Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

A STUDY OF QOS SUPPORT FOR REAL TIME MULTIMEDIA COMMUNICATION OVER IEEE802.11 WLAN

A thesis presented in partial fulfillment of the

requirements for the degree of

Master of Engineering

In

Computer Systems Engineering

Massey University, Albany

New Zealand

Kun Qian

2005

Abstract

Quality of Service (QoS) is becoming a key problem for Real Time (RT) traffic transmitted over Wireless Local Area Network (WLAN). In this project the recent proposals for enhanced QoS performance for RT multimedia is evaluated and analyzed. Two simulation models for EDCF and HCF protocols are explored using OPNET and NS-2 simulation packages respectively. From the results of the simulation, we have studied the limitations of the 802.11e standard for RT multimedia communication and analysed the reasons of the limitations happened and proposed the solutions for improvement.

Since RT multimedia communication encompasses time-sensitive traffic, the measure of quality of service generally is minimal delay (latency) and delay variation (jitter). 802.11 WLAN standard focuses on the PHY layer and the MAC layer. The transmitted data rate on PHY layer are increased on standards 802.11b, a, g, j, n by different code mapping technologies while 802.11e is developed specially for the QoS performance of RT-traffics at the MAC layer.

Enhancing the MAC layer protocols are the significant topic for guaranteeing the QoS performance of RT-traffics. The original MAC protocols of 802.11 are DCF (Distributed Coordination Function) and PCF (Point Coordinator Function). They cannot achieve the required QoS performance for the RT-traffic transmission. IEEE802.11e draft has developed EDCF and HCF instead. Simulation results of EDCF and HCF models that we explored by OPNET and NS-2, show that minimal latency and jitter can be achieved. However, the limitations of EDCF and HCF are identified from the simulation results. EDCF is not stable under the high network loading. The channel utilization is low by both protocols. Furthermore, the fairness

index is very poor by the HCF. It means the low priority traffic should starve in the WLAN network. All these limitations are due to the priority mechanism of the protocols.

We propose a future work to develop dynamic self-adaptive 802.11e protocol as practical research directions. Because of the uncertainly in the EDCF in the heavy loading, we can add some parameters to the traffic loading and channel condition efficiently. We provide indications for adding some parameters to increase the EDCF performance and channel utilization. Because all the limitations are due to the priority mechanism, the other direction is doing away with the priority rule for reasonable bandwidth allocation.

We have established that the channel utilization can be increased and collision time can be reduced for RT-traffics over the EDCF protocol. These parameters can include loading rate, collision rate and total throughput saturation. Further simulation should look for optimum values for the parameters. Because of the huge polling-induced overheads, HCF has the unsatisfied tradeoff. This leads to poor fairness and poor throughput. By developing enhanced HCF it may be possible to enhance the RI polling interval and TXOP allocation mechanism to get better fairness index and channel utilization. From the simulation, we noticed that the traffics deployment could affect the total QoS performance, an indication to explore whether the classification of traffics deployments to different categories is a good idea. With different load-based traffic categories, QoS may be enhanced by appropriate bandwidth allocation Strategy.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Dr. Mohammad A. Rashid whose constant encouragement, insightful comments, and valuable suggestions helped me in all the time of research for and writing of this thesis. I really enjoyed working with him.

I also would like to thank Dr. Tom Moir, senior lecturer of Computer Systems Engineering at Massey University, Albany campus for providing the knowledge of the signal processing and constant encouragements during the research work.

I am also grateful to some people who provided me with the facilities for my project. They are: Mr. Shamir Bishay, Ms. Freda Anderson, and Ms. Sonya Eastmond. They always helped me and gave me conveniences for my project.

I would have never finished this thesis without the encouragement and help of many relatives, friends and colleagues to whom I am very much indebted. I must thank my parents since they always have loved me, believed in me, and encouraged me in my study. Especially, I would like to give my special thanks to my wife Teresa whose support; understanding and encouragements enabled me to complete this work.

Table of contents

Page

Abstract i
Acknowledgements iii
Table of contents iv
List of figures vii
List of tables ix
Glossary x

CHAPTER 1 INTRODUCTION 1
1.1 backgrounds 1
1.2 Scope and objectives of the thesis 1
1.3 Outline of the thesis 4

CHAPTER 2 AN OVERVIEW OF WLAN STANDARDS 6
2.1 Topologies and Operating modes of WLAN 6
2.1.1 Topologies 6
2.1.2 IEEE802.11 operations 8
2.2 WLAN standards 9
2.2.1 WiFi (IEEE802.11b) 10
2.2.2 Bluetooth 11
2.2.3 HomeRF 12
2.2.4 Comparison of WiFi, Bluetooth and HomeRF 12
2.3 IEEE802.16 and WiMAX 12
2.3.1 IEEE802.16 12
2.3.2 WiMAX Forum 15
2.4 The future of wireless LAN 15
2.4.1 UWB 15
2.4.2 FSO 17
2.4.3 100Mbps WLANs 18
Summary 18

CHAPTER 3 NETWORK QUALITY OF SERVICE 19
3.1 Categorizing traffic 19
3.2 The basic concept of QoS 21
3.2.1 QoS overview 22
3.2.2 QoS versus Bandwidth 22
3.3 QoS parameters in the network 23
3.3.1 Network availability 23
3.3.2 Bandwidth 24
3.3.3 Delay 24
3.3.4 Jitter 25
3.3.5 Packet Loss 27

Summary 27

CHAPTER 4 QOS OF 802.11 WLAN 28
4.1 Medium access Mechanism 28
4.1.1 CSMA/CD & CSMA/CA 29
4.1.2 DCF 31
4.1.3 PCF 33
4.2 MAC Layer operations 36
4.2.1 RTS/CTS 36
4.2.2 802.11 frame fragmentation 37
4.2.3 Station connectivity 37
4.3 802.11 MAC Frame Format 38
4.3.1 General MAC frame format 39
4.3.2 802.11 Control frame 40
4.4 QoS limitations of 802.11 WLAN 42
Summary 43

CHAPTER 5 ENHANCEMENT OF QOS OVER 802.11 WLAN 45
5.1 challenges for RT-traffic QoS in 802.11 Network 45
5.2 DCF-based QoS enhancement schemes 46
5.2.1 DFS schemes 46
5.2.2 DENG schemes 50
5.2.3 IACC schemes 51
5.2.4 Blackburst schemes 51
5.2.5 VMAC schemes 53
5.3 PCF-base enhancement schemes 54
5.3.1 Super-poll scheme 54
5.3.2 other enhance PCF schemes 57
5.4 Up Coming IEEE 802.11e 60
5.4.1 EDCF 61
5.4.2 HCF 66
Summary 70

CHAPTER 6 SIMULATION WITH OPNET AND NS-2 FOR PERFORMANCE STUDY 72
6.1 Simulation with OPNET 73
6.1.1 Performance of RT-traffic over DCF 75
6.1.1.1 Performance of RT-traffic over DCF base on 802.11b 77
6.1.1.2 Performance of RT-traffic over DCF base on 802.11a 79
6.1.2 802.11e EDCF model develop 82
6.1.2.1 source_qos , sink_qos model and wlan_mac_intf_qos model 83
6.1.2.2 wlan_mac_qos model 84
6.1.2.3 QSTA attribute and node interface 90
6.1.3 Performance of RT-traffic over EDCF 91
6.1.3.1 Delays over EDCF 93

6.1.3.2 Loading Throughputs over EDCF 95
6.2 Simulation with NS-2 97
6.2.1 Explore HCF model by NS2 simulator 97
6.2.1.1 Discrete events machine (DEM) 97
6.2.1.2 Abundance object library 97
6.2.1.3 Dividual object model 97
6.2.1.4 Open source code 98
6.2.1.5 The parameter for Simulation of HCF by NS-2 99
6.2.2 The performance of RT-traffic over HCF 100
6.2.2.1 Delay of RT-traffic over HCF 101
6.2.2.2 Throughput of RT-traffic over HCF 103

CHAPTER 7 SIMULATION DATA ANALYSIS AND RESULTS 105
7.1 Simulation data analysis 106
7.2 Throughput and Fairness study 110
7.3 The future of QoS over WALN 113
7.3.1 Development of the PHY layer 113
7.3.2 Development of the MAC layer 114

CHARPTER 8 CONCLUSIONS 119
8.1 Research Conclusion 119
8.2 Research Contribution 121
8.3 Future research directions 123

References xv
Appendix A: The relative data xxi
Appendix B: The detail Figure xxvii
Appendix C: Key C/C++ source code xxix

List of figures

Figure 1-1 802.11 WLAN protocols in Network Architecture 2
Figure 2-1: Infrastructure Mode [7] 6
Figure 2-2: ESS Structure [7] 7
Figure 2-3 Ad-hoc Mode [7] 7
Figure 2-4 the wireless universe [11] 10
Figure 4-1 Frame Transmission after Random Back-off 32
Figure 4-2 the DCF Medium Access Process 33
Figure 4-3 the VFP and CP Timeline 34
Figure 4-4 the PCF Process 35
Figure 4-5 Frame Fragment 37
Figure 4-6 the General 802.11 MAC Frame 40
Figure 4-7 the Frame Control Sub-fields 40
Figure 4-8 RTS Frame 41
Figure 4-9 CF-END and CF-END+CF-ACK Frame 41
Figure 5-1 Different IFS [76] 63
Figure 5-2 HCF Mechanisms [62] 68
Figure 6-1 DCF Model 75
Figure 6-3 Time Delays in 802.11b Under the Light Loading. 77
Figure 6-4 Time Delays in 802.11b under the Heavy Loading 78
Figure 6-6 Throughputs in 802.11b Under the Heavy Loading 78
Figure 6-7 Traffic Delays in 802.11a under the Light Loading 79
Figure 6-8 Time Delays in 802.11a under the Heavy Loading 80
Figure 6-9 throughputs in 802.11a under the light loading and heavy loading 80
Figure 6-10 Performance of RT-traffic Over DCF Base on 802.11g 81
Figure 6-11 WLAN Station Model 82
Figure 6-12 QSTA Node Model 83
Figure 6-13 QSTA Interface 83
Figure 6-14 the Source_qos Process 84
Figure 6-15 the Sink_qos Process 84
Figure 6-16 the Waln_mac_intf_qos Process 84
Figure 6-17 the Waln_mac_qos Process 85
Figure 6-18 SV Block for EDCF 85
Figure 6-19 a Part of the Function Code of EDCF 86
Figure 6-20 Model Attribute 90
Figure 6-21 Type of Service 90
Figure 6-22 the Local Statistics Determined 90
Figure 6-24 Traffic delay in the 10% loading 93
Figure 6-25 Traffic Delay in Scenario 3 93
Figure 6-26 Traffic Delay in the in Scenario 9 94
Figure 6-27 Traffic Loading Throughputs in scenario 1 95
Figure 6-28 Traffic Loading Throughputs in Scenario 3 95
Figure 6-29 Traffic Loading Throughput in Scenario 10 96

Figure 6- 30 the HCF Exploring Process by NS-2 ..... *98*
Figure 6- 31the Delay in the Light Loading (10%) *101*
Figure 6-32 the Delay in the Heavy Loading (70%) *101*
Figure 6-33 the Total Delay of HCF *102*
Figure 6- 34 the Throughputs in the Light Loading (10%) *103*
Figure 6- 35 the Throughputs in the Heavy Loading (70%) *103*
Figure 6- 36 the Total Throughput *104*
Figure 7-1 Voice Mean Delay *106*
Figure 7-2 Voice Mean Jitter *107*
Figure 7-3 Voice Packet Loss Ratio *108*
Figure 7-4 Video Mean Delays *108*
Figure 7-5 Video Mean Jitter *109*
Figure 7-6 Video Packet Loss Ratio *110*
Figure 7-7 the Total Throughputs *111*
Figure 7-8 the Fairness Index of EDCF *112*

List of tables

Table 2-1 Compare with WiFi, Bluetooth and HomeRF *12*
Table 2-2 the Physical Layer Characteristic of 802.16 ..... *14*
Table 2-3 Comparing with IEEE 802.16 and IEEE 802.11 ..... *14*
Table 5-1 802.11e TC-to-AC Mapping [56] *62*
Table 5-2 Different CW and AIFS as the AC [77] *64*
Table 5-3 different TXOP Limit [78] *65*
Table 6-1 the Parameter for EDCF *91*
Table 6-2 Traffic Loading for EDCF Simulation *92*
Table 6-3 the Parameters of HCF Simulation *100*

Glossary

3G	Third-generation
AAS	Arbitrary Antenna System
ABR	Available Bit Rate
AC	Access Categories
ADPCM	Adaptive Differential Pulse Code Modulation
AGC	Automatic Gain Control
AIFS	Arbitrary Inter-Frame Spacing
AP	Access Point
ARQ	Automatic Repeat Request
BAD	Broad-Address
BE	Best Effort
BEB	Binary Exponential Backoff
BER	Bit Error Rate
BK	Background Traffics
BSS	Basic Service Set
BSSID	Basic Service Set Identity
BSSID	Basic Service Set ID
BTS	Base Transaction Station
BWA	Broadband Wireless Access
CA	Collision Avoidance
CAP	Controlled Access Period
CAS	Channel-Associated Signal
CBR	Constant Bit Rate
CCA	Clear Channel Assessment
CCK	Complementary Code Keying
CDMA	Code Division Multiple Access
CELP	Code Excited Linear Prediction
CF	Contention Period
CF	Compensating Fiber
CFP	Contention Free Period
CIR	Channel Impulse Response
CMOS	Complementary Metal Oxide Semiconductor
CP	Contention Period
CPE	Customer Premises Equipment
CRC	Cyclic Redundancy Check
CS/CCA	Carrier Sense/Clear Channel Assessment

CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CW	Contention Window
DA	Destination Address
DARPA	Defense Advanced Research Projects Agency
DBPSK	Differential Binary Phase Shift Keying
DAC	Distributed Admission Control
DCF	Distributed Coordination Function
DDN	Digital Data Network
DE	Discard Eligibility
DED	Discrete Event Driven
DEM	Discrete Events Machine
DFS	Dynamic Frequency Selection
DFS	Distributed Fair Scheduling
DFT	Discrete Fourier Transform
DIFS	DCF Inter-Frame Space
DQPSK	Differential Quadrature Phase Shift Keying
DS	Distribution System
DSS	Distribute Service System
DSSS	Direct Sequence Spread Spectrum
DTIM	Delivery Traffic Indication Map
EDCA	EDCF Admission Control
EDCF	Enhance DCF
EENAD	End to End Network Architecture Design
ERP	Extended Rate PHY
ESS	Extend Service Set
FCC	Federal Communications Commission
FCS	Frame Check Sequence
FEQ	Frequency Domain Equalizer
FFT	Fast Fourier Transform Algorithm
FHSS	Frequency Hopping Spread Spectrum
FSK	Frequency Shift Keying
FSM	Finite State Machine
FSO	Free Space Optics
FTAM	File Transfer, Access, and Management
FWT	Fast Walsh Transform
GAD	Group-Address
GFSK	Gaussian Frequency Shift Keying
GPRS	General Packet Radio Service
GPS	Generalized Processor Sharing

HC	**Hybrid Coordinator**
HCCA	**HCF Controlled Channel Access**
HCF	**Hybrid Coordination Function**
HEC	**Header Error Control**
HIPERLANS	**High-Performance Radio Local Area Networks**
HomeRF	**Home Radio Frequency**
HTTP	**Hyper Text Transport Protocol**
IE	**Information Element**
IEEE	**Institute of Electrical and Electronics Engineering**
IETF	**Internet Engineering Task Force**
IFFT	**Inverse FFT**
IntServ	**Integrated Service**
IR	**Infra-Red**
ISI	**Intersymbol interference**
ISM	**Instrumentation, Scientific, and Medical**
ISO	**International Organization for Standardization**
ITU	**International Telecommunication Union**
ITU-T	**International Telecommunications Union – Telecommunications**
LAN	**Local Area Network**
LD-CELP	**Low Delay CELP**
LLC	**Link Layer Control**
LMDS	**Local Multipoint Distribution Service**
LOS	**Line-of-Sight**
LSB	**Least Significant Bit**
MA	**Multiple Access**
MAC	**Medium Access Control**
MIB	**Management Information Base**
MIMO	**Multiple-Input Multiple-Output**
MMDS	**Multichannel Multipoint Distribution Service**
MPDU	**MAC Protocol Data Unit**
NAODA	**Network Application Optimization and Deployment Analysis**
NAV	**Network Allocation Vector**
NLOS	**No-Line-of-Sight**
NS	**Network Simulation**
OFDM	**Orthogonal Frequency Division Multiplexing**
OSI	**Open System Interconnection**
PBCC	**Packet Binary Convolutional Coding**

PC	**Point Coordinator**
PCF	**Point Coordination Function**
PDA	**Personal Digital Assistants**
PDO	**Protocol Development and Optimization**
PF	**Persistence Factor**
PHY	**Physical Layer**
PIFS	**Priority Inter-Frame Space**
PLCP	**Physical Layer Convergence Protocol**
PLW	**PSDU Length Word**
PMD	**Physical Medium Dependent**
PN	**Pseudo-Noise**
POTS	**Plain Old Telephone Service**
PPDU	**PLCP Protocol Data Unit**
PSDU	**PLCP Service Data Unit**
PSF	**PLCP Signal Field**
QAM	**Quadrature Amplitude Modulation**
QAP	**Quality Access Point**
QoS	**Quality of Service**
QPSK	**Quadrature Phase Shift Keying**
QSTA	**Quality Station**
RA	**Receiver Address**
RF	**Radio Frequency**
RSI	**Required Service Interval**
RSVP	**Resource Reservation Setup Protocol**
RT	**Real Time**
RTP	**Real time Transport Protocol**
RTS/CTS	**Request to Send/Clear to Send**
SA	**Source Address**
SAD	**Single-Address**
SAP	**Service Access Point**
SCFQ	**Self-Clocked Faire Queue Model**
SFD	**Start of Frame Delimiter**
SFQ	**Start-Time Fair Queuing**
SI	**Service Interval**
SIFS	**Short Interframe Space**
SLA	**Service Level Agreement**
SLA	**System Literary Analysis**
SLSND	**System Level Simulation for Network Devices**
SNA	**System Network Architecture**

SNR	**Signal-to-Noise Rate**
SRTS	**Synchronous Residual Time Stamping**
STA	**Wireless Station**
STC	**Space Time Code**
SWAP	**Share Wireless Application Protocol**
TA	**Transmitter Address**
TC	**Traffic Classes**
TBTT	**Target Beacon Transmit Time**
TCP	**Transport Control Protocol**
TDM	**Time Division Multiplexing**
TDMA	**Time Division Multiple Access**
TOS	**Type of Service**
TSPEC	**Traffic Specification**
TU	**Time Unit**
TXOP	**Transmission Opportunity**
UDP	**Universal Data Gram Protocol**
U-NII	**Unlicensed National Information Infrastructure**
UWB	**Ultra Wide Band**
VBR	**Variable Bit Rate**
VFRAD	**Voice Frame Relay Access Device**
VINT	**Virtual Internet Test-bed**
VMAC	**Virtual MAC**
VOATM	**Voice Over Asynchronous Transfer Mode**
VOFR	**Voice Over Frame Relay**
VOIP	**Voice Over Internet Protocol**
VS	**Virtual Source**
WAP	**Wireless Application Protocol**
WEP	**Wired Equivalent Privacy**
Wi-Fi	**Wireless Fidelity**
WiMAX	**Worldwide Interoperability for Microwave Access)**
WLAN	**Wireless Local Area Network**
WM	**Wireless Medium**
WMAN	**Wireless Metropolitan Area Network**
WPAN	**Wireless Personal Area Network**

CHAPTER 1

INTRODUCTION

1.1 BACKGROUNDS

With the explosive growth of the IP-based network, Quality of Service (QoS) is becoming a key issue for Real Time (RT) traffic transmitted over Wireless Local Area Network (WLAN).

Because IEEE802.11 technology can provide cheap and flexible wireless access capability, more and more vendors believe that it will play an important role in the future wireless communication system.

In the past, IEEE802.11 just provided better quality of service for the conventional data transmission. After all, WLAN has different architecture against wired LAN: its transmission medium is by radio frequency (RF) or Infra-Red (IR). These results into lower bandwidth and higher loss rate or bite error rate (BER). The original 802.11 standard is designed for best-effort data transmission. However, real-time voice, audio and video traffic are successfully transmitted over wired IP-based network today. It requires ***WLAN*** to provide good QoS performance for the RT-traffic. More and more people began to research performance of RT-traffic over WLAN with enhanced QoS.

1.2 SCOPE AND OBJECTIVES OF THE THESIS

Figure 1-1 presents the main network protocol architecture. IEEE802.11 WLAN covers Physical layer and the MAC layer of the Data Link layer. The QoS performance of RT-traffic is related with every layer of the network architecture. While this study is focused on the PHY layer and MAC sub-Layer of the OSI (Open System Interconnection) standard.

WLAN PHY layer is required to transmit a bit stream over physical Wireless mediums. The physical layer has two sub-layers called PLCP (Physical Layer Convergence Protocol) and PMD (Physical Medium Dependent) [1]. There are three types of physical

layers. Two of them are used a radio frequency and one is for infrared. The PLCP sublayer provides a carrier sense signal, called clear channel assessment (CCA) and provides a common PHY service access point (SAP) independent of transmission technology. The PMD is the layer responsible for the modulation and encoding/decoding of the signal. The PLCP and PMD sub-layers vary based on 802.11 types [2]. There are four basic PHY concepts and building blocks in the different PMDs that each 802.11 PHY provides. They are scrambling, coding, interleaving, symbol mapping and modulation. The different symbol mapping and modulation is the main difference among the IEEE802.11 protocols based PHY layer.

