interrupts to the CPU that can be used in the event that the CPU services the McBSP.

The McBSP interfaces to external devices through the external peripheral interface provided on the EVM. The McBSP uses 7 pins to interface to external devices. These pins are:

- ❑ CLKR - Receive clock
- ❑ CLKX - Transmit clock
- ❑ CLKS - External clock
- ❑ DR - Received serial data
- ❑ DX - Transmitted serial data
- ❑ FSR - Receive frame synchronization
- ❑ FSX - Transmit frame synchronization

Two McBSP devices on separate TI 'C6000 series boards can be connected to each other to form a high-speed communications port. One McBSP simply generates the clock and frame pulses, and is therefore referred to as the clock and frame master. The other McBSP is the clock and frame slave, and does not generate any clock or frame synchronisation pulses [TI455a].

Multiple TI 'C6000 boards can all be connected via a TDM data bus in a similar fashion [TI491a].

University of Cape Town

Appendix C: Data Structures used by the MAC Framework

A number of data structures have been defined for use in the MAC framework. These structures are initialised or modified by the MAC framework code. The declarations of these structures can be found in “global.h”

A brief summary of each data structure is presented below. The buffers used in the MAC framework will also be elaborated upon.

C.1 Data Structures

- ❑ **struct type_timer**

This structure is used to store values used to calculate the time elapsed for any section of MAC framework processing

- ❑ **struct type_stats**

This structure is used to hold the globally accessible statistics information about the number of packets created, the number of packet slots wasted etc.

- ❑ **struct type_llcell**

Every LLC packet uses this structure. It contains fields for all sections of an LLC packet containing an ATM cell.

- ❑ **struct type_elementholder**

Element holders are necessary to allow for the very efficient enqueue and dequeue operations implemented in the buffers of the MAC framework. Essentially, an element holder points to an element sized block of memory that is globally accessible and is not freed while the MAC framework is running.

- ❑ **struct type_cellbuffer**

This structure contains all of the control variables for each buffer, as well as an uninitialised pointer to an array of elements. The use of this structure will be elaborated on later in this appendix.

❑ **struct type_PiggyBack**

The piggyback structure contains fields for the information to be attached to each MPDU. This information will be used by the scheduler to allocate bandwidth and/or to acknowledge receipt of a packet.

❑ **struct type_MPDUa**

This structure contains an array of LLC packets as well as a piggyback structure.

❑ **struct type_connection**

Each connection uses this structure to store connection dependant variables such as the PCR, MCR, CDV, connection ID and type of the connection. This structure also contains ingress, egress and MPDU buffers, as well as an element holder structure for enqueueing and dequeueing MPDUs.

❑ **struct type_frameheader**

This structure contains all fields needed by the frame header. This includes variables specifying the different sizes of each frame section and the transmit permissions for the CBR, and VBR uplink sections of the frame.

❑ **struct type_supheader**

This structure is similar to the frame header, except it contains information about the supplemental uplink period.

❑ **struct type_Frame**

The frame structure can be used to hold an entire MAC frame. The size of each data section of the frame is allocated at run time, and is dependant on the LLC packet size and the number of MPDU slots in that section.

❑ **struct type_MSG**

This structure is used to hold the messages that are posted to the mobile and base-station controller's mailboxes.

C.2 Buffers used by the MAC framework

The buffers used in the MAC framework have been briefly discussed in chapter 4. This section will elaborate on the earlier discussion.

The MAC framework buffers are implemented as fixed sized (fixed at initialisation time) arrays with control variables allowing them to be treated as circular buffers. As mentioned in chapter 3, this type of buffer requires very little dynamic memory allocation and traversing, when compared to linked lists, and very little re-organisation, compared to a normal array where data must be re-shuffled frequently.

Figure 18 shows the state of an initialised buffer before any elements (cells in this case) have been enqueued. The buffer array is actually an array of pointers to elements.

