DUBLIN CITY UNIVERSITY
SCHOOL OF ELECTRONIC ENGINEERING

Simulation of IEEE 802.11e in GloMoSim

Daragh Kelleher
August 2003

MASTER OF ENGINEERING

IN

ELECTRONIC SYSTEMS

Supervised by Dr. J. Murphy

Acknowledgements

I would like to thank my supervisor Dr. John Murphy and Dr. Sean Murphy for their guidance, enthusiasm and commitment to this project. Thanks are also due to Prof. C. McCorkell for supporting this work to date.

Declaration

I hereby declare that, except where otherwise indicated, this document is entirely my own work and has not been submitted in whole or in part to any other university.

Signed: .... Date: ....

Abstract

The forthcoming IEEE 802.11e supplementary standard will introduce Quality of Service functionality to the IEEE 802.11 family of Wireless Local Area Network standards.

This project endeavours to simulate this Quality of Service functionality under a variety of different traffic conditions. A secondary objective of this project is to gain experience of simulating wireless networks using the GloMoSim simulator.

The method of the solution used is to upgrade the existing IEEE 802.11 libraries contained in GloMoSim to agree fully with the most important parts of the IEEE 802.11e supplementary standard. Specifically, this involves the design of an Enhanced Distributed Co-ordination Function.

The results obtained show that the introduction of a prioritisation mechanism to the classes of traffic offered to the network has the effect of significantly modifying the throughput and voice packet end-to-end delay.

Thus it is possible to conclude that, with IEEE 802.11e, reasonable Quality of Service guarantees can be made to applications that have a time critical requirement.

Table Of Contents

Table of Figures

Chapter 1 - Introduction

Layer two of the 7-layer Open Systems Interconnection (OSI) model for data communication between peer-to-peer entities describes the Data Link Layer. For Local and Metropolitan Area Networks, the IEEE 802.x group of standards further divides Layer 2 into the Logical Link Control (LLC) and Medium Access Control (MAC) sublayers.

The MAC sublayer describes the strategy for obtaining control of, and releasing of, the physical medium. Specifically the IEEE 802.11 standard defines the operations of the MAC sublayer and the Physical Layer (PHY), when the physical medium is wireless.

There are a growing number of amendments to the original IEEE 802.11 standard that address different issues such as Quality of Service (802.11e), security (802.11i), and increasing data rates (802.11g). IEEE 802.11e is designed to work with the following variants of Physical Layer:

IEEE 802.11 – Uses a Direct Sequence Spread Spectrum Modulation (DSSS) scheme[1] at the Physical Layer. This supplementary standard operates in the unlicensed 2.4 GHz frequency band with a data transfer rate of 1-2 Mbps.

Advantages: the most widely available standard.

Disadvantages: lower data rates than 802.11a and 802.11b.

IEEE 802.11a – Uses an Orthogonal Frequency Division Modulation (OFDM) scheme at the Physical Layer. This supplementary standard operates in the unlicensed 5 GHz frequency band and offers a peak performance of 54 Mbps.

Advantages: there is less interference in the 5Ghz band.

Disadvantages: less range than 802.11b.

IEEE 802.11b – Uses a Direct Sequence Spread Spectrum Modulation (DSSS) scheme at the Physical Layer. This supplementary standard operates in the unlicensed 2.4 GHz frequency band but with a data transfer rate of 11 Mbps.

Advantages: the most widely available standard.

Disadvantages: lower data rates than 802.11a, susceptible to interference.

---

[1] This is a similar modulation technique to the one used in the 3GPP (3rd Generation Partnership Project) WCDMA (Wideband Code Division Multiple Access) mobile radio standard.

1.1 The Problem Statement

IEEE 802.11b and IEEE 802.11a are the most recent MAC and Physical Layer specifications from IEEE for wireless local area networks. Services such as File Transfer Protocol, available in pre-802.11e standards, need reliability but not necessarily Quality of Service.

Examples of applications that have tight Quality of Service requirements are Voice, Video and most Continuous Bit Rate (CBR) sources that have a ‘real-time critical’ aspect.

These standards do not provide any Quality of Service (QoS) guarantees to their users, they provide best effort data transfer only. Networks that do not provide QoS mechanisms can suffer from mean delay, delay variation (jitter) and data loss. This is because packets waiting to access the network are not scheduled and so *contention* may occur if two nodes attempt to transmit at the same time. Packets losing contention are buffered until the transmission medium is available. This results in delay variation since the amount of time spent in the buffer depends on the other traffic present.

A network that operates with Quality of Service can guarantee bandwidth to ‘real-time critical’ applications by assigning them a higher priority than applications without this constraint. Different types of traffic classes with different priority levels can be charged at a price appropriate to their Quality of Service agreement.

The most efficient route through the network is also important as Figure 1.1 illustrates.

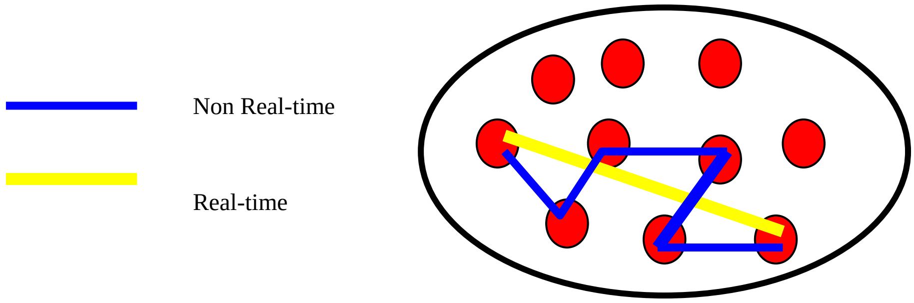

Figure 1.1 Dependency of observable delay on route chosen through the network

Other problems that must be overcome before network Quality of Service guarantees can be made include roaming, interoperability/handover to other systems such as General Packet Radio Service (GPRS) or Universal Mobile Telecommunication System (UMTS), hidden terminals, fluctuating link characteristics and the dynamics of the nodes. This project investigates whether the introduction of the IEEE 802.11e supplementary standard can address *some* of these issues and provide Quality of Service.

1.2 The Document Structure

The layout of the remaining chapters in this document is as follows:

In Chapter 2 a description of the basic components of a Wireless Local Area Network is provided. The legacy IEEE 802.11 standard is introduced including a description of the two main functions, the Point Co-ordination Function and the Distributed Co-ordination Function.

The most important aspects of the Draft IEEE 802.11e supplementary standard are described in Chapter 3. The Enhanced Distributed Co-ordination Function is described in detail whereas the Hybrid Co-ordination Function is briefly described.

In Chapter 4 the Parsec programming language is referred to and is used as an aid to introducing GloMoSim – the discrete event simulator used in the project. The remainder of the chapter is devoted to a description of GloMoSim.

The implementation part of this project is described in Chapter 5. Specifically, this chapter deals with the design issues encountered when incorporating an EDCF function into the simulator.

In Chapter 6 full details of the eight tests performed in the simulation part of the project are provided. Graphs of the results are illustrated and their significance is discussed.

Conclusions on the success, or otherwise, of the project are presented in Chapter 7 along with suggestions for further research that could be initiated in order to improve understanding in this area.

Finally Appendix A contains a list of the software resource requirements in order to run the GloMoSim simulator and Appendix B contains state diagrams of the GloMoSim implementation of the IEEE 802.11 MAC layer code.

1.3 Summary

This introductory chapter briefly describes where the IEEE 802.11 standard fits in the commonly referenced 7-layer OSI stack. The main features of the numerous IEEE 802.11 supplementary standards are introduced. The importance of providing Quality of Service in local area networks supporting real-time critical applications is emphasised. Finally the structure of the remaining parts of this document are described.

Chapter 2 – The Legacy IEEE 802.11 Standard

2.1 Introduction

In section 2.2 of this chapter a general description of the components of a typical Wireless Local Area Network is given. Emphasis is placed on the principal network elements and further detail can be obtained by referring to the standard, [2]. Section 2.3 describes the MAC frame types and their formats.

Finally, in section 2.4, a functional description of the IEEE 802.11 MAC is introduced. It should be noted that a greater emphasis is placed on the DCF function. The PCF function is mentioned only briefly since it is not dealt with in later chapters and is not implemented in the simulator.

2.2 General Description of a Wireless Local Area Network

The Independent Basic Service Set

The formation of the simplest form of WLAN can be explained by the diagram in Figure 2.1[2]. Mobile terminals, called Stations (STA), are free to move within a coverage area, called an Independent Basic Service Set (IBSS), and still remain in communication with one another. The STAs can communicate directly on a peer to peer basis. This type of network where no preplanning is involved is called an ad-hoc network[3].

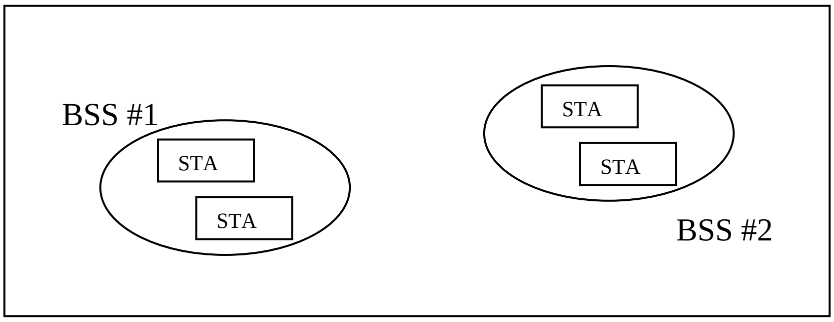

Figure 2.1 Two Independent Basic Service Sets (IBSS)

---

[2] A similar diagram exists in [2], chapter 5.2.

[3] The term ad-hoc network can have different interpretations. The description in Chapter 2.2 is consistent with [1]. Another interpretation is a scenario where a continually changing topology requires the continuous alteration of routes through the network without prior notification.

The Infrastructure Basic Service Set

In order for a station to communicate beyond the IBSS some form of connection must be made with a Distribution System (DS). The STA nominated to do this is called the Access Point (AP). The relationship between an AP and other STA in the IBSS is analogous to a Mobile Base Station in a mobile telephony network – all communication between STA must be via the AP. Thus the AP can be considered as the point of entry/exit to an extended network. The DS may be wired or wireless.

Figure 2.2[4] depicts the connection of an IBSS to a Distribution System to become an *Infrastructure BSS*.

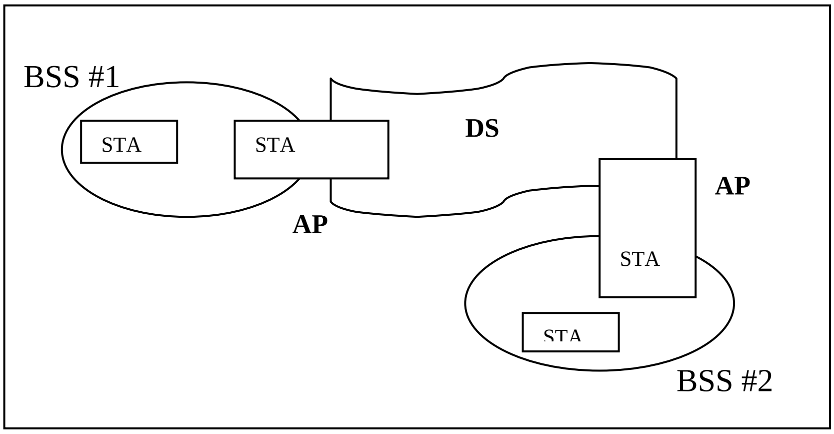

Figure 2.2 Two *Infrastructure* Basic Service Sets

2.3 The IEEE 802.11 MAC Frames and Formats

The frame types in IEEE 802.11 can be broadly placed into Management, Control and Data categories. These frames can be sent alone or in combination, e.g., in certain circumstances it is possible to send a Data frame + CF-Ack together as a valid frame combination. This is known as *piggybacking*. Please consult [Table 1, chapter 7.1.3.1.5, 2] for an exhaustive list of frame types and valid combinations. Figure 2.3 below illustrates the standard frame format of the IEEE 802.11 MAC. The numbers in brackets are the lengths of each field in octets. Not all frame types will include all fields. The Frame Control, Duration, Address1 and FCS are the only mandatory fields for all frame types.

---

[4] A similar diagram exists in [2], chapter 5.2.2.

Frame Control (2)	Duration Id (2)	Address 1 (6)	Address 2 (6)	Address 3 (6)	Sequence Control	Address 3 (6)	Frame Body (0-2312)	FCS (4)

Figure 2.3 Frame format of IEEE 802.11 MAC

The Frame Control field holds the frame type. The principal frame types are:

1. Data Frames,
2. Management frames,
3. RTS/CTS Frames,
4. ACK Frames.

Furthermore, frames exist to support the optional PCF function during a Contention Free Period (see chapter 2.4.2) such as:

1. Beacon frames,
2. CF-Poll,
3. CF-End.

2.4 MAC sublayer functional description

The fundamental mechanism for MAC admission in IEEE 802.11 is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [5]. Each Station (STA) in the Basic Service Set (BSS) will only access the medium if there are no ongoing transmissions. Two transmission modes exist; for Contention Periods (CP) and Contention Free Periods (CFP). These are time multiplexed in a superframe as shown in Figure 2.4[6].

---

[5] This is roughly similar to CSMA/CD (Collision Detect) in the legacy IEEE 802.3 Ethernet standard for wired networks, except that in CA no collisions should occur, and so in theory retransmissions are not needed.

[6] A similar diagram exists in [2], chapter 9.3.1

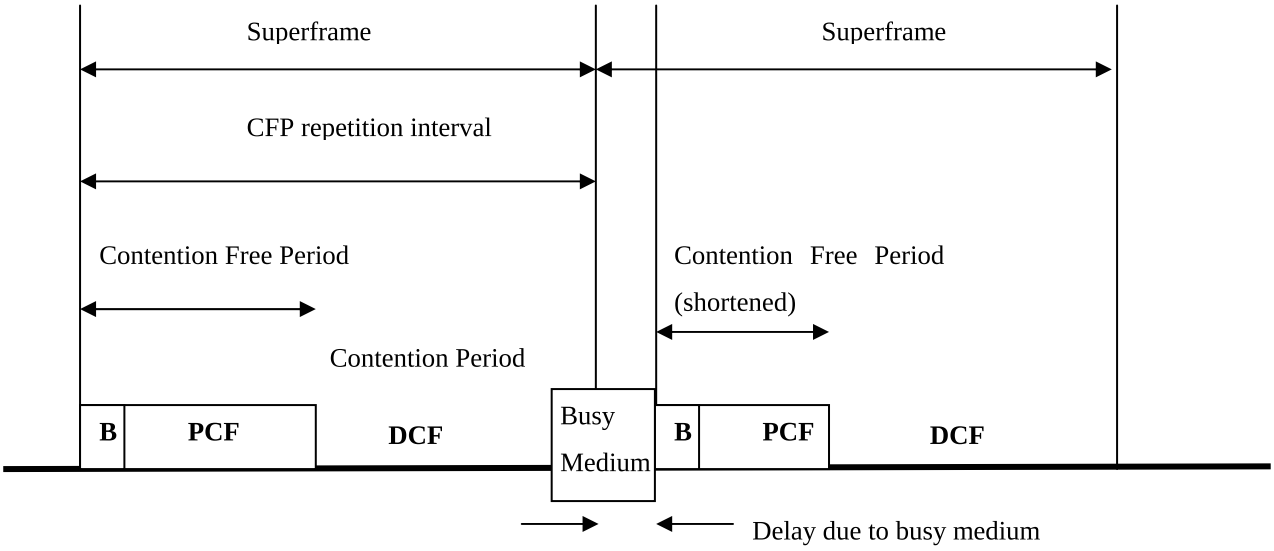

Figure 2.4 Contention Periods and Contention Free Periods

These transmission modes are discussed in more detail in the following sub-chapters.

2.4.1 Distributed Co-ordination Function (DCF)

The DCF is used during a Contention Period. Figure 2.5[7] depicts a typical DCF mechanism after the source has won contention. The STA senses the medium is unused if after a DIFS time interval (DCF InterFrame Space) it remains unused. Then it starts a backoff counter and waits for a further random time before initiating the frame transmission sequence. This prevents packets from two STAs colliding - in the case where both STAs simultaneously sense the medium is unused and simultaneously transmit.

The STA then has access to the network until it has transmitted all parts of a single MAC Service Data Unit (MSDU), but this is subject to maximum time limit. An MSDU is part of the Frame Body field and can be up to 2304 bytes long[8]. The receiving STA sends back an ACK after one SIFS (Short IFS) time interval. This always gets sent immediately since a SIFS has a smaller duration than the DIFS. If another STA had also sensed the idle medium and had waited for a DIFS period but still lost contention then that STA stops, but does not

---

[7] A similar diagram exists in [chapter 9.2.5.4, 2]

[8] The remaining 8 octets of the Frame Body are used for privacy functionality.

reset its backoff counter. In this way it has a better chance of winning contention after the next DIFS has elapsed, since the counter will resume at the frozen value.