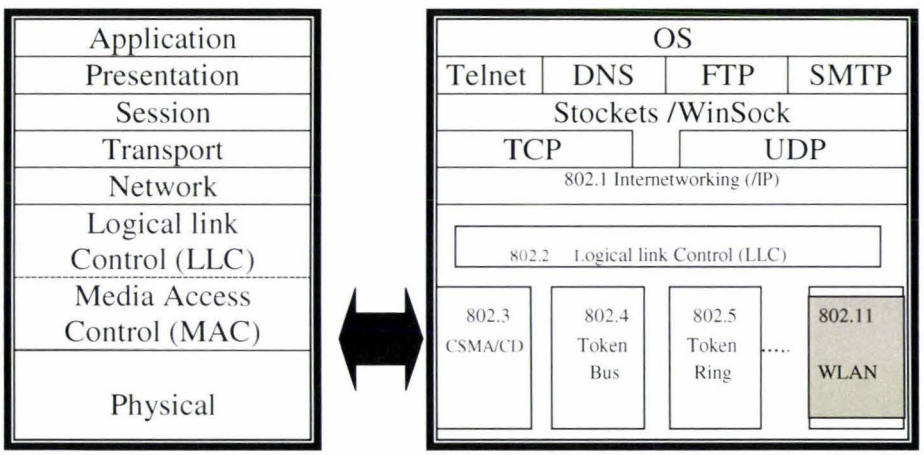


Figure 1-1 802.11 WLAN protocols in Network Architecture

According to Shannon's Law, the capacity of the signal transmission speed rate is directly proportioned to the transmission bandwidth [3]. It became a base design idea for QoS enhancement exploring the physical layer of WLAN. 802.11b, 802.11a, 802.11g, 802.11h and 802.11n increase the data rate from 1 Mbps to more than 100 Mbps with different symbol mapping and modulation technologies [4].

From figure 1-1, the Data Link layer includes the LLC sub-layer and MAC sub-layer, 802.11 WLAN does not cover the LLC sub-layer. The 802.11 MAC layer provides functionality to allow reliable data delivery for the upper layers over the wireless PHY media. The data delivery itself is based on an asynchronous, best-effort, connectionless

delivery of MAC layer data. The 802.11 MAC provides a controlled access method to the shared wireless media.

Originally, the 802.11 MAC layer mechanism is based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoid). It has two approaches: Distributed Coordination Function (DCF) and Point Coordination Function (PCF) [5]. DCF does not support any QoS guarantees for RT-traffic and PCF has limitations leading to poor RT-traffic QoS performance. Recently, different kinds of QoS enhancement schemes for both infrastructure and ad-hoc modes have been proposed for 802.11 MAC layer. On the basis of these results, the 802.11 group was drafted a protocol 802.11e for RT-traffic QoS over WLAN and finial published on 21 November 2005.

Generally, the MAC technology is the key for the QoS performance of RT-traffic over WLAN 802.11[6]. In the WLAN 802.11 family, the 802.11e is being drafted specially to deal with the QoS question.

The main objective of the thesis is to present and evaluate the QoS performance of RT traffic over the IEEE802.11 WLAN, focusing on the MAC layer of protocol. For studying the RT-traffic's OoS enhancement in 802.11WLAN, it gives a survey and analysis for the architecture and the mechanism from current 802.11 WLAN and some QoS enhancement proposals. This thesis attempts to find the limitations of the upcoming 802.11e, proposes and studies some solutions for enhancing the QoS performance of RT-traffic. The main contributions are as follows:

- A critical study of performance evaluation of the RT-traffic over QoS based on the original IEEE802.11 protocols.
- An in-depth analysis of the current proposed ideas for the MAC layer to enhance performance of RT-traffic over WLAN.
- Performance evaluation of the RT-traffic with QoS enhancement for the upcoming 802.11e protocol.
- Analysis and simulation performance of the RT-traffic. Analysis and simulation of the original IEEE802.11MAC protocols and PHY layer protocols for enhancement of QoS.

- Explore the EDCF model by OPNET and the HCF model by NS-2
- Analysis and simulation of the QoS performance of the RT-traffic based on the EDCF and HCF.
- Limitations analysis and resolution proposes of the upcoming 802.11e for RT-traffic QoS performance over WLAN.

In order to cover the above topics, the thesis is divided into eight chapters. The outline of the chapter is given in the next section.

1.3 OUTLINE OF THE THESIS

Chapter 1 provides the technology background and overview of the thesis. It presented the Scope and object and outline of the thesis. The main sub-topics of this thesis are issued. The outline of the thesis is given.

Chapter 2 introduces some detail for the WLAN characteristic. Meanwhile, it provides the basic knowledge for IEEE802.11 and compares the current wireless scheme to illustrate what is the advantage and disadvantages for traffic transmission.

Chapter 3 provides basic concepts of the QoS and RT-traffic used in this thesis. It presents the basic measured factors of QoS in the network for the following chapters. It analyses the QoS performances of RT-traffic against best-effort data traffic. The chapter presents the RT-traffic concept and its QoS requirements. It analyses the detail of the parameters and finally, explains the wired QoS technologies to fetch up the same topic in the next chapter for WLAN.

Chapter 4 describes the issues of WLAN QoS, it illustrates traffic over MAC layer and analyses the existing protocols: DCF and PCF. Finally, based on the mechanisms, it determines the limitations for RT-traffic QoS over WLAN.

Chapter 5 focuses on the QoS enhancement based on 802.11 WLAN. It continues to study MAC layer protocols. Based on the challenges in MAC layer, the chapter analyses

the advantages and disadvantage for some proposed enhancement schemes based on PCF and DCF. In fact, the upcoming 802.11e exactly is based on these proposed schemes. Based on these analyses, the chapter illustrates the upcoming 802.11e for RT-traffic QoS performance. It analyses the mechanism of 802.11e from the AC definition, AIFS, back-off mechanism, TXOP-EDCF and admission control for EDCF. The chapter also analyses the access occur, admission control and the algorithm schedule of 802.11e draft for HCF.

Based on the above knowledge and analysis, the ways are three methods to study the 802.11e physically. The three methods are: mathematical analysis method, experimental method and simulation method. After analyses, we decide to employ the simulation method to study the protocol.

In chapter 6, we explore the models of the EDCF and HCF by OPNET and NS-2 respectively. We simulate the protocols to measure the QoS for RT-traffic performance over WLAN. The chapter focuses on analyzing delay, jitter, and throughput and packet loss rate. These key QoS factors covered by 802.11 protocols without some measures are just for high layer in OSI system. As we know, the RT-traffic are the time-sensitive traffic; delay is the most important factor.

In chapter 7, based on the data of the simulation in chapter 6, we do further analysis for the topic. Because nowadays the upcoming 802.11e is unstable in some cases, the chapter attempts to find the limitations in the 802.11e. By the mathematical model analysis, the chapter presents some relationships among the parameters in the 802.11e mechanism. At the end of this chapter, some prediction of the RT-traffic over WLAN and some proposals for solution to the 802.11e limitations are given.

Chapter 8 gives the conclusion for the thesis. It presents the contributions of this thesis in this research field and gives some research direction for further work related to QoS performance of RT-traffic over 802.11 WLAN.

CHAPTER 2

AN OVERVIEW OF WLAN STANDARDS

2.1 TOPOLOGIES AND OPERATING MODES OF WLAN

2.1.1 Topologies

Actually, IEEE 802.11 defines two topologies for the WLAN: infrastructure and ad hoc mode.

Infrastructure mode

Infrastructure mode contains: AP (Access Point), STA (Wireless Station) and DSS (Distribute Service System). DSS can be divided into BSS (Basic Service Set) and ESS (Extend Service Set) by the covered area. AP is also called the wireless hub. In infrastructure mode, AP can connect with the wired network infrastructure for data receiving, data buffer and data transmission. The wireless AP can cover tens or hundreds of users. The covered radius is about hundreds of meters. (Figure 2-1)

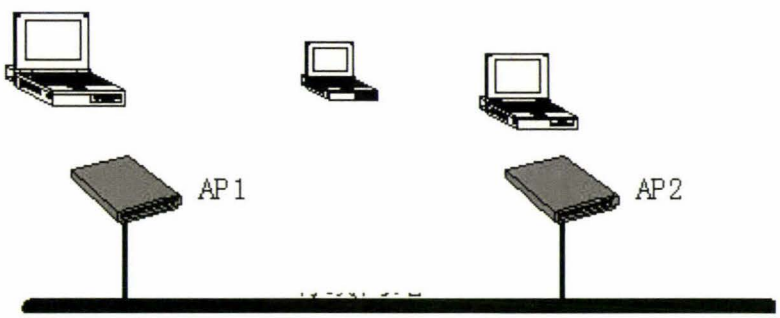


Figure 2-1: Infrastructure Mode [7]

An AP and associated STA construct the BSS. Any STA in the BBS is connected with this AP in anytime. In other words, the covered cell area by an AP is BSS. The STA uses BSSID (Basic Service Set Identity) to connect to the AP. In IEEE802.11, the BSSID is the MAC address of the AP.

ESS is a kind of constructed network composed multi-APs and its distributed system. All APs must share an ESSID, in other words, ESS contains many BSSs. Because of the characteristic of the IEEE802.11, the ESS just contains the Physical Layer and Data Link Layer (Figure 2-2).

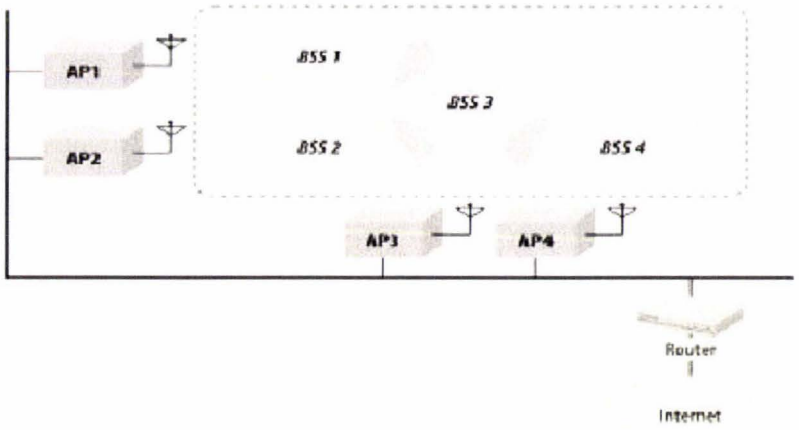


Figure 2-2: ESS Structure [7]

Ad-hoc mode

Ad-hoc mode (also called peer-to-peer mode or an Independent Basic Service Set, or IBSS) is a set of 802.11 wireless stations that can communicate directly with each other without an access point or any connected wire (Figure 2-3). The covered area by ad-hoc is called IBSS.

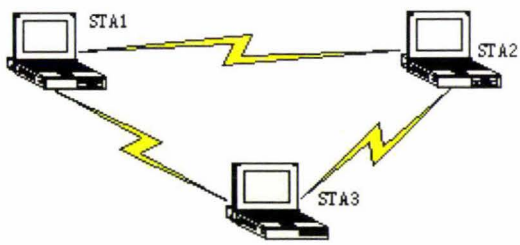


Figure 2-3 Ad-hoc Mode [7]

An ad-hoc network is composed of a group of wireless terminals with the wireless port. Those PCs have the same group name, ESSID and passport. Ad-hoc network provides flexible access at anytime. It just needs two or more wireless ports in the available area. They are independent networks. This constructed network even needn't any precondition as the request in network management. However, ad-hoc network must "see" other points; otherwise, it thinks the network is break. So an Ad-hoc network is used in less clients' network circumstance, say 4 or 8 clients, at the same time they are nearer enough.

This mode is efficient for setting up a wireless network anywhere quickly and easily. Because a wireless infrastructure does not exist physically, it is always used in a hotel room, convention center, or airport.

2.1.2 IEEE802.11 operations

The operation of WLAN can be divided into two processes. One is a STA enters a BSS and another is a STA from are BSS to other BSS (roaming). Once the power of STA is turned on, the STA can be in a sleep mode. Meanwhile, the STA need the synchronous signal to cooperate with the AP. Actually, the STA keeps synchronous with the AP by active scan or passive scan.

Active scan means that after the STA open a successful association, the STA scans all channels. In every scan, STA's scan area is a group of channels. When STA finds one idle channel, it broadcasts with the ESSID, the AP answers by the signal. **Passive scan** means that the AP sends the beacon signal per 100 milliseconds. The signal contains the time stamp for STA synchronization. It can support rate and other information to identify. The STA receives the beacon signal to start the association process [8].

To defend the unauthorized entrance, after the STA is fixed on the AP and take the synchronous signal, AP and STA commence to change the verifiable information. The verifiable service provides that the WLAN can control who can enter the network. The process also can setup the authorized mark in the WLAN.

Passing the authorization verification, association starts. The association is for building up the mapping relation between AP and STA. In fact, make the wireless connection to wire connection. The distributed system distributes this mapping relation to all APs in ESS. A STA can associate with just one AP. In the associative process, the STA and AP arrange the signal as strong or weak. Fro example the IEEE802.11b, the changed rate is 11Mbps, 5.5, 2 Mbps, 1 Mbps.

To connect STA from one BSS to another BSS needs reassociation. Reassociation means the process, which the STA associate with the new AP when the STA from one of the BSS in the ESS associates to other BSS. Reassociation always starts from the STA.

IEEE802.11 defines that every STA associates with one AP. If a STA from one BSS to other BSS, it's in the roaming. This process of dynamically associating and reassociating with APs allows network managers to setup WLANs with very broad coverage by creating a series of overlapping 802.11b cells throughout a building or across a campus. To be successful, the IT manager ideally will employ "channel reuse," taking care to set up each access point on an 802.11 DSSS channel that does not overlap with a channel used by a neighboring access point. The roaming may be base roaming and extending roaming. The base roaming means that STA roam in an ESS. Extending roaming means STA from a BSS in ESS to a BSS in other ESS, 802.11does not ensure the above layer connect in Extending roaming.

2.2 WLAN STANDARDS

In fact, IEEE802.11 is not the only standard for WLAN. With such a wide range of possible applications, it is understandable that a number of different wireless data communication solutions would arise. Of course, the IEEE 802.11 is better one of these. The other situation, these include HIPERLANS (High-Performance Radio Local Area Networks[9]), HomeRF (Home Radio Frequency), Bluetooth [10], GPRS (General Packet Radio Service), 3G (third-Generation Cell), and FSO (Free-Space Optics). With overlapping functionality, a seemingly endless stream of technical variants, and differing states of commercial maturity, it seems a daunting task to fit them together. For these reasons, the IEEE 802.11 standards have emerged as a key enabler for the growing universe of wireless data applications.

A traditional way to divide wireless standard is by distance (Figure 2-4). Although longer-distance technologies would seem to provide the same capabilities as shorter-distance technologies, factors such as frequency availability, transmission speed, interference, cost, and government regulation have resulted in specific application for each technology. A short-range wireless network with communication distance of 10

meters or less has come to be known as a WPAN (Wireless Personal Area Network). The most notable WPAN is Bluetooth. The IEEE 802.15 Working Group is developing standards for WPANs. A WLAN generally handles distances of several hundred meters or less and typically links devices within a building or campus. In this group, we have IEEE802.11, HIPERLAN, and HomeRF. A WMAN (Wireless Metropolitan Area Network) provides broadband wireless access for small regional areas such as cities or metropolitan areas. Wireless MAN protocols include LMDS (Local Multipoint Distribution Service) and MMDS (Multichannel Multipoint Distribution Service). In fact, this kind of traditional division is not perfect. The technologies in these standards overlap.

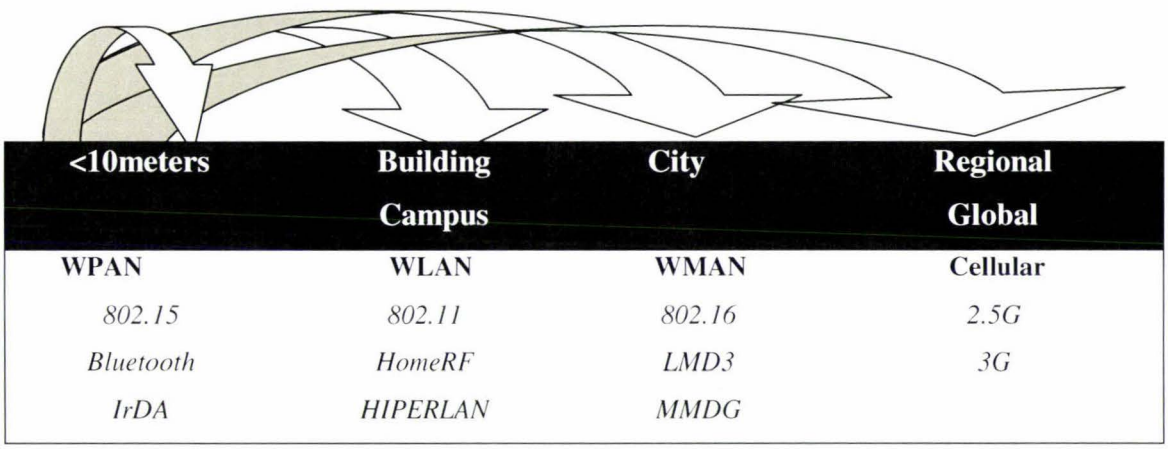


Figure 2-4 the wireless universe [11]

2.2.1 WiFi (IEEE802.11b)

IEEE 802.11b, also called Wi-Fi (Wireless Fidelity) is based on the code modulation DSSS (Direct Sequence Spread Spectrum) and the code technology CCK (Complementary Code Keying). WiFi is the broadest application in WLAN now [12]. It will be discussed detail in the next chapter.

The theoretical rate of WiFi is 11Mpbs. However, this is not the actual information rate. 11Mbps means the whole physical layer rate. Most of them are used for the protocol itself. The application rate is not so much. In fact, the highest application rate is 6Mpbs. If the interference increased, the rate should be reduced quickly. The 50% interference code rate should result to 2/3 reducing, it should be 2Mbps. WiFi is very difficult to break

through the bottleneck about the low rate. 802.11b is compatible with the old style DSSS system. It cannot be compatible with the system that is based on FHSS (Frequency Hopping Spread Spectrum).

The QoS and the low rate will become the main hindrance for 802.11b development. Meanwhile, IEEE802.11b uses the 2.4GHz ISM band, and many home electric applications also use this band. So the 802.11b must face more interference. The security of the 802.11b is not good. It uses WEP (Wired Equivalent Privacy) encryption to resolve the problem. IEEE is developing 802.11i to increase the safety of WLAN.

2.2.2 Bluetooth

Bluetooth technology is ideal for the WPAN originally. The Bluetooth 1.0 was issued in December 1999 [13]. It defines many applications that use the WAP (Wireless Application Protocol) to connect to the Internet. It can increase the new functions for cell phone system, wireless communication system, WLAN and Internet. It can be used for home automation, home entertainment, electrical business, wireless portfolio, digital facilities, industry control and intelligent building, etc. The IEEE802.15 group is setting down a WPAN standard, with rate more than 20Mb/s. It is based on the Bluetooth technology. The main purpose is to realize the perfect jointing between WPAN and WLAN. After one year, the conclusion was that the base MAC structure of IEEE802.11 with the three kinds of physical medium layers does not fit to use in WPAN. 802.15 was issued in March 2001, although it is later than 802.11. However, the market quotient of 802.11 in home wireless communication is still less. The Bluetooth decided to use the 2.4GHz band [14].

Quality of service is an important issue when designing any communication link. The Bluetooth specification provide QoS configuration to allow the properties of links to be configured according to the requirements of higher layer applications or protocols. The properties that can configure depend on the peak bandwidth, delay requirements and delay variations.

2.2.3 HomeRF

HomeRF has an integrated system for home intranet. The system includes the connection protocol between peripheral equipment and host computer, the connection protocol between peripheral equipments, the connection protocol between host computer and central control, enter network and PSIN, etc. The rate of HomeRF can be 8M~11Mbps. Like Bluetooth, HomeRF permits up to 5 equipments to connect each other.

2.2.4 Comparison of WiFi, Bluetooth and HomeRF

802.11b(Wifi), Bluetooth and HomeRF have different characteristics respectively in WLAN. We can see them from the table below (table 2-1).

Table 2-1 Compare with WiFi, Bluetooth and HomeRF

	802.11b	HomeRF	Bluetooth
Transmission Speed	11Mpbs	1,2,10Mpbs	30-400Kpbs
Application area	Building and Campus LAN	Home office, Personal Home Network	
Terminal Type	Laptop, PC PALM PC Internet Gateway	Laptop, PC, Modem, Telephone, Mobile, Internet Gateway	Laptop, Cellular Mobile, PALM PC, BP-Call, Car
Transmission Protocol	DSSS	FHSS	RFSS
Supportive Corporations	Cisco, Luccent, 3Com,WECA consortium	Dell, Home Group, Intel, Motorola, Proxim	Bluetooth Group, Ericsson, Motorola, Nokia
Reference Link	www.wireless-ethnet.com	www.homerf.org	www.bluetooth .com

2.3 IEEE802.16 AND WiMAX

The technology of IEEE802.16 is for WMAN. It is developing quickly now. More and more users are interested in IEEE802.16 for wireless communication products. As the WiFi alliance for WLAN, the WiMAX forum is setup for the IEEE802.16 development.