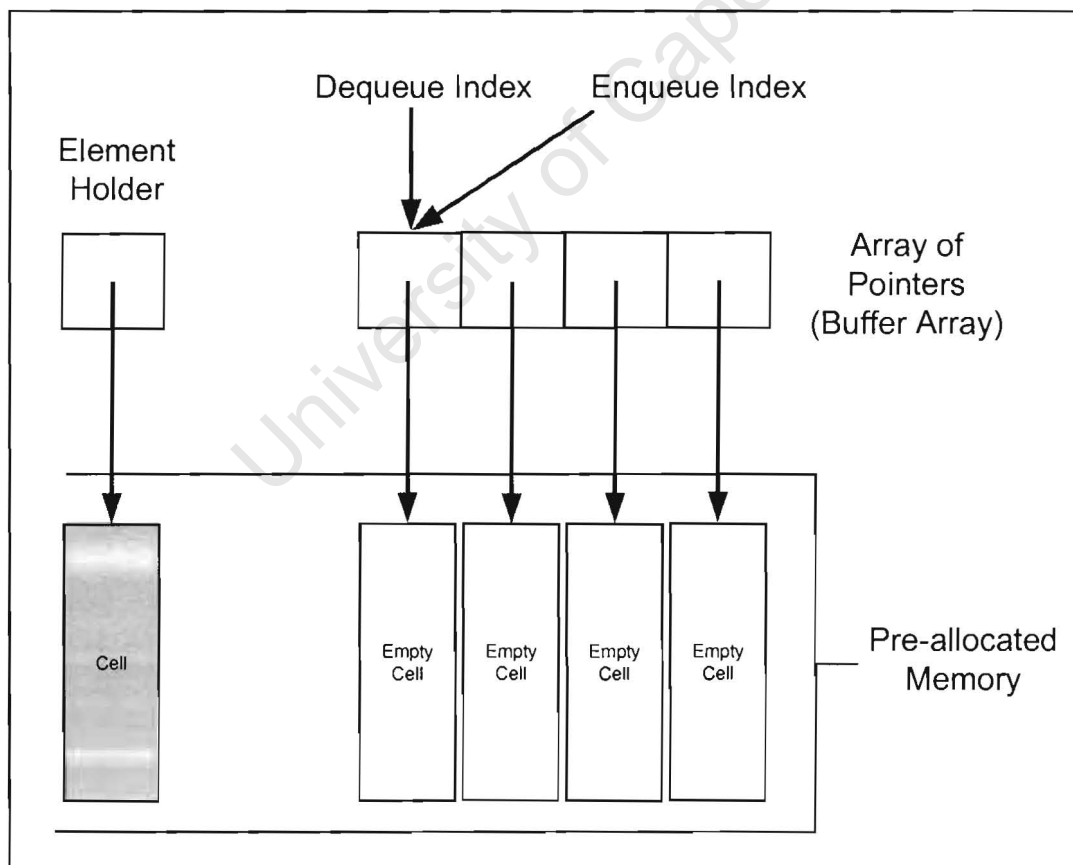


Figure 18: Buffer State before Enqueue

When an element is enqueued the buffer state changes to that shown in Figure 19. The enqueue index moves to the next position in the array of pointers. The pointer in the original enqueue position has been modified to point to the element being enqueued. The element holder pointer is modified to point to the empty element that was in the buffer. This ensures that the element holder pointer still points to valid memory space.