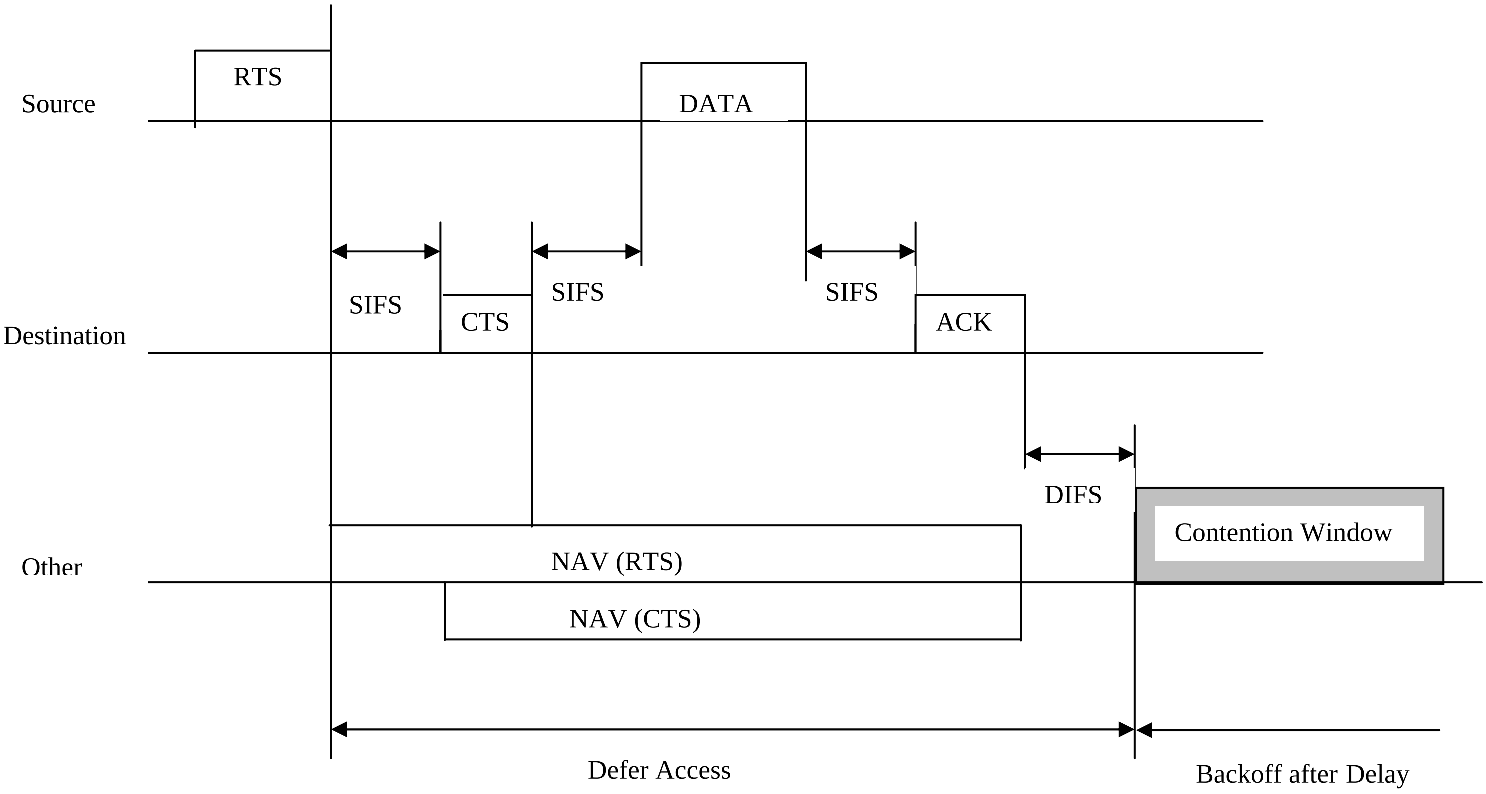

Figure 2.5 The DCF in operation

Carrier Sensing

Each STA regards the medium as either in state IDLE or BUSY and sets a Network Allocation Vector (NAV) during which time access to the medium is forbidden. The duration of the NAV is based on information received by the STA as it listens to the Request To Send (RTS) frame sent by the originating STA and the Clear To Send (CTS) frame sent by the terminating STA. When the NAV timer expires all listening STAs assume the medium is in state IDLE and recommence their DIFS timers.

Interframe Spacing

The time intervals between the different frames are called Inter Frame Spaces (IFS).   These intervals are used as waiting periods prior to transmitting a frame and therefore due their different lengths enforce priority.

The different types of IFS are as follows:

1. Short IFS (SIFS) is used before an ACK frame, before a CTS frame and before a Data frame. Since it is the shortest IFS it guarantees that the STA will maintain control of the medium for the duration of the frame exchange sequence,
2. DCF IFS (DIFS) is used to by all STAs to sense the medium is idle before transmitting a DATA frame,
3. Extended IFS (EIFS) is used when an incorrect transmission (as detected by an incorrect Frame Check Sequence (FCS) bit) is detected by the terminating PHY. If the transmission medium remains idle for this period, retransmission of the erroneous frame is permitted. An EIFS is larger than a DIFS.

Table C on page 31 shows the IFS values used in the simulator. They are based on the default values of the Management Information Base (MIB) as presented in Annex D of [1] and [2].

Determination of Backoff Time

After a DIFS time has expired the additional random backoff time must pass before a STA can begin a frame exchange sequence. This is to prevent collisions when more than one STA begins the frame exchange sequence at the same time. It is derived as follows:

$$\text{Backoff Time} = \text{Rand}(\ ) \times \text{aSlotTime} \qquad (1)$$

Where, `Rand ()` = a pseudorandom integer in the range [0, CW]

`aSlotTime` = the slot time duration of the PHY

`aCWmin <= CW <= aCWmax`.

After a successful MSDU frame transmission the Contention Window is reset to aCW_{min}. However if a frame transmission is unsuccessful a retry counter is incremented and the Contention Window is also increased to the next value in a series as follows:

$$CW = 2^{2+i} - 1 \qquad (2)$$

where i is the number of transmission attempts.

After further unsuccessful transmission attempts – and provided the retry counter has not reached its upper limit – the CW eventually reaches aCW_{max}. This is 1023 slots for IEEE 802.11. A retry is classed as the entire frame exchange sequence and it can be seen that by increasing the Contention Window retransmission attempts are given increasingly lower priority the more often they are unsuccessful! [9]

2.4.2 Point Co-ordination Function (PCF)

The PCF is an optional extension to the DCF and is used during Contention Free Periods (refer again to Figure 2.4 on page 8). It is controlled by the STA that is also the Access Point (AP) for the Infrastructure BSS. This node is called the Point Controller (PC). This implies that the PCF cannot be supported within a simple IBSS since there is no AP in an IBSS. The AP maintains a list of STAs that have registered that they wish to transmit, and polls them to begin transmitting to the medium without disturbance.

PCF Access Procedure

The PC assumes control during the Contention Free Period by sensing the medium is idle for a PIFS duration. The PIFS duration is shorter than the DIFS and so the PC gains control of the medium before STA using DCF. At the start of the PCF a Beacon Frame is transmitted to all STAs within range containing parameters that describe the Contention Free Period. After a further SIFS period the PC will transmit one of the following frames:

1. a CF-Poll , to request a STA to begin transmission,
2. a Data frame, if the PC needs to send it’s own data,
3. a Data frame + CF-Poll, a piggybacking mechanism,
4. a CF-end frame, when there is no traffic scheduled for the CFP.

NAV operation during a CFP

One of the parameters in the Beacon Frame, the *CFPMaxDuration*, is used to set the NAV of each STA so that no STA can take control of the Contention Free Period away from the Point Co-ordinator. Since the CFP can be of variable duration, *CFPMaxDuration* can have a minimum and maximum value and this is documented in [Annex D, 2].

2.4.3 Combining DCF and PCF

DCF is the most favourable mode in a wireless network where the topology is changing in a random unplanned fashion (sometimes called an *ad-hoc* network). PCF is favourable when the network has a permanent infrastructure because more guarantees can be made on maximum throughput etc. and the fair sharing of resources amongst all STAs is possible.

---

[9] Refer also to [chapter 9.2.4 figure 50, 2] if further explanation is required.

The PCF duration per superframe is variable and controlled by a parameter to tailor its contribution to networks with different characteristics. PCF is the method in legacy IEEE 802.11 for providing a limited Quality of Service. In practice PCF + DCF is not used as often as DCF alone.

2.5 Summary

In this chapter a very brief summary of the high level components of a wireless network is given. The distinction between an IBSS and an Infrastructure BSS is important to remember for later chapters in this document, since the GloMoSim simulator supports design of the former but not the latter type of service set[10] (although it is possible to have mixed wired-wireless networks). Further detail on issues not directly relevant to this report such as the concept of an Extended Service Set and Overlapping Basic Service Sets are available in [chapter 5, 2].

The most important part of the chapter was a detailed description of the MAC sublayer and it's two constituent functions, namely PCF and DCF. It should be noted that in practice DCF is the most widely used function of the two and therefore more effort has been made in explaining its operation.

DCF is also important because it acts as the foundation for the Enhanced DCF. EDCF is introduced in Chapter 3 as the principle mechanism for providing Quality of Service.

Please refer to [chapter 9, 2] for further information to accompany this chapter.

---

[10] It is possible to nominate a server node as a type of Access Point provided the routing is configured so that all clients communicate only with the serve - but EDCF rules still apply.

Chapter 3 – The IEEE 802.11e Supplementary Standard

3.1 Introduction

In the previous chapter the legacy IEEE 802.11 MAC is described. In this chapter the Quality of Service aspects of the IEEE 802.11e supplementary standard are introduced.

Section 3.2 briefly mentions the new QoS fields added to the IEEE 802.11 MAC frames.

Section 3.3.1 introduces the dominant new function in the standard, the enhanced version of the DCF, called EDCF.

Section 3.3.2 briefly mentions the Hybrid Co-ordination Function. This function aims to improve upon the PCF and addresses some of the limitations of the PCF. However, like the PCF, it does not feature in the simulator and therefore is only briefly described for completeness.

The legacy standard is not rendered obsolete because of the introduction of the IEEE 802.11e amendment and an upgraded BSS, called a QBSS, may be a mixture of IEEE 802.11e compliant QSTAs and non-conformant legacy STAs. Therefore Section 3.5 addresses the issue of co-existence and backward compatibility between the three functions, EDCF, PCF and HCF.

3.2 The IEEE 802.11e MAC Frames and Formats

Chapter 2.3 on page 6 described the legacy IEEE 802.11e frame types and formats. With IEEE 802.11e new frame types are required so that it is possible to distinguish between frames from STAs and frames from QSTAs. As before *piggybacking* is permitted. New frame types can be sent alone or in combination, e.g., in certain circumstances it is possible to send a new QoS Data frame + CF-Ack together as a valid frame combination. Please consult [Table 1 chapter 7.1.3.1.5, 2] for an amended exhaustive list of frame types and valid combinations. Figure 2.3 below illustrates the standard frame format of the IEEE 802.11e MAC. A new two octet field called QoS Data has been added. It is only used in new QoS supporting frame types.

Frame Control (2)	Duration Id (2)	Address 1 (6)	Address 2 (6)	Address 3 (6)	Sequence Control (2)	Address 3 (6)	Q0S Control (2)	Frame Body (0-2312)	FCS (4)

Figure 3.1 Frame format of IEEE 802.11e MAC

The Frame Control field holds the frame type. The new QoS frame types are:

1. QoS Data.

Furthermore, frames exist to support the HCF function during a Contention Free Period (see chapter 3.3.2) such as:

2. QoS CF-Ack,
3. QoS CF-Poll,
4. QoS Null.

The QoS Control Field

This field is used primarily to identify the traffic category to which the frame belongs. The field consists of subfields that contain various Quality of Service information as follows:

- the priority of the frame [0..7],
- the TXOP duration requested during a Contention Free period (from non-HC),
- the TXOP duration allowed during the CFP (from HC only),
- the amount of buffered traffic in the queue of the sending STA.

For further explanation please refer to [chapter 7.1.3.5, 1].

3.3 MAC Sublayer Functional Description

3.3.1 Enhanced Distributed Co-ordination Function (EDCF)

The EDCF mode used in the Contention Period introduces the notion of Access Categories (AC) which are used too help define priorities for different types of traffic. There are four ACs (the designations are called Best Effort, Video Probe, Video and Voice). Each AC has separate parameters for controlling access to the medium; these are the AIFS (Access IFS) and the Backoff[11]. The range [0, CW] defines the value of the Backoff and CW is chosen from the range $CW_{min} <= CW <= CW_{max}$. However there are now separate CW_{max} and CW_{min} values per Access Category.

---

[11] There are three more parameters defined in [1] - TXOPBudget, TXOPLimit and Load – but these are less significant and not handled by the simulator.

The AIFS replaces the DIFS for a given Access Category. For example AIFS_0 is used for AC_0, …, AIFS_3 is used for AC_3 etc.

Priority is introduced when Access Categories with lower parameter values experience lower waiting times and are thus scheduled ahead of Access Categories with higher backoff times. In Figure 3.2[12], AC_3 is the highest priority because AIFS_3 is of shorter duration than AIFS_1 and AIFS_2. Furthermore the Contention Window range [0, CW] is more likely to be shorter for a higher priority Access Category because CW_{min} and CW_{max} are smaller for the highest priority AC. Table C lists the default parameter values.

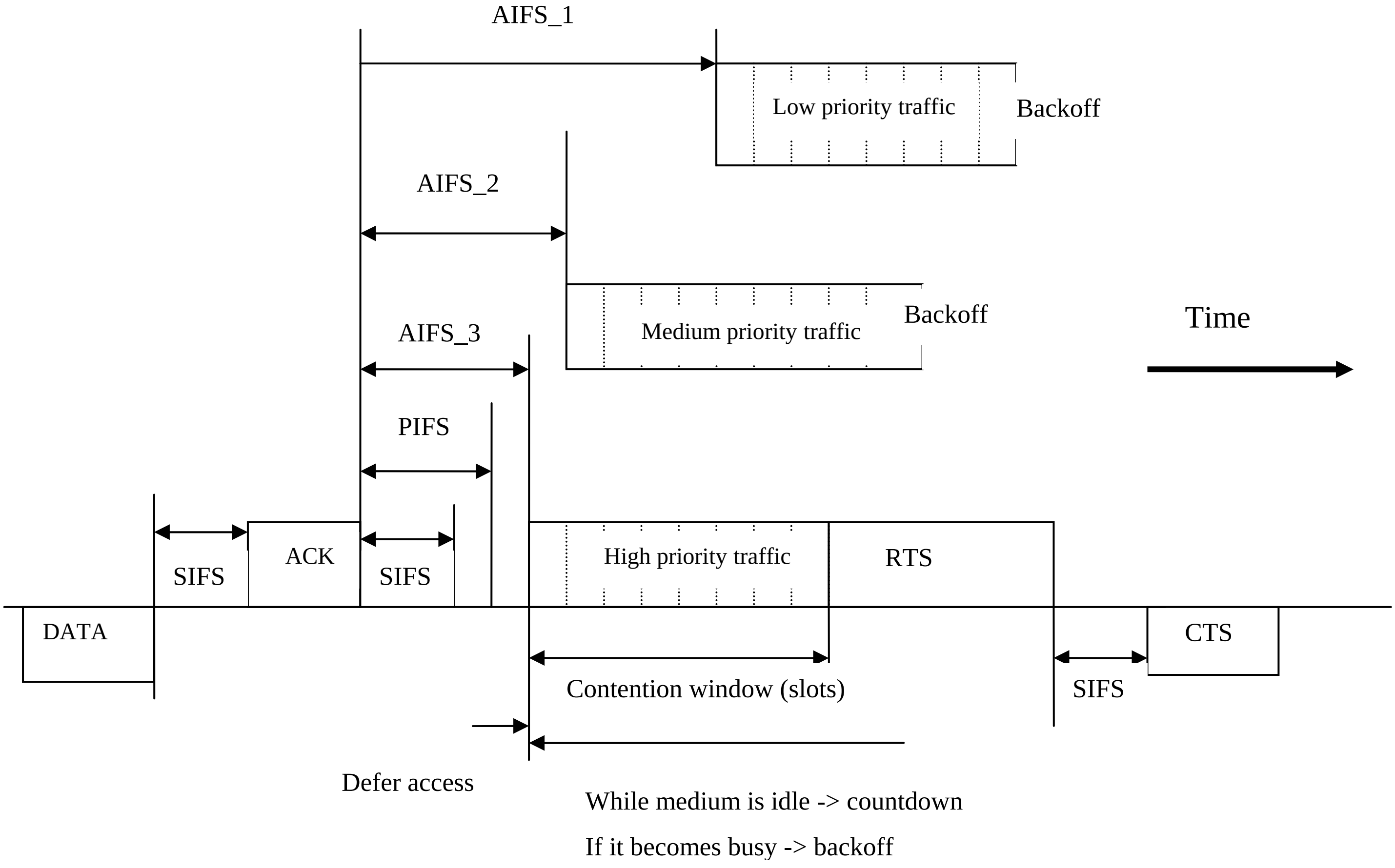

Figure 3.2 Access Categories and their influence on Backoff

---

[12] There is a similar figure in [4].

As well as AC's there also further priority values assigned to individual Traffic Streams (TS). In total there are 8 priorities across the four AC's. The relationship between priority and Access Category is shown in Table A[13].

Priority (Same as 802.1D Priority)	802.1D Designation	Access Category (AC)	Designation (Informative)
1	BK	0	Best Effort
2	-	0	Best Effort
0	BE	0	Best Effort
3	EE	1	Video Probe
4	CL	2	Video
5	VI	2	Video
6	VO	3	Voice
7	NC	3	Voice

Table A Priority to Access Category Mappings

With 802.11e it is now possible to have *intra*-STA contention, as well as *inter*-STA contention, when the backoff counters of two TSs, within the different AC's, within the same STA, expire simultaneously. This conflict is resolved in favour of the higher priority TS. However the probability of this occurring is very low.

3.3.2 Hybrid Co-ordination Function (HCF)

The Hybrid Controller (HC) is the equivalent of the Point Controller in the HCF function. The biggest difference between the Hybrid Controller and the Point Controller is that the HC may operate in both the Contention Period and the Contention Free Period. As the draft standard quite succinctly states:

> *…all STAs and QSTAs inherently obey the medium access rules of the HCF, because each frame transmitted under HCF by the HC or by a non-AP QSTA*

---

[13] This is further described in, [1], *chapter 6.1.1.3, table 0.1*. These priorities are inherited from the IEEE 802.1D standard for MAC bridging.

> *contains a duration value to cause STAs and QSTAs in the BSS to set their NAV to protect the frames expected to follow that frame.* [1, chapter 9.10.2]

As a result even though the Beacon frame had indicated the end of the Contention Free Period the HC can still maintain control of the Contention Period by setting the NAV during the Contention Period. Figure 3.3 illustrates the new 802.11e superframe structure[14] and shows an example of HCF intervention with a QoS CF-Poll frame during the Contention Free Period.