2.3.1 IEEE802.16

IEEE802.16 was approved in December 2001. It is for 10~66GHz high band in LoS (Line-of-Sight) operation of the WMAN. The 802.16 includes 802.16a, 802.16Revd, 802. 16e. 802.16a is designed for 2~11 GHz band in NLoS(No-Line-of-Sight) broadband operation to connect the network system. IEEE approved it in January 2003. 802.16RevD is the developed 802.16a. It is focused on CPE (Customer Premises Equipment). It was approved in October 2004. 802.16e is the high extension of the 802.16 a/d. The designed

idea is enhancing the data transmission capability based on the 802.16 standard. It is estimated that would be approved in the later half of 2005. Development of WMAN is required by the BWA (Broadband Wireless Access) market increasingly. The 802.11x and other technologies were used for the BWA recently and they are successful. However, from the whole design idea and characteristic of the WLAN is not good for the BWA application. When WLAN is used outdoor environment, the bandwidth and customer number limits it. Meanwhile, the communication distance is a big problem for the WLAN [15]. Based on this situation, IEEE made 802.16 to solve the QoS and physical circumstance (outdoor radio frequency transmission) for the BWA and 'last mile' requirement to connect the market. IEEE802.16 can provide a global standard for these enterprises and home clients without broadband service. They can get the broadband service by the Communication Corporation and ISP through the WMAN.

The equipments designed as the IEEE802.16 can be used instead of DSL and T1/E1 in 'last mile' communication area. The 802.16 includes a protocol, which supports low time delay for RT multimedia traffic. It also supports NLOS broadband connection between terminal equipments and BTS (Base Transaction Station). A BTS can support hundreds even thousands of clients. 802.16 can provide high QoS for RT multimedia traffic.

802.16a, 802.16RevD and 802.16e have the same Physical Layer and MAC layer. The Physical Layer Protocol is 256 points FFT (Fast Fourier Transform Algorithm) OFDM (Orthogonal Frequency Division Multiplexing) standard.

If it uses CDMA (Code Division Multiple Access), for preventing the interference, the RF bandwidth must be higher than data throughput. If the wireless bandwidth is lower than 11GHz, it cannot do undoubtedly. The reason is for the data rate of 70Mbps, it need to provide more than 200MHz RF bandwidth to do the transaction plus and NLOS function.

For reliability, 802.16 has some designed characteristics: flexible channel bandwidth, self-adaptation burst signal profile, adopts Reed-Solomon, correct convolution code, AAS (Arbitrary Antenna System) (can change distance and capacity), DFS (Dynamic

Frequency Selection, to reduce the interference), STC (Space Time Code, increase performance in attenuation circumstance by the time and space code respectively). The Physical Layer characteristic of 802.16 is showed in table 2-2.

Table 2-2 the Physical Layer Characteristic of 802.16

Characteristic	Advantage
256 FFT OFDM wave	***Resolve multi-path in outdoor LOS and NLOS***
Self-adaptation modulation and changeable FCS	Maximum rate and high reliable for RF link
Support TDD and FDD duplex mode	Adapting different mode in globe
Flexible channel width (3.5,5,10MHz)	Flexible, it can be used in different place in globe.
Support intelligence antenna system	Control interference and increase system expansion ability

Most of the wireless network works on shared medium. So they need a kind of controlled access mechanism for their clients. The MAC Layer of 802.16 assigns capacity to their clients by TDMA protocol in point-to-point network topology, which is controlled by the BTS. By TDMA mechanism, 802.16 system provide not only high rate service with SLA (Service Level Agreement), but also can control QoS, not just only control the priority.

Comparing IEEE 802.16 and IEEE 802.11

We explain it by the table 2-3:

Table 2-3 Comparing with IEEE 802.16 and IEEE 802.11

	802.11	802.16	Technology
Communication distance	>300 inch	30mile, normal 4-6mile	802.16 is 256 FFT. 802.11 is 64FFT
Cover area	Short, indoor	Outdoor NLOS, support advanced antenna technology	802.16 system gain enhance, penetrable capacity better
Expansible	For LAN, clients number can from 1 expand to tens Fixed channel bandwidth (20 MHz)	Can to hundreds clients, expand unlimitedly. Flexible channel bandwidth (from 1.5MHz to 20 MHz)	802.11MAC is CSMA/CA protocol 802.16 MAC is dynamic TDMA 802.11 just can use unlicensed band, channels less 802.16 can use all channels
Bit Rate	2.7bit/s/Hz	5bit/s/Hz	
QoS	No QoS normally	MAC include QoS	802.11 MAC base on (CSMA/CA) 8-2.16 base on dynamical TDMA, bandwidth can be reassigned

From QoS, 802.16 connect medium by agree/require protocol. It supports different service level. The protocol was divided into up link and down link. There are TDM data streams in down link and up link by TDMA. It supports centralization dispatching for the time sensitive service to access without collision. 802.16 MAC improved system throughput and efficient bandwidth and ensure the time delay not too long. TDM/TDMA technology also supports multicast and broadcast service. Generally, for WALN, the core technology is CSMA/CA, so to realize QoS which must be other ways.

2.3.2 WiMAX Forum

WiMAX (Worldwide Interoperability for Microwave Access) forum is set up in April 2001 for 10~66 GHz band IEEE802.16. The primary function of WiMAX is to establish a worldwide Interoperability uniform standard based on IEEE802.16 and ETSIHIPERMAN standard. The uniform standard can ensure the system components developed by different producers conform interoperability.

2.4 THE FUTURE OF WIRELESS LAN

It is very difficult to predict the future about the WLAN. We just can forecast from the general trends of the WLAN development. For more and more RT-traffic over the WLAN, we must advance the wide band and must increase data rate. That's sure the WLAN's trend. In this section we discuss three technologies about the future of WLAN. They are UWB (Ultra Wide Band), FSO (Free Space Optics) and 100Mbps.

2.4.1 UWB

UWB is not designed for WLAN. It is an extension of Bluetooth with much higher rates that uses short duration and low power pulse. It is suitable for WPAN today. However, this upcoming technology might influence the WLAN technology.

UWB (Ultra Wide Band) is a new technique for which wide relative bandwidth signals generated via short, low power pulse and that could allow high-bandwidth, interference-resilient communication. UWB can penetrate the wall. This technology originally was used in military affairs. FCC (Federal Communications Commission) lifted a ban in

February 2004. In fact, UWB is not only used for wireless communication but also radar, scouting orientation. Military projects use UWB for accurate measure of distance and position. For example, the car collision orientation detector is. Meanwhile, for the ultra width bandwidth and penetration, UWB can be used radar to find the picture underground and behind the impediment. FCC gives 3.1GHz-10.6GHz for civil use [17]. The smallest bandwidth is 500MHz. FCC requires that the emission power must be lower -41.3dBm/MHz (about 1mW/MHz). UWB's general rate is hundreds of Mbps in several meters. It is far higher than the rate of the WLAN.

The UWB is different from the WLAN. It uses FHSS and DSSS. However, it is different from 802.11 and Bluetooth by a few allocated frequency bands to transmit signal, UWB adopt all frequency bands. UWB signal is short, just ms. so the power is less. Because the bandwidth of UWB is huge, so it is suitable for huge data transmission, especial for the RT multimedia traffic. Meanwhile, UWB can better collaborate with other wireless technologies. The power of distributive transmission is very low; just like the noise to other wireless signals.

UWB calls for the design of RF devices of extremely wide wireless bandwidth. It is the nature of the technology that it will always be operating in the presence of interference at power levels much higher than the desired signal.
The bandwidth of the signal requires faster processing than maybe done digitally today.
As with the RF challenge, there will be a similar challenge to design antennas with the desired bandwidth.

UWB is only an FCC initiative, so major global standardization effort is necessary.
From the short term (say in 3-5 years), UWB can see the extension of the Bluetooth. However, for the future, UWB must be advantage complement for the 802.11.
UWB does not modulate and demodulate complicated carrier signal, so it does not need mixer, filter, RF/IF converter and local oscillator. Because there aren't those complicated parts, the power is very lower. It is easy to integrate into CMOS (Complementary Metal Oxide Semiconductor). The more important is the technology of the UWB physical layer.

If it is possible, 802.11 can be based on the UWB physical layer, by UWB technology for transmission.

2.4.2 FSO

FSO is not a new technology. It existed in 1980. In the past 4 years, it was developed quickly. In the American '911' affair, many people dead. Meanwhile many enterprises near the Manhattan city were broken to their communication. After then, many new corporation use FSO to provide the wireless broadband access service. It is proved that FSO is better in resolving the 'last mile' in broadband access.

FSO is a kind of technology combined of optical fiber technology and wireless communication. It transmits signals by air, not by the optical fiber. Today, many enterprises and organizations do not have optical fiber line, but they need the higher data rate than T1 or T3. FSO can do the job instead. It can provide more than 1Gbps rate. It is expected to have well useful role in the future WLAN.

For the future, the main technological challenges that FSO links face are as follow; Fog, which consists of tiny water droplets and can absorb, scatter, or reflect light, is the major challenge. Other forms of weather, such as rain and shower, have a lesser effect, although very heavy rain or blizzard conditions can also break down a link.

Absorption, which is a function of the wavelength of the light in use, can decrease the power of the light beam. Absorption most often comes from fog or aerosols such as dust, sea salt or man-made pollutants.

Scatter, especially when the scattering particle is similar to the wavelength, can significantly attenuate the beam intensity because it redirects energy in random directions. The scattering particles could be fog, haze, or pollutants. As the link range increase, so do the scattering losses.

Physical objects, such as birds, can actually temporarily interrupt FSO links.

Building sway can disturb the alignment of the transmitter and receiver and disrupt the link.

Turbulence, which occurs when heated objects create moving air pockets of differing temperatures, causes time-varying changes in the index of refraction at the air-pocket interfaces. It can result in beam wander as it randomly reflects through the pockets, scintillation in the form of intensity fluctuations, and increased beam spread.

2.4.3 100Mbps WLANs

In the near future, we might see the formation of higher throughput task group in 802.11[18]. It is anticipated that such a task group would not only focus on achieving a 100 Mbps data rate, but also strive for a 100Mbps throughput experience for users-because it is what they have come to modifications to the 802.11 physical layer and the 802.11 MAC. The group must weigh questions of coexistence and backward compatibility in addition to those of basic viability regarding spectral efficiency, range, and power consumption.

From a usage-profile perspective, the two main drives for 100 Mbps throughput WLANs will be throughput equivalence with wired 100BASE-T Ethernet and wireless multimedia for the home. The former will further the cause of the fully wireless office because wireless will provide high-quality audio and video to all parts of the home without wiring and also support Internet surfing.

Summary

The IEEE 802.11 covers two layers: PHY layer and MAC layer of the OSI system. There are two topologies in 802.11: infrastructure mode and ad hoc mode. Recently, heterogonous standards are set for wireless LAN. IEEE802.11 will play the most important role in the wireless communication market. Because of its inherent characteristic, the 802.11 MAC layer is more complicated than the 802.3 MAC layer. The QoS performance of RT-traffic will become the new challenges for the WLAN.

CHAPTER 3

NETWORK QUALITY OF SERVICE

In 2004, VOIP (Voice over Internet Protocol) became the most popular technology in the telephone market. VOIP is not a new network, but a new real time application on IP network. In the past, circuit switching was used for voice transmission, and the dedicated communications path established for the conversation. VOIP is actually a converged network strategy. When VOIP is used on the LAN, voice is converted to stream IP packets and sent over an Ethernet network. This is a kind of real-time traffic; it would meet QoS requirements and is comparable in performance to conventional circuit-switched telephone networks.

Actually, the QoS requirements of real-time traffic are different to conventional data over digital network. The real time traffic is a kind of time-delay sensitive traffic. In old circuit switch and telephone system, it provided a dedicated path for voice transmission, so we needn't consider the QoS for RT-traffic. However, although IP-based system provides a flexible transmission path, different types of traffic are transmitted by one rule for transmission. Because of different character of RT-traffic, the performance over the network is poor. VOIP just resolve the problem by QoS technology. In this chapter, we will study some concept of the QoS in the network.

3.1 CATEGORIZING TRAFFIC

As end-user expectation or application requirement [19], we can categorize applications into four different traffic categories: Interactive, Responsive, Timely, and Network Control.

Interactive application contains: VOIP, interactive game, video conferencing etc. As the name 'interactive' indicates, it means two or more people actively participating. So this kind of traffic needs respond in 'real time'. In [20], 'real time' is defined: that is minimal

delay (latency) and delay variation (jitter) between the sender and receiver. Because the interactive application operates in real-time, packet loss must be minimized. For example, if you talk with your friend, he can't hear parts from your words, he may not catch what you mean. In other words, the quality of service of the conversation is not satisfactory.
As network architecture, interactive applications typically are UDP-based (Universal Data gram Protocol) [21]. The UDP-based transmission cannot retransmit lost or dropped packets as with TCP-based (Transport Control Protocol) traffic. As RT-traffic is time-sensitive, packet retransmission after certain delay would not be beneficial. For example, you talk with your friend and lost some words. If you repeated it later, your friend maybe confused with recent words. Real time Transport Protocol (RTP) [4] is an Internet Engineering Task Force (IETF) protocol that provides end-to-end functionality for transport of real time audio and video data, and has been widely accepted.

Responsive application contains: streaming audio/video, client/server transaction etc. Stream media application includes Internet radio (talk radio and music) and audio/video broadcasts.

'Responsive' indicates the traffic between a person and a networked device application. Terminal user requires these applications to be 'responsive' so a request sent to the networking device requires a relatively quick response back to the sender. The traffic is similar to 'Real-Time' traffic. So we can call it 'near real-time' traffic.
Responsive application can use either UDP or TCP-based transport. Stream media applications typically use UDP (but can also use TCP). Web-based applications are based on the Hyper Text Transport Protocol (HTTP) and always use TCP. For Web-based applications, packet loss is managed by TCP that retransmits lost packets. The application-level protocol control how often packet is retransmitted. Otherwise, the lost packets are discarded, resulting in some distortion in the media.

Timely application contains E-mail, non-critical OAM (Operation and Maintenance) etc. It is transmitted between a person and networked device. The difference against the Responsive traffic is that timely application does not require 'near real-time' performance

but do require the information to be delivered in a timely manner. The time application is the called 'best-effect traffic' normally.

Network control application contains critical alarm, routing, billing, critical OAM (Operation and Maintenance) etc. The network control applications are used to control the operation and administration of the network, for example ACK. These applications can subdivide into those required for critical and standard network operating conditions.

There are many other categories for different traffic. These categorizations are according to similar principle. The most widely available schemes [3] categorie traffic to three categories: CBR (constant bit rate), VBR (variable bit rate), and ABR (available bit rate).

CBR traffic is always transmitted as constant bit rate traffic in networks, generally, the RT-traffic (audio traffic and video codes) are CBR traffic. These traffic must be assigned with enough required bandwidth without extra bandwidth. In the past, CBR usually run in circuit-switched network. Then they can be transmitted by dedicated bandwidth.

VBR traffic contains the near-RT-traffic. They are more "burst" in nature and fluctuate between low and high bandwidth requirements, so the transmitted rate can be a little bit variable.

ABR traffic is the best-effect traffic. It is the same as the Timely application. The name suggests that it can function with a wide range of available bandwidth.

3.2 THE BASIC CONCEPT OF QOS

Quality of service (QoS) involves a broad range of technologies, architecture, and protocols. In this section, we will discuss QoS from three parts: the general concept of QoS, the relation between QoS and bandwidth, and the QoS performance measure.

3.2.1 QoS overview

According to [22], QoS refers to the ability of a network to provide better service to selected network traffic over various technologies. The theory behind is that different traffic has different requirements in terms of bandwidth, delay, loss, and availability. Recently, IP-based network is widespread used over the world. Because, IP-based network has its advantage: network equipment becomes easier to maintain, resulting in lower operational costs. Since the IP-based network is by the connectionless protocol, the transmission paths become unpredicted. However, for CBR and VBR traffic transmission, the situation is different. For example, the original voice is transmitted with the circuit switch. It means the transmission has its dedicated path. QoS technologies played a smaller role because the traffic was similar behavior and the dedicated connection suffered less delay to meet the required behavior of the particular application. But in IP-based network, it must support different type of traffic. RT-traffic need low latency, otherwise, the terminal user quality of the RT-traffic may be poor. We can consider a voice RT-traffic. Voice traffic is transmitted in original telephone system using TDM (Time Division Multiplexing) technology. It can ensure the traffic has a very deterministic behavior, a low delay and no loss. However, as the IP-based network characteristic, it cannot provide the TDM-voice behavior. QoS techniques can be applied to the best-effort IP network to make it capable of supporting VOIP with acceptable, consistent, and predictable voice quality. Therefore, QoS technology plays a crucial role to ensure that different traffic can be properly supported in a multi-service IP network.

3.2.2 QoS versus Bandwidth

From Shannon's Law:

$$C = B\log_2(1+SNR) \ldots\ldots\ldots\ldots \text{(2-1)}$$

Where C is the capacity of the channel in *bps* and B is the bandwidth of the channel in *Hertz*.

If the ratio of Signal to Noise (SNR) is fixed, the capacity is directly proportioned to the bandwidth. One-view believes that QoS is not needed as increasing the bandwidth will suffice and provide good QoS for all traffic. They think that transmission can use added bandwidth simply instead of the complicated implementation of QoS. We can't say this

argument is incorrect. If all network connections have infinite bandwidth such that networks never became congested, then one would not need to apply QoS technologies. However, it is difficult in the actual operation. High-bandwidth connections are not available throughout the network, from the traffic's source to the destination. This is especially true for an access network where the most commonly available bandwidth is typically only hundreds of kbps. Furthermore, bandwidth discontinuities in the network are potential congestion points resulting in variable and unpredictable QoS that a user or application experience.

3.3 QoS PARAMETERS IN THE NETWORK

Many QoS parameters can be measured and monitored to determine whether a service level offered or received is being achieved. Actually, the QoS performance is related with every layer of the standard OSI network model. For the high layer, Silence suppression, Echo solution, traffic Compression, Synchronization are the technologies to enhance the RT-traffic's QOS performance. In this section, we discuss the following parameters in both of PHY layer and MAC layer that are relevant to 802.11standand. These parameters are bandwidth, delay, jitter, and packet loss.

3.3.1 Network availability

Network availability is a significant parameter to affect QoS. Simply say, if the network is unavailable, even during brief periods of time, the user or application may achieve unpredictable or undesirable performance. It can result into poor QoS performance. Network availability is the summation of the availability of many items that are used to create a network. These include networking device redundancy, resilient networking protocol, and multiple physical connections etc. Network operators can increase their network's availability by implementing varying degrees of each of these items. The greatest challenge for network operators today is to provide highly available IP networks.

3.3.2 Bandwidth

Bandwidth is probably the second most significant parameter that affects QoS. Bandwidth allocation can be subdivided into two types: available bandwidth, guaranteed bandwidth. Available bandwidth is not always equally share by all customers. This allows all users to compete for available bandwidth. They get more or less bandwidth depending upon the amount of traffic from other users on the network at any given time. We can think the available bandwidth is the maximum bandwidth that the network provides. Guaranteed bandwidth has the high QoS for clients, network operators offer a service that provides a guaranteed minimum bandwidth and burst bandwidth in the SLA (System Literary Analysis). Because the service is guaranteed, the service is priced higher than the available bandwidth service. The network operator must ensure that those who subscribe to this guaranteed bandwidth service get preferential treatment (QoS bandwidth guarantee) over the available bandwidth subscribers. In some cases, the network operator separates the subscribers by different physical or logical networks. Burst bandwidth can be specified in term of amount and duration of excess bandwidth (burst) above the guaranteed minimum. QoS mechanisms may be activated to discard traffic that is consistently above the guaranteed minimum bandwidth that the subscriber agreed to in the SLA.

3.3.3 Delay

Network delay is the transit time that traffic was transmitted from the sender point to the destination point of the network. Delay can cause significant QoS issues with RT-traffic transmission that simply time-out and fail under excessive delay conditions. Some traffic can compensate for small amounts of delay but once a certain amount is exceeded, the QoS becomes compromised. For example, some networking equipment can 'spoof' an SNA (System Network Architecture) session on a host by providing local acknowledgements when the network delay would cause the SNA session to time out.

Delay can be both fixed and variable. There are packetization delay, transmission delay, propagation delay, store-and-forward delay and processing delay. The latency metric captures the one-way delay encountered by a datagram from the instant it is transmitted

by the framework at the sending end point, to the time instant it is received by the framework in the receiving end point. For the one-way delay measurement to be accurate, a synchronization mechanism such as NTP (Network Time Protocol) [22] is required. The total delay (latency) in a network will be the accumulated sum of them. Propagation delay is the length of time it takes information to travel the distance of the line. This is fixed by the speed of light therefore. Propagation delay is fixed by the speed of light therefore.

Transmission delay is the length of time it takes to send the packet across the given media. It is determined by the speed of the media and the size of the packet. Processing delay is the time required by a networking device for route lookup, changing the header, and other switching tasks.

Network delay is common problems in the performance of RT- traffic like video or audio streaming. Applications can be divided into data-only applications, voice-only applications, video-only applications or truly mixed multimedia traffic. Real-time, interactive applications such as desktop conferencing are sensitive to accumulated delay, which is referred to as latency.

For some RT-traffic, a bounded delay is allowed. For voice communication the end-to-end latency is below 300ms [8]. The budget for national network latency for an intercontinental voice packet will be far below this value after considering for the large satellite latency. Most applications will perform acceptably within one-way-delay value of 150 ms while for highly interactive data and conversational applications, a delay value of less than 100 ms may be desirable

3.3.4 Jitter

Jitter is the measure of delay variation between consecutive packets for a given traffic flow. Jitter has a pronounced defect on RT-traffic such as voice and video. These real-time applications expect to receive packets at a fairly constant rate with fixed delay between consecutive packets. As the arrival rate varies, the jitter impacts the

application's performance. A minimal on mount of jitter maybe acceptable but as jitter increase, the application may become unusable. All networks introduce some jitter because of variability in delay introduced by each network node as packets are queued. However, as long as the jitter is bounded, QoS can be maintained.

RT-traffic are particularly disruptive to cause audible pops and clicks. As discussed above, the fundamental architecture of the IP network is different from circuit-switched networks, so is variance in propagation jitter. As we know, audio encoding/decoding is a basic synchronous process, which the analog signal in sampled at prescribed intervals, the samples (or their encoding) are transmitted, and the received samples are clocked out to reproduce the analog signal.