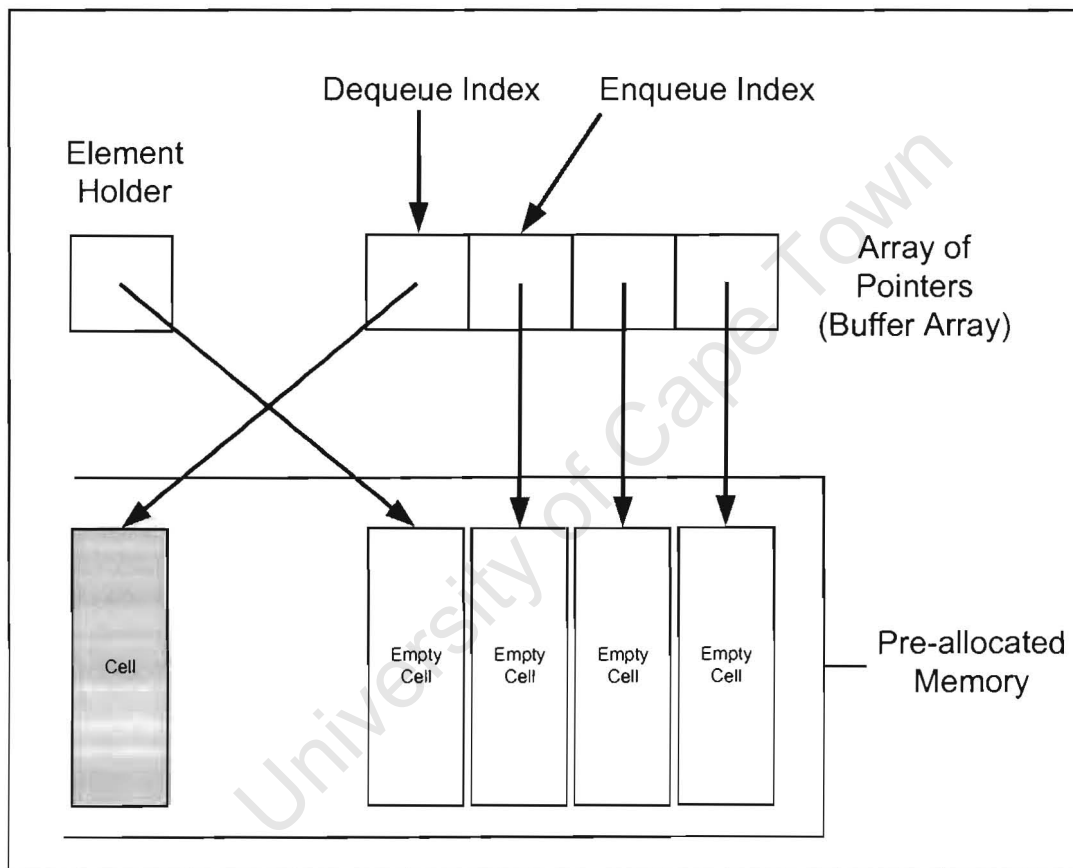


Figure 19: Buffer State After a Single Enqueue

When the enqueue index reaches the end of the array it will wrap around to the starting position again. A wraparound state variable keeps track of whether the buffer is wrapped or not, as different enqueue and dequeue algorithms are needed if the buffer is wrapped or not.

To dequeue the element just enqueued in the above figures one would simply reverse the enqueue procedure and increment the dequeue index. The buffer would then be empty again.

Note that the element is never moved from one memory location to another during the enqueue or dequeue procedure. This factor improves the overall speed at which the MAC framework is capable of processing data at by approximately 41%.

University of Cape Town

Appendix D: Overview of the MAC Framework Code

This appendix mentions all files containing code relevant to the MAC framework. The relevance of each file and each function in the files is discussed to aid understanding of the code. This will also aid those wishing to continue development of the WATM test-bed.

C.1 Buffers.c

The buffers.c file contains all code needed to initialise and use the buffers defined in the MAC framework. The enqueue and dequeue procedures only copy pointers instead of copying an entire element. This results in an increase of about 41% in the speed at which the MAC framework can process data.

This file contains the following functions:

- ❑ **void initbuffer(struct type_cellbuffer *buf, int size, int length)**
This function allocates all the memory that a specified buffer will need. The size of the elements to be buffered is specified as well as the number of elements to be contained by the buffer.
- ❑ **int enqueue(struct type_cellbuffer* buf, void** element)**
This function enqueues an element into the buffer by swapping the addresses pointed to by the element pointer and the pointer in the buffer that points to an unused element in the buffer.
- ❑ **int dequeue(struct type_cellbuffer* buf, void** element)**
This function dequeues an element from the buffer using a similar procedure to the enqueue function.
- ❑ **int flushbuffer(struct type_cellbuffer* buf)**

This function cleans out a buffer by resetting its enqueue, dequeue, numelements and wraparound variables.