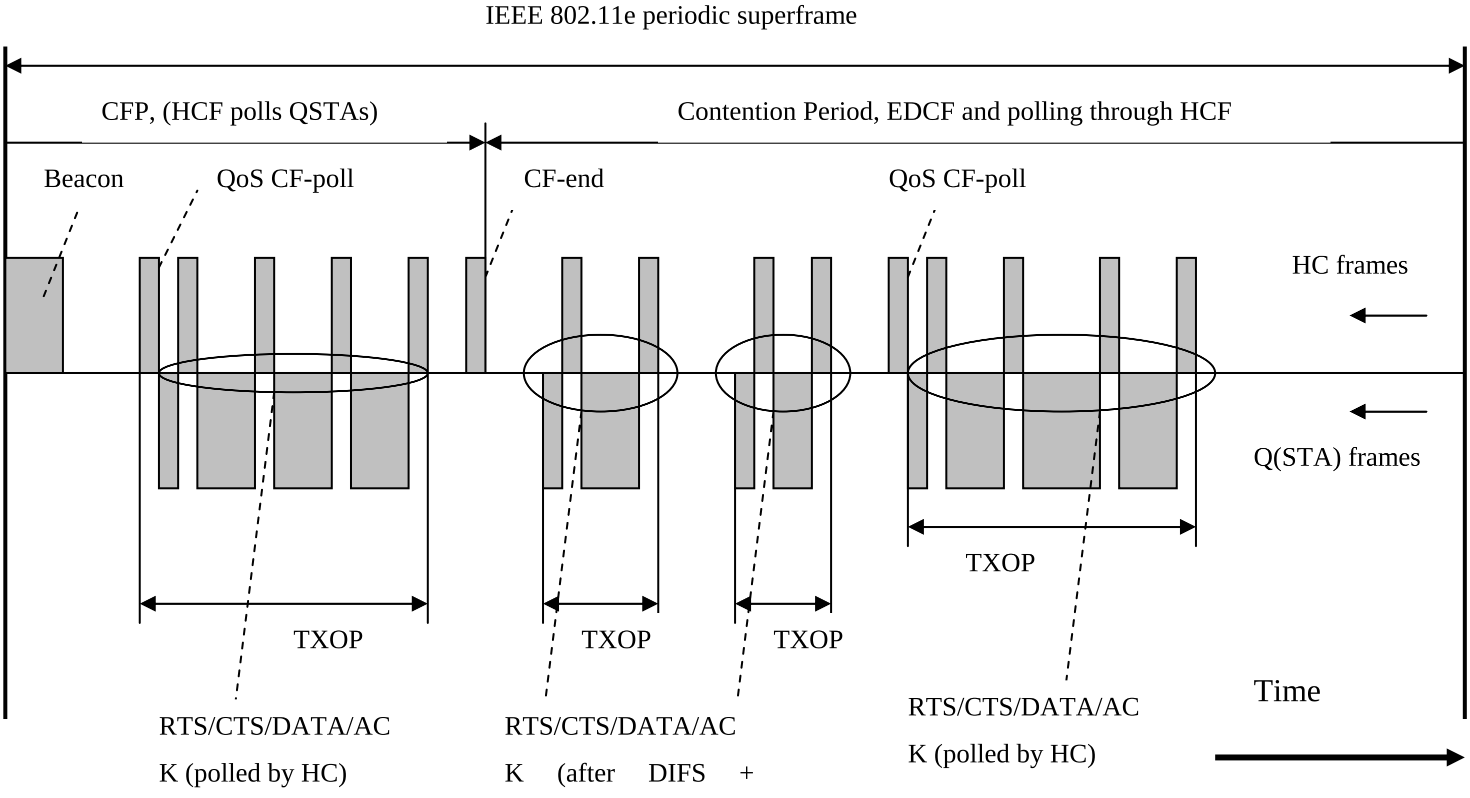

Figure 3.3 The IEEE 802.11e superframe structure

3.3.3 Co-existence of DCF, EDCF, PCF and HCF

An STA using the DCF function can co-exist with and operate concurrently with either a Hybrid Controller using the HCF function or a Point Controller using the PCF function. If a PC operates in the BSS then the DCF and PCF functions alternate with a Contention Period and a Contention Free Period. However if a HC operates in the QBSS the non-QoS STA

---

[14] There is a similar diagram in [4].

treats the HC as if it were a PC, i.e., the HC must be able to receive legacy IEEE 802.11 frame types as well as new QoS frame types. The non-QoS STA will use the DCF access method only during the Contention Period.

3.3.4 Other Quality of Service enhancing features

EDCF-TXOP bursting

EDCF Transmission Opportunity (TXOP) bursting is a feature allowing a QSTA access to the medium for more than one MSDU transmission attempt after contention has been won. Higher priority traffic streams are given more time to burst than lower priority traffic streams. This feature should significantly improve link utilisation for high priority traffic.

In Annex D of the IEEE 802.11e MIB, the parameter that specifies the upper limit that a QSTA can transmit for in EDCF mode is called *dot11EDCFTableTXOPLimit.* The default value for this parameter is defined per Access Category[15] as follows:

- **AC_0**: *dot11EDCFTableTXOPLimit* = 0μs, only one MSDU can be sent.
- **AC_1**: *dot11EDCFTableTXOPLimit* = 1500μs for 802.11b or 3000μs for 802.11a.
- **AC_2**: *dot11EDCFTableTXOPLimit* = 3000μs for 802.11b or 6000μs for 802.11a.
- **AC_3**: *dot11EDCFTableTXOPLimit* = 1500μs for 802.11b or 3000μs for 802.11a.

The second and subsequent frames in an EDCF TXOP burst must be of the same Access Category as the first frame.

Direct Link Protocol

This protocol is introduced in 802.11e to allow QSTAs that are part of an Infrastructure BSS to send and receive data on a peer to peer basis without sending messages via the AP. The Direct Link Protocol is necessary because a QSTA may be in power saving mode and only the AP can wake up the QSTA. Figure 3.4[16] assists in describing the following steps in the protocol's invocation.

1. QSTA 1 wishes to send a frame to QSTA 2 but must first inform the AP by sending a DLP request frame.
2. The AP informs QSTA 2 of the DLP request.
3. If the DLP request is accepted QSTA 2 responds to the AP with details of it's capabilities and also it's addressing information. The DLP is now activated.

---

[15] The values for AC_3 seem strange. It would be logical to give AC_3 the largest EDCF TXOP.

[16] A similar diagram appears in [1], chapter 5.8.

4. The AP informs QSTA 1 that QSTA 2 is ready to receive a frame. The AP also sends QSTA 1 details of the capabilities of QSTA 2.
5. QSTA 1 engages QSTA 2 in peer to peer dialog. A timer parameter is set in QSTA 2 specifying a period during which it cannot return to power saving mode. After a timeout the DLP is deactivated and peer-to-peer communication is no longer possible.

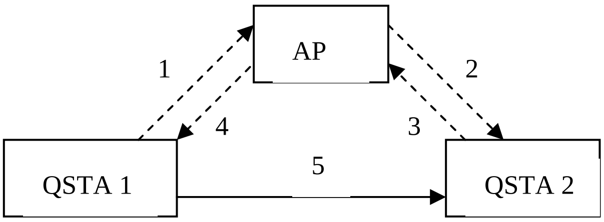

Figure 3.4 The Direct Link Protocol

3.4 Summary

This chapter describes the main Quality of Service features of the draft IEEE 802.11e supplementary standard.

The Hybrid Co-ordination Function is the umbrella term for the total functionality of the IEEE 802.11e MAC. It firstly comprises a period of controlled access where a Hybrid Controller seizes control of the medium and either polls QSTAs to begin a TXOP or otherwise begins a TXOP for itself. This controlled access can now be either in the CP or the CFP. Secondly it comprises a period of contention access using the EDCF function where QSTAs vie for, and obtain, access to the medium based on the priority of the traffic they wish to send.

The standard also defines new Quality of Service frame types and combinations to support these new functions. These are briefly described and attention is drawn to the Quality of Service fields that are of interest.

For the remainder of this report the focus is on the contention access parts of the superframe, i.e., the implementation, testing, and simulation of the EDCF.

Chapter 4 – The GloMoSim Simulator & PARSEC Language

4.1 Introduction

GloMoSim is a discrete event simulator developed by the Parallel Computing Laboratory at UCLA[17]. It is developed using the PARSEC programming language.

Section 4.2 introduces PARSEC and explains the extent of its influence over the simulator.

Section 4.3 examines the interesting aspects of GloMoSim on a layer by layer basis.

4.2 A Brief Presentation of Parsec

PARSEC (Parallel Simulation Environment for Complex Systems) is a C-based language that can be used to simulate discrete events. The most important concept is that of an *entity*. A PARSEC program is built up of groups of *entities* that perform buffered message passing between one another. A *message type* consists of a name and a parameter list and is similar to a struct in C.

An entity sends a message by executing a **send** statement, for example:

send *message_type* **to** *destination* **after** *delay;*

An entity receives a message with the **receive** statement, for example:

receive (*message_type1*) [**when** *guard1*] *statement1* ;
or receive (*message_type2*) [**when** *guard2*] *statement2* ;
.
.
or receive (*message_typen*) [**when** *guardn*] *statementn* ;

A PARSEC entity is similar in structure to a standard C function and it describes a class of objects. Every PARSEC program must contain a *driver entity* whose purpose is similar to the function ***main( )*** in C. Execution of a PARSEC program begins by executing the first statement in the body of the driver entity.

There are two variants of the Parsec compiler available, Parallel and Sequential. Parallel Parsec allows simulations to be distributed across more than one processor by partitioning

---

[17] It has also been developed into a commercial product called *QualNet* by a UCLA start-up company called *Scalable Network Technologies*, http://www.scalable-networks.com.

the nodes over each processor. This has the advantage of increased scalability and faster simulation times and is supposedly the major attraction of PARSEC.

However in this project it is the sequential version of the compiler that is used[18]. There is only one partition on a single processor that simulates all the nodes in the network. Therefore within GloMoSim there are two entities, the driver and the single partition. Figure 4.1 shows the signalling between the entities. After receiving the StartSim message partitionEntityName goes into a loop waiting for events to occur that will trigger message sending and reception. When no more events are scheduled the simulation ends.

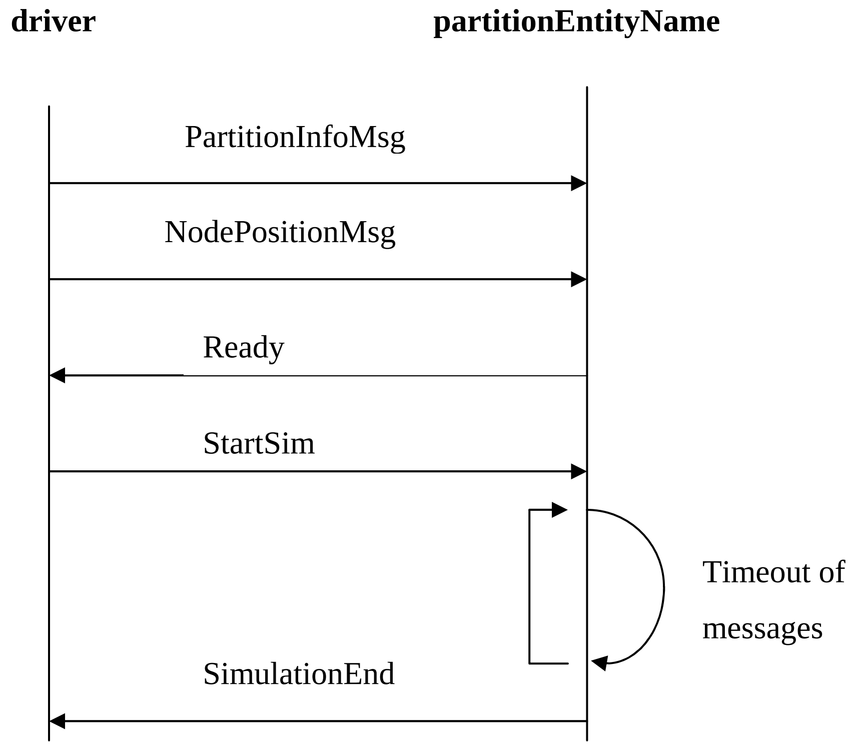

Figure 4.1 Message passing between the GloMoSim simulator entities

The major differences between a sequential Parsec and the C language from a designers perspective are the additional libraries of functions to interface with the simulation clock, pseudorandom number generating functions and message communication.

The sequential model uses a data structure called a Global Event List or *SplayTree* that stores all the time dependent events that occur in the system in their timestamp order. Examples of such events are timeouts of IFS of Backoffs, generations of packets at the

---

[18] The parallel version is not freely available from UCLA until competence has been demonstrated with sequential Parsec! In any case the sequential version was deemed adequate for the needs of this project.

Application Layer at defined interdeparture times and simulation of propagation delay at the Channel Layer. Messages are implicitly time stamped with the current value of the simulation clock. However if a message needs to be scheduled for a future time, a delay must also be specified. For further information, please refer to the User Manual, [10].

4.3 The Layered Architecture of GloMoSim

GloMoSim (Global Mobile system Simulator) consists of libraries of PARSEC programs that simulate wireless and wired networks. The simulator is built using a layered approach that tries to match the OSI 7-layer network architecture.

Many of the layers have simulated more than one protocol, e.g. at the MAC layer there is Wired, CSMA, TSMA, FAMA, 802.11 and now 802.11E. Users can, to a limited extent, specify the protocol they wish to use at the given layer. At the Application Layer it is also possible to choose multiple protocols simultaneously for various types of traffic.

4.3.1 The Application Layer

The protocols available are as follows:

Continuous Bit Rate (CBR)

This is the application that is used to provide the periodic high priority voice and video traffic for the simulation. It is possible to specify item size, inter-departure time, number of items to send and start/end times.

Hyper Text Transfer Protocol (HTTP)

A set of Cumulative Distribution Function (CDF) tables exists for the size of HTTP items retrieved, number of items per 'Web page', think time, and user browsing behaviour. They are based on empirical measurements of real web traffic, consult [21] for further details. The output of a pseudorandom number generator function and linear interpolation is used to choose values from the CDF tables. For the simulation HTTP traffic is used exclusively to provide the low priority, best effort data traffic.

File Transfer Protocol (FTP)

The inter-departure time, duration and size of packets are randomly determined from *tcplib* – a freely available library for generating realistic TCP/IP network traffic. Please refer to [20] for further information on *tcplib*. FTP uses *tcplib* to randomly determine the amount of application layer items to send. The size of each item is always randomly determined by *tcplib*.

TELNET

A TELNET application also exists but its use is considered inappropriate for this project.

4.3.2 The Transport Layer

The Transport Layer provides two protocols either Transmission Control Protocol TCP or User Datagram Protocol (UDP). The choice of protocol is hard-coded into the simulator i.e. CBR traffic always uses the UDP while all other traffic (HTTP, FTP etc.) uses TCP.

The TCP code is adapted from FreeBSD v2.2.2 source code, consult [19] for more information.

4.3.3 The Network Layer

The only protocol supported at the Network Layer is the Internet Protocol (IP).

Packets that arrive at the network layer are queued in a First-In-First-Out (FIFO) fashion waiting until the MAC layer is ready to service a packet. The size of the queue is adjustable.

The network layer also implements choice of routing protocols, e.g., AODV, DSR, ODMRP and ZRP. However, unless otherwise stated, static routing has been chosen for the simulations

4.3.4 The MAC Layer

The MAC Layer of interest is the 802.11 (other available protocols have been mentioned previously). Here only the DCF function is implemented. Therefore there is no Contention Free Period and it is not possible to define a fully functional AP[19]. The only frame formats that are supported are the RTS, CTS and ACK Control Frames and Data Frames. Table B shows how GloMoSim represents the frame formats of the IEEE 802.11 MAC. Refer to [2] and to page 13 for a review of other frame formats and types.

---

[19] It is shown in Chapter 6 how the behaviour of an AP can be sufficiently approximated for the purposes of the simulation using static routing rules.

```
/*
 * CTS and  ACK frames.
*  Note: All frames types must match the short
*  control (this one) exactly for its first four
*  (universal) fields.
*/

typedef struct _Mac802_11ESCtrlFrame
{                                    //  Should Be  Actually
   unsigned short frameType;         //     2          2
   char Padding[2];                  //     0          2
   int duration;                     //     2          4
   NODE_ADDR destAddr;               //     6          4
   // char FCS;                      //     4          0
} M802_11EShortControlFrame;         //---------------------
                                     //    14         12
```

```
/*  QoS Data frame header. */
typedef struct _Mac802_11EFrameHdr
{                                    //  Should Be  Actually
  unsigned short frameType;          //     2          2
  char Padding1[2];                  //     0          2
  int duration;                      //     2          4
  NODE_ADDR destAddr;                //     6          4

  NODE_ADDR sourceAddr;              //     6          4
  //NODE_ADDR Address3               //     6          0
  unsigned short seqNo;              //     -          -
  unsigned char  fragId;             //     2          3
  unsigned short QoSControl;         //     2          2
  char Padding2[5];                  //     0          5
  char FCS[4];                       //     4          4
} M802_11EFrameHdr;                  //---------------------
                                     //    30         34
```

Table B PARSEC data structures representing IEEE 802.11e frame formats

A state machine diagram visually describing the operation of the 802.11 MAC is available in Appendix B, Figure 7.1and Figure 7.2.

4.3.5 The Radio and Channel Layers

The Radio and Channel Layers are based upon the original legacy 802.11 standard; there is no support for 802.11a or 802.11b. This means the simulator’s frequency band is ~ 2.4Ghz and the theoretical maximum bitrate is 2 Mbits/sec.

4.4 Representation of Important Data in the Simulator

Data structures (i.e. C *structs*) are used to represent many of the fundamental elements in the system e.g. nodes, channel types, messages and data relevant to a particular layer. The complete state of each node is maintained by separate data structures. When the simulation code of a node is being executed it does not have access to the data structures of the other nodes in the simulation (which is unfortunate for statistics collection). Structures are used to represent data specific to different Nodes, Messages, Timers and each of the simulation layers. Figure 4.2 shows the messages that are passed between the Application Layer, Transport Layer and Network Layer.

Each of the layers are represented by families of functions. A special initialisation function for each layer is called at the start of the simulation to read important data from the input file to 'set the scene' for the simulation, e.g., the protocol layer chosen, the size of the Network Layer queues, the type of routing chosen etc. Functions are also available to collect key statistics at each of the layers at the end of the simulation

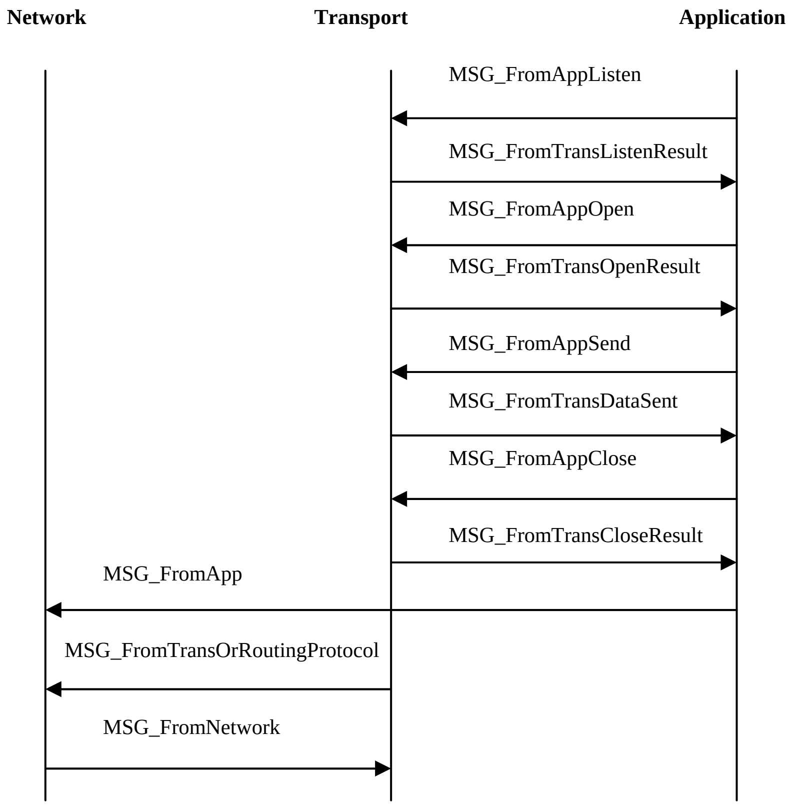

Figure 4.2 Message passing between Application Layer, Transport Layer and Network Layer

4.5 Use Case Study: Protocol Data Unit (PDU) Movement

All applications have a start time configured in the *app.conf* input file. This means that the application layer will send it's first message to the partition entity after the start time timeout occurs.