To make matters worse, individual packets of audio may experience more or less delay as they travel the network. This causes variations in the inter-arrival time. Some packets may experience "infinite delay" (be lost) or such long delay that they become useless or stale.

The main technique for dealing with the jitter buffer and some amount of audio information (PCM samples, say) is buffered up before playing the samples out. If an incoming packet is delayed somewhat, then the information already in the buffer can be played out until either the buffer empties or the information arrives. Because the packet are presumable coming at a fixed rate, when one or more packets arrived late, some subsequence packets should arrive "early" (or on time) to refill the buffer. If the buffer runs dry, some type of audio fill-in must be supplied. As network jitter gets worse, the size of the jitter buffer must be increased to avoid too many under-runs. Unfortunately, the use of a jitter buffer introduces a delay proportional to the buffer's target size. Some products (e.g., Microsoft NetMeeting) create an extremely large jitter buffer on the order of 200-300ms in order to handle jitter from virtually any type of connection, which introduces a "fixed" delay degradation of the conversation.

Generally, Packets of multimedia data may be discarded due to delays beyond tolerable delay limits or due to network errors. The amount of latency and jitter that can be tolerated is strictly limited for some applications. ABR traffic is such as traditional data traffic. CBR traffic such as output from speech or video codecs in LAN TV can use circuit-switched as well as packet-switched network solutions providing low, average and peak traffic transfer capabilities having statistical quality control. CBR traffic produced by traditional codecs generates constant stream of bits requiring circuit-switched network with guaranteed minimum bandwidth.

3.3.5 Packet Loss

Loss can occur due to errors introduced by the physical transmission medium. For example, most landline connections have loss as measured in the Bit Error Rate (BER). However, wireless connections such as satellite, mobile, or fixed wireless networks has a high BER that varies due to environment or geographical conditions such as fog, rain, RF interference, cell hand off during roaming, and physical obstacles such as trees, buildings and mountains. Loss can also occur when congested network nodes drop packets. Because congestion has a direct impact on packet loss, congestion avoidance mechanisms are often deployed.

Summary

QoS is the key problem for RT-traffic transmission over IP-based network. As the end-user expectations or traffic QoS requirements, we can categorize traffic to four different traffic categories: interactive applications (CBR, RT-time traffic), responsive applications (VBR, nearly RT-time traffic), Timely application (ABR) and Network control applications. The QoS of interactive applications ask minimal delay and delay variation. The respond applications require relatively low packet delay, jitter, and loss. Timely applications (ABR) require lowest QoS for time delay within a reasonable bound. For QoS of network control applications, they require a relatively low amount of delay. Jitter can be negligible.

CHAPTER 4

QOS OF 802.11 WLAN

As discussed in chapter 2, WLAN has different characteristic and operation model against wired network. In chapter 3, after discussed some QoS guarantee technologies in wired network, we can find that these technologies are based on specific route path. It is impossible in the WLAN. As we know, 802.11 WLAN covers the PHY layer and the MAC layer. The QoS guarantee can be considered from both of them.

For the PHY layer, the basic design method is bandwidth enhancement. 802.11b, a, g, j and n [24][25][26][27] is based on the PHY layer. In this thesis, we focus on the MAC layer of the 802.11 WLAN. MAC layer is the most important research task to explore the extensibility of the technology to real time traffic over WLAN. The research focuses on Medium Access technologies. Before consideration of QoS of Real-Time traffic, the Medium access based on CSMA/CD is very perfect for common traffic transmission.

4.1 Medium access Mechanism

The target of MAC is to better share a single channel by multiple independent data sources. One of the development research tasks for shared network is the design of the multiple access protocol. The key measures for a MAC protocol for traffic is that whether achieve high channel utilization under high load with low access delay. In fact, different MAC protocol results to different utilization and delay. That is the design rule of the protocol that controls each host accessing the channel. It is also the retransmission schedule goal after an unsuccessful access attempt. In the earlier IEEE802.11 standard for data communication, the protocols are always DCF (Distributed Coordination Function) and based on CSMA/CA and PCF (Point coordination Function) for infrastructure WLAN.

4.1.1 CSMA/CD & CSMA/CA

CSMA/CD

Unlike the human ear that is much better at picking out a voice from among multiple simultaneous speakers, any electronic receiver in Ethernet at selecting one signal from among many cannot do. Thus every local area network using a shared medium design must incorporate a contention management time [28]. No matter how many devices share the medium, they have to wait their turn. But there are two problems: when is the appropriate time they can speak and how long they must wait to avoid collision. The CSMA/CD designs a procedure approach to solve it.

Carrier Sense (CS) All stations listen to the cable continuously to determine whether the Ethernet is currently in use. If the medium is idle, the device can transmit.

Multiple Access (MA) It means the medium supports many users at the same time [29]. However, there is a more powerful design element than might be apparent at the first. Networks degrade (or experience reduced performance) in different ways, depending on their design. Another, there is a potential problem. If all those attached devices need to transmit lots of data, great congestion will ensure. However, that might be acceptable for some networks-not the congestion, but the threat of it. Some networks are built to support primarily word processing, an occasional file transfer, and perhaps some electronic mail. With such networks, the light load per machine allows many devices to be present without degrading the network. Network that will experience much traffic per device need to be engineered more carefully, perhaps by being divided up into multiple bridge segments.

Collision Detection So ***CSMA/CD*** must use the collision detection mechanism, which requires all stations listen whilst transmitting. If the medium is idle, it can transmit. If busy, the stations listen for idle, and then it transmits. If collision detected, stop sending data and start sending a random jam to ensure all stations detect the collision, and then wait for a random retransmission delay as the Binary exponential back off Algorithm.

The Binary exponential back off Algorithm is R_t=RND (0~2^n)*timeslot, n=10 for $n \geq 10$[30].

For the I-time attempt, choose a random number in the interval and wait that number of slot and try again if collision again. When the collisions occur more than 16 times, the transmission is given up.

CSMA/CA

CSMA/CA is based on CSMA/CD. But its mechanism is different with CSMA/CD. However, WLAN has itself characteristic different with Ethernet. Since the radio signal level has a wide dynamic range, collision detection is not possible in the same way as with the wired equivalent. Collision detection limits the damage of a collision, improving the channel utilizing and access delay figures. Without it, collision must be avoided by randomizing the transmission time at moments of high collision probability to obtain acceptable performance of traffic over WALN. Without collision detection, colliding transmissions continue to their conclusion despite the inevitable rejection of the contending frames. So for avoiding collision [31], CSMA/CA is different than CSMA/CD:

1) Before transmission, the station must give how long must be spent during this transmission. This give any potential user an idea of how long to wait before they have an opportunity to transmit.

2) Stations cannot transmit until the previous announced transmission elapsed.

3) Stations are unaware whether their voices are received while they are transmitting unless they receive acknowledgement from the receiver when they are done.

4) If two stations happen to start transmitting at the same time, they are unaware they are speaking over each other. The transmission station knows collision happened because they do not receive acknowledgement.

5) The stations wait a random amount of time and attempt to retransmission again, should they not receive acknowledgement. This is smaller compare to the CSMA/CD.

There are problems that treating another transmission as a collision leads to the effect exist in CSMA/CA and CSMA/CD, know as "channel [32] effect", whereby a successfully transmitting host is able to send subsequent frames un-contended. The currently transmitting station will reset its contention window to the minimum giving it a significant advantage over other stations that are backing off their contenting windows. The result on channel efficiency is not detrimental as a station that 'wins' a round of contention can transmit for longer without having to waste bandwidth in other contests; but the effect on delay and delay variance is extremely detrimental.

4.1.2 DCF

There are two-access mechanisms in IEEE802.11 MAC sub-layer. They are DCF and PCF. DCF is based on CSMA/CA protocol, and it is for asynchronous data transmission (or best-effort service). For the topology of wireless LAN, DCF be can used for both ad hoc and infrastructure modes.

DCF basic mechanism

The DCF is almost identical to the basic CSMA/CA but incorporates ideas from earlier wireless multiple access protocols MACA [33] and MACAW [34]. For the CSMA/CA in WLAN, a station wants possess the medium to transmit data must sense whether the medium is idle. In CSMA/CD, it is detected by the electric power variation. However, in DCF, it uses the virtual carrier-sense function, the network allocation vector (NAV) instead. The NAV is a timer that is updated by data frames transmitted on the medium. For example, in an infrastructure BSS, a station send a frame to the receiver by the broadcast style. All stations in the BSS can receive this frame; the frame contains a duration field. This duration covers the whole time of that transmission and the expected acknowledgment. It indicates that the duration is the carries time. Then every station can 'sense' the carrier.

DCF Process

In DCF operation, a station wanting to transmit a frame must wait a specific amount of time after the medium becomes available. This time value is the DIFS (DCF inter-frame space). Once the DIFS interval elapses, the medium becomes available for station access

contention. However, if there are many stations in this BSS, because of the waiting time are the same, it causes collisions inevitably. For this reason, DCF uses a random back-off timer. The random back-off algorithm selects a value from 0 to the contention window (CW) value randomly. Different vendor sets the default CW values. The values are stored in the station NIC. The range of the values for random back-off start at 0 slot times and increment up to the maximum value, which is a moving ceiling start at CW_{min} and stopping at a maximum value known as CW_{max}. A station randomly selects a value between 0 and the current value of the CW. The random value is the number of 802.11 slot times the station must wait during the medium idle CW before it may transmit. A slot time value derived from the PHY based on RF characteristics of the BSS. The mechanism is shown in Figure 4-1

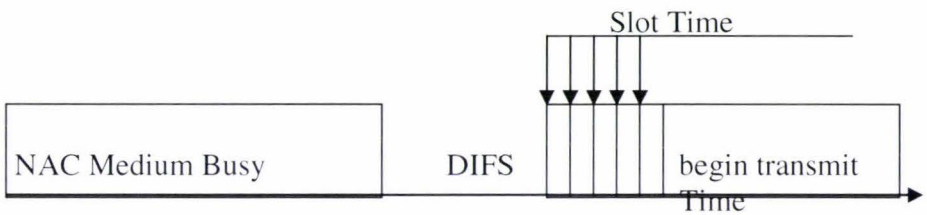


Figure 4-1 Frame Transmission after Random Back-off

The DCF Medium Access Process is shown in figure 4-2, PDF use freezes the back-off timer instead of incrementing the back-off time, when a station senses the carrier in back off mode. Use of this idea not only removes the channel capture effect but ensure that a station that has been waiting for longer is more likely to reach the end of its back-off period and transmit than newly contending [35] stations.

The DCF specifies an acknowledgement based loss detection and retransmission strategy, which is applicable to unicast traffic only. No loss repair mechanism is provided for multicast traffic.

In fact, The CW (Contention Window) is set to a recommended minimum value of 16, and doubled upon timeout when waiting for acknowledgement. For multicast, as no acknowledgement is used, the minimum value is the only value the contention window will be set to, and so is very important parameter. By increasing the size of the minimum contention window, delay should decrease.

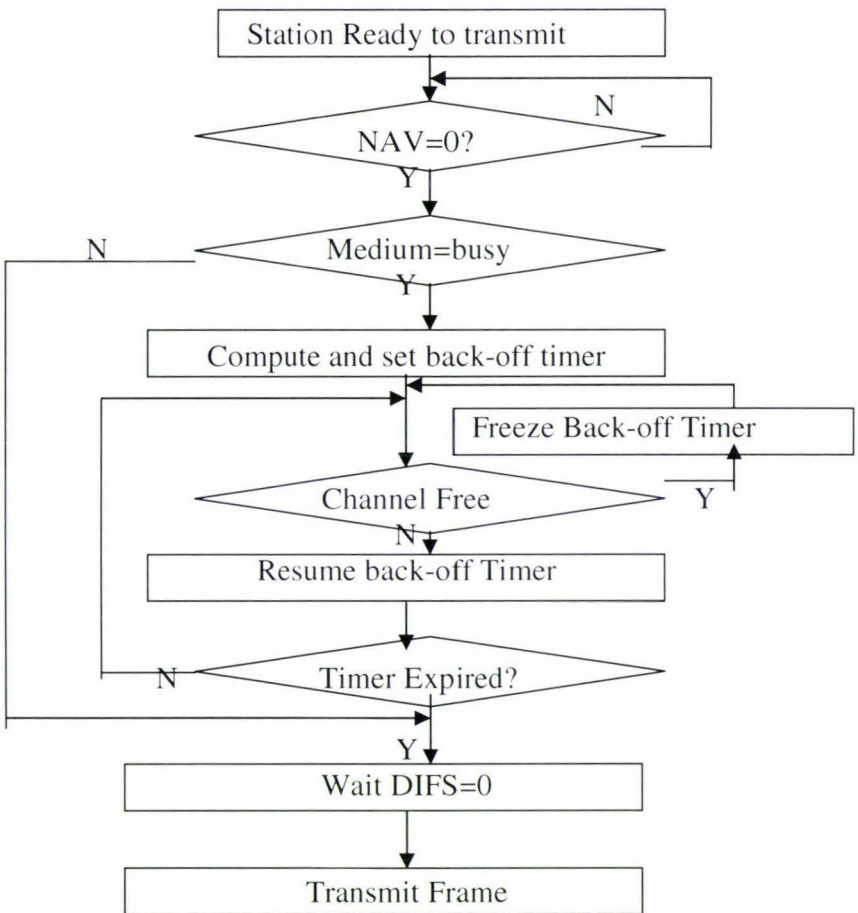


Figure 4-2 the DCF Medium Access Process

4.1.3 PCF

PCF is another access control protocol. PCF uses a central-controlled polling method to support synchronous data transmission. Unlike DCF, the PCF is only applicable in wireless network with access point, i.e. an infrastructure network [8]. In ad hoc mode, all wireless stations within the communication range can communicate directly with each other, whereas in infrastructure mode, an AP is needed to connect all stations to a DS (Distribution System), and each station can communicate with other through AP. For

real-time traffic, PCF provides contention-free frame delivery to and from AP. However, for this reason, it has to increase protocol header in PCF mechanism. The reason made PCF not widely deployed.

PCF basic mechanism

The concept of CFP (Contention Free Period) is deployed by PCF. The CFP is the window of time for PCF operation. The CFP begins at set intervals following a beacon frame containing a delivery traffic indication map (DTIM) information element. The network administrator determines the frequency of CFPs. Once the CFP begins, the AP assumes the role of the PC (and as such, PCF just can be in infrastructure BSSs). Each station in the BSS sets its NAV to the CFPmaxDuration value. This value is included in the CF parameter set information element. The CFPMax duration defines the time value that is the maximum duration for CFP. The PC (Point Controller) can end the CFP intervals, and beacon frames sent during the CFP contain the CFPDurationRemaining field to update station NAVs of the remaining duration of the CFP. Figure 4-3 depicts the CFP and contention period (CP) as a function of time.

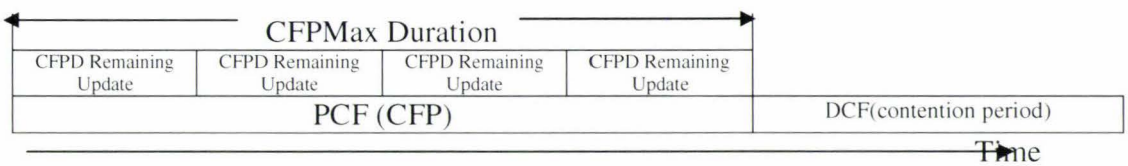


Figure 4-3 the VFP and CP Timeline

Different from DCF operation, PCF administers all stations in the BSS by AP without allowing them to freely access the medium and transmit data. Station can only send data (one frame at a time) when the PC polls them. The PC can send frames to stations, poll stations for frame transmission, acknowledge frames requiring MAC-level acknowledgements, or end the CFP.

PCF process

When the CFP begins, the PC must access the medium in the same manner as a DCF station. The difference with DCF stations is that, the PC attempts to access the medium after waiting an interval of time known as the priority inter-frame space (PIFS). The PIFS

interval is one slot time longer than the SIFS interval and one slot time shorter than the DIFS interval, allowing PCF station to access the medium before DCF stations yet still allowing control frames, such as acknowledgment frames, to have the highest probability of gaining access to the medium.

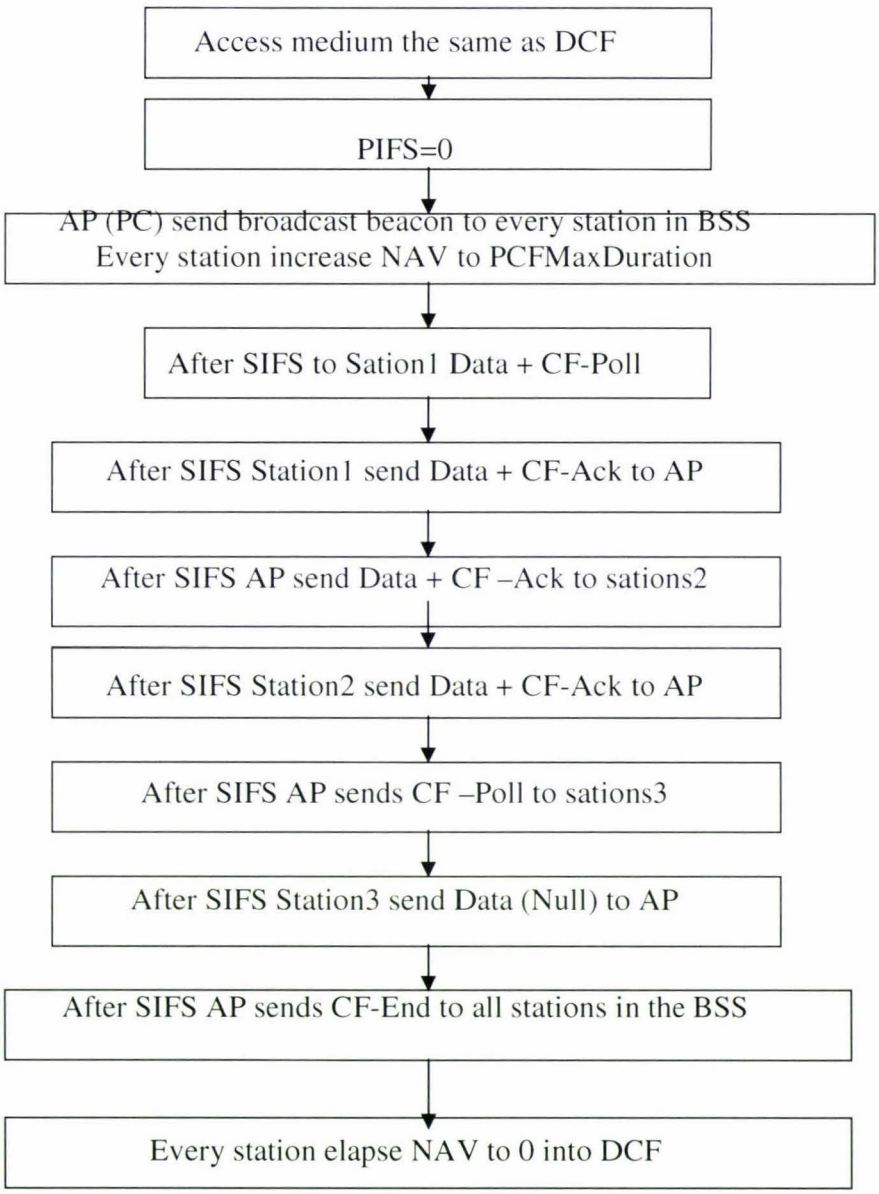


Figure 4-4 the PCF Process

After waiting a PIFS interval, the PC sends the initial beacon frame contain the CF parameter information element. The PC waits for one SIFS interval subsequent to the beacon frame transmission and then sends one of the following to a CF-Pollable station:

A data frame, a poll frame (CF-poll), a combination data and poll frame (Data +CF-poll), a CFP end frame (CF-End). If the PC has no frame to send and no CF-Pollable stations to poll, the CFP is considered null, and immediately following the beacon frame; the PC sends a CF-End frame terminating the CFP.

The process of PCF is shown in figure 4-4, we assume that there are stations in a BSS, and station 1 wants to send a message to station2 and station 3 has not any massage to send.

Note in PCF process every station ignores NAV, when they send an Ack to the AP. The transmission of the additional polling and ACK message required by the PCF is optimized through piggybacking multiple messages in a single transmission. For example, the PC may append both ACKS of previous transmission to avoid waiting the inter-frame interval specified for individual frame transmissions.

4.2 MAC Layer operations

4.2.1 RTS/CTS

Another Mac-layer operation problem is the "hidden node". Two stations on opposite sides of an access point can both "hear" activity from an access point, but not from each other, usually due to distance or an obstruction. To solve this problem, 802.11 specifies an optional RTS/CTS (Request to Send/Clear to send) protocol at the MAC layer. When this feature is in use, a sending station transmits an RTS and waits for the AP to reply with the CTS. Since all stations in the network can hear the AP, the CTS cause them to delay any intended transmission, allowing the sending station to transmit and receive a packet acknowledgement without any chance of collision. Since RTS/CTS adds additional over-head to the network by temporarily reserving medium, it is typically used only on the largest-size packets, for which retransmission would be expensive from a bandwidth standpoint.