C.2 Init.c

This file contains all of the initialisation routines for the MAC framework. This includes routines to set up individual connections and to allocate all memory needed by the MAC framework.

The following functions are in this file:

- **void initvar()**

This function initialises all general variables used in the MAC framework and allocates memory for MAC frames, enqueue and dequeue elements etc.

- **void initconnection(struct type_connection *connection, int type, int id, int MACid, int pcr)**

This function initialises all variables pertaining to a specific connection and calls the initbuffer function for the ingress, egress and MPDU buffers for that connection.

C.3 Mcbsp.c

This file contains functions to initialise the McBSP and to perform transfers of data using the McBSP. It contains the following functions:

- **void MCBSPinit()**

This function simply sends a series of commands to the McBSPs to start them. This function does not need to configure the McBSPs because this configuration has already been performed using the graphical configuration tool in CCS2.

- **void DMAGo(Uint32 *dest, Uint32 *source, int numelements)**

This function originally used the DMA controller to transfer a block of elements to the McBSP for transmission. Later, it was modified to use the CPU to directly service the McBSP. This change did not result in a change in performance. Two McBSPs are used in tandem, with a different McBSP transmitting sequential elements. This approach improves the speed at which data can be handled.

C.4 All.c

This file contains all of the MAC framework algorithms, including the base-station and mobile controller functions, the LLC packet generation code, the code for the high resolution timing functionality and the statistics generation code. It contains the following functions:

- ❑ **double returtime()**
This function returns a microseconds time that is used to evaluate the speed performance of the MAC framework.
- ❑ **void tx_cellcreate(int tablenum)**
This function creates packets for a specific connection. It then implements the ingress portion of the LLC abstraction layer by placing the created packet in the enqueue element holder and enqueueing the element in the correct connection's ingress buffer.
- ❑ **void tx_trafficgen()**
This function calls tx_cellcreate for each connection.
- ❑ **void tx_packer()**
This function dequeues packets from the ingress buffer of each connection and packs the packets into MPDUs. It then enqueues complete MPDUs into the MPDU buffer for that particular connection.
- ❑ **void tx_bsDownlinkWriter()**
This function cycles through the connections on the base-station in a weighted round robin fashion, dequeuing MPDUs from the connection's MPDU buffer and placing them in the relevant blocks of the downlink section of the frame. The function then calls the PHY layer abstraction code to transmit the different downlink frame blocks.
- ❑ **void rx_mbDownlinkReader()**
This function (run by the mobile controller) receives the data transmitted by tx_bsDownlinkWriter from the PHY abstraction layer. It then unpacks the MPDUs and places the packets into the correct connection's egress packet buffer.
- ❑ **void tx_mbUplinkWriter()**

Similar to `tx_bsDownlinkWriter`, except this function creates and transmits the uplink frame block, and is run by the mobile controller.

❑ **`void rx_bsUplinkReader()`**

Similar to `rx_mbDownlinkReader`, except this function unpacks the uplink MPDUs and places the packets in the relevant connection's egress buffer and is called from the base-station controller.

❑ **`void tx_mbSupWriter()`**

This function is similar to `tx_mbUplinkWriter`, except it creates and transmits the supplemental uplink block of the frame.

❑ **`void rx_bsSupReader()`**

Similar to `rx_bsUplinkReader`, except this function unpacks the supplementary uplink block of the frame and places the packets in the relevant connection's egress buffer.

❑ **`void statsClear()`**

This function resets all statistics gathered by the MAC framework.

❑ **`void statsUpdate()`**

This function performs a number of tasks. Firstly, it performs the egress LLC abstraction layer functionality by dequeuing each packet from the egress packet buffers and placing the packet in a pre-allocated memory location. This function then performs the packet sink functionality by taking each packet from the abstraction layer and analysing its contents for statistics calculation purposes. The function then recalculates all of the statistics gathered by the MAC framework and logs all of the statistics to a log within CCS2.