The *splaytree* functionality is used to delay the sending of a message until a future time.

The packet size decreases and increases as it moves up and down through the layers respectively.

The Message structure contains a ‘packet’ variable of type string. This is the packet without overhead. The Message structure also contains a string called ‘info’ which can be used as a container for internal simulator control information.

The functions GLOMO_MsgAllocPacket( ), GLOMO_MsgAddHeader( ) and GLOMO_MsgRemoveHeader( ) perform the insertion and deletion of layer headers as the packet moves through the layers.

The passing of Protocol Data Units (PDU) across the boundary between layers conforms to an Application Programming Interface (API) consisting of three functions, whose prototypes are as follows:

```
GLOMO_MsgSend     (GlomoNode *node, Message *msg, clocktype delay)
GLOMO_CallLayer   (GlomoNode *node, Message *msg)
GLOMO_xxxLayer20  (GlomoNode *node, Message *msg)
```

Figure 4.3 takes a snapshot of the GloMoSim functions that are called for a packets typical transition down the stack, from Application Layer to Channel Layer.

---

[20] ‘xxx’ is the layer type e.g. Application, Transport etc.

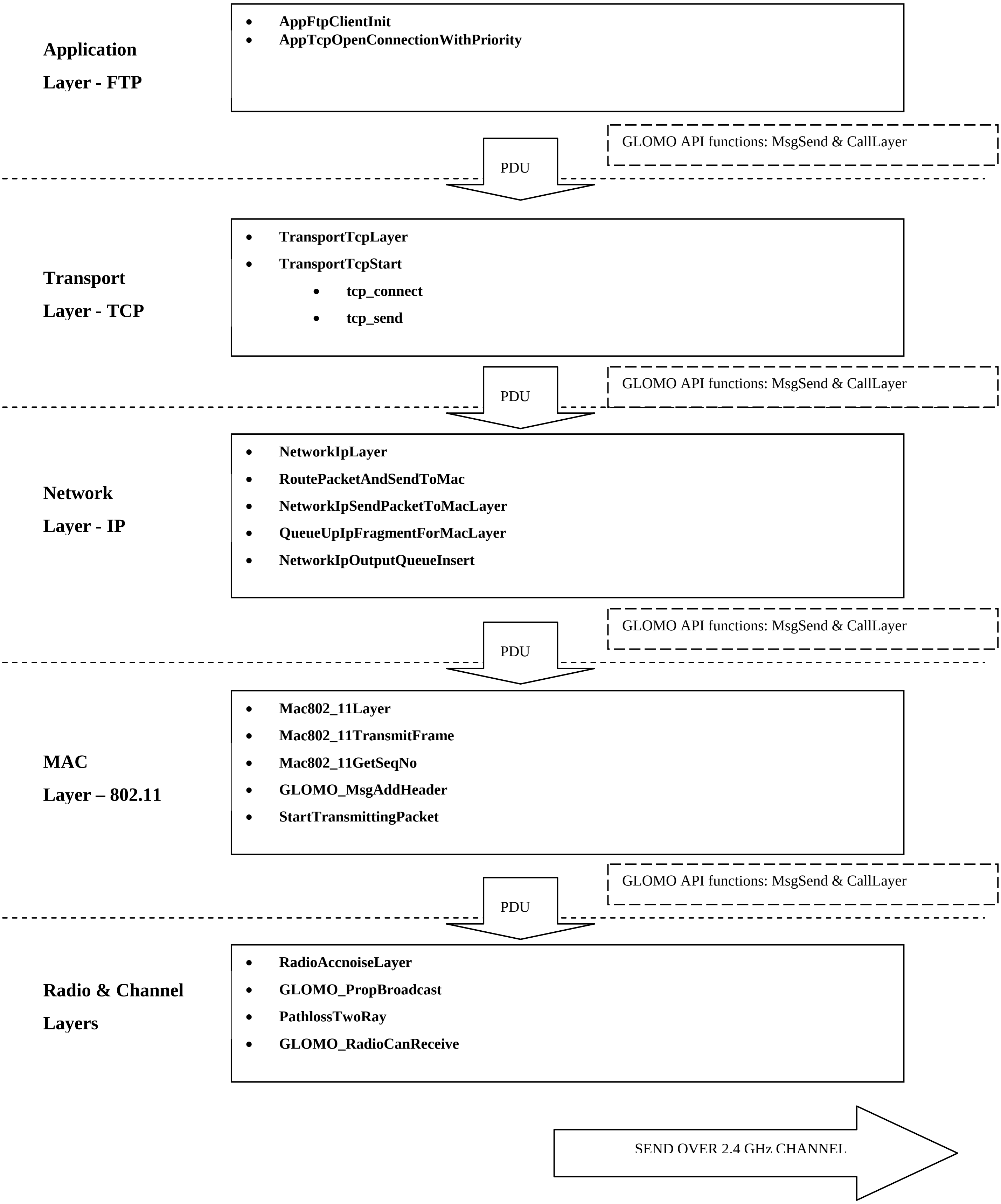

Figure 4.3 Functions called when Originating Node generates an item

4.6 Summary

This chapter briefly describes the Parsec programming language that can be used to simulate discrete events. It is noted that PARSEC functionality has a limited impact on the performance of the GloMoSim simulator when it’s sequential form is used. Parsec programs can be treated as C programs with enhanced timer capabilities and a message passing interface.

The GloMoSim simulator is described at a high level on a layer by layer basis. The operation of the MAC layer is most easily represented by the state machines in Appendix B. A description is given of how important data is represented in the simulator is given. Finally a description of the functions called by each layer during a typical message sending is given and the API that allows PDUs to pass between layers is identified.

In the next chapter, each of the layers that require re-design is presented.

Chapter 5 – Implementation of IEEE 802.11e in GloMoSim

5.1 Introduction

The objective of this chapter is to detail the design work involved in this project. Using GloMoSim v2.03 - incorporating the 802.11 MAC DCF function – as a design base the task was to design a new EDCF function at the MAC layer while also making any necessary adaptations to the higher layers.

Subchapter 5.2.1 describes the necessary alterations to the I/O handling code to support a new priority parameter and a new protocol definition.

Subchapters 5.2.2 to 5.2.4 outline the modifications to the Application, Network and MAC Layers respectively in terms of new data, queue handling techniques, timer values, alteration to frame formats and statistics collection.

5.2 Designing an EDCF function

Figure 5.1 on page 32 shows the new queuing concept at the network layer and how each queue is mapped to an access category at the MAC layer. The comparison is noted between the legacy 802.11 structure on the left and the 802.11e structure.

5.2.1 Adaptation of File I/O handling

The code to read the files that specify the input parameters to the simulation are updated.

This allows a new protocol named “802.11E”, to be declared for the MAC layer in the ‘CONFIG.IN’ file.

In addition, a new parameter named “priority” can now be assigned to each of the traffic sources CBR, FTP, HTTP and TELNET in the ‘APP.CONF’ file.

For example, to use CBR, the following format is needed:

```
      CBR <src> <dest> <items to send> <item size>
                <interval> <start time> <end time> <priority>
where
<src> is the client node,
<dest> is the server node,
<items to send> is how many application layer items to send,
<item size> is size of each application layer item,
<interval> is the interdeparture time between the application layer items,
<start time> is when to start CBR during the simulation,
```

```
<end time> is when to terminate CBR during the simulation,
<priority> is in range [7(highest)...0(lowest)].
```

Note in this updated version of the simulator, when choosing MAC as 802.11E the priority is adjustable within the range BUT when choosing MAC as legacy 802.11 the priority must be set to the same value for all sources.

5.2.2 Adaptation of the Application Layer

Originating Side

The source node is normally deemed the client. Each traffic source has separate functions to implement the protocol it uses. Having accepted the priority of the items to send, each of the traffic sources passes the items to the Transport Layer with the appropriate overhead and also an internal value for priority. At this stage the priority value is mapped into an internal representation of priority (i.e. highest priority 1,…., lowest priority 8) in order to simplify the queuing process at the network layer.

The CBR client is also adapted to send the priority information inside the packet along with a timestamp of the current simulator clock value. This allows the total delay, average end-to-end packet delay and delay variation to be determined for each priority at the server on the terminating side.

Terminating Side

At the CBR server it is now possible to calculate whether the average delay is too large by deducting the received packet timestamp from the current simulator clock value. The priority received from the Transport Layer is unmapped back to it’s external representation.

New functionality is also introduced to calculate and display the Delay Variation, Maximum Packet Delay and Minimum Packet Delay. These delay statistics and the priority of the session are incorporated into the existing functions for printing application layer statistics.

At the HTTP server the average offered load per client/server pair and the ratio Uplink:Downlink bytes sent is calculated and recorded as a new statistic. All new statistics are ported to the output file ‘GLOMO.STAT’.

5.2.3 Adaptation of the Network Layer

The network layer is updated with eight new FIFO queue’s representing the eight possible priorities of the different traffic sources. See Figure 5.1 at the ‘NW Layer’.

The priority value is stored in the unused ‘Type of Service‘ field of the IP header. In a multiple hop scenario packets that are received by nodes that are not destination nodes can now be sent up to the network layer. Then the *ipheader* -> *ip_tos* (Type of Service field of IP header) is read to put the packet back in the correct priority queue for forwarding on to the next hop node.

5.2.4 Adaptation of the MAC Layer

It is helpful to read this section in conjunction with the MAC State machine in Appendix B, Figure 7.1 and Figure 7.2.

Transmitting Side

The MAC Data frame format is adjusted to include two bytes Quality of Service data in order to conform to the standard, see page 13.

A new enumerated data type is introduced to represent the Access Category. New data is also introduced to represent the Backoff, the paused Backoff, and the Contention Window on a per Access Category basis.

New initial values for the AIFS times per Access Category are chosen to match the default values recommended in the MIB [Annex D, 1].

New initial values are also chosen for the maximum and minimum Contention Window limits per Access Category to match the default values of the MIB. Table C presents the values of these default settings.

```
aCWmax = 1023, aCWmin = 31, aSlotTime = 20 μs (based on 801.11)

CW  min  [ac_0,  ac_1,  ac_2,  ac_3]  =  [aCWmin,  aCWmin,  ((aCWmin  +  1)/  2)  -1,
((aCWmin + 1)/ 4) -1]
CW max [ac_0, ac_1, ac_2, ac_3] = [aCWmax, aCWmax, aCWmin, ((aCWmin + 1)/ 2) -1]
AIFS [ac_0, ac_1, ac_2, ac_3]   = [2, 1, 1, 1]
Note: all values are {n x aSlotTime}
```

Table C IEEE 802.11e MAC parameter settings from latest draft standard, February 2003

When the MAC layer receives an indication from the network layer that there are packets waiting to be serviced the following steps take place. The MAC checks that its NAV is not set [State: S_WF_NAV] and that the Radio is idle [State: S_IDLE]. If true, the MAC layer invokes a function in the Network Layer[21] to return the identity of the highest priority non-empty queue. This information is used to initialise the Access Category for the processing of the next frame. The MAC layer then sets the AIFS timer for the Access Category by sending a timeout message to itself with the AIFS delay [State: S_WF_DIFS_OR_EIFS]. When AIFS timer expires the MAC goes into backoff mode by setting the backoff timer specific to this Access Category equal to the minimum Contention Window value that is also specific to this Access Category [State: S_BO]. When the backoff timer expires, and assuming the Radio is still idle, the MAC calls another function in the Network Layer to process the first packet in the highest priority queue for this Access Category. It reads the destination address of the packet and begins the inter-node frame exchange sequence by sending an RTS message [State: X_RTS].

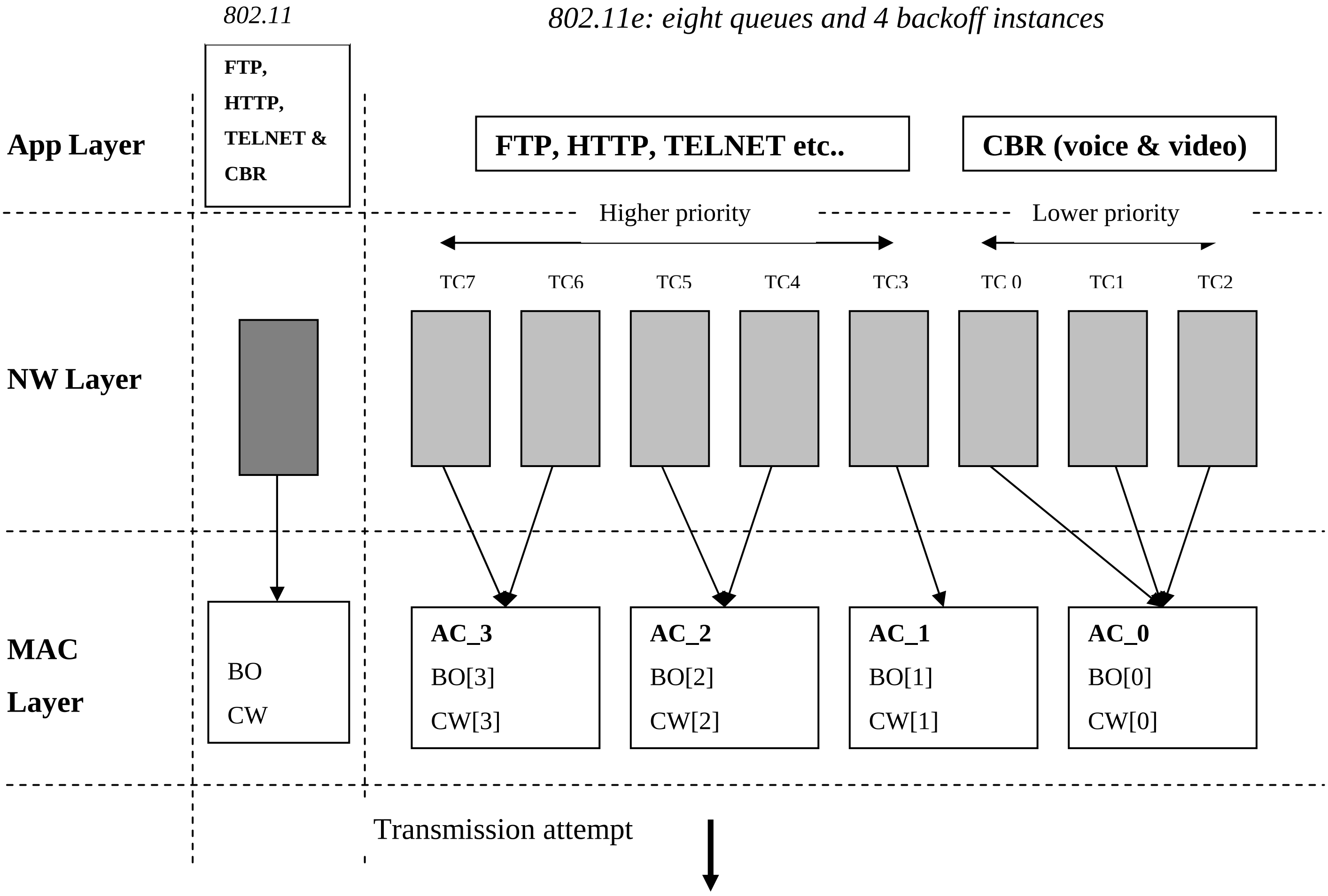

Figure 5.1 Queuing structure in IEEE 802.11e MAC with EDCF

---

[21] This is an exception to the general design rule that a layer may not call a function belonging to another layer. All communication between layers is normally performed via the API.

5.3 Summary

This chapter contains a functional description of the implementation of the IEEE 802.11e EDCF function in the GloMoSim simulator. It is now possible to define 802.11E as a valid MAC protocol in the 'CONFIG.IN' file.

Modifications are required at the Application Layer to handle the passing of new priority values to lower layers and new statistics have been formulated. At the Network Layer a new queuing mechanism has been implemented. At the MAC Layer the Access Category concept is introduced and new data is initialised.

The next chapter describes the verification process for the new function and discusses in detail whether the results obtained from a series of test simulations are confirmation that the EDCF function is successful in providing Quality of Service.

Chapter 6 – Simulations, Results and Discussion

6.1 Introduction

Subchapter 6.2 describes the types of traffic sources that are used for the simulations.

Subchapters 6.3 - 6.10 documents eight tests that aim to simulate the operation of the EDCF in different circumstances. In some tests EDCF is also compared against legacy DCF or against previous versions of the 802.11e Draft Standard.

Each of the following *Test* subchapters consists of an objective, a description of the scenario, a graph or table of the result obtained and a discussion on their meaning.

The majority of tests are based around a scenario where the offered load is increased towards the maximum accepted load of the link. Measurements of delay, delay variation, packet loss and throughput are used to discover the effects of prioritisation on different types of traffic sources.

The following assumptions are made about the simulations unless otherwise stated:

- the choice of 802.11e Quality of Service specific parameters (as listed in Table C) are based on Draft version D4.2, February 2003, of [1],
- the basic network topology consists of Node 0 as a server and a varying number of client nodes (depending on the scenario for the test). All communication is between the server and it's clients, i.e., there is no communication directly between clients[22].
- all stations are within receiving range of each other, are stationary and implement a static routing plan based on an input routing file,
- the geographic node placement is determined randomly by the simulator,
- there is a maximum of one hop between the client and the server,
- all frames including Data frames are transmitted at rate of 2 Mbit/sec,
- the maximum size of each of the eight Network Layer priority queues is 100,
- the RTS/CTS mechanism is always on,
- MAC level fragmentation is disabled,

---

[22] In this configuration Node 0 can be considered to be a simple Access Point to an Infrastructure. However there is no HCF functionality in Node 0.