4.2.2 802.11 frame fragmentation

802.11 MAC provides for another operation method for the large size packets. It is the Frame Fragmentation. A value called the fragmentation threshold specifies that frames over a specified size should be divided into multiple transmissions. The frame header contains a sequence control field that shows the order of the fragments. Fragments constituting a frame are transmitted immediately after one another without any contention for the medium. Each fragment has its own CRC (cyclic redundancy code), and an individual ACK is transmitted for each fragment. The fragment transmissions are separated by the appropriate frame interval spaces. The transmission of a sequence of fragments is call a frame burst. If an error occurs on a fragment, subsequent fragments are not transmitted until the previous frame is acknowledged. The retransmission and back-off rules apply to fragmented frame transmissions. Duration information in the fragment and ACK frames sets the NAV. Broadcast and multicast frames are not fragmented even if their size exceeds the fragmentation threshold. Fragmentation is shown in Figure 4-5

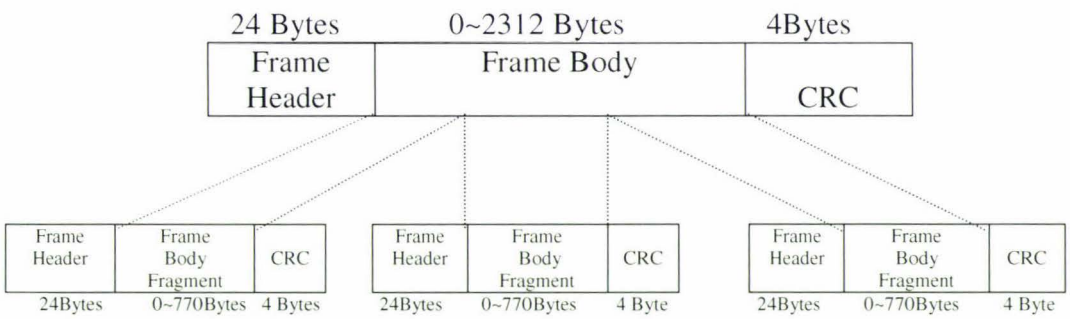


Figure 4-5 Frame Fragment

4.2.3 Station connectivity

There are three exchanges between the wireless stations and the AP when a wireless station joins a BSS. They are **the probe process**; **the authentication process** and **the association process**.

The probe process is used for the station to connect with the most appropriate AP at the situation where there are many APs for choosing.

In this stage, station can send the **probe request frame** on every channel it is allowed to use. The probe request frame contains two key fields: SSID element and support rates element. It gives APs information what services set the station belongs and data rates the station support respectively. Client station sends probe requests frames blindly, meaning they don't know anything about the APs they are probing for. When an AP receives a probe request frame that successfully passed a frame check sequence (FCS), it replied a **probe response frame**. There are some fields in the probe request frame: Timestamp field, it is for the value of the TSFTIMER of the frame sender. It is used to synchronize the clock of the client station to the AP. Beacon interval field; it is for the number of time units (TUs) between beacons. ATU is 1024 microseconds. Capability information field, it is for Mac and PHY layer capabilities. SSID element; the AP is configured with the SSID. Support rates element; the AP support data rate. PHY parameter set element; either frequency hopping or direct sequence. This element provides PHY-specific information to the client station. When the client station receives the probe response frames, the station compares them and determines which AP to associate with. The compared content always contains matching SSIDs, signal strength, and vendor proprietary extensions.
The authentication process is for WLAN security. 802.11 authentication consists of two authentication modes: open authentication and shared-key authentication. 802.11 authentication is oriented around device authentication and determines whether the device is allowed on the network. Authentication is simplified to an authentication request and an authentication response.

The association process is initiated by a wireless station with an association request frame containing the capability information of the client and completed by the AP in an association response frame. The association response indicates success or failure as well as a reason code.

4.3 802.11 MAC Frame Format

There are three types of 802.11 MAC frames; there are control frames, management frames and data frames. They are all various frames based on the general MAC frame.

4.3.1 General MAC frame format

We can see the general 802.11 MAC frame format from figure 4-6. The Address 2, 3, and 4, Sequence Control, and Frame body fields are not found in every frame. The frame control field is 2 Bytes in length, and it contains basic frame control information, (shown in Figure 4-7) including protocol version, the frame type (data, MAC control, or MAC management) and subtype, if the frame is originated from or is bound to the DS and if the frame is encrypted. The duration/ID field normally indicates the duration of the remainder of a frame exchange sequence and is used to control the virtual carrier sense mechanism as previously described.

The address fields, if present, contain one of the following 48-bit IEEE 802 Link Layer address: Destination Address, Source Address, Receiver Address, Transmitter Address, and Basic Service Set ID (BSSID). For infrastructure networks, the BSSID is the Link Layer address of the AP. For ad hoc networks, the BSSID is a random number generated at the time the ad hoc network is formed. The receiver, transmitter, and BSSID addresses are the MAC addresses of stations joined to the BSS that are transmitting or receiving the frame over the wireless Ethernet. Destination and Source address are the MAC addresses of stations, wireless or otherwise, that are the ultimate destination and source of the frame. In those cases where two address are the same (for example, the Receiver station and the Destination station are one and the same), then a single address field is used. Four address fields are present only in the uncommon case where the DS is implemented with an 802.11 wireless Ethernet, and only for frames traversing the DS. A more typical case involves a frame origination from a wireless station in an infrastructure BSS that is bound for a station on a wired network such as an IEEE 802.3 wired Ethernet. In this situation, the Address 1 field contains the BSSID, the Address 2 filed contains the address of the source/transmitter station, the Address 3 field contains the address of the destination station, and the Address 4 field is not present. Including both the BSSID and the Destination Address (or Source Address for frames flowing to the BSS) in the frame avoids requiring the AP to maintain a list of MAC addresses of stations that are not in the BSS.

The Sequence Control field is 2 bytes in length, and it contains the Sequence Number and Fragment Number sub-fields. Receiving stations use this field to properly reassemble multi-fragment frames and to identify and discard duplicate frame fragments.

The Frame Body is an optional field that contains the MAC frame payload. For 802.11 MAC management type frames, the Frame Body contains information elements that are specific to the subtype. The FCS field contains a 4 Byte CRC. The CRC calculation includes the entire MAC frame field.

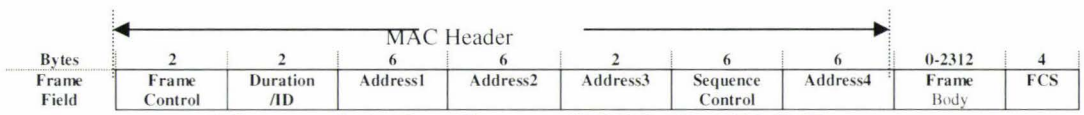


Figure 4-6 the General 802.11 MAC Frame

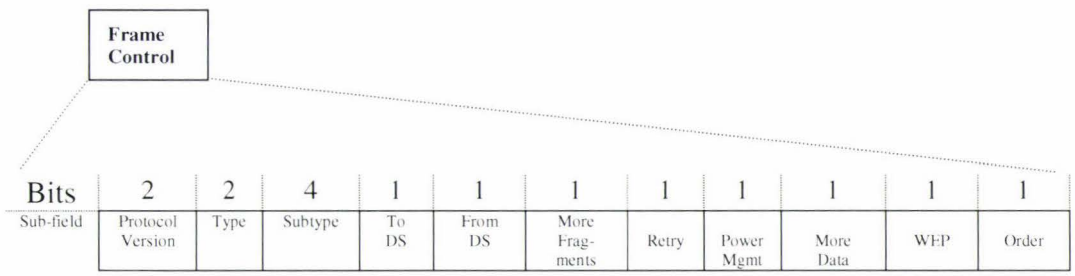


Figure 4-7 the Frame Control Sub-fields

4.3.2 802.11 Control frame

There are six kinds of frame in 802.11-control frame; they are PS-Poll (Power save poll), RTS, CTS, ACK, CF-End (Contention-free End) and CF-End+ CF- Ack (CF-End + contention-free acknowledgement)

The PS-Poll frame is indicator to the AP that a wireless station in power save mode is requesting that any frame buffered on the AP be delivered.

The RTS Frame is the request to reserve the wireless medium as a part of the 802.11 medium access mechanism. The frame is shown in Figure 4-8, the duration sub-field is

the time required for the station's frame exchange to take place. It includes the time to transmit the RTS frame, the time to receive the CTS frame (including the SIFS interval), the time to transmit the data frame (including the SIFS interval), and the time to receive the ACK frame (including the SIFS interval). It's measured in microseconds. RA (Receiver address) field indicates that the MAC address of the intended recipient of the frame. TA (Transmitter address) indicates that the MAC address of the transmitter of the frame sender.

Frame Control 2 Bytes	Duration 2 Bytes	RA 6 Bytes	TA 6 Bytes	FCS 4 Bytes

Figure 4-8 RTS Frame

The CTS frame is response to an RTS frame. It is an indication to receiving station that the medium has been reserved for the specified duration; it is constructed very similar as RTS without TA sub-field. The difference with RTS, the duration field is that for the value obtain from the duration field of the immediately previous.

The ACK Frame is the same architecture as the CTS. Comparing with RTS, the duration field is for ACK frame is usually 0 because the frame it is acknowledging includes the transmission time for the SIFS interval and the ACK frame in its duration field.

The CF-End and CF-ACK frames are specific to PCF operation. They indicate the end of the contention-free period, and the CF-END +CF- ACK also include an acknowledgement of the last frame received by the PC. Figure 4-9 shows the frame format. Where, the duration set to 0. The receiver address field is the destination MAC address of the intended recipient of the frame. In the case of the CF-End frames, it is the broadcast MAC address because every station set should receive the notification. The BSSID is the MAC address of the AP.

Frame Control 2 Bytes	Duration 2 Bytes	RA 6 Bytes	BSSID 6 Bytes	FCS 4 Bytes

Figure 4-9 CF-END and CF-END+CF-ACK Frame

802.11 Management Frame Field and Elements

802.11 management frames leverage field from the generic MAC frame detailed earlier and also utilize data structures

4.4 QoS LIMITATIONS OF 802.11 WLAN

The QoS performance of RT-traffic is challenge for the MAC layer in 802.11WLAN. In particular, wireless links have specific characteristic such as high loss rate, bursts of frame loss, high latency and jitter. Furthermore, wireless link characteristics vary over time and location. There are several ways to characterize QoS in WLANs such as parameterized or prioritized QoS as proposed in [49].

Generally, QoS is the ability of a network element (such as an application, a host or a router) to provide some levels of assurance for consistent network delivery. Parameterized QoS is a strict QoS requirement that is expressed in terms of quantitative values, such as data rate, delay bound, and jitter bound. In a traffic specification (TSPEC), these values are expected to be met within the MAC data service in the transfer of data frames between peer stations. Prioritized QoS is expressed in term of relative delivery priority, which is to be used within the MAC data service in the transfer of data frames between peer stations. With a prioritized QoS scheme, the values of QoS parameters such as data rate, delay bound, and jitter bound, can vary during the transfer of data frames, without renegotiating the TSPEC between the station and the AP. According to the above definitions of QoS, we can know the limitation of DCF and PCF.

DCF is designed for best-effort service; it has been superseded by other standard. Generally, RT-traffic require specified bandwidth, low delay and jitter, but can tolerate some loss. In DCF, all the stations compete for the transmission with the same priorities. In other words, it cannot guarantee RT-traffic for high-priority RT-Traffic.
PCF has been designed to try to support RT-Traffic. However, the performance of the RT-traffic under this mechanism is very poor. The reasons are: because the central

polling scheme is questionable, all the communication between any stations must be through the AP, so channel resource must be wasted. The situation is serious in heavy loading [37] [38] [39]. The other reason is that: the cooperation between CP and CFP modes may lead to unpredictable beacon delays [9] [10]: the AP schedules the next beacon transmission at the next TBTT (Target Beacon Transmit Time), the beacon frame maybe delayed. In the current 802.11 legacy standard, stations are allowed to transmit even if the frame transmission cannot terminate before the upcoming TBTT [40]. The duration of the beacon to be sent after the TBTT defers performance of multimedia application. In the worst case, the maximum delay of a beacon frame can be 4.9 ms in 802.11a, and the average delay of a beacon frame can reach up to 250***ms*** [9]. The final reason is difficult to predict the transmission time of a polled station. Because the frame's length is between 0~2304 Bytes, maybe have a variable transmission time. Meanwhile, the PHY rate of the polled station can be changed according to the varying characteristics of the channel.

Summary

The IEEE802.11 MAC layer protocol is based on two mediums access coordination functions. They are the basic DCF and the optional PCF. For the QoS performance, the mechanisms of the DCF and PCF are perfect for the good convention data transmitting. However, because the RT-traffic requires specified bandwidth, low delay and jitter, their limitations are obvious.

DCF cannot provide priority for RT-traffic; it means RT-traffic must share the limited channel resource with other traffic as the same priorities. The result is high delay and jitter.

PCF use the Poll mechanism with consideration for the RT-Traffic QoS performance. However, the QoS performance of the RT-traffic by using PCF is still poor. The mechanism's limitation is due to the channel resource wasted, unpredictable beacon delays and the difficult prediction of the transmitted time.

In short, neither DCF nor PCF can provide good QoS performance for RT-traffic over 802.11 WLAN. More and more researchers proposed new and modified schemes for the RT-traffic QoS performance enhancement over the MAC layer based on DCF and PCF limitations. The advantage and disadvantage of these schemes' mechanisms are analyzed in the next chapter.

CHAPTER 5

ENHANCEMENT OF QOS OVER 802.11 WLAN

Because the original IEEE802.11 can't guarantee the enough QoS for the RT-traffic, many schemes for RT-traffic over the 802.11 WLAN has been proposed to enhance the RT-traffic's QoS performance. The IEEE 802.11 working group was consided a new standard to enhance the QoS of the RT-traffic over the WLAN. It is the 802.11e. *T*he standard is published on 12 November 2005. All proposals for the QoS performance enhancement support latency-sensitive application, ensure the reasonable delay, jitter and throughput, meanwhile, some proposal consider the reasonable fairness for those latency-insensitive traffic (such as best-effort, Http, etc).

5.1 CHALLENGES FOR RT-TRAFFIC QOS IN 802.11 NETWORK

The WLAN is a shared, half-duplex medium, while most wired Ethernets are full duplex. This characteristic decides that MAC mechanisms just allow for only one station on the medium at a given time, whether it's the access point (AP) or workstation. While the wired Ethernet, say in particular 802.3x full-duplex operation, create a point-to-point link between Ethernet, it means the wired LAN medium theoretically can get two times normal bandwidth. Furthermore, a station can transmit the data without contending with the station on the other side of the link. Of course, it also depends on the wired network's configuration. However, in 802.11 WLAN, these contentions occur not only between the AP and workstation but also among the workstations. So PCF protocol tries to reduce the contention by the poll mechanism. It's available in very low client count BSSs data. However, it was found to cause more degradation in overall throughput than the normal contention-based access in DCF. We can recall there are three limitations of the PCF we just discussed in the pervious chapter, they are: (1) every connection must be through the AP result to network resource wasted. (2) The beacon frame maybe delay by the 802.11 legacy standard that station are allowed to transmit even if the frame transmission cannot terminate before the upcoming TBTT. It can result to the unpredictable beacon delay. (3)

It is difficult to predict the transmission time of a polled station. The reason is that the frame has different length, meanwhile the PHY rate of the polled station can affect throughput. For the DCF, the protocol supports RT-traffic transmission over the WLAN with the worse QOS. Because DCF is not priority mechanism, so the RT-traffic have the same opportunity to transmit with those best-effort traffic. The time delay is the same. However, the RT-traffic is a kind of latency-sensitive application, so they cannot ensure the RT-traffic QoS performance over the 802.11 WLAN.

For enhancing the RT-traffic QoS performance over the WLAN, many kinds of QoS schemes were proposed for IEEE 802.11 recently. Those enhance schemes use the new algorithms in the MAC layer to achieve the better RT-traffic QoS performance over the WLAN. They define the parameters for how a station or a flow should access the wireless medium. We can classify those schemes into DCF-based and PCF-based enhancement schemes.

5.2 DCF-based QoS enhancement schemes

DCF-based means the QoS enhancement schemes based on DCF protocol. As the DCF can be used in both ad-hoc and infrastructure topology in WLAN, recent research works mainly focus on the DCF-based enhancement schemes. We can classify them by whether the schemes provide only one priority for one station or introduce multiple priority queues in each station. We can call them station-based or queue-based schemes.

5.2.1 DFS schemes

The distributed fair scheduling (DFS) was introduced in Reference [50] by the Microsoft Communications Projects Research Group. DFS is a schedule for the fairness allocation. With fair schedule, it means to allocate bandwidth in reasonable proportion to weights of the packet flows sharing the channel. Meanwhile, it must ensure the priority for RT-traffic. An interesting feature of DFS is that it can be implemented with simple modification to the IEEE802.11 standard. DFS is based on one station to ensure the priorities and the fairness for enhanced IEEE 802.11 standard by self-clocked faire queue model (SCFQ [64]). Fair queuing is one of the important factors for QoS over the WLAN.

Much research has discussed the "fair queuing" algorithms for achieving a fair allocation of bandwidth on a shared medium.

GPS

SCFQ model comes from the GPS (Generalized Processor Sharing) discipline, which is the typically faire queuing algorithm. Following [51], the process is showed below:

GPS's formula is:

$$\frac{W_i(t_1,t_2)}{W_j(t_1,t_2)} \geq \frac{\phi_i}{\phi_j}, \forall_j \qquad (5\text{-}1)$$

Where, ϕ_i is the weight associated with flow i (i =1, 2,.....n)
$W_i(t_1,t_2)$ is the amount of the flow i traffic.
ϕ_j is the weight associated with flow j (j =1,2,.....n)
$W_j(t_1,t_2)$ is the amount of the flow i traffic.

If the flow i , j is back logged in interval $[t_1,t_2]$, and the condition formula (5-1) is true, we can say it is the fair queuing. From formula (5-1), we can find that the fairness is valid regardless how small the interval $[t_1, t2]$ is, however the algorithm is for centralized network, it means it just for the infrastructure topology of the WLAN.

SCFQ Mechanism

SCFQ can be in a ***distributing*** manner. The two improvement based on GPS discipline are:

1) SCFQ implement distributed fair schedule that can also be extended to other fair queuing algorithms, such as Start-time Fair Queuing (SFQ) [42].

2) As the SCFQ, the distributed implementation behaves differently against the GPS.

The SCFQ mechanism is shown below briefly:

Firstly, we just in a virtual clock is maintained by the central coordinator, SCFQ use some variations explained below:

$v(t)$ denotes the virtual time at real time t
P_i^k denotes the kth packet arrives on flow i
A_i^k denotes the real time at which packet P_i^k arrives
L_i^k denotes size of the packet P_i^k

S_i^k denote the start tag

F_i^k denote the finish tag

Let $F_i^0 = 0, \forall_i$

1) On arrival of packet P_i^k, the packet is stamped with tag S_i^k

$$S_i^k = \max\{v(A_i^k), F_i^{k-1}\} \qquad (5\text{-}2)$$

Also, F_i^k the finish tag of P_i^k is calculated as

$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i} \qquad (5\text{-}3)$$

2) The SCFQ sets a virtual clock initially; the virtual clock is set to 0, i.e., $v(0) = 0$. The virtual time is updated only when a new packet is transmitted. When a packet begins transmission on the output link, the virtual clock is set equal to the finish tag of the packet.

3) Packets are transmitted on the link in the increasing order of the finish tags. Ties are broken arbitrarily.

In (5-2), (5-3), we also used other algorithms, such as SFQ, WFQ, WF2Q, etc to calculate the start and finished tags [42] [43] [44] [45].

DFS (Distributed Fair Scheduling) Mechanism

DFS is based on SCFQ; DFS transmit the packet whose finished tag is smallest, as well as updating the virtual time. A distributed mechanism uses the back-off interval mechanism from IEEE802.11 MAC to do the smallest finished tag. Then let the back-off interval that is proportional to the finish tag of packet to be transmitted. Following [42],

$$F_i^k = S_i^k + Scaling\_Factor * \frac{L_i^k}{\phi_i} = v(f_i^k) + Scaling\_Factor * \frac{L_i^k}{\phi_i} \qquad (5\text{-}4)$$

Where, each node maintains a local virtual clock $v_i(t)$, $v_i(0) = 0$. P_i^k denotes the k-th packet arriving at the flow at the node i on the LAN. Each transmitted packet is tagged with its finish tag. When at time t node i hears or transmits a packet with finish tag Z, node i sets its virtual clock v_i equal to $\max(v_i(t), Z)$. When packet P_i^k reaches the front of its flow at node i, the packet is stamped with start tag S_i^k, calculated as, $S_i^k = v(f_i^k)$ where f_i^k denotes the real time when packet P_i^k reaches the front of the flow. In (5-4), we know, F_i^k is the finish tag, where appropriate choice of the *Scaling_Factor* allows us to choose a suitable scale for the virtual time.

The next step of DFS is that to choose a back-off interval such that a packet with smaller finish tag will ideally be assigned a smaller back-off interval. This step is performed at time f_i^k. Specifically, node i picks a back-off interval B_i for packet P_i^k as a function of F_i^k and the current virtual time $v_i(f_i^k)$ as follows:

$$B_i = [F_i^k - v(f_i^k)]slots \qquad (5\text{-}5)$$

From (5-4), (5-5), we can get formula (5-6)

$$B_i = [scaling\_factor * \frac{L_i^k}{\phi_i}] \qquad (5\text{-}6)$$

From (5-6), if we randomize the B_i value and use ρ denotes the random variable with mean 1, we can get formula (5-7):

$$B_i = [\rho * B_i] \qquad (5\text{-}7)$$

Then we can use (5-7) to handle the collisions.

In fact, from formula (5-4) to (5-7), we just discuss DFS in linear scheme. From (5-5), (5-6), we can know that in the linear scheme, back-off interval B_i is a linear function of finish tag, and directly proportional to (1/flow weight). This can make the back-off intervals large, when flow weights are small. So DFS use exponential mapping scheme to resolve the limitation. By the use exponential mapping scheme, following [45] and formula (5-5), (5-6), (5-7), we can get the formula (5-8):

$$B_i = \gamma(\Delta) = \begin{cases} \Delta & if\ \Delta, Threshold \\ [Threshold + (1 - e^{-k_2 * (\Delta - Threshold)})] & otherwise \end{cases} \qquad (5\text{-}8)$$

Where,

Δ denotes the back-off interval obtained in (5-7) using the linear scheme,

$r(\Delta)$ is the another function to obtain the actual back-off interval B_i to be used for medium access.