❑ **`void bscontroller(Arg id_arg)`**

This function is run as the "main" function of the base-station MAC process. It is synchronised with the mobile MAC controller through the use of mailboxes. The relevant functions to process or generate the MAC frame are called at the relevant times.

❑ **`void mbcontroller(Arg id_arg)`**

As above, except this function is run as the "main" function of the mobile terminal MAC process.

□ **void main()**

This function simply calls the initialisation routines for all of the global variables and connections that are used by the MAC framework. After the main function's tasks are complete the mobile and base-station controllers take over the execution time of the DSP and the main is no longer run.

University of Cape Town

Appendix E: The Accompanying CD-ROM

A CD-ROM accompanies this thesis. This CD-ROM contains the following information:

- ❑ This thesis document in “.doc” format can be found in the “Thesis Document” directory.
- ❑ Many of the documents referenced in this thesis can be found in the “Referenced papers and other useful documents” directory.
- ❑ Documentation on TU’s research, and more particularly on the “RaDevil” wireless research, can be found in the “Technical University of Berlin” directory.
- ❑ Documentation on various embedded systems, including all of the Texas Instruments’ documentation can be found in the “Referenced papers and other useful documents\Microprocessor Information” folder.
- ❑ Source code for the MAC framework can be found in the “MAC framework source” directory.
- ❑ The images used in this thesis have been included in various formats in the “Thesis Document\Images” directory.

References

- [Abra01] N. Abramson, "The ALOHA System: Another Alternative for Computer Communications", AFIPS Conference Proceedings, 1970.
- [Akyi01] Ian F. Akyildiz, Janise McNair, Loren Carrasco, Ramon Puigjaner, Yelena Yesha, "Medium Access Control Protocols for Multimedia Traffic in Wireless Networks", IEEE Network Magazine, Vol. 13, No. 4, pp. 39-47, July/August 1999.
- [Akyo01] Bora Aydin Akyol, Doctorate Dissertation, Chapter 5, "Medium Access in the Wireless ATM Network", Stanford University, June 1997.
- [ATMF01] The ATM Forum Technical Committee, "Wireless ATM Capability Set 1 Specification – Draft (BTD-WATM-01.13)", January 13, 2000.
- [ATMF02] The ATM Forum Technical Committee, "Wireless ATM Radio Access Layer Requirements", August 9-23 1996.
- [Ayan01] Ender Ayanoglu, Kai Y. Eng, and Mark J. Karol, "Wireless ATM: Limits, Challenges, and Proposals", IEEE Personal Communications August 1996. pp 18-34.
- [Bett01] Christian Bettstetter, Hans-Jorg Bogel, Jorg Ebershacher, "GSM phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface", IEEE Communications Surveys, Third Quarter 1999, vol.2 no. 3.
- [Bing01] Benny Bing, "High-Speed Wireless ATM and LANs", Artech House inc.
- [BRAN01] Broadband Radio Access Networks (BRAN), "HIPERLAN Type 2 - Data Link Control (DLC) Layer - Part 1: Basic Data Transport Functions", available: <http://www.etsi.org>
- [Bree01] Breeze Wireless Communications Ltd, "Radio Signal Propagation", sigprop.pdf, available: <http://www.mibo.si/pdf/sigprop.pdf>