- EDCF TXOP bursting and Forward Error Correction schemes have not been implemented,
- the following parameters are valid for the PHY:

```
RADIO-TX-POWER            15.0 dBm,
RADIO-RX-SENSITIVITY     -91.0 dBm,
RADIO-RX-THRESHOLD       -81.0 dBm.
```

6.2 Traffic Sources

There are three basic traffic sources used in the simulations as follows:

Low Priority Best Effort Data

The GloMoSim HTTP traffic source code is adapted from the work published by Bruce Mah, "An Empirical Model of HTTP Network Traffic" [21].

Real traffic traces from a subnet carrying HTTP traffic are used to compile a variety of empirical probability distributions describing various aspects of web client behaviour. These distributions are then used to create a synthesised workload i.e. a series of CDFs for different components of the model.

The key parameters in determining the offered load are described below (based on the values of the CDFs [23]).

The mean request size from client to server is 320 Bytes and the mean reply size from server to client is 8-10 kBytes. The mean number of files per Web Page is 2.8-3.2 (where there is a main file followed by embedded images etc.).

A separate request and reply is made for each file per Web Page. Therefore the mean total traffic load (uplink and downlink) per Web Page is 26-32 kBytes.

The Think Time per Web Page is the average time between Web Page requests. However, it is possible to increase the offered load to the network in a controlled fashion by reducing the Think Time[24]. For example, if the Think Time is set to one second then the Offered Load on a single connection between client and server for a single Web Page is 26-32 kBytes/sec (or 208-256 kbits/sec).

---

[23] They are also documented in [21].

[24] This is necessary to compensate for the lower volumes of traffic that characterised the time when the measurements were taken, c.1995, when the mean Think Time was 15 seconds.

The following Think Times generate the following approximate Offered Loads to the Network:

- 1 second Think Time generates ~230kbit/sec (26-32 kBytes/sec),
- 2 second Think Time generates ~115kbit/sec,
- 4 second Think Time generates ~58kbit/sec,
- 8 second Think Time generates ~28kbit/sec,
- 16 second Think Time generates ~14kbit/sec.

In the simulations run for this project, unless otherwise stated, a Think Time of four seconds is chosen and this results in an offered load of *approximately* 52-64 kbit/sec for each source assuming the experiments are run for a sufficiently long duration. ‘Test 4’ on page 45 is used to determine what the duration should be. For further information please consult [21].

Low Priority is assumed to mean priority 1, Access Category 0 for each of the following tests.

Medium Priority Video

In order to produce a medium priority video source a CBR stream with a packet length of 1500 bytes and an inter-departure time of 32ms is used to produce a bitrate of 384 kbit/s, excluding overhead added by the lower layers. This is appropriate to model a low quality video source[25] such as H.323 or MPEG-1. Medium Priority is assumed to mean priority 4, Access Category 2 for each of the following tests.

It is explicitly stated in the test concerned if different values than these are used.

High Priority Voice

A high priority isochronous voice source is generated using a CBR stream with a packet length of 60 bytes and an inter-departure time of 20 ms to produce a bitrate of 24 kbit/s, excluding overhead added by the lower layers. This is appropriate to model ITU-T G.711 speech codec [22], and is similar to the model used by [12]. High Priority is assumed to mean priority 7, Access Category 3 for each of the following tests.

It is explicitly stated in the test concerned if different values to these are used.

---

[25] A high quality video source (e.g. DVD) requires a ~5Mbit/s bitrate and therefore could not be chosen for 802.11.

6.3 Test 1: Demonstration of the basic effects of QoS parameters

6.3.1 Scenario

This test was used as part of the initial basic testing process to verify the correct operation of the new Quality of Service parameters for EDCF with 802.11e as the MAC protocol. These parameters are given in Table C on page 31. Node 0 is the destination/server node. There are eight source/client nodes each generating a single CBR stream of packet length 1000 bytes and inter-departure time of one second. Each source node is assigned one of the eight priorities, i.e., Node 1 = priority 0, Node 2 = priority 1, …, Node 8 = priority 7.

As a means of comparison, the same test is then run again but instead using 802.11 as the MAC protocol.

The simulation time is 10 minutes.

6.3.2 Results

The average delay per data item is calculated in the simulator as a running average over all the items sent from the Application Layer. The delay is the time between the data item being passed as a Protocol Data Unit to the Transport Layer on the transmitting side and being received as a Protocol Data Unit in the Application Layer on the terminating side.

Figure 6.1 shows the results obtained comparing the legacy 802.11 protocol with DCF against the new 802.11e protocol with EDCF.

The average delay is fairly consistent between the competing nodes using legacy 802.11. However, when different priorities are introduced the delay for the lowest priority traffic stream is almost doubled while the delay for the highest priority traffic stream is almost halved.

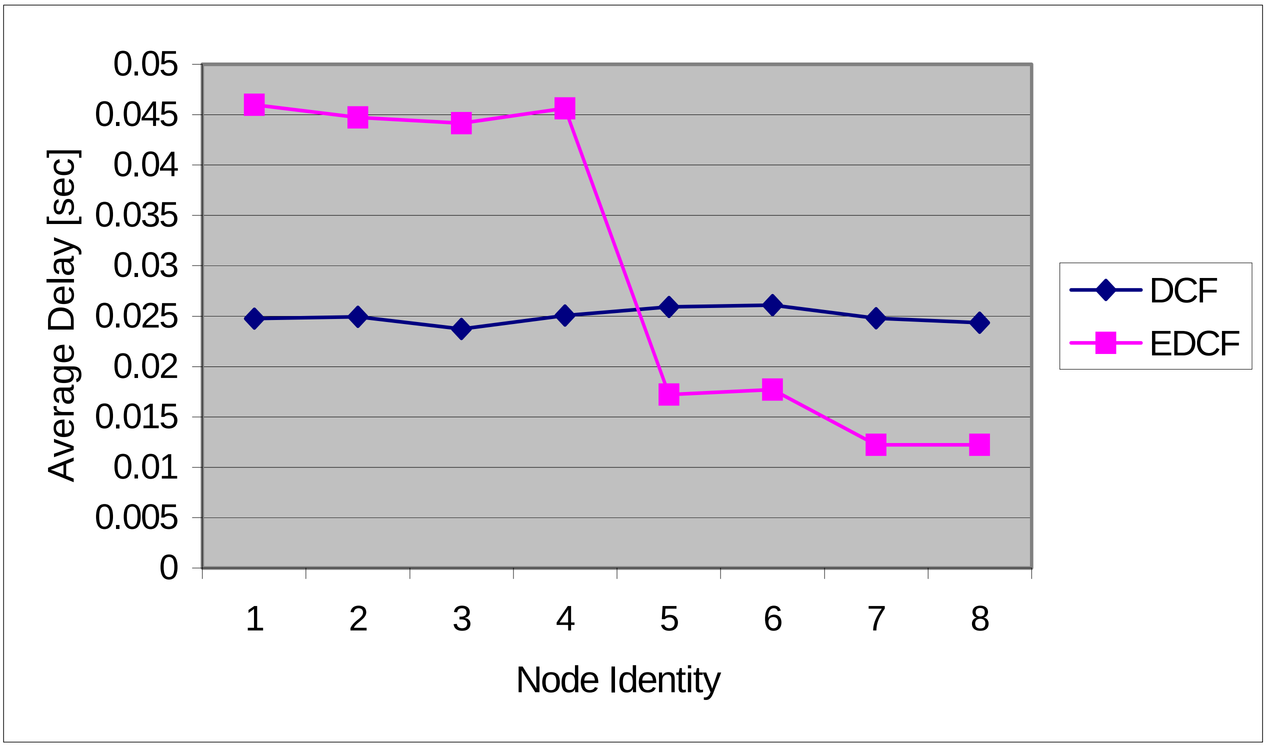

Figure 6.1 Delivery delays in a QBSS with eight client nodes as observed by server node

6.3.3 Discussion

The purpose of this test was simply to observe the effects that different IFS, CW_{min} and CW_{max} values have on the average end-to-end delay observed at the destination node.

By reducing the IFS time and the random backoff times (which are based on Contention Window values) the higher priority traffic experiences lower waiting times in accessing the idle medium. Consequently the low priority traffic suffers by having to wait longer in the queue before being serviced.

It should be noted that this test is somewhat contrived because each of the sending nodes is synchronised. In reality this is unlikely and the possibility could arise where a low priority source could gain access to an idle medium, without delay, during the time when the other nodes have completed their previous transmission and not started their next.

For simplicity CBR traffic sources were chosen for every priority source in this test. This removes any doubt that the difference in delays could have been due to different traffic sources being used. In the remaining tests CBR is only used for higher priority traffic sources that normally display CBR behaviour.

6.4 Test 2: Maximum Possible Link Utilisation for CBR traffic

6.4.1 Scenario

Another basic test of some interest is to deduce the Link Utilisation of the 802.11e MAC. The MAC layer is based on a *Stop and Wait* scheme whereby the sending node transmits a frame and waits for an acknowledgement before sending the next frame.

This scenario consists of two nodes. The simulation is set up with an originating node continuously sending a single CBR source, with an inter-departure time of 0 seconds, to a destination node, i.e., there is no idle (non-sensing) time.

The time to put a packet on the air, t_i , is defined as:

$$t_i = (\text{ packet size in bits / bitrate of the link }) \quad (3)$$

The round trip delay, t_{out} , is the length of time before the transmitter receives the ACK. In the 802.11e MAC, t_{out} is calculated as follows:

$$t_{out} = (\text{ AIFS delay + Random Backoff + RTS frame transmission time + CTS frame transmission time + ACK frame transmission time + All propagation times + transmitter processing + receiver processing }) \quad (4)$$

Fortunately, t_{out} is measured by the simulator.

The minimum time between transfer of successive DATA packets, t_T , is:

$$t_T = t_i + t_{out} \quad (5)$$

The size of the link, a, is the ratio:

$$a = (t_T / t_i) \quad (6)$$

According to [14], the link utilisation (or normalised throughput) is given by:

$$\text{Link utilisation} = (1 - p) / a \quad (7)$$

where p is the probability of bit error.

The following assumptions are also made:

- The calculations are made with a negligible probability of bit error, i.e., p = 0.
- The overhead added by lower layers is small compared to the packet size.

6.4.2 Results

Table D illustrates the maximum link utilisation for high and low priority CBR sources with varying packet sizes of 60, 1500 and 2000 bytes[26]. Although in remaining simulations low priority sources will be HTTP based it is nevertheless interesting here to note the differences in link utilisation values, depending on priority, for sources of the same type.

Priority	Packet Size	t_i	t_T	a	Link Utilisation, 1 / a	% Normalised Throughput
high priority [7]	60 bytes	0.00024	0.001496	6.232513	0.160448936	16.04489361
	1500 bytes	0.006	0.013016	2.169301	0.460978089	46.09780895
	2000 bytes	0.008	0.017016	2.126975	0.470151188	47.01511883
low priority [0]	60 bytes	0.00024	0.001516	6.315846	0.15833192	15.83319204
	1500 bytes	0.006	0.013036	2.172634	0.46027084	46.02708402
	2000 bytes	0.008	0.017036	0.036277	0.469599232	46.95992317

Table D Maximum achievable link utilisation for CBR sources in IEEE 802.11e

---

[26] The maximum allowed packet size in GloMoSim is 2048 bytes.

6.4.3 Discussion

From Table D it is obvious that the link utilisation is very dependent on packet size. Larger packet sizes benefit from higher utilisation because there is more data per transmission instance. However there is a greater risk of bit error which will lead to frame error – although for simplicity in the calculations above, bit error was assumed negligible. The reverse is true for smaller size packets where a low link utilisation and low bit/frame error is achieved. The slightly lower link utilisation for different priority packets of the same size can be attributed to the differing, priority dependent values for AIFS, CW_{max} and CW_{min}.

The figures for link utilisation are also highly dependent on many parameter values that are based on the underlying Physical Layer. The results presented in Table D are based on an 802.11e MAC layer with a legacy IEEE 802.11 2Mbit/s Physical Layer. Different results would be expected for 802.11a or 802.11b Physical Layers.

Using legacy 802.11 the maximum link utilisation expected in the remainder of the tests is approximately 48% or approximately 1 Mbit/s. An offered load to the network beyond approximately 1 Mbit/s should result in a loss in throughput of lower priority sources.

However this is only valid if the traffic sources are HTTP or CBR video where the packets are on average are large. If the network was exclusively fully loaded with small voice packets the link utilisation is only approximately 16%. For legacy 802.11 this suggests that the maximum number of duplex voice calls (2 x 24 kbit/sec) that can be supported from a bitrate perspective is approximately seven.

6.5 Test 3: Verification of the HTTP offered load

6.5.1 Scenario

In chapter 6.2 the HTTP traffic source used in GloMoSim was described. The difficulty with using CDFs of real network traces is that it is not possible to specify in advance the exact arrival times and packet lengths and therefore the bitrate of the offered load.

Before proceeding with the remaining tests it is important to verify, for example, that the values for mean request/reply length and mean number of files per Web Page are generally close to those documented in [21]. Otherwise it would be impossible to know with confidence the offered load[27]. In particular, if a Think Time of ‘x’ seconds is used, what is the average offered load (combined for uplink and downlink) to the network?

It is important to see how long the simulations should run for before the median offered load approaches the mean. It is also interesting to note the ratio between offered load in the uplink and downlink direction. If it can be shown that the downlink traffic represents >95% of the offered load then the uplink traffic can be neglected in the calculation of throughput[28].

Node 0 is again the server node. Ten client nodes each request Web Pages from the server. The uplink direction is from client to server while the downlink direction is from server to client.

6.5.2 Results

Simulation runs with durations of 10min, 30min and 300min were recorded. Think Times of 10s, 4s and 2s were used at each simulation time. In the simulations where Think Time was four seconds or greater, the percentage throughput at each client ranged between 99.66% and 100%. This high value is anticipated since the aggregate load over all client/server links never exceeded 70% of the maximum link rate of 2Mbit/s. However for a Think Time of two seconds, and with ten client nodes, significant throughput degradation was observed as expected because the load on the link exceeded the maximum bitrate of the link.

A selection of results is shown in Figure 6.2. The y-axis measures the offered down link load in bytes across each of the ten client nodes. These ten loads are averaged and the

---

[27] If the offered load is unknown then deciding how many sources are necessary to load the network to a certain percentage is impossible.

[28] This is desirable because it is not possible to automatically calculate both uplink and downlink throughput within the model for a given connection because the statistics are based in different node and layer structures.

results are written in the panel of Figure 6.2, as ‘Av. Load per Node’. The total downlink load on the link is found by dividing this value by the Think Time and the result is written in the panel of Figure 6.2, as ‘Total DL Load on Link’ in kbit/sec.

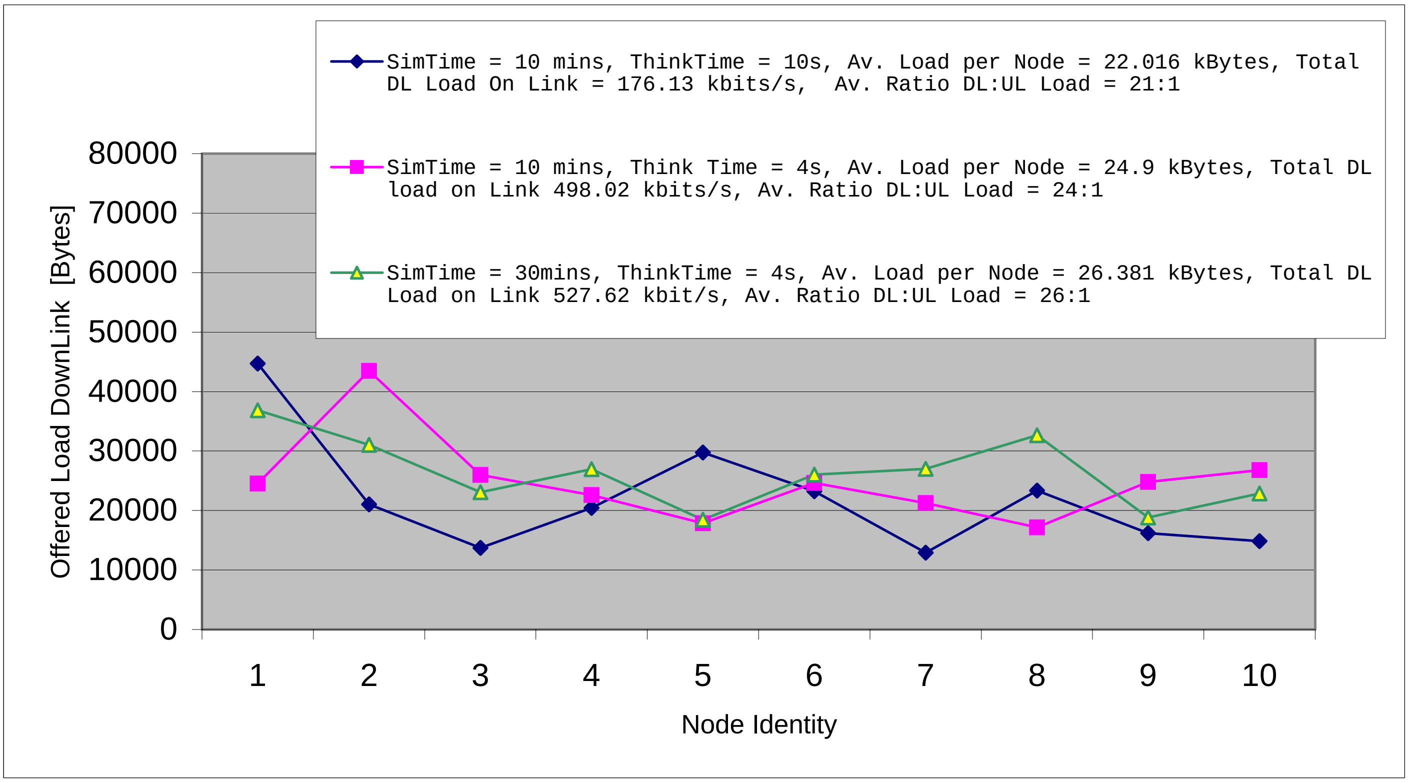

Figure 6.2 Total Downlink Offered Load per Node

6.5.3 Discussion

The estimation of the HTTP based offered load per client/server pair, before the remaining simulations take place, is helpful in determining the number of clients required to produce a particular offered load to the network and in calculating the appropriate traffic mix between CBR and HTTP sources.