K_1, K_2 are constant parameters.

Generally, Distributed Fair Scheduling (DFS) to allocate wireless resources in proportion to the priorities of the packet flows sharing the wireless channels by SCFQ. This mode focuses on the calculation of the value of the back-off interval based on an exponential function instead of a simple linear function that is currently adopted by the IEEE 802.11 standard. This Exponential Mapping Scheme overcomes the shortcomings of the IEEE

802.11 standard that could not perform fair resource allocation among different multimedia flows on short time scales. In DFS, the back-off process is always initiated before transmitting a frame. Contrary to 802.11 DCF, the back-off interval is calculated as a function of packet size and weight of the station, which can be linear, exponential, or square-root function. As this scheme, stations get low priority by low weights generate longer back-off intervals than those with high weights, meanwhile Fairness is achieved by considering the packet size in the calculation of the back-off interval; a new back-off interval is calculated using the original back-off algorithm of the IEEE 802.11 DCF. Simply to say, Back-off periods depends on packet size by Self-Clocked Fair Queuing model. The pros side of this scheme is (1) DFS has good throughput utilization. (2) DFS can provide the priority mechanism. (3) DFS can avoid the contention better (4) DFS can ensure the good fairness. However, the high complexity implementation of this scheme limits deployment.

5.2.2 DENG schemes

DENG proposed a scheme that provides the priority based on IEEE802.11 DCF mechanism. It uses different IFS and different back-off algorithm to achieve the priority base on DCF.

J. Deng found the probability of packet collision in a mobile network could be high as 60% [46], although the RTS/CTS were used in wireless LAN. Like this situation, the RT-traffic cannot ensure QoS absolutely. The Deng [47] scheme was proposed to enhance the RT-traffic QoS. This scheme provides different IFS lengths. This difference is in PIFS or DIFS meanwhile define the different random back-off time. Deng scheme provides a kind of algorithm to combine different PIFS and DIFS with random back-off time. The scheme classifies the applications into four class properties. The first class is C_1=PIFS+B_1, where $B_1 \le Integer[rd \times 2^{2+i}/2]$, where rd is a uniform random variable in (0,1), the second class is C_2=PIFS+B_2, where $B_2 \le Integer[2^{2+i}+[rd \times 2^{2+i/2}]$; the third class is C_3=DIFS+B_3, where $B_3 \le Integer[rd \times 2^{2+i}/2]$; the fourth class is C_4=DIFS+B_4, where $B_4 \le Integer[rd \times 2^{2+i}/2]$. Because of PIFS shorter than DIFS, from the time longer, we can know $C_4 > C_3 > C_2 > C_1$. A station uses PIFS and short random back-off time

algorithm gets the highest priority. Whereas, the station uses DIFS and long back-off time algorithm just get the lower priority. Because of this algorithm, the high priority stations have short waiting time to access the medium. Meanwhile, the collision happened, high priority stations have more chances to access the medium than the low priority. The scheme is very efficient to the latency-sensitive applications; RT-traffic can obtain the more change to transmission in the WLAN. However, when the high priority stations have none frame to transmission, the low priority stations still in long back-off time. It is not reasonable and wastes the large network resource. In fact, this is the early consideration for the application classification and gives different wait time. Later, it is adopted in EDCF of IEEE802.11e. Of course the later is very reasonable algorithm.

5.2.3 IACC schemes

IACC scheme was proposed by I. Aad and C. Castelluccia [48] in 2001. This scheme also provide priorities base on IEEE802.11 DCF access method. The scheme has three choice parts: as different priorities define the different contention windows to stations. However, this scheme performs well with UDP traffic while it performs badly with TCP traffic. The reason is that all TCP ACKs are sent with the same priorities, which affects the differentiation mechanism. The second part is that each of the schemes is that low priority traffic suffers as long as high priority frames are queued. Contrary to the first part, there is no back-off problem with TCP, but TCP ACKS also reduce the effects of service differentiation because all ACKs have the same priorities. With the final part, works well for TCP and UDP flows, each station is assigned a different can transmit more information per medium access than low priority stations. This mechanism is used to increase both transmission reliability and differentiation. However, the weakness of part three is the bad performance in a noisy environment; long packets are more likely to be corrupted than short ones, this result to the poor performance of the RT-traffic in WLAN.

5.2.4 Blackburst schemes

Blackburst is a kind of jam [49]. The length of the balckburst is determined by the time the station has waited to access the medium. It is calculated as a number of black slots. So the scheme is called Blackburst scheme.

Black burst scheme classify all stations into high priority stations and low priority stations. The main mechanism is for high priority the scheme imposes all high priority station, which want to transmit the frame with the equal constant intervals t_{sch}. Then the priority station sends the jam as t_{sch} to the medium, and calculates the jam's larger to decide which station can send the frame first. The step is:

1) When a high priority station wants to send a frame, if the medium is busy, the station waits for the medium to be idle for PIFS.

2) Then the station sends a jam to the channel. The jam is called Black burst. The medium is entering a black burst contention phase now. As the time of the station send the jam, the length of the black burst is different each other. The station determines the length of the black burst by the time that the station has waited to access the medium. The length of the black burst can be denoted by $B_i=N*S_i$, where S_i is the black slot, the N denotes the integer number.

3) After sending a B_i jam, the station waits for a short time to listen the medium whether idle. If the medium is idle, the station transmit frame immediately.

4) If the medium is not idle, the medium check which black burst is longer and win to enter whose contention period, and the shorter black burst station has to wait the longer one until the medium becomes idle again and enters another burst contention period.

5) When the station finished transmission of a frame, the station schedules the nest transmission attempt t_{sch} seconds in the future.

For the low priority stations, the Black burst scheme defines the ordinary CSMA/CA access method of IEEE802.11 for them. So the scheme has the nice effect that real-time flows will synchronize, and share the medium in a time division multiple access (TDMA) fashion [50]. So unless some low priority traffic comes and disturbs the order, very few black burst contention period will have to be initiated once the stations have been synchronized.

Because of reducing the collisions in DCF, the Black burst scheme can provide better QoS performance for RT-traffic over WLAN than CSMA/CA.

For enhancing QoS, the Black burst scheme reduces the traffic delay and jitter.
The weakness of the Black burst scheme is that it requires constant access intervals, for high-priority traffic load; otherwise the performance degrades considerably.

5.2.5 VMAC schemes

VMAC [50] means Virtual Medium Access Control. Because the VMAC algorithm operates in parallel to the MAC in the mobile host but does not handle real packet transmission like in MAC. This is why it is called virtual MAC.

In fact, VMAC is a fully distributed service quality estimation, radio monitoring, and admission control approach to support service differentiation. A virtual MAC (VMAC) algorithm monitors the radio channel and estimates locally achievable service levels. The VMAC estimates MAC level statistics related to service quality such as delay, jitter, packet collision, and packet loss.

The advantage of virtual MAC is that it can estimate higher order statistics than first-order performance statistics without too much overheads. By this way, more sophisticated analysis and traffic control methods can be applied. Moreover, a virtual source (VS) algorithm can utilize the VMAC to estimate application-level service quality. The VS allows application parameters to be tuned in response to dynamic channel conditions based on VS allows application parameters to be tuned in response to dynamic channel conditions based on "virtual delay curves."

The goal of the VMAC is to estimate QoS parameters in the radio channel accurately since relative service differentiation is not enough for real-time services. Moreover, this scheme uses the following back-off timer differentiation:

$$CW_{\min}^{high\_pri} < CW_{\min}^{low\_pri}, CW_{\max}^{high\_pri} < CW_{\max}^{low\_pri}.$$

Where, CW is the size of the contestant window.
From the mechanism of the VMAC, we can find: when these distributed virtual algorithms are applied to the admission control of the radio channel, then a globally

stable state can be maintained without the need for complex between $CW_{\min}^{high\_pri}$ and $CW_{\min}^{low\_pri}$, i.e., decreasing $CW_{\min}^{high\_pri}$ and increasing $CW_{\min}^{low\_pri}$ provide high priority traffic lower delay than before, and low priority traffic higher delay than before.

The weakness of the VMAC is that it is too much complexion interactions between application and MAC layers.

5.3 PCF-BASE ENHANCEMENT SCHEMES

We have discussed some enhancement schemes based on DCF in the previous sections. However, because the PCF is the optional schedule, just few researchers try enhancing the protocol. In fact, these schemes are suitable just under some special conditions.

5.3.1 Super-poll scheme

As discussed in 5.1, BSS overlap, hidden terminal effects and noise characteristics of the wireless environment are the main challenges for RT-traffic QoS over the WLAN based on IEEE802.11. Although PCF polling based protocol designed for RT-traffic support, but it has the poor performance without considering the poll lost. Actually, PCF is very sensitive to lost polls, especially in the noise environment that including BSS overlap and hidden terminal. Robust Superpoll scheme proposed approach that the PC broadcasts at the beginning of the contention free period a Superpoll. To make the scheme more reliable, each packet includes identities of remaining stations to be polled in the list. Therefore, stations have multiple opportunities to receive the poll. Then the poll transmission is more reliable than PCF in the 802.11 standard.

As we known, the PC polls each station individually. Super-poll base protocol in which the list of stations allocated the right for transmission in a certain period is announced at the beginning of a period, have been proposed for communication networks that have very large propagation delays, e.g., satellite communication. The mechanism of the Super-poll scheme must base on the assumptions of the IEEE802.11. From the PCF we know:

1) PC maintains a list of wireless stations within its BSS and updates it whenever a new station joins or a station leaves the BSS.

2) The stations register for PCF service during the PCF service during the DCF period.

3) The PC determines the polling sequence.

4) A wireless station that receives the poll will transmit for the time allocated to it.

There are some problems for the mechanism:

1) When the WLAN channel is very noisy, it result a lost poll in the specific station loosing its turn during the current super-frame. So it has to increase the probability that poll is indeed received by the station.

2) We must consider the quality of the channel, or the noise, may be different between any two stations in the WLAN: the quality of the channel depends on the characteristics of the propagation path between the stations. This fact indicate to the possibility that the poll can be received at the intended station indirectly, i.e., not necessary from the Point Coordinator but from another station in the WLAN.

From the two points above, the Super Poll based protocol's mechanism is as following:
The Point Coordinator (PC) computes the identity of the stations that will be polled during the current PCF interval. The PC broadcasts an initial Super-Poll that includes this list of stations to be polled during the current PCF interval.
Using Chaining concept to alleviate the problem of the lost poll, each station that transmits during the PCF period will incorporate in each data packet a Super-Poll.

The Schemes made some mathematical models to calculate the Channel efficiency [51].
It uses U_{single} to denote the channel efficiency for the Single Poll and use $U_{SuperPoll}$ to denote for Super-poll.

The first step calculates the average number of bytes per transmission:

$$O = \frac{(G-1)+(G-2)+\cdots+0}{G} = \frac{G-1}{2} \qquad (5\text{-}9)$$

Where, O denotes the average number of bytes per data packet required to represent the polling list.
G denotes the stations in the polling list.

If considering this overhead, O, and the overhead from the initial Super- Poll transmitted by the PC, given by $sPTS/G$, the average number of the bytes per transmission is given by: $D+A+(G-1)/2+sPTS/G$.

Where, D denotes the number of bytes per packet.

A denotes the number of bytes in the ACK packet.

The following step is to calculate the probability that a poll is not received. For single polling case, P denotes it. In the Super-Poll protocol, P_j denotes the probability that the j-th station in the Super-Poll list will not be able to receive the Super-Poll until its time to transmit. From the classical probability rule and this probability computation is recursive, we can obtain:

$$P_1 = Pr(st\#1 \text{ does not receive SuperPoll}) = Pr(st\#1 \text{ does not receive SuperPoll from PC}) = P$$

Let

$$P_{2a}=Pr(st\#2 \text{ does not receive SuperPoll from PC}) = P,$$

$$P_{2b}=Pr(st\#2 \text{ does not receive SuperPoll from st\#1}) = Pr(st\#2 \text{ does not receive SuperPoll from st\#1} \mid st\#1 \text{ does not receive SuperPoll})* Pr(st\#1 \text{ does not receive SuperPoll}) + Pr(st\#2 \text{ does not receive SuperPoll from st\#1} \mid st\#1 \text{ receives SuperPoll})*Pr(st\#1 \text{ receives SuperPoll}) \quad (5\text{-}10)$$

$$=(1)(P_1)+(P)(1-P_1)=P_1(1-P)+P$$

Therefore,

$$P_2= Pr(st\#2 \text{ does not receive SuperPoll}) =P_{2a}*P_{2b}=P*(P1(1-P)+P),$$

$$P_3= Pr(st\#3 \text{ does not receive SuperPoll}) =P(P_1(1-P)+P)*(P_2(1-P)+P)$$

.

.

.

In general:

$$P_1 = P$$
$$P_j = P_{j-1}(P_{j-1}(1-P)+P).for.j = 2,3,...,G. \quad (5\text{-}11)$$

So we can obtain the average probability that a station will not receive the Super-Poll, P^* denotes it.

$$P^* = \frac{\sum_{i=1}^{G} P_i}{G} \qquad (5\text{-}12)$$

Following this, we can calculate the gain efficiency:
From the PDF, we can obtain:

$$U_{single} = \frac{D(1-r)(1-P)}{D+A+PTS-Dr} \qquad (5\text{-}13)$$

If we let Y_i denote the number of bytes used by ***i***th order in the polling list and we let Y_G denotes the number of bytes used by the group size of G. We can obtain:

$$Y_G = \tau + \delta \qquad (5\text{-}14)$$

Where, let

$$\alpha = (1-P)(1-P^*),$$
$$\beta = sPTS/G + (G-1)/2,$$
$$\tau = \alpha(Y_{G-1} + (1-r)D + A + 1 + \beta),$$
$$\delta = (1-\alpha)(G(D+A) - r(D) + 1 + \beta)$$

So the channel efficiency for the Super-Poll protocol $U_{SuperPoll}$ can be obtained as:

$$U_{SuperPoll} = \frac{GD(1-r)(1-P^*)}{Y_G} \qquad (5\text{-}15)$$

So the efficiency gain ***E*** is obtained below:

$$E = \frac{U_{SuperPoll} - U_{single}}{U_{single}} = \frac{G(D+A+PTS-Dr)}{Y_G}\frac{(1-P^*)}{(1-P)} - 1 \qquad (5\text{-}16)$$

From formula (5-16), we can find that the Super-Poll scheme has the potential to improve the IEEE 802.11 PCF support for multimedia. However, it doesn't solve the limitation mentioned in chapter 4.

5.3.2 other enhance PCF schemes

Some enhanced PCF schemes are proposed to improve the utilization of the wireless channel and support certain QoS of RT-traffic over WLAN based on PCF IEEE802.11 standard.

Round-Robin Scheme

The scheme [52] is very simple; it just improves from the mechanism of the frame transmission based on PCF. From the discussion of 5.13, we can understand that when the PC (point coordinator), an AP that can control the PCF, gains access to the medium, it

first finds the lowest address of stations, and then, after checking whether there is any data for this address in its queue, determines the type of frames to transmit.

For increase utilization of the wireless channel, there are four types of frames the PC can choose to transmit: CF-POLL, CF-ACK-POLL, CF-DATA-POLL and CF-DATA-ACK-POLL. The PC always piggybacks the acknowledgment message if the PC needs responding to a station just transmitting a data frame. Whether the PC receives the acknowledgement from the station or not, the PC should change the polling address in turn if there is any other station in this BSS. After serving all stations in this BSS, the PC chooses the lowest address of stations again until the duration of this CFP has reached the maximum CFP duration. Now, we can understand that Round-Robin Scheme can be more efficient based on PCF.

FIFO Scheme

FIFO means first in first out. The scheme also just change the part of the frame based on the PCF. According to the order of frames in the PC's queue, the PC determines next transmitting data and poll address. Again, in order to promote the utilization pf the wireless channel, the PC always piggyback the polling message in transmitting frames, and piggybacks the acknowledgement message if the PC needs to response a station transmitting a data frame, just now.

Different from the round-robin scheme, if the receiving station has to acknowledgment back, the PC assumes that the transmission has encountered an error and will try again until the retry-count has reached the limit. Keeping transmitting data in queue and polling the corresponding station, the PC will work as the Round-Robin Scheme if there is no data in queue. Under the duration of polling stations in round-robin order, the PC resumes adopting the FIFO Scheme if any frame enters its queue. Obviously, FIFO has the more reliability than Round-Robin scheme. FIFO scheme achieves the highest throughput. The weakness of the FIFO is its weak fairness.

Priority Scheme

From the PCF mechanism, the AP sends priority-based polling packets to a succession of stations in the wireless BSS, which can assign stations different priorities. IEEE802.11

does not specify how the AP determines the polling sequence. The priority scheme [52] provides simple support of priority transmission. The scheme function follows:

1) Every session's type of traffic is given by the up-stream gateway or registered when stations associate with the PC.

2) When entering the PCF duration, the PC first check the stations of highest TOS (Type of Service), and then, if this type of stations have data to send or receive, transmit data frame or poll frame to this station with highest TOS.

3) After serving the stations with highest TOS, the PC will serve the stations with secondary TOS in turn.

4) Until serving all stations with TOS larger than 1, the PC will serve the station with Best-Effort traffic and entering into PC's queue.

Like the two schemes described above, in order to promote the utilization of the wireless channel, the PC always piggybacks the polling message in transmitting frames, and piggyback the acknowledgement message if the PC needs to response a station transmitting a data frame just now. Unusually, in this scheme, the PC needs to keep track of the flag, which indicates whether a certain station has data to send or not. This flag is the "more data sub-field" in a MAC header. A mobile station that is polled by the PC during a CFP may use this sub-field to indicate to the PC that there is at least one more frame buffered at the mobile station to be sent to the PC. Even though there is no data to the station with certain TOS values, PC should still send a Poll frame to this station as long as the flag has been set "true". From the mechanism of the priority scheme, we can find that the scheme can support at low cost QoS of traffic but may severely affect low priority and best-effort traffic to compensate the losses of high priority traffic. So, it can enhance the RT-traffic QoS over WLAN 802.11 PCF.

Priority-ELF scheme

ELF means Effort Limited Fair [53]. ELF scheme model focuses on the insight that, in a wireless environment, we must distinguish between "efforts" (air time spent on a flow) and "outcome"(actual useful throughput achieved by the flow).

While effort equals outcome in a wire-line environment, they can be substantially different in a wireless environment. An ELF scheme strives to limits on the effort spent on each flow using a per-flow power factor setting.

The ELF scheduling can be used in wireless network to guarantee that all flows experiencing an error rate below a per-flow threshold receive their expected service, defined as a specified rate for reserved flows or a specified share of best-effort capacity for best-effort flows. Here priority-ELF scheme combines the ELF's novel notion with our priority scheme.

Similar to the priority scheme, when entering the PCF duration, the PC first check the stations of highest TOS, and then, if this type of stations have data to send or receive, transmit data frame or poll frame to this station with highest TOS. Additionally, the PC also checks the values of the counters which record the amount of frames transmitted by each station in a certain period is larger than one. If not, the PC will not transmit data or poll frame to a station even though the PC has data for it in queue or it has frames to send. This condition happens when there are some link errors causing the effort (air on a flow) higher than the threshold of its power factor. This condition could also happen when the source suddenly produces too many frames to transmit. The priority-ELF scheme also prevents the PC from wasting too much time in an error-prone flow. In fact, priority-ELF scheme is based on the priority scheme with adding the limitation on the effort spent. For this reason, the priority-ELF scheme becomes fairer than priority. Meanwhile, the scheme achieves high utilization of the wireless channel link.

5.4 Up Coming IEEE 802.11e

For enhancing the 802.11 MAC to include bi-directional quality of service to support latency-sensitive application such as voice and video; IEEE formed 802.11e-working group in September 1999. The group is in charge of the 802.11e building-up. In fact, 802.11e [54] is an extension of MAC layer to the legacy 802.11a/b/g standard to provide QoS for RT-traffic. The first draft of 802.11 was available in January 2001. Two new protocol of MAC layer was proposed in this draft. They are hybrid coordination function (HCF) and EDCF (Enhanced DCF). In fact, EDCF is not an independent protocol. It is

the part of the HCF. HCF combines aspects of both two modes: contention mode and controlled access mode. In contention mode, 802.11e employ the EDCF protocol. EDCF can be regarded as a "soft" QoS assurance mechanism in the sense that a traffic class can statistical reduce its transmission delay by categorizing itself into a higher priority traffic class in its contention for the channel. For explaining clearly, we discuss EDCF and HCF in this section respectively.

So far, IEEE does not approve 802.11e. It looks like still in debate with some instability and other problems [55] [56] [57]. So recently, the research of the performance of QoS over WLAN base on IEEE802.11 sounds very significance.

5.4.1 EDCF

EDCF is based on DCF of 802.11. From section 5.12, we know the mechanism of DCF. DCF is a listen-before-transmission scheme indeed. In this mode, if the medium is idle, the station that wants to transmit frame waits for DIFS interval, and then transmits a packet immediately; otherwise, the back-off procedure is employed. The back-off time is computed as:

Back-off Time=Random ()*Slot Time (5-17)

Where Random () is a pseudorandom integer drawn from a uniform distribution over the interval [0.CW] CW is an integer between [CWmin] and [CWmax]. Where, Slot time, [CWmin] and [CWmax] is determined by the PHY layer characteristic. The different traffic have the same mechanisms in the DCF. In other words, the different traffic have the same waiting IFS time (DIFS) and the 'same' back-off time. It means all stations and traffic have the same priority to access the wireless medium. From the mechanism of the DCF, we know easily: the QoS is not supported with the use of DCF. For those time-sensitive traffic, WLAN must ensure the less packet delay and jitter. The RT-traffic should be having the higher priority to be transmitted than those time-insensitive traffic (say best-effort services); EDCF provides this priority mechanism for different traffic. For the priority mechanism, EDCF use static parameter setting for differentiating

between traffic categories. Those parameters include: AIFS (Arbitrary Inter-frame Spacing), CW.