- [Chan01] Ajay Chandra, V. Gummalla, John O. Limb, "Wireless Medium Access Control Protocols", IEEE Communications Surveys, Second Quarter 2000.
- [CRG01] The Communications Research Group, UCT, "Broadband Networks and ATM", 2002.
- [Davo01] F. Davoli, P. Maryni, C. Nobile, "ISA2: A 'Two-Phase' MAC Protocol for Multimedia Wireless Networks", Globecomm '98.
- [Dobb01] Oliver Cole, "White-Box Testing", Dr. Dobbs' Journal, March 2000, available: <http://www.ddj.com/documents/s=887/ddj0003a/0003a.htm>
- [Elber01] Jamal Elbergali, Neco Ventura, "A New Inter-Cluster Handoff Technique to Support Real Time Applications in Wireless ATM Networks", Univ. of Cape Town, South Africa, ICWN 2002, p 373-378, June 24-27, 2002, Las Vegas, Nevada, USA.
- [Frat01] Luigi Fratta, Paolo Giacomazzi, Luigi Mesumeci, "PRAS: A MAC Protocol for Wireless ATM Networks", Globecom '99.
- [Geie01] Jim Greier, "The BIG Question: 802.11a or 802.11b", available: http://www.80211-planet.com/columns/article/0,4000,1781_961181,00.html
- [Geor01] Nikolaos Georganopoulos, A. H. Aaghvami, "MAC Scheduler using Polling with Priorities for Future Wireless Multimedia Networks", Globecom'99.
- [Hang01] Chi Hang Tse, Brahim Bensaou, Kee Chaing Chua, "Efficient Distributed Scheduling Architecture for Wireless ATM networks", IEICE TRANS. COMMUN. Vol.E83-B, No.2, February 2000.
- [Hac001] Anna Hac, "Multimedia Application Support for Wireless ATM", Prentice Hall, 2000.
- [Hoen01] Christian Hoene, "Prototyping Environment for Link/MAC Layer Protocols", available: <http://www-tnk.ee.tu-berlin.de/~hoene/macboard/sld001.htm>
- [Holle01] Kevin Holley, Tim Costello, "The Evolution of GSM Data Towards UMTS", 1998 GSM World Congress.

- [Hoss01] AZM Ekram Hossain, V. K. Bhargava, "Scheduling Multiservice Traffic for Wireless ATM Transmission over TDMA/TDD Channels", Globecomm '99.
- [How01] Dale How, Patrick Atkins, Neco Ventura, "A MAC Framework for Multimedia Application Support in WATM", SATNAC 2002.
- [Iera01] Antonio Iera, Antonella Mollinaro, Salvatore Marano, "Multimedia QoS Provisioning in Wireless-ATM Networks", Globecomm '99.
- [Inte01] Intersil, PRISM wireless LAN, available:
<http://www.intersil.com/design/prism/index.asp>
- [IEEE01] IEEE working group for wireless LAN's, 802.11a and 802.11b specifications, available: <http://standards.ieee.org/catalog/olis/lanman.html>
- [Khan01] Jamil Y. Khan, "an Adaptive Reservation Packet Switched Multiple Access Protocol for a Wireless Network", ICPWC 2000.
- [Kim001] Won-Tae Kom, Yong-Jin Park and Jae-II Jung, "MMAC: An Efficient WMATM MAC for Multicast Communications", Globecomm '99.
- [Kubb01] Osama Kubbar and Hussein T. Mouftah, "Multiple Access Control Protocols for Wireless ATM: Problems Definition and Design Objectives", IEEE Communications Magazine. November 1997, pg. 93-99.
- [Mikk01] Jouni Mikkonen, Nokia Mobile Phones, "Wireless ATM Overview".
- [Ohta01] Atsuchi Ohea, Masafumi Yoshioka, Masahiro Umehira, "PRIME ARQ: A Novel ARQ Scheme for High-Speed Wireless ATM", IEICE Trans. Commun. Vol.E84-B, No. 3 March 2001.
- [Petr01] Dietmar Petras, "Medium Access Control Protocol for wireless, transparent ATM access", Aachen University of Technology, Chair of Communication Networks.
- [Rama01] Vijaya Chandran Ramasami, "Orthogonal Frequency Division Multiplexing", available: www.ittc.ukans.edu/~rvc/acads/ofdmreport.pdf
- [Sanc01] Jaime Sanchez, Ralph Martinez, Michael W. Marcellin, "A Survey of MAC Protocols Proposed for Wireless ATM", IEEE Network, November/December 1997, pg 52.