The mean offered load per Web page as described in [21] is in the range 26-32 kBytes (and has a median value less than 26kBytes). In Figure 6.2, using a simulation time of 30 min and a Think Time of four seconds the lower bound is approached @ 26.38 kBytes. This produces an average offered load of ~ 53kbits/s towards each client.

Therefore future simulations will run for a minimum of 30 minutes to ensure that the average offered load is known to be reasonably accurate.

The average ratio of downlink load to uplink load is always greater than 24:1 (i.e. > 96%) when using a Think Time of four seconds. Therefore the following assumption is made: *in future simulations it is acceptable to neglect the uplink contribution in the calculation of offered load and throughput.*

6.6 Test 4: Average Delay of Voice and Video Sources as HTTP Sources Increase

6.6.1 Scenario

A QBSS is set up with a single, high priority, voice source in one client node and a single, medium priority, video source in another client node. This test investigates the effect adding an increasing number of bursty, low priority, best effort HTTP sources to the QBSS. Each of these HTTP data sources is located on a separate client node. The Think Time is 8 seconds which means the average offered load per data source is approximately in the range 26-32kbps. The server, Node 0, is the destination for all sources.

In terms of link utilisation, when only one data source is added to the network the proportion of video to voice to data is in the ratio 15:3:1 approximately. Similarly in the maximally loaded case (with as many data sources added as possible), in terms of link utilisation the proportion of video to voice to data is approximately 3:1:3.

The simulation is also run using the legacy 802.11 MAC without a priority mechanism as a means of comparing both.

The simulation time per simulation run is 30 minutes.

6.6.2 Results

The average end-to-end delay of both the voice and video sources is measured at the destination Node 0 as the number of data sources added to the network grows. This is illustrated in Figure 6.3. The graph displays results for both the legacy 802.11 MAC using DCF and the priority enhanced 802.11e MAC using EDCF. The trend observed is that using the DCF function, the average delay for the Video source and particularly for Voice source is significantly higher than when using EDCF. In fact the average delay for both Voice and Video does not exceed 4ms and 10ms respectively using EDCF, even for a maximally loaded network.

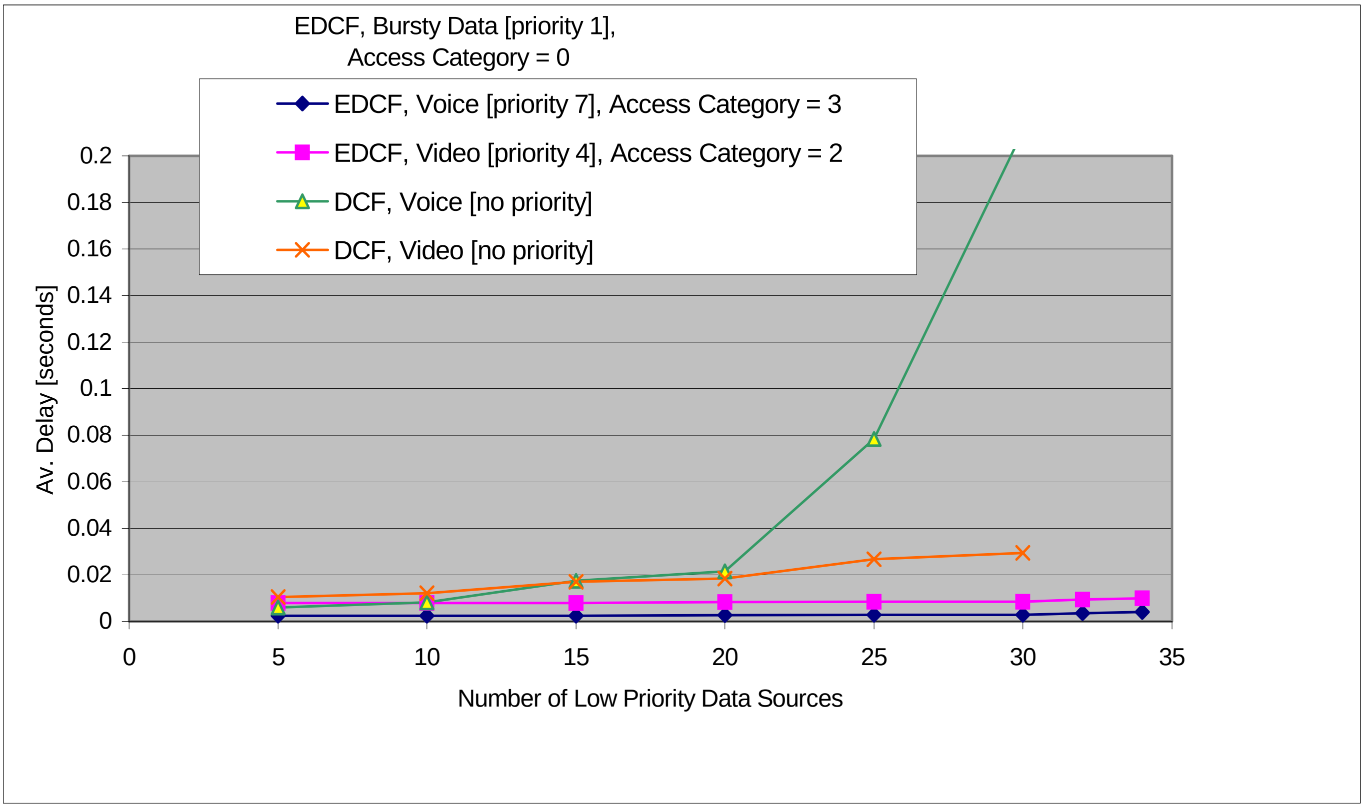

Figure 6.3 Average Delay for one voice source and one video source with increasing numbers of data sources

6.6.3 Discussion

In this configuration the traffic mix, i.e. the ratio voice:video:data, changes as the number of low priority traffic streams increases.

According to [18], the maximum acceptable end-to-end delay for a voice conversation over an IP based network is 150 ms. However since the likelihood is that there will be further hops beyond Node 0 before the voice call terminates a lower acceptable delay is more appropriate. Thus 40 ms is assumed.

Also according to [18], the maximum acceptable end-to-end delay for a one-way video session is < 10ms.

[21] noted that the median Think Time for the HTTP sources was about 15 seconds. The experiments here use a Think Time of 8 seconds to compensate for the increased size of Web Pages in the years since the study was made. This is based on an estimation that the average size of a web page has approximately doubled since the study initiated in [21] was

made (due mainly to the increased number of embedded images per page). Assuming this compensation is reasonable then the approximate number of data sessions that can be supported in the network up to maximum load is approximately 16. Beyond approximately 16 data sessions the downlink throughput measured at each client begins to fall off. Also with out a priority mechanism in place the average delay for DCF voice and video sources increases. In particular the video delay is greater than the 10ms constraint and the voice delay rapidly exceeds the 40 ms constraint. However from Figure 6.3 it is observed that using 802.11e the higher priority voice and video are given primary access to the network ensuring that even in the fully loaded case the average delay always remains acceptably low for Quality of Service to be assured under the constraints of [18].

6.7 Test 5: Delay Variation and Maximum Delay of Voice and Video as HTTP Sources Increase

6.7.1 Scenario

Using an identical scenario to Test 5 the delay variation and maximum delay are measured. The maximum delay is the largest delay experienced across all of the items in a single CBR session. The delay variation of a CBR session is the difference between the largest maximum delay experienced across all items and the smallest minimum delay experienced across all items. According to [18] the delay variation of a voice source ideally should remain close to 1 ms for acceptable perceived end user quality in either a two-way voice conversation or a one-way voice messaging application.

6.7.2 Results

Figure 6.4 on page 49 shows the maximum delay for both voice and video sources using both the legacy DCF function and the EDCF function. The graph illustrates that without a prioritisation mechanism the voice and video sources must contend with the data sources on equal terms to gain a transmission opportunity and suffer large maximum delays because of this. In contrast the prioritisation mechanism of EDCF ensures that the both the voice and video sources have a very low maximum delay.

Figure 6.5 on page 49 shows a similar result for delay variation. Using the DCF function, without a prioritisation mechanism in place the delay variation is never below 50ms for either voice or video sources. Even when the network is less than approximately 75% loaded (with 5 data sources) the delay variation is too large. However with the EDCF function, the delay variation for both voice and video sessions approaches an acceptably low value regardless of the percentage load on the network.

6.7.3 Discussion

This test demonstrates the improvements to Quality of Service, in terms of reduction in delay variation of high and medium priority traffic sources, as a result of the EDCF function. Added to the findings of Test 4 on average end-to-end delay it is concluded that EDCF can provide a complete Quality of Service guarantee to delay sensitive traffic even in a highly loaded network.

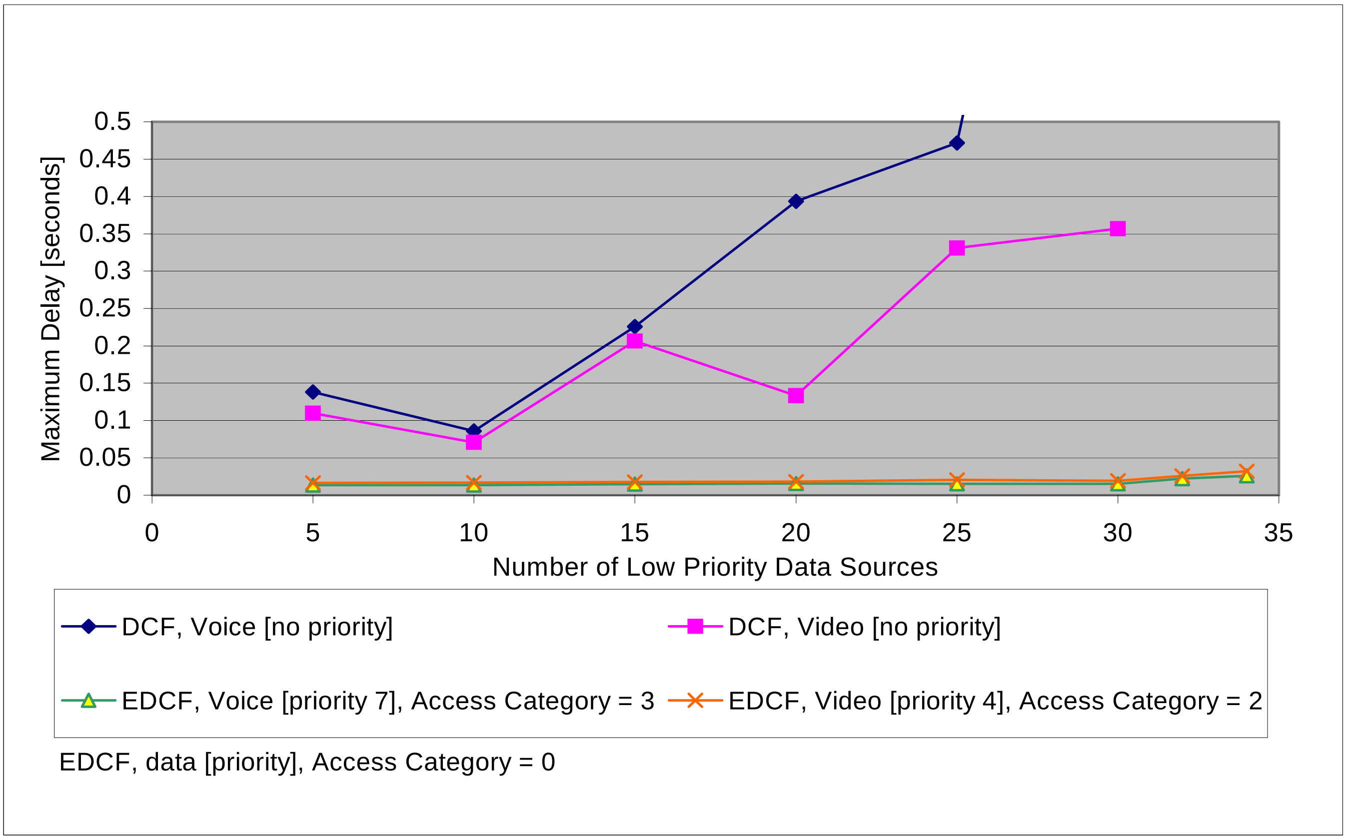

Figure 6.4 Maximum Delay for one Voice Source and one Video Source with Increasing Data Sources

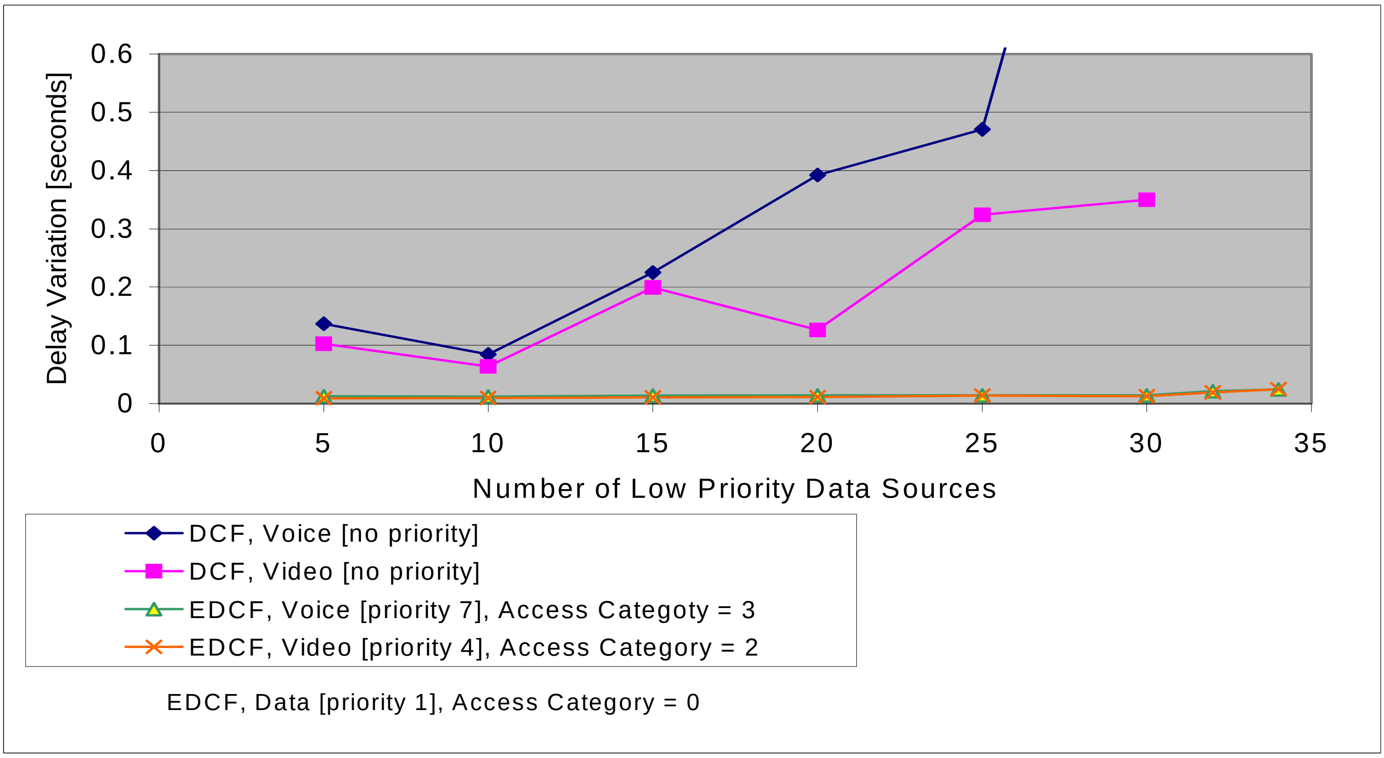

Figure 6.5 Delay Variation for one voice source and one video source with increasing data sources

6.8 Test 6: Estimation of Downlink Throughput of Low Priority HTTP Sources

6.8.1 Scenario

Using an identical scenario as Test 5 and Test 6, the purpose of this test is to measure the throughput and thus percentage packet loss for the low priority data sources in the downlink direction. In a data application, like HTTP, there can be no tolerance for packet loss.
A measurement is also made to deduce if the offered load at the server, towards each client, is maintained at approximately 26 kbit/s as the load on the network increases.

The downlink throughput across all client/server pairs is defined as follows:

$$\begin{array}{l}\text{Offered Load Downlink} \\ \text{(calculated at server)}\end{array} = \begin{array}{l}\text{Total Bytes sent to each} \\ \text{client (downlink)}\end{array} \tag{8}$$

$$\begin{array}{l}\text{Downlink Throughput} \\ \text{(calculated at} \\ \text{each client)}\end{array} = \frac{\begin{array}{l}\text{Total Bytes received} \\ \text{at client(downlink)}\end{array}}{\begin{array}{l}\text{Offered Load Downlink} \\ \text{(calculated at server)}\end{array}} \tag{9}$$

The throughput figure for each client is aggregated manually to form the total throughput for the network and is expressed as a percentage.