AC Definition

For the priority mechanism, the EDCF must classify the applications used in the WLAN firstly. In fact, the draft 802.11e specification attempts to provide classification for up to eight classes of data. EDCF and HCF polled access leverage these eight classes, known as traffic classes (TC). The TC actually comes for the IEEE802.1D standard. In other words, from this class; WLAN is better connective with the wire-network. Meanwhile, the draft categorizes traffic from QoS-enable clients into four broader categories know as access categories (AC). It is shown in table 5-1: [57]

Table 5-1 802.11e TC-to-AC Mapping [56]

802.1D value/TC	Common usage	AC
1	Low priority (Background)	0
2	Low priority (not defined)	0
0	Best effort (BE)	0
3	Signal/control (CL)	1
4	Video probe (<100ms latency)	2
5	Video (<10ms latency)	3
6	Voice	3
7	Network control	3

AIFS

From the QoS mechanism, any system supports QoS needs three key component points: firstly, it classifies the traffic; secondly, it must mark the traffic with the appropriate QoS value; finally, it must to differentiate and prioritize the traffic, based on the QoS value. From the table 5-1, the 802.11e draft has classified the traffic from the 802.1D. So, EDCF provides the mechanism to differentiate and prioritize the traffic for transmission.

There are two new concepts introduced in EDCF.

One is Arbitration inter-frame space (AIFS) [58]. In fact, AIFS is the one of the IFS was introduced in pervious chapter. The size of AIFS is different based on different traffic AC. We can recall that there are different IFS in the 802.11. They are: SIFS, PIFS, DIFS and AIFS.

SIFS (short IFS): this is the shortest IFS. It is used for transmission of high priority frames: acknowledgement of DATA frame, CTS frames, PCF frames and all DCF DATA frames except the first fragment of a burst.

PIFS (PCF IFS): longer length than SIFS. Use for PCF protocol, after this inter-frame spaces interval expires, any PCF mode frame can be transmitted.

DIFS (DCF IFS): longer length than PIFS. Use for DCF protocol, after this inter-frame spaces interval expires, any DCF mode frame can be transmitted.

SIFS, PIFS, DIFS are the same for different traffic, so for the IFS expired mechanism, PCF and DCF can not provides any priority for the QoS services. AIFS provides different queue IFS by different AC. AIFS is defined as:

$$AIFS_i = SIFS + aAIFS_i * aslotTime \quad \text{..... (5-18)}$$

Where, ***i*** denotes different AC weight; the default value of the $aAIFS_i$ is defined as either 1 or 2 [84]. When $aAIFS_i$ =1, AIFS value actually equal to PIFS, it is always for high priority AC1, AC2, AC3. Otherwise, $aAIFS_i$ =2, AIFS value actually equal to DIFS, it is always for high priority AC0. When the medium has been idle longer than $AIFS_i$ +slotTime, the frame is transmitted immediately. If the channel is busy, the arriving packet in each AC has to wait until the medium becomes idle and then defer for AIFS + SlotTime. The different IFS are shows in figure 5-1.

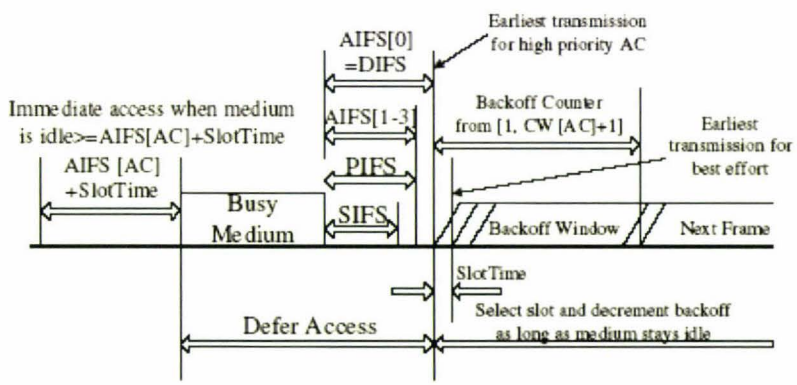


Figure 5-1 Different IFS [76]

From the figure 5-1, when the medium is idle, the higher priority (AC1, AC2, AC3) frames just need wait T_h=PIFS + SlotTime, default is T_h =DIFS. However, the lower priority frames (AC0), must wait T_l=DIFS + SlotTime. Obviously, $T_l > T_l$. In the other words, the higher priority frames that have smaller AIFS access the channel firstly.

EDCF back-off mechanism

EDCF is not only to differentiate and prioritize the traffic by the IFS, also in back-off mechanism.

EDCF is based on the DCF, so the back-off mechanism is the similar as DCF [59]. The formula (5-17) is also applied for EDCF. However there are two different points:

(1) Firstly, the different CW sizes for different traffic AC. As we know, if the collision happened, the framed, wanted to send, must wait a back-off time. For the formula (5-17), the time is a pseudorandom integer drawn from a uniform distribution over the interval [0, CW] CW is an integer between [CWmin] and [CWmax]. In DCF, Slot time, [CWmin] and [CWmax] is the same for different traffic. It is determined by the PHY layer characteristic. However, in EDCF, [CWmin] and [CWmax] are different. It is shown in table 5-2.

Table 5-2 Different CW and AIFS as the AC [77]

AC	CWmin	CWmax	AIFS
0	Standards 802.11 CWmin	Standards 802.11 CWmax	2
1	Standards 802.11 CWmin	Standards 802.11 CWmax	1
2	((CWmin+1)/2)-1	Standards 802.11 CWmin	1
3	((CWmin+1)/4)-1	((CWmin+1)/2)-1	1

(2) Secondly, TXOP (transmission opportunity) is adopted in EDCF. TXOP is the moment in time when a station can begin transmitting frame. TXOP is a given bounded-duration time in which the station may transmit a sequence of SIFS-separated DATA frame exchanges. Unlike basic medium access for DCF, where each frame and accompanying acknowledgment contends for the medium, a TXOP can facilitate multiple frames/acknowledgments as long as they fit within the duration of the TXOP. In fact, the schedule is designed for avoid the virtual collision. If the back-off counters of two or more parallel ACs in one station reach zero at the same time, the TXOP just is gotten by the highest priority AC. At the same time, the other colliding ACs doubles the CW size as if there is a true collision happened. It looks like in DCF. The TXOP ends when there are no more frames to be transmitted or when the TXOP maximum duration expires. The

default TXOP maximum duration is given by the MIB (Management Information Base) variable **dot11DefaultCPTXOPLimit**. It is shown in table 5-3.

Table 5-3 different TXOP Limit [78]

AC	TXOP Limit (802.11b)	TXOP Limit (802.11a/g)
0	0	0
1	3.0milliseconds (ms)	1.5ms
2	6.0 ms	3.0ms
3	3.0 ms	1.5ms

From the table 5-2 and 5-3; we can get the different AC parameters. AC (0) matches standard DCF values with the exception of the AIFS, which has a value of DIFS+1 slot time. TXOP duration limit of 0 means just allows a single frame to be transmitted. AC (1) has the same channel-access parameters as an 802.11 DCF station, with the exception of a TXOP duration that allows for multiple frames to be transmitted and acknowledged. AC (2) has a smaller contention window than lower-priority ACs and a longer TXOP. If we consider the default initial CWmin value is typically 7 slot times, from formula CWmin (AC(2)) = ((CWmin+1)/2)-1=3. Obviously, With AC (2), the CWmin value of 7 changes to 3. The station now only has to select a back-off value ranging from 0 to 3, a much shorter time windows. The CWmax value is also different, now using the CWmin value of 7. In the case, after the station has back-off and reached the CWmax value it increments the retry counter much faster. AC (3) has the shortest contention window of the ACs but also a shorter TXOP duration limit as well.

TXOP-EDCF

For the better performance of RT-traffic in WLAN, 802.11e standard also allows the QAP (quality access point) to adapt these parameters dynamically depending on network conditions [60]. To improve the throughput performance, EDCF packet bursting can be used in 802.11e. It means once a station has gained an EDCF-TXOP, it can be allowed to send more than one frame without contending for the medium again. After getting access to the medium, the station can send multiple frames as long as the total access time does not exceed the TXOP Limit bound determined by QAP. To ensure that no other station interrupts the packet bursting, SIFS is used between packet bursts. If a collision occurs,

the EDCF bursting is terminated. This mechanism can reduce the network overhead and increase throughput by multiple transmissions using SIFS and burst acknowledgements.

Admission Control of EDCF

For the pervious section, we understand that EDCF provides the priority for the RT-traffic well. However, too much high priority traffic also results to the poor performance of the QoS. In other words, how to ensure the performance of the real-time traffic in the overload circumstance. For example, a BSS can accommodate a maximum of five simultaneous VOIP calls. When the sixth VOIP call wants to send in the BSS, all the calls have the poor performance. Because of they have the same priority, so the situation occurs possibly.

Actually, EDCF admission control (EDCA) provides a schedule to solve the problem. It's the distributed admission control (DCA) [61].

DCA functions at a high level by monitoring and measuring the percentage of utilization of the medium for each AC. The unused percentage of the medium is referred to as the available budget for the AC. This available budget always notices the QSTA (QoS Station) in the QoS parameter information element (IE) in the QAP beacon. The general structure of the IE was discussed in chapter 5. When the budget starts to approach 0, stations will not to send the new frame. In the other words the TXOP of the station cannot be increase. It can ensure the existing RT-traffic performance over QoS.

5.4.2 HCF

Actually, EDCF is just the part of HCF. It is adopted in contention mode. The section should be called HCF in controlled access mode indeed. HCF mechanism is base on PCF, and overcome PCF limitations. There are some differences between both of HCF and PCF. HCF is more flexible and more suitable for RT-traffic priority; HCF solves the three problems of PCF. In chapter 5, we have discussed the limitations of HCF. There are three main problems: (1) resource waste. (2) The unpredictable beacon delay. (3) Difficult to predict the transmission time of a polled station. For those problems, HCF (1) HCF allows the direct link between peer stations without going through the QAP. It can

save many wastes to increase the throughput. (2) QSTA is prohibited to transmit a packet if the frame transmission cannot be finished before the next beacon. It can solve the next beacon unpredicted delay. (3) A TXOPLimit is used to bound the transmission time of a polled station. It can solve the final problem.

HCF Access Occur

From the access occurring, HCF is different from DCF. DCF just occurs during contention free period (CFP), however HCF can occur during both contention period (CF) and CFP. It always coexists with EDCF in CP well.

During the CP, QAP start several contention-free bursts, call controlled access period (CAP). If the channel is idle, CAP can start after PIFS, as we know, PIFS<DIFS<AIFS, so HCF are the high priority to start its HCF controlled channel access (HCCA). HCF defines a schedule can let HCCA start whenever it wishes flexibly. The schedule is very simple: defining a latter periodically after a beacon frame.

During the CFP, the beacon is sent by QAP, the beacon includes the PCF compensating fiber (CF) parameter set IE that specific the start time and duration of a CFP. The QAP contains a logical entity known as the hybrid coordinator (HC) that keeps tracks of HCD client stations and schedules the polling intervals. Then the HC offers a TXOP to QSTA by sending QoS CF-Polls to them. QSTA station must reply back within a SIFS time interval with data frames or with a QoS null frame, indicating the station has no traffic or the frames it desires to send is too large to do so in the time allotted in the TXOP. Finally, the CFP ends when the HC sends a CF-end frame, or the CFP duration expires. After CFP, the EDCF and HCCA alternate in a beacon interval. Because the 802.11e draft defines EDCA is mandatory, so a variable $T_{CAPLimit}$ in 802.11e draft bound the maximum duration of HCCA.

The mechanism is shown in figure 5-2 [62].

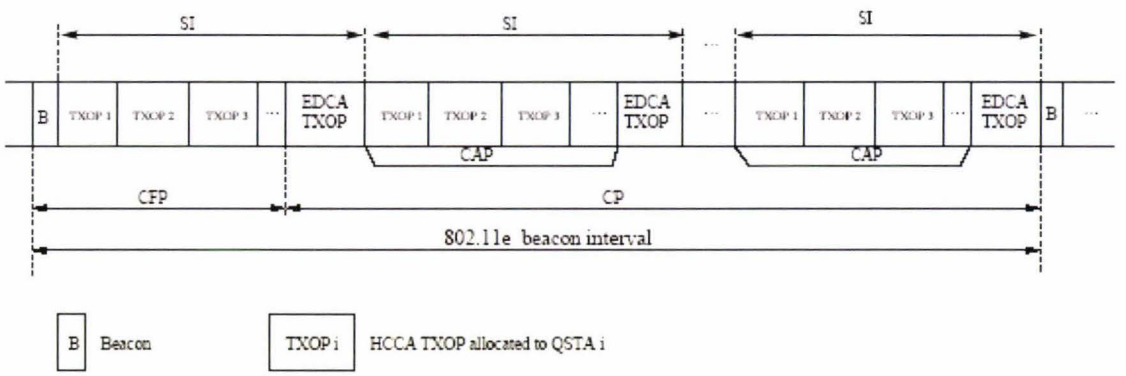


Figure 5-2 HCF Mechanisms [62]

HCF Admission Control

Admission control is the most important part in HCF. As we know, EDCF special admission control is the DAC. It is the new concept in 802.11e different from early ones. In HCF, the admission control mechanism is more robust and effective than DAC. HCF uses the traffic specification (TSPEC) as negotiation between the QAP and the QSTAs. HCF requires that the QSTAs request particular reservation parameters for the application traffic stream from the QAP. The QAP can evaluate the requested traffic stream. The QAP can evaluate and determine whether there are enough budgets available on the wireless medium to facilitate the requested traffic stream. Then QAP can determine accept, reject, or even offer an alternative set of parameters to the QSTAs. The TSPEC allows QSTAs to specify the QoS parameter requirement of a traffic stream such as: frame/stream 802.1D priority; frame size; frame rate; data rate and the delay bound and the maximum required service interval (RSI). Where, the RSI is the maximum time duration between the start of successive TXOPs that can be tolerated by the application. When the RSI is requested by the QSTA, the QAP calculates the TXOP and allocates it to every QSTAs by the RSI simply.

HCF algorithm schedule of 802.11e draft

In fact, HCF algorithm is calculating and allocating every TXOP to those QSTAs correctly. The process is as following:

1) QSTA sends the TSPEC to the QAP.

2) QAP determines the minimum value of the all the maximum RSIs required by different traffic streams.

3) Selecting service interval (SI), Which is duration that the highest sub multiple value of the 802.11e beacon interval. It should be less than the minimum of all the maximum RSIs.

4) The whole 802.11e beacon interval is cut into several SIs and QSTAs are polled accordingly during each elected SI. Where, the selected is the time between the start of successive TXOPs allocated to the QSTA, which is the same for all the QSTAs.

5) QAP calculates the different TXOP values allocated to the different traffic streams for different QSTAs.

Following [101], we can use the formula to represents it:

The first step calculates the number of the packets:

$$N_{i,j} = [\frac{\overline{\rho}_{i,j} SI}{M_{i,j}}] \qquad \text{..... (5-18)}$$

Where,

$N_{i,j}$ denotes the number of packets arriving in the traffic stream j in the QSTA i during the selected SI.

$\overline{\rho}_{i,j}$ denotes the mean data rate request of the applications from the traffic stream j in the QSTA i.

$M_{i,j}$ denotes the normal MSDU size for this queue.

The second step calculate the allocate TXOP.

$$T_{i,j} = \max(\frac{N_{i,j} M_{i,j}}{R} + O, \frac{M_{\max}}{R} + O) \qquad \text{..... (5-19)}$$

Where,

R denotes the PHY layer transmission rate

$M_{\max}$ denotes the maximum MSDU size.

O denotes the transmission overheads due to PHY/MAC layer headers, IFSs, ACKs and poll frames.

6) Summing all the TXOP values of different traffic streams in the QSTA i.

$$TXOP_i = \sum_{j=1}^{J_i} T_{i,j} \qquad \text{..... (5-20)}$$

Where, J_i denotes the number of different traffic streams in QSTA i.

As this schedule, QAP allocates the all TXOP to different QSTAs (shown as figure 5-2).

So when a new traffic flow tries to enter the BSS, it is very simple to check it:

$$\frac{TXOP_{K+1}}{SI} + \sum_{i=1}^{K} \frac{TXOP_i}{SI} \leq \frac{T_{CAPLimit}}{T_{Beacon}} \qquad \text{..... (5-21)}$$

Where, the $\sum_{i=1}^{k} \frac{TXOP_i}{SI}$ denotes the total fraction of transmission time reserved for HCF of all ***K*** QSTAs in the beacon interval. If the equal is true the K+1-th traffic can get in, otherwise, it can't get in. It can ensure the QoS performance of the RT-traffic over WLAN in HCF.

Summary

For enhancement of RT-traffic QoS performance over the MAC layer, many schemes were proposed based on mechanisms of the DCF and PCF.

Because the DCF is the basic protocol, most schedules were proposed based on DCF. DFS, DENG, IACC, Black burst, and VMAC are the available schemes. Deng achieved the good performance for high priority traffic. DFS achieved fairness. IACC provided good service differentiation. Black burst reduced the RT-traffic's delay. VMAC implemented the channel condition. But all of them have their limitations inside.

The enhanced schedules based on the PCF cannot solve the limitations of the PCF, but they also provid some good mechanisms to enhance the RT-traffic QoS performance over WLAN.

Upcoming 802.11e provides technical methods from some enhance schemes. For resolving the priority limitation in DCF, based on Deng and ICCA, it employed EDCF as the protocol in CP interval. The main difference against the DCF, it divided the traffic into eight classes as the 802.1d with their different AIFS. Meanwhile, EDCF defines different CW interval to different traffic as their AC respectively. It can provide reasonable priority for RT-traffic. The main attraction in 802.11e defines that the EDCF is the part of the HCF. HCF can control the access mode in FCP interval, and solves the

problems of PCF. Different with PCF, HCF can occur during both the CF and CFP. By calculating and allocating every TXOP to the QSTAs reasonably, HCF guarantee the low delay and jitter of RT-traffic transmitted over WLAN.

As the theoretical mechanism of 802.11e, we find it is prefect to guarantee the low delay and jitter, the main parameters of RT-traffic. However, from DFS proposal, the fairness is another important parameter of the QoS in WLAN network. We should study the 802.11e by some physical model to calculate delay, jitter and fairness index of RT-traffic QoS performance parameters. The advantage and limitations of the 802.11e draft will be further studied in the next chapters by physical model.

CHAPTER 6
SIMULATION WITH OPNET AND NS-2 FOR PERFORMANCE STUDY

Many factors can affect the QoS performance of the RT-traffic over WLAN. As discussing in the pervious chapters, because of most of the RT-traffic are time-sensitive applications; transmitted delay is the main factor affecting the QoS of the RT-traffic performance based on the IEEE802.11. IEEE group developed PHY layer of the 802.11 standards for increasing the transmission data rate. Generally, RT-traffic ask the higher data rate. For example, the higher-solution videoconference must be transmitted with 30 frames/ second for the frame inter-arrival time information, and one frame size is 352*240 pixels. It means the data rate is more than 2.5Mbps required. Obviously, the low rate WLAN cannot guarantee their QoS performance. Although 802.11a/g can get 54Mpbs, 802.11a/b/g standard's MAC layer is the same. They are based on DCF and optional protocol PCF.

Actually, bandwidth is not enough in mostly existing WLAN. Under this situation, developing MAC layer becomes the focus problem to resolve the RT-traffic QoS performance. IEEE 802.11e gives two protocols that are the EDCF and HCF. We have discussed the principles of them clearly in chapter 5. However, how is the RT-traffic performance physically in the WLAN? We need to study by some methods. As my considerations, there are three optional methods to study it:

1) Mathematical analysis method: we can determine and analyze the research objects of the WLAN basically. Then, based on some reasonable assumptions and principlea, describe the study object and WLAN system, and extract the mathematic analysis model of the study object. Then resolve the question (say delay, throughput and fairness) based on the mathematic model.

2) Experimental method: we can buildup the WLAN test-bed based on the protocols. Designing the programmable WLAN network card, and use C or assembly language write the 802.11 a/b/g in the programmable chip in PHY layer with the connected hardware and write 802.11 DCF in MAC layer. Then we can design the upcoming

802.11e programmable and write it in the programmable chip. Use video, voice and background traffic to study its performance.

3) Simulation method: we can buildup the system model by the network simulation software, and then run the models in the PC to get the interested results and study them.

For the method 1, the disadvantage is obviously: the availability and precision are limited by the effect of the assumptions. In reference [63], Jie et al. propose a mathematical model by the Markov transfer possibility chain to analyses the EDCF protocol.

For the method 2, the experimental cost is too expensive. Re-deploying and sharing the existed source is very difficult. The whole built system should be not flexible to study the performance of the RT-traffic in the WLAN. Meanwhile, the test-bad are very difficult to build up to big enough. It is very difficult to achieve the compatible between kinds of traffic and different protocols.

The method 3 is the best choice. Simulation can build up the network model as requirements. It can get plentiful results to study the QoS performance of RT-traffic based on WALN. There are two effective network simulation software tools today. They are OPNET and NS-2. This research project is based on the simulation results by building WALN model use the two software tools.