- [Sanc02] Ricardo J. Sanchez, Fadi F. Wahhab, Joseph B. Evans Victor S. Frost, Gary J. Minden, "Design and Evaluation of an Adaptive Data Link Control Protocol for Wireless ATM Networks ", Globecom '98.
- [Scha01] Alan Y. Schaevitz, Introduction to ATM, ATM Tutorial, ATM Conference, 1997.
- [Scou01] John Scourias, "Overview of the Global System for Mobile Communications", University of Waterloo, Available:
<http://ccnga.uwaterloo.ca/~jscouria/GSM/gsmreport.html>
- [Sheu01] Shiann-Tsou Sheu, Shih-Chian Wu, "An Adaptive Cell Checking Controller for Wireless ATM Networks", IEICE/IEEE Joint Special Issue on Recent Progress in ATM Technologies. IEICE Trans. Commun. Vol. E83-B, No.2, February 2000.
- [Siem01] Siemens white paper, "The New Path from GSM to UMTS 'Soft Evolution' of Mobile Radio Networks", 2001.
- [Sing01] Jatinder Pal Singh, Sachin Adlakha, Subrat Kar, Surendra Prasad, "Proposal, Simulation and Performance Appraisal of an Optimal Medium Access Protocol for Wireless ATM Networks", ICPWC 2000.
- [Sobi01] Daniel Søbirk, Johan M Karlsson, Department of Communication Systems, Lund University, Sweden, Lars Falk, Christer Lind, Telia Research AB, Sweden, "An Overview of Proposed MAC Algorithms for Wireless ATM".
- [Speth01] Michael Speth, "OFDM Receivers for Broadband-Transmission", available:
"http://www.ert.rwth-aachen.de/Projekte/Theo/OFDM/www_ofdm.html
- [Stubb01] Adam Stubblefield, John Ioannidis, Aviel D. Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP", AT&T Labs Technical Report TD-4ZCPZZ, August 6, 2001.
- [Suh001] Young-Joo Suh, "Efficient Wireless LAN MAC Protocols for Ad-Hoc Networks", IEICE Trans. Commun. Vol. E84-B, No.3, March 2001.
- [Sung01] Sungwon Lee, Young-Jae Song, Dong Ho Cho, Yong-Bae Dhong, Jung-Won Yang, "Wireless ATM MAC Layer Protocol using WDWEDF and

- Two-phase Scheduling Algorithm”, IEICE Transactions on Communications Vol. E81-B, No.12 December 1998 pg 2432 – 2443.
- [TI190d] Texas Instruments, “TMS320C6000 Peripherals Reference Guide”, spru190D.pdf, available: <http://www.ti.com>
- [TI269d] Texas Instruments, “TMS320C6201/6701 Evaluation Module Users Guide”, spru269d.pdf, available: <http://www.ti.com>
- [TI301c] Texas Instruments, “TMS320C6000 Code Composer Studio Tutorial”, spru301c.pdf, available: <http://www.ti.com>
- [TI305] Texas Instruments “TMS320C6201/6701 Evaluation Module Technical Reference”, spru305.pdf, available: <http://www.ti.com>
- [TI423a] Texas Instruments, “TMS320 DSP/BIOS Users Guide”, spru423A.pdf, available: <http://www.ti.com>
- [TI455a] Texas Instruments, “The McBSP as a high speed communications port”, spra455a.pdf, available: <http://www.ti.com>
- [TI491a] Texas Instruments, “TMS320C6000 McBSP as a TDM Highway”, spra491a.pdf, available: <http://www.ti.com>
- [TI529a] Texas Instruments, “TMS320C6000 DMA Example Applications”, spru529a.pdf, available: <http://www.ti.com>
- [UMTS01] UMTS World, “CDMA Overview”, available: <http://www.umtsworld.com/technology/cdmabasics.htm>
- [WAVE01] WAVE Report, OFDM Tutorial, available: <http://www.wave-report.com/tutorials/OFDM.htm>
- [Yoon01] Sunkeun Yoon, Saewoong Bahk, “A MAC protocol based on FEC for real time traffic in wireless ATM networks”, Globecom ’98.
- [Zhou01] Sihui Zhou, John Deane, Alex Lam, David Skellern, Terry Percival, “An Efficient Media Access Control Protocol for Wireless ATM Networks”, Globecom ’98.