6.8.2 Results

Figure 6.6 illustrates the percentage throughput for the low priority data source. There is approximately zero HTTP data loss in the network until roughly 12 low priority data sources are added to the network

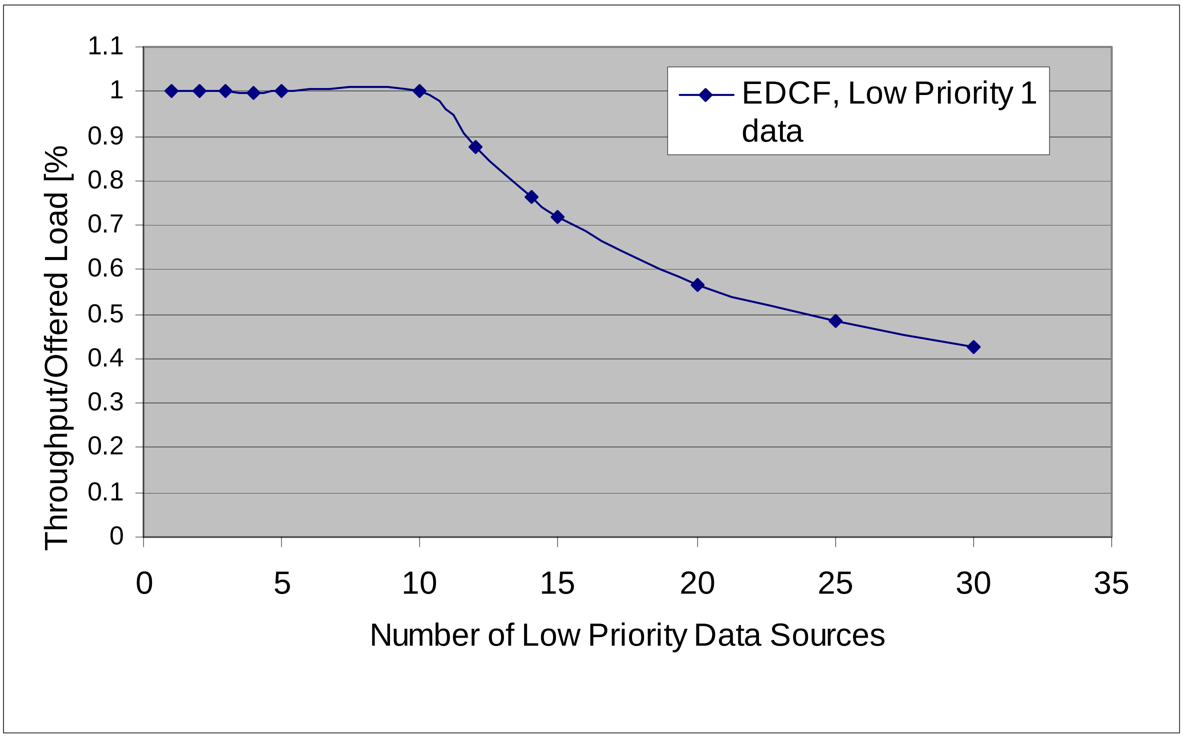

Figure 6.6 Percentage Throughput of Low Priority Data sources

6.8.3 Discussion

An inherent feature of 802.11 and 802.11e is that it is never possible to achieve a full link utilisation. For example, one hundred competing sources each delivering 20kbit/s from the application layer could never achieve 100% throughput on a 2Mbit/s legacy 802.11 radio link. This is because of the packet overhead introduced by the lower layers and most importantly because of the time delays introduced by the *Stop and Wait* scheme and Carrier Sense mechanism (e.g. the random backoff times, the Inter-Frame Space times etc).

Hence, Figure 6.6 demonstrates that the throughput begins to fall significantly below 100% after 12 low priority data sources are added to the network.

Another potential negative phenomenon in a highly loaded network using the HTTP model is that the average downlink HTTP offered load towards each client could decrease as the overall offered load to the network across all traffic types increases. This decrease could occur because of collisions in the uplink direction (causing the packet not to be received) from clients making Web Page requests to the server. These collisions result in fewer responses to Web Page requests being sent by the server to the clients.

6.9 Test 7: Case Study of Residential User *(incorporating a comparison of Draft Standards)*

6.9.1 Scenario

This test considers the typical requirements of a Residential User with a wireless network, i.e. a QBSS, set up in their house. Such a user may simultaneously require some streaming video capabilities (e.g. watching music videos, web based television[29] etc.), multiple low priority data connections (e.g. World Wide Web) and multiple voice connections.

In the following scenario, the QBSS consists of an Access Point designated as Node 0, one client node accepting a medium priority video stream from the access point and five client nodes requesting and receiving low priority Web data from the server. The Think Time for the Web based data is four seconds.

Pairs of voice sources, one uplink and one downlink to provide full duplex communication, are successively added to the network. The average delay of the voice sources in both uplink and downlink directions are recorded to estimate the number of voice calls that the network can support in this configuration.

According to [18] the delay in streaming video applications is of less importance than real time video conferencing applications due to the absence of a conversational element. [18] suggests that the acceptable delay threshold for streaming video is ten seconds. This is because of software available at the receiver that can compensate for delay and delay variation. Therefore a more relevant measurement of performance for the streaming video in this test is throughput.

All prior tests in this chapter have used the February 2003 version of the draft 802.11e standard, [1]. The parameters specific to Quality of Service are described in Table C, on page 31.

However the values of these parameters have changed considerably from previous drafts and remain the subject of interest. The parameters for the October 2002 version of the draft 802.11e standard are listed below in Table E and show that CW_{max} is the latest parameter to change for access categories AC_2 (from 1023 slots to 31 slots) and AC_3 (from 511 slots to 15 slots).

---

[29] With a maximum bitrate for legacy 802.11 of 2Mbit/s the possibility of having a high quality video link e.g. DVD is impossible since this requires about 5Mbit/s.

```
aCWmax = 1023, aCWmin = 31, aSlotTime = 20 µs (based on 801.11)

CW min [ac_0, ac_1, ac_2, ac_3] = [aCWmin, aCWmin, ((aCWmin + 1)/ 2) -1,
((aCWmin + 1)/ 4) -1]
CW max [ac_0, ac_1, ac_2, ac_3] = [aCWmax, aCWmax, aCWmax,
((aCWmax + 1)/ 2) -1 ]
AIFS [ac_0, ac_1, ac_2, ac_3]   = [2, 1, 1, 1]
Note: all values are {n x aSlotTime}
```

Table E IEEE 802.11e MAC parameter settings from previous draft standard, October 2002

Therefore a secondary, but important, part of this test is a comparison of the 802.11e parameters from February 2003 with those of the previous draft issue from October 2002, to see if any gain was achieved from changing these parameters. The gain achieved is in terms of voice delay reduction and video throughput increase.

6.9.2 Results

Figure 6.7[30] shows the average delay for a voice call in both the uplink and downlink direction as the number of voice calls is increased. Applying the constraint from previous tests for maximum tolerated average delay of 40 ms; it can be seen that the maximum number of voice calls that can be supported is different depending on which set of parameters is used for Quality of Service. The parameter values from the October 2002 draft can support only one voice call before both the uplink and downlink delay becomes too large. However the parameter values for the February 2003 draft can support three simultaneous voice calls. After this the downlink delay becomes too large.

Figure 6.8 shows the average throughput of the streaming video session as the number of voice calls is increased towards the maximum load the network can support. The graph shows that using the older parameter values from the October 2002 draft of the standard the throughput of the video traffic stream deteriorates rapidly when more than one duplex voice session is added to the network. However the parameter values for the February 2003 draft maintain an approximately zero packet loss up to four duplex calls being added to the network.

---

[30] Note that with one exception (voice calls = 1) there is more than one voice source in the uplink direction and in the downlink direction. Therefore the largest of the Average delay figures for each direction is placed in the graph.

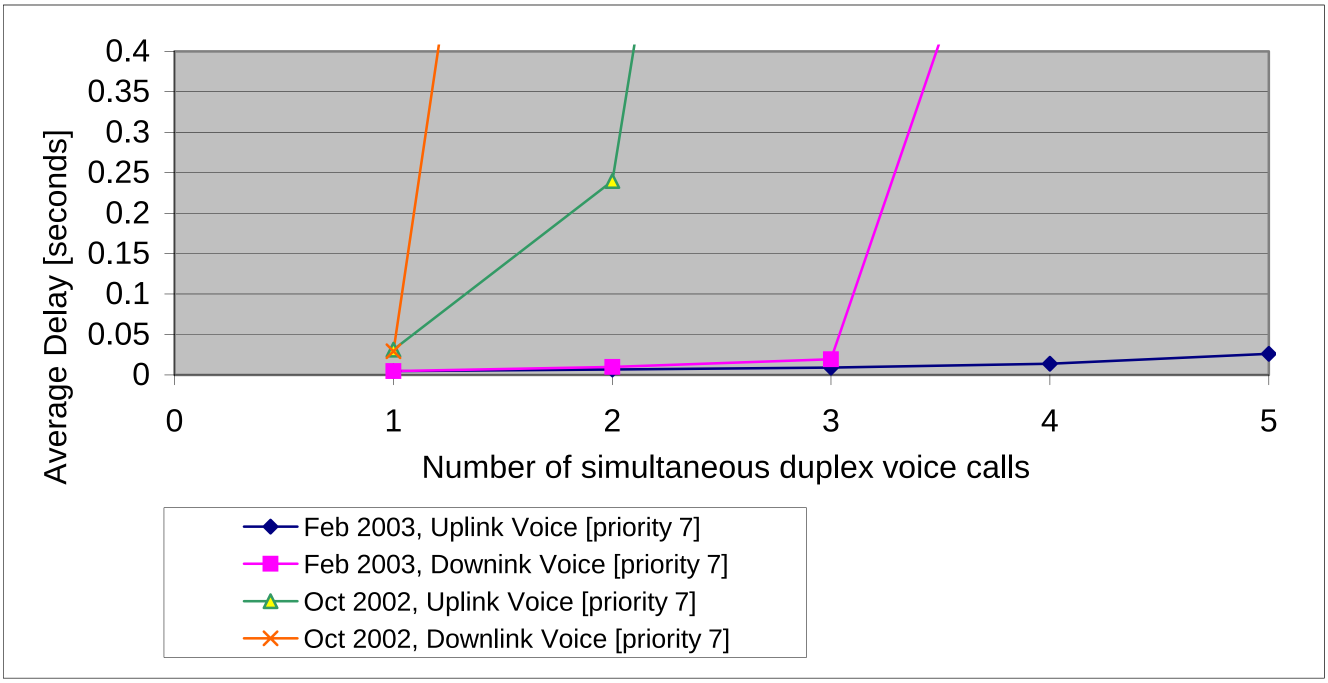

Figure 6.7 Average Delay of simultaneous duplex voice calls in a Residential QBSS

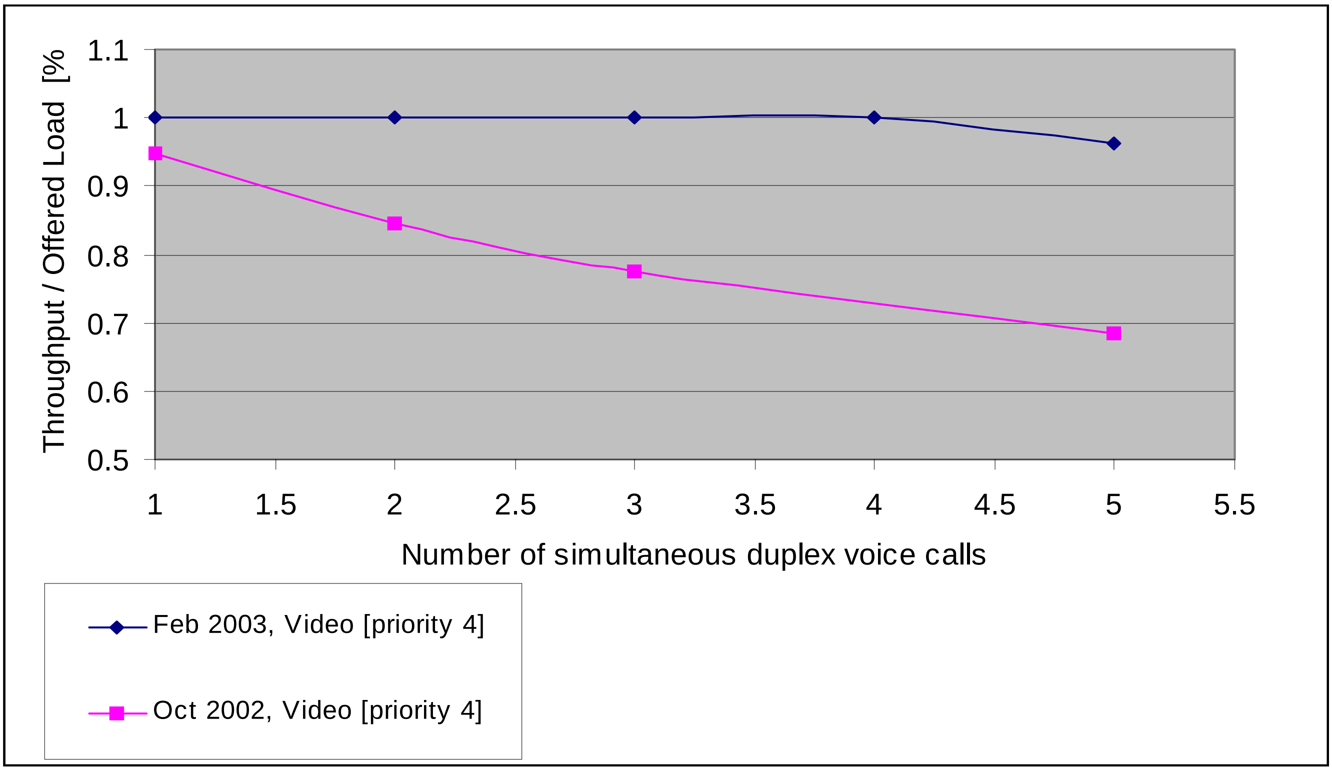

Figure 6.8 Average throughput of medium priority streaming video sessions in a Residential QBSS

6.9.3 Discussion

The excessive load on the AP causes the average delay in the downlink direction to rise substantially faster than in the uplink direction.

The result in Figure 6.7 is an acceptable configuration for a residential QBSS. More voice connections could be supported by removing the medium priority video or by offering just one low priority data connection. However three simultaneous voice calls seems sufficient in most residential scenarios as is given by the February 2003 simulation.

Unfortunately a successful outcome to this Test cannot be confirmed without checking the throughput values for lower priority traffic sources.

Figure 6.8 shows the throughput for the medium priority video traffic. As can be seen, beyond three voice calls in the February 2003 case and one voice call in the October 2002 case the throughput of the medium priority video session begins to fall off. This is because the network has gone beyond its full capacity. As was observed in Test 2 on page 40 the link utilisation for networks with *Stop and Wait* schemes rarely go above approximately 50%.

On the subject of comparison between the two versions of the draft standard it is obvious that the most recent draft from February 2003 is an improvement on its predecessor. By reducing the CW_{max} value substantially for AC_3 and AC_4 the random backoff times for the highest priority traffic streams are potentially reduced. This is because in the event of retransmissions the backoff window can only increase to a maximum of 31 slots instead of 1023 slots for AC_2 and to a maximum of 15 slots instead of 1023 slots for AC_3. This has the effect of allowing more voice traffic streams to be set up, see Figure 6.7, below the acceptable delay threshold and also maintaining a higher throughput of video traffic, see Figure 6.8, as the load on the network increases. However the trade-off comes in the form of increased delay of the low priority data sources (not a big problem) and reduced throughput values (a big problem).

An important issue not illustrated in the graphs is the loss in throughput of the low priority data sources as the load on the network exceeds the maximum allowable. The reduction in throughput is greater than for the medium priority video shown in Figure 6.8. This is because the data traffic has the lowest priority of all traffic sources.

6.10 Test 8: Case Study of Business User

6.10.1 Scenario

In contrast to the previous test, a QBSS comprising of business users, perhaps operating in an airport or hotel, will have different requirements. A larger number of voice connections should be offered than in the residential users case and also a modest number of data sessions should be supported. In a moderately highly loaded legacy network, with many users, video connections will consume too much of the bit rate and should not be tolerated.

The scenario for this test comprises of a QBSS with a single AP located at the server Node-0. Ten QSTAs are configured as clients each requesting low priority HTTP sessions with the server. An average offered load of 13kbps per session is achieved by setting the Think Time to 16 seconds. Additionally QSTAs are gradually added to the network each supporting two CBR high priority voice sessions (one each for the uplink and downlink) towards the server Node.

6.10.2 Results

Figure 6.9 illustrates the average end-to-end delay[31] of the duplex, high priority voice calls as observed at the terminating server Node. The delay remains acceptably low until five duplex calls are added to the network. Beyond this the downlink average delay becomes too large. The uplink average delay becomes too large when more than seven duplex calls are added to the network. A possible reason why the downlink delay rises before the uplink delay is because the server Node is dealing with the downlink traffic of all nodes - unlike the client node, which only has one session to deal with.

---

[31] Note that with one exception (Voice Calls = 1) there is more than one voice source in the uplink direction and in the downlink direction. Therefore the largest of the Average delay figures for each direction is placed in the graph.

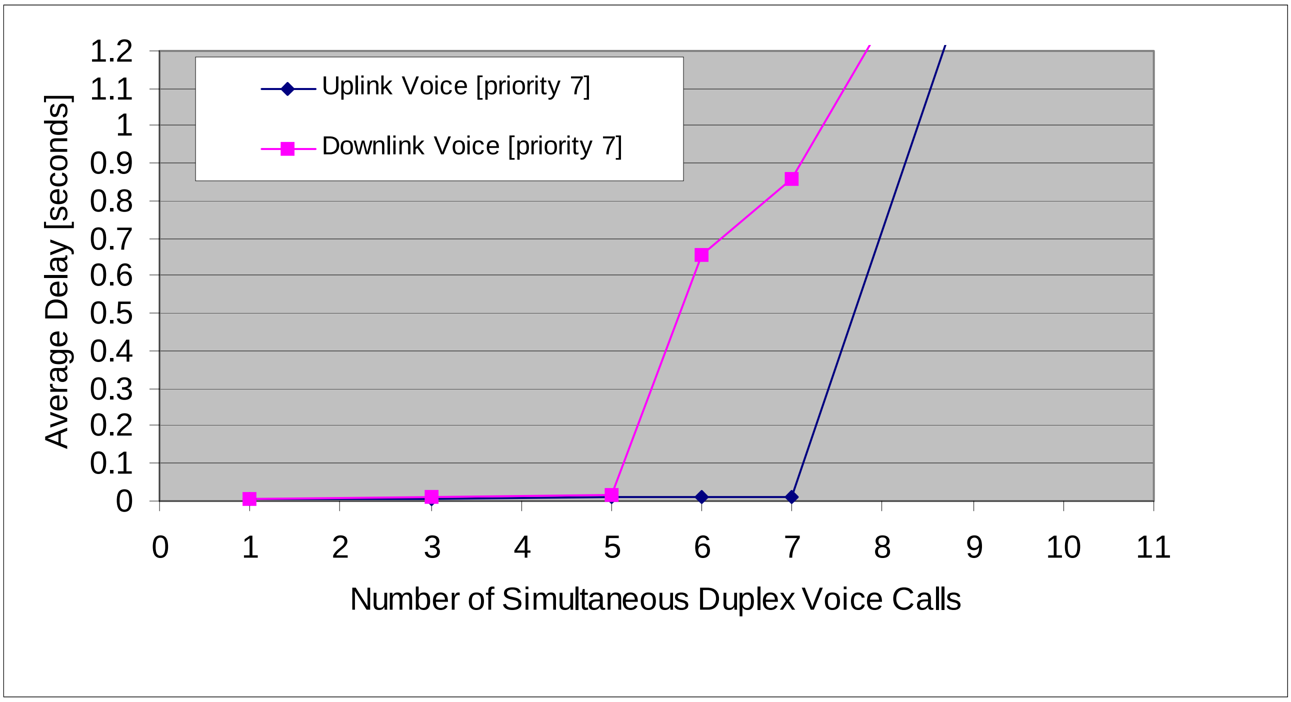

Figure 6.9 Average Delay of Simultaneous Duplex Voice Calls in an Urban QBSS

6.10.3 Discussion

It may seem unusual that only 10 data sources each of approximately 13kbit/s and 5 duplex voice calls each of (24 x 2) kbit/s is enough to fully load the network. This only adds up to approximately 370kbit/s yet the maximum bitrate of the link is 2Mbit/s. However this can be explained partly from the results of Test 2 on page 40. Test 2 demonstrated that the link utilisation for a high priority CBR voice source, of packet size 60 bytes, is approximately 16%. Small packet sizes, such as those used to model a voice call, will result in an inefficient network utilisation. Consideration should be given to upgrading to a 54Mbit/s 802.11a network if support for a large number of voice calls is a requirement.

6.11 Summary

This chapter presents 8 Test Simulations that are used to judge the effectiveness of introducing Quality of Service functionality to a wireless local network.

Test 1, Test 2 and Test 3 form a set of fundamental tests that are used to ascertain that the basic functions of the simulator are behaving as expected. Test 1 demonstrated that a set of CBR traffic streams using an 802.11e MAC layer exhibit an average delay that is dependent on the assigned priority of the traffic stream. In contrast CBR traffic streams in a legacy 802.11 network that does not assign priorities have a roughly similar average delay. Test 2 confirmed that in a *Stop and Wait* scheme such as the 802.11 MAC the link utilisation is heavily dependent on the packet size. Furthermore using the largest permitted packet size (~2000 bytes), where the size of the link, a ~ 2, the maximum link utilisation is less than 50%. Test 3 studied the performance of the HTTP source model used in GloMoSim; in particular whether the CDF values for various parameters agreed with the mean values reported in [21]. If run for a sufficiently long simulation time the HTTP offered load approaches the mean values reported in [21].

Test 4, Test 5 and Test 6 use an identical scenario of a wireless network that supports a single CBR high priority voice session and a single CBR, medium priority, video session. To this QBSS an increasing number of low priority HTTP data sessions are added until the network becomes fully loaded. Test 4 demonstrates that the average delay is maintained below acceptable levels for high priority voice sources using EDCF but not for sources using DCF. Test 5 demonstrated that the delay variation was unacceptably high for voice sources with no priority using the DCF function. However, using EDCF, the delay variation is sufficiently low for high priority voice sources. Test 6 shows that the throughput of HTTP sessions deteriorates as the network becomes fully loaded because they have the lowest priority. This is the trade-off for having low delay high priority traffic sources present in the network.

Test 7 and Test 8 choose two common situations where an 802.11 network might be used; a residential QBSS based in a users home and a QBSS in a busy public environment such as a hotel or airport. In both tests the network is suitably configured with low priority HTTP data sessions and in Test 7 a medium priority video session is also added. Both tests then evaluate how many simultaneous high priority duplex voice calls can be supported before the average delay becomes too large. Test 7 also compares two issues of the draft 802.11e standard and concludes that the default parameter values in the latest of the drafts offers a superior performance in terms of reducing voice delay and video throughput.

Chapter 7 - Conclusions and Further Research

7.1 Introduction

This is the final chapter of this document. Subchapter 7.2 documents the conclusions that can be drawn from the project. Subchapter 7.3 describes possible implementation applications, improvements and future directions for this work.

7.2 Conclusions

It is evident from the Tests performed in the previous chapter that the introduction of a prioritisation mechanism into a Wireless Local Area Network does have an influence the performance of that network. The behaviour of different traffic classes in terms of delay, delay variation, throughput and packet loss can be influenced strongly by applying certain priorities to them. However this prioritisation process is only effective if used selectively. Assigning everything with a high priority, for example, will achieve nothing. Furthermore, the ratio of traffic sources with the most stringent real-time critical requirement to those with lesser such requirements should be small.

The choice of eight priorities is a result of inheriting the priorities of the 802.1D standard for MAC bridging. However such a large number of priorities may seem unnecessary at the moment since the number of different application types loosely belong to voice, video or data. For instance there is no difference between two priorities in the same Access Category as far as Carrier Sensing and Backoff times are concerned. However the higher of two priorities within the same Access Category will have all its queued frames serviced before the lower priority traffic gets an opportunity. This may be necessary to distinguish between Network Control frames and Voice frames in the highest Access Category for example. The number of priorities is not as important as the number of Access Categories; in this respect four is a sufficiently low number to be able to make clear distinctions in service.

With the success of the EDCF implementation it remains to be seen how much value could be added with a HCF function. If there is only a minimal improvement in terms of delay and throughput then the extra complexity in terms of new frame types, additional control signalling etc. may not be worthwhile. The complexity of design would extend into the management of a network with many more tuneable parameters. Furthermore if these parameters are not self-regulating then the network will be insensitive to changes in levels of usage. An example was presented, in [5], of how Quality of Service could not be met when two HCs from overlapping QBSSs poll at the same time. The HCF function requires all stations to follow the co-ordination of the HC; so this worst case scenario where the polling frames collide and the throughput drops down to zero does not seem negligible. This situation could never occur with EDCF.

One of the simulations in this project demonstrated that altering a single EDCF Quality of Service parameter had a significant effect on the performance of the system. It seems possible that further EDCF parameter optimisation could show similar gains in Quality of Service performance. The underlying principle is that to make a traffic source high priority is to ensure that it experiences the shortest delay in accessing the network. Therefore there still seems to be scope for reducing the waiting times of high priority traffic in the latest draft of the standard. An obvious improvement is to make CW_{min} as small as possible for the highest Access Category. There is no apparent reason why it could not be set to zero for instance.

The maximum bitrate of the legacy IEEE 802.11 standard that was used during this project is approximately 2Mbps. This proved to be very limiting during the simulations as it allowed far less scope in simulating high bitrate, multimedia type, traffic sources. As was demonstrated in the Tests of the previous chapter, a Stop and Wait scheme like IEEE 802.11 can at best offer a link utilisation approaching 50%. The situation worsens when packet sizes (like voice packets) are small - link utilisation drops to less than 20%. In wireless networks Quality of Service demanding high priority traffic sources (that have a poor link utilisation) are increasing in number. There is a compelling reason to upgrade the simulator in any future work to match this trend.

Finally some comments on the GloMoSim simulator. The quality of the coding in the simulator is very good. Although a lack of commenting is prevalent in many places this is compensated for with highly intuitive, repetitive naming conventions for data and functions. The structure of the libraries is also very clear and the different protocols that can be used at different layers are well partitioned (no sharing of functions) thus removing ambiguity. Redundancy promoting readability has been favoured over extremely compact code. Unfortunately there is very little supporting documentation for GloMoSim so a reasonable length of time is initially required by designers to familiarise themselves with the architecture of the simulator. Although there is a Visualisation Tool available it was found to have no benefit during this project[32]. There are some shortcomings in the simulator. A simple unidirectional traffic source at the application layer that generates Poisson arrivals but still has the possibility to roughly predetermine the offered bitrate of the source would be useful in this project. Although the simulator has the ability to handle many nodes simultaneously it does not cope as easily when multiple traffic sources originate from the same node (increasing the queue size substantially at the network layer alleviates this problem slightly). Also it is observed that some statistics can potentially overflow for very long simulation times.

7.3 Further Research

The following list addresses areas of further research and specifically how these studies can be applied to the GloMoSim simulator:

1. Upgrade to 802.11a. This would require modification of the GloMoSim Channel Layer, Radio Layer and MAC layer. It is anticipated that upgrading to 802.11b would be less effort since it operates in the same frequency band as legacy 802.11.
2. Implementation of a Point Co-ordination Function and then upgrade of this to a Hybrid Co-ordination Function. This implies the support of an Access Point in the design, i.e., a node must be nominated as the AP in the CONFIG.IN file and this node should support Access Point functionality (such as the recognition of new frame types and formats) as well as Point Co-ordination functionality.
3. Further investigation of parameter optimisation. Further deviation from the default values that are presented in the MIB of [1] would be interesting to see whether the performance of the system could be improved, in terms of packet delay & throughput etc.
4. Implement EDCF-TXOP bursting. This is where a node, upon capturing the channel through EDCF, may be given the opportunity to transmit multiple MSDUs subject to a time limit, called *TXOPlimit*. This time limit is dependent on the Access Category with higher priority traffic allowed to burst for a longer duration. While this is advantageous in terms of increased throughput for the higher rated Access Categories it adds further delay to low priority traffic. The current version of the simulator only allows the transmission of a single MSDU before the EDCF process begins again.

---

[32] Many of the options that are promised for GUI based design in GloMoSim seem to have been held back for the QualNet product.

5. A potential weakness in the Tests performed in this project is the assumption that the HTTP traffic volumes of 1995 are still valid today. Therefore the Cumulative Distribution Functions that describe the nature of the HTTP traffic could be updated. This is a superior solution to adjusting the Think Time as was done in this project.
6. Debug the fragmentation code in the simulator. The code to implement fragmentation and defragmentation in 802.11 has been ‘commented out’ by the original designer and therefore is not used. Fragmentation is the process of breaking up an MSDU into smaller MSDUs and is used to increase the reliability of transmission in scenarios where the radio environment cannot guarantee the reliable transmission of large MSDUs. Refer to [2], chapter 9.1.4, for a more detailed overview of fragmentation.
7. This project is empirical in nature, i.e., it is based on experiment and not on theory. Generation of a mathematical model to support the simulation would give additional confidence in the results obtained from the simulator.
8. There are additional features already designed within GloMoSim that were not considered during this project, for example, enabling mobility of the nodes, multiple hopping between source and destination nodes using different routing algorithms etc. The next step in Quality of Service studies could include simulations where these features are activated.
9. Investigation of the effects on Quality of Service when two adjacent QBSSs are overlapping. This is of particular interest when a HCF function has been implemented.

7.4 Summary

This chapter ends the work of this project by adding some final conclusions to accompany the discussions in Chapter 6. Suggestions are made for improvements to the GloMoSim simulator and also for further research in this area.

References

[1] IEEE Std 802.11WG, Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/D4.2, February 2003.

[2] Information technology – Telecommunications and information exchange between systems – Local and Metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, International Standard ISO/IEC 8802-11: 1999(E), ANSI/IEEE Std 802.11, 1999 Edition.

[3] A. Grilo and M. Nunes, "Performance Evaluation of IEEE 802.11e", http://citeseer.nj.nec.com/552673.html, 2002, (June 2003).

[4] S. Mangold, S. Choi, P. May, O. Klein, G. Hertz and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service", in Proc. European Wireless '02, Florence, Italy, February2002.

[5] S.Mangold, "IEEE 802.11e: Coexistence of Overlapping Basic Service Sets", Proc. of the Mobile Venue'02, pp. 131135, Athens, Greece, May 2002.

[6] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks", Proceedings of the 12th Workshop on Parallel and Distributed Simulations – PADS '98, May 26-29, 1998.

[7] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin and H. Song, "Parsec: A Parallel Simulation Environment for Complex Systems", Computer, Vol. 31(10), October 1998, pp. 77-85.

[8] T. Bostrom, T. Goldbeck-Lowe and R. Keller, "Ericsson Mobile Operator WLAN solution".

[9] J. Zyren and A. Petrick, "IEEE 802.11 Tutorial", http://easy.intranet.gr/paper_6.pdf, April 2003.

[10] R. Bagrodia and R. Meyer, "PARSEC User Manual, For Parsec Release 1.1", http://pcl.cs.ucla.edu/projects/parsec/manual, September 1999.

[11] "GloMoSim User Manual, version 1.2", http://pcl.cs.ucla.edu/projects/glomosim/GloMoSimManual.html, April 2003.

[12] H. L. Truong and G. Vannuccini, “The IEEE 802.11e MAC for Quality of Service in Wireless LANs”, International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet SSGRR 2003w L'Aquila, Italy, January 6-12, 2003.

[13] M.Collier, “Broadband Networks”, Course Notes for EE552, MEng in Electronic Systems, Dublin City University, 2000.

[14] J. Murphy, “Data Networks”, Course Notes for EE545, MEng in Electronic Systems, Dublin City University, 1999.

[15] E. Royer and C. Toh, “ A Review of Current Routing Protocols for Mobile Wireless Ad-Hoc Networks”, MOBILCOM 1999.

[16] T. Lohmar and R. Keller, “QoS Interworking between WLAN and UMTS Core Network”, In Workshop on IP Quality of Service in Wireless and Mobile Networks (IQWIM99), 1999.
Available at http://www.comcar.de/IQWiM99

[17] S. Mangold, S. Choi, P. May, O. Klein and G. Hertz, “IEEE 802.11E - Fair Resource Sharing between Overlapping Basic Service Sets”, *Proceedings* of the PIMRC 2002, Published: Lisbon, Portugal (September/2002).

[18] P. Coverdale, “Itu-T Study Group 12: Multimedia QoS requirements from a user perspective”, *Workshop on QoS and user perceived transmission quality in evolving networks*, Oct 2001, http://www.itu.int/itudoc/itu-t/workshop/qos/s2p1.html, (August 2003).

[19] http://www.freebsd.org, August 2003.

[20] Peter B. Danzig, Sugih Jamin, “tcplib: A Library of Internetwork Traffic Characteristics”, http://citeseer.nj.nec.com/danzig91tcplib.html, 1991.

[21] B. Mah, “An Empirical Model of HTTP Network Traffic“, *Proceedings of INFOCOM’97*, April 7-11, 1997 in Kobe, Japan.

[22] ITU-T Recommendations G.711 (11/88): “Pulse Code Modulation (PCM) of voice frequencies”.

Appendix A

Resource requirements

This is an entirely software based simulation project. Thus there are no hardware requirements such as laboratory equipment. The simulations are run on a TOSHIBA Tecra 8000 laptop with Intel Pentium II processor and with Windows NT4.0 operating system.

The software requirements are as follows:
Microsoft Visual C++, version 6.0 [33].
Parsec Compiler, version 1.1
This is available for download from the PCL (Parallel Computing Laboratory) website at UCLA:
http://pcl.cs.ucla.edu/projects/parsec
GloMoSim simulation software, version 2.03
This software is also available at the PCL website: http://pcl.cs.ucla.edu/projects/glomosim

Java Development Kit [34], version 1.2.2.
This software is also available at the Sun Microsystems website: http://www.java.sun.com

Documentation Resources

The final draft of the IEEE 802.11e standard has not yet been published. However the latest draft standard, [2], is available at http://www.ieee.org, and then following the link, which is password protected, to the 802.11e Task Group.

Documentation about the GloMoSim simulation package is available at, http://pcl.cs.ucla.edu/papers

---

[33] A ‘C programming language’ compiler and libraries are a pre-requisite for compiling PARSEC programs.
[34] The JDK 1.2.2 is needed to run the GloMoSim VT (Visualisation Tool).

Appendix B

Figure 7.1 IEEE 802.11 MAC STATE MACHINE, Transmitting Side

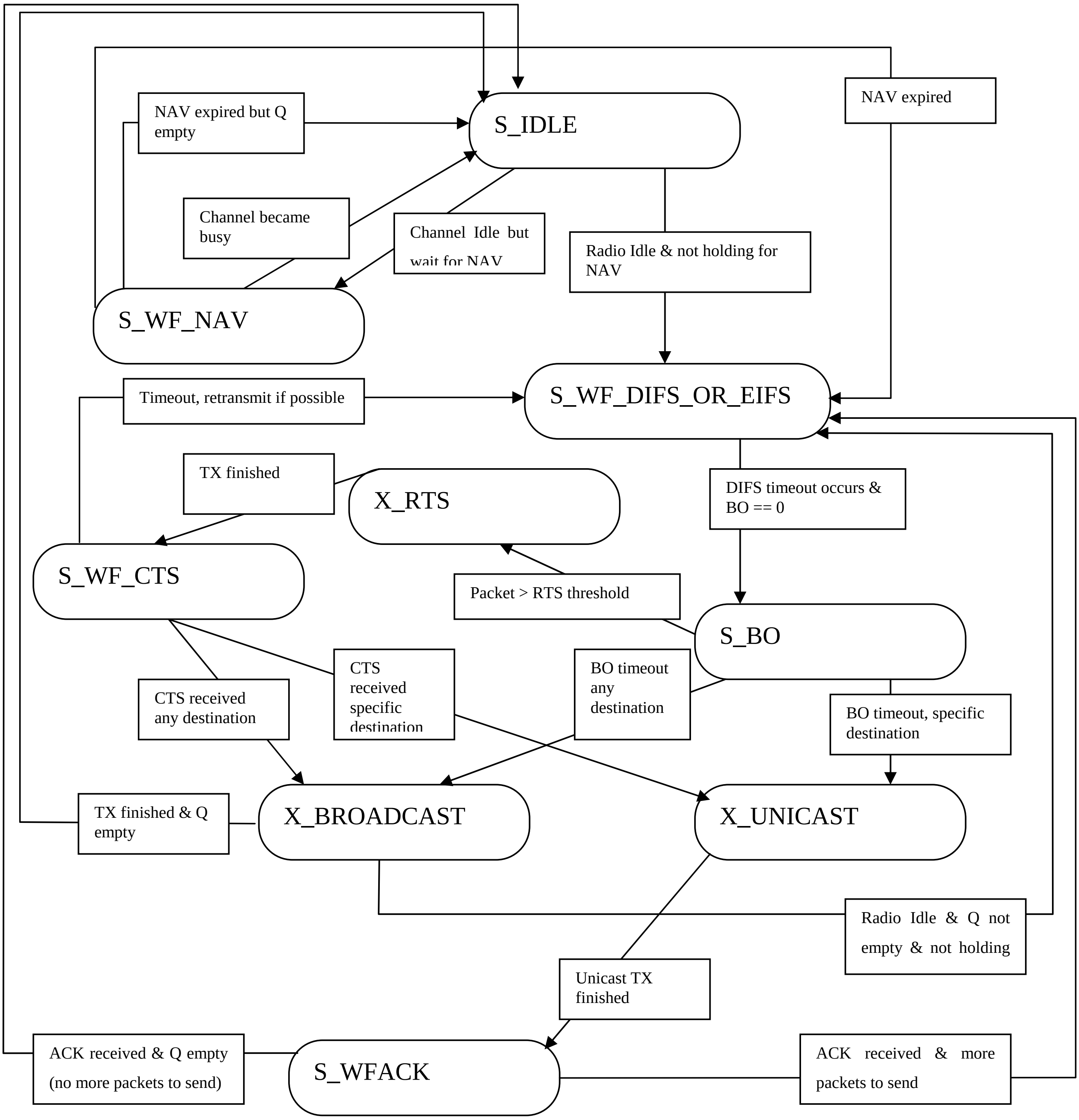

Figure 7.2 IEEE 802.11 MAC STATE MACHINE, Receiving Side