6.1 Simulation with OPNET

OPNET is developed by two experts of the MIT University in 1986. OPNET earned many awards recently to prove OPNET can simulate network performance and get the correct result to study the kinds of networks. In the industry, many big corporations (says Cisco, AT&T etc.) employed OPNET to do simulations and modulations. In OPNET products, Modeler can simulate all kinds of network performance. Modeler contains EENAD (End to End Network Architecture Design), SLSND (System Level Simulation for Network Devices), PDO (Protocol Development and Optimization) and NAODA (Network Application Optimization and Deployment Analysis). Modeler employs the hierarchical network modeling [64]. For the protocol design, it is all as the OSI standard.

Modeler employs the object-oriented modeling; the same category nodes employ the same node model. There are different object in the node, we can set up the parameter for every node. By the function provided by the OPNET, We can easily get every object ID to operate them. For example, we can use *op_pro_id(pro_handle)* to obtain a unique integer identifier for a process. And use the *op_topo_assoc(objid, direction, objmtype, index)* to obtain the object ID of an object's association having the specified type.
OPNET events start by the FSM (Finite State Machine Modeling) mechanism, it avoids driving from time; build up model by the event drives. OPNET employ the DED (Discrete Event Driven) simulation mechanism. Compare the time driving; the computing capacity is increased effectively.

For studying the performance of the RT-traffic, we construct a WLAN model by the OPNET. The model is the general Ad-hoc WLAN; in the WLAN we employ 4 category traffic as the 802.11e requires. We chose the true value from the voice (G.729), the videoconference, BK (background traffic) and BE (Best- effort). From the pervious chapter, we can know their AC exactly from 3 to 0.

To study the results of the performance of the RT-traffic QoS over the WLAN 802.11, we form two scenarios to do simulation: DCF and EDCF from MAC layer. DCF protocol exists in the newest version (10.5) OPNET. However, EDCF is not approved yet by the IEEE802.11 [65], so we need to develop it base on the DCF. In every scenarios, we changed different PHY layer from 802.11b, a, g to study the performances of the RT-traffic over the QoS. As we know, the time-delay is the important factor for the performance of the RT-traffic. However, from chapter 1, we know that for a network, the throughput and fairness is very important factors relation with the RT-traffic over the WLAN. How to reasonably increase the performance for WLAN must be considered. In fact, the target of 802.11e is to reduce the time delay for the RT-traffic. However, by providing priority for different traffic, it results many problems in throughput and fairness. I guess it is the reason why IEEE does not approve the 802.11e as quickly as expected.

For study of the throughput, we actually do many experiments by the model. For the limited pages, we cannot provide all results in the thesis. In every stage, we just provide two cases: the light and the heavy model figures. And we exported graph data to spreadsheet by the Microsoft Excel and analyzed them in the MATLAB.

6.1.1 Performance of RT-traffic over DCF

Since DCF is the standard model existed in the OPNET 8.0, the performance of RT-traffic over DCF can be validated by the standard model from OPNET 8.0. The DCF protocol can be set up by the MAC layer in every station. The 4-category traffic in four WLAN advance stations respectively. It is shown in figure 6-1.

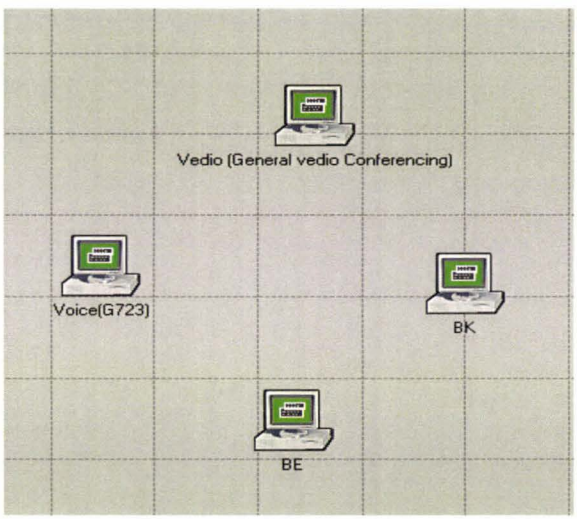


Figure 6-1 DCF Model

We deploy the attributes for every node, and determine their topology and choose the individual DES statistics and configure simulation.

Figure 6-2 shows one of the station's attributes:

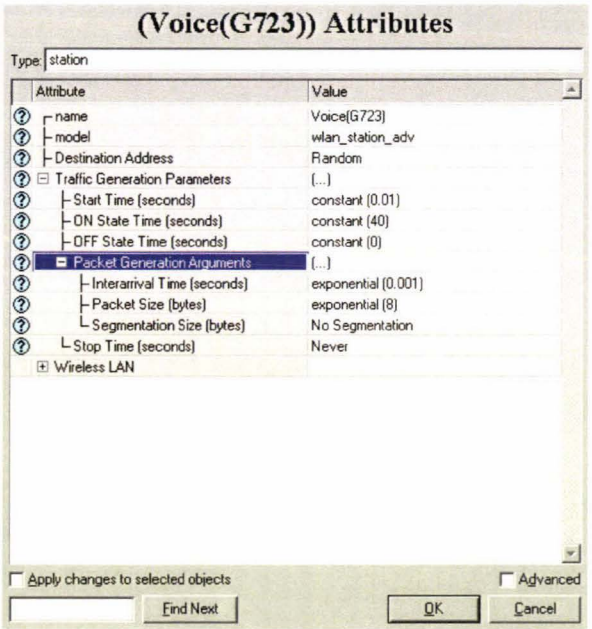


Figure 6-2: Voice Attribute Deployment

For every node:

We chose the start time from 0.01 second. And On Stat Time is 40 sec by the constant distribute; Off Stat Time is constant (0).

We use that every packet must be generated in 0.001 seconds by the exponential arrived rate without fragment.

For the light loading scenario: throughput, we determined the BE (Best Effort) traffic is 1111bps (138.88byte/second), it is the FTP down light load [66].

BK (Background) traffic is the 44 bps (5.5byte/second). It is the e-mail traffic.

Videoconference is the general definition type; it is 128*240 pixels/ frame and the 10 frames /second. The loading throughput is 307200bps (38400byte/second)

The voice is the G729 [67] encode speech, it is about 64kbps (8kByte/second).

For the heavy loading scenario: we increase the BE and BK's loading more and more until to 11.11Mbps (1.389Mbyte/second) and the 10.24Mbps. And the videoconference is the high definition type; it is 352*240 pixels/frame, 30frame/second. The loading throughput is 2534400bps (316800byte/second)

Results of the simulation are shown in the following sections.

6.1.1.1 Performance of RT-traffic over DCF base on 802.11b

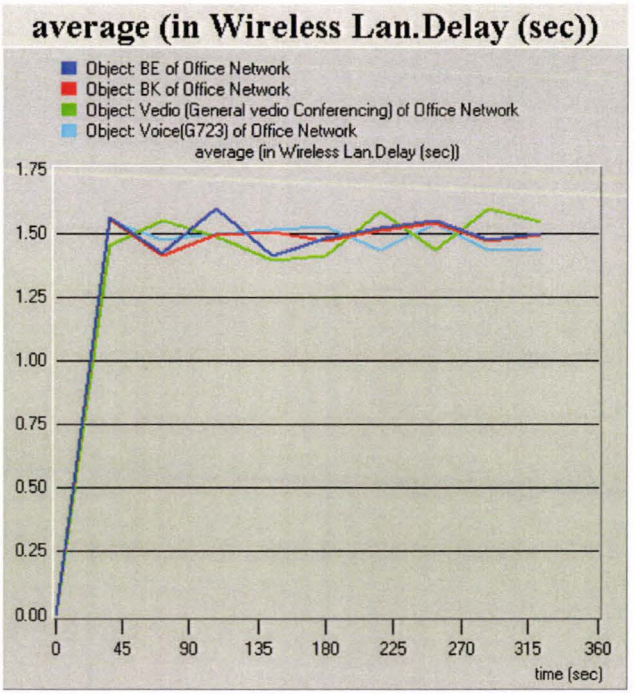


Figure 6-3 Time Delays in 802.11b Under the Light Loading.

Analysis: from the figure 6-3, we can find that in 802.11 b the RT-traffic's QoS cannot be guaranteed. The delay is same. As we know; the delay time of AC2 RT-traffic cannot exceed 200ms. AC3 traffic cannot exceed 150ms. In this figure, all traffic delay is more than 1000ms. Furthermore, the WLAN just is loaded the low background. From the figure, we can find that: since the delay time of different traffic are around 1.5 second, the QoS performances of the RT-traffic based on the DCF of 802.11b aren't acceptable.

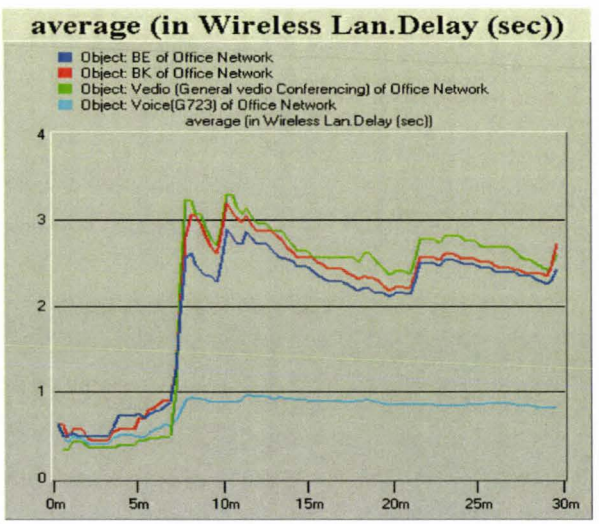


Figure 6-4 Time Delays in 802.11b under the Heavy Loading

Analysis: the delay for all is larger than 0.5 second. It is impossible to accept.

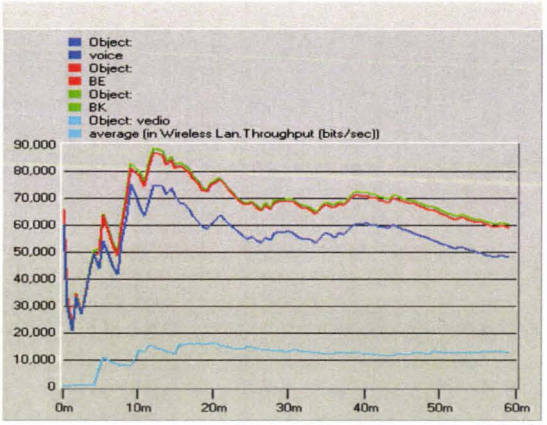


Figure 6-5 Throughputs in 802.11b under the Light Loading

Analysis*:* From the figure, because of the loading that we setup is light, so the different traffic have different throughput loading.

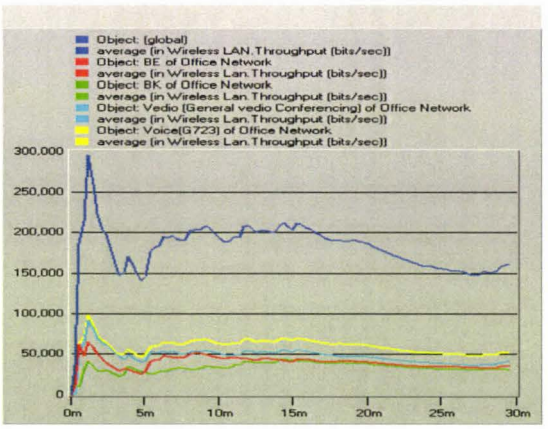


Figure 6-6 Throughputs in 802.11b Under the Heavy Loading

Analysis: under the heavy loading, the throughput tends to be similar; in fact, the RT-traffic cannot ensure the QoS at all.

6.1.1.2 Performance of RT-traffic over DCF base on 802.11a

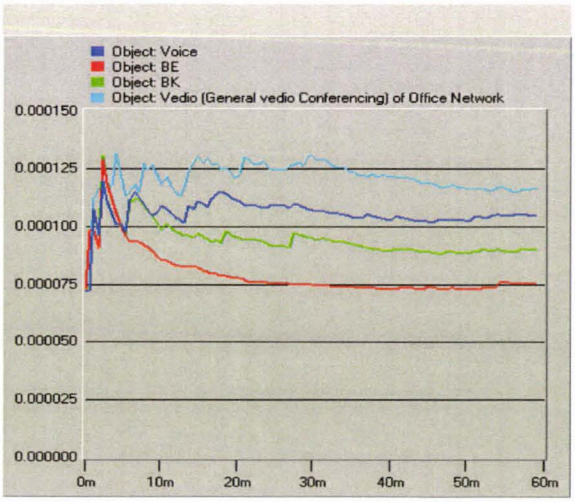


Figure 6-7 Traffic Delays in 802.11a under the Light Loading

Analysis: Delay is stable. In 802.11a the RT-traffic QoS can be guaranteed. It means that if we had enough bandwidth, the QoS is not a problem.

In fact, so light loading is impossible in the actual WLAN indeed. It just is the ideal situations.

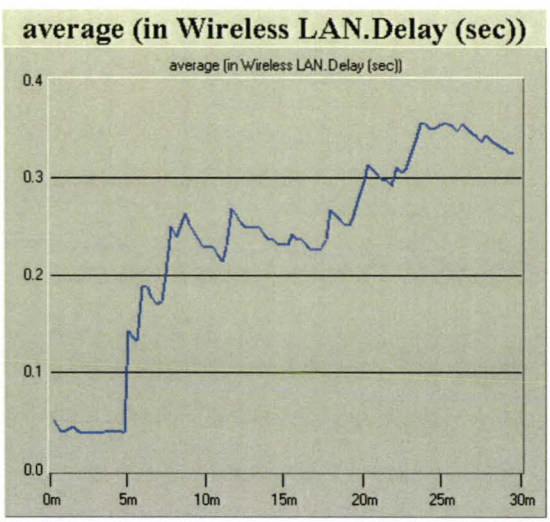
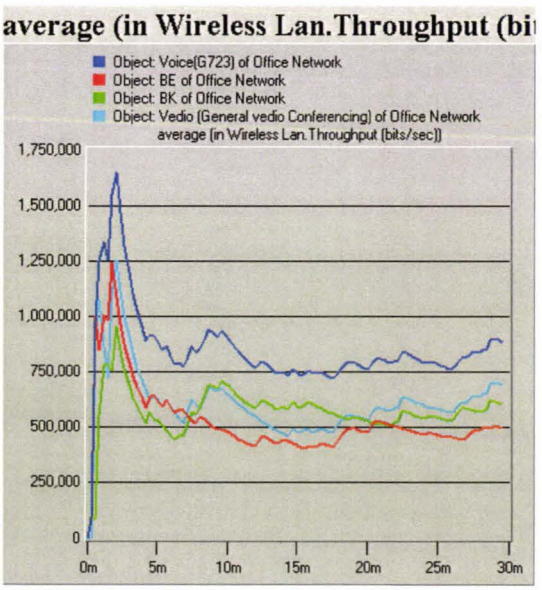
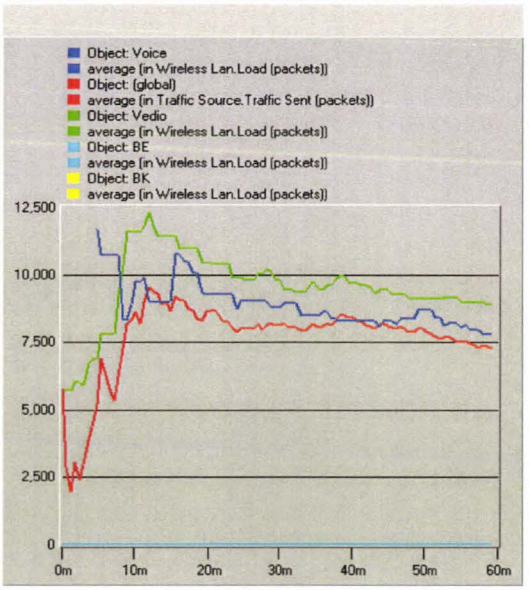


Figure 6-8 Time Delays in 802.11a under the Heavy Loading

Analysis: the delay is not stable now, and the different traffic are facing the same delay time.



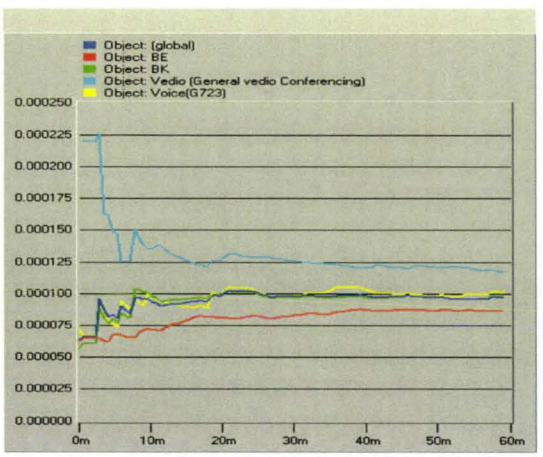


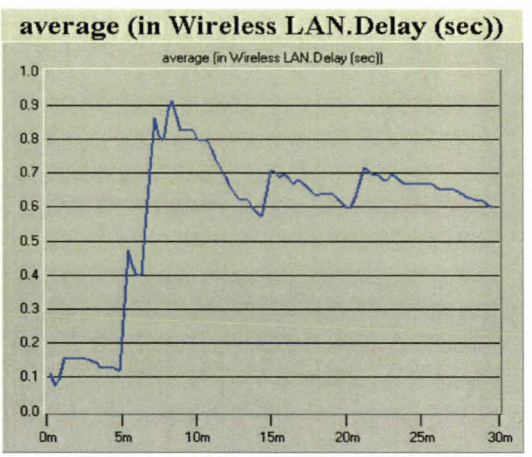
(a) Light Background Loading **(b) Heavy Background Loading**

Figure 6-9 throughputs in 802.11a under the light loading and heavy loading

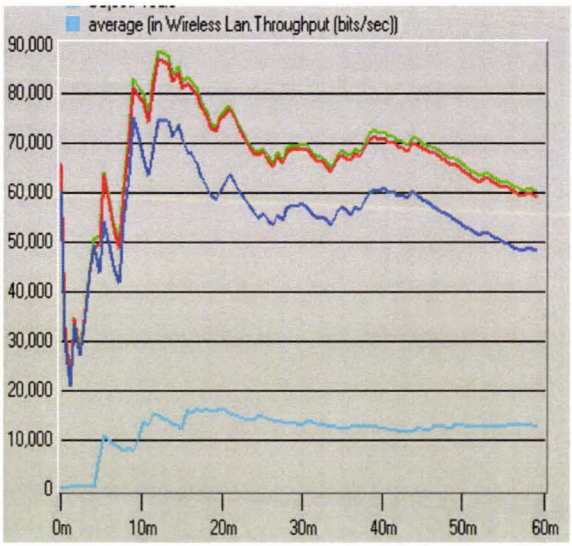
Analysis: From the figure, we can find that the situation is similar as to the 802.11a. When the load is heavy, the throughput tends to be same. The reason is there is no priority in DCF.

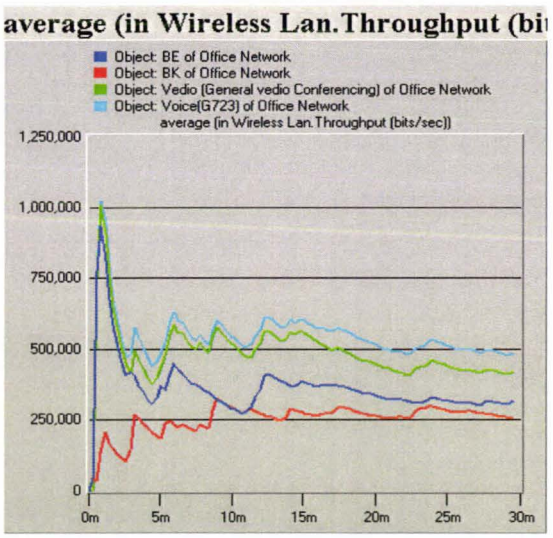
6.1.1.3 Performance of RT-traffic over DCF base on 802.11g





(a) Delay in Light Background Loading **(b) Delay in Heavy Loading**





(c) Throughputs in Light Loading **(d) Throughputs in Heavy Loading**

Figure 6-10 Performance of RT-traffic Over DCF Base on 802.11g

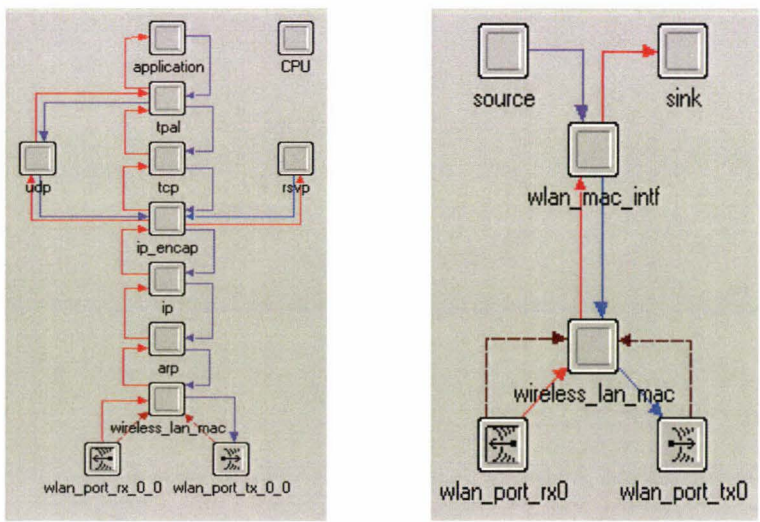
Analysis: The performance of the RT-traffic over the 802.11g is similar as over the 802.11a. There is little difference performances between them. We can compare figure 6-7 to 6-10(a), we can find, under the light loading, the average delay of the RT-traffic is less than in 802.11a, of course they all can ensure the QoS. Compare figure 6-8 to 6-10(b), in the heavy loading, 802.11a monotone increase to lost control, the 802.11g tends to be stable, however, it increase faster than 802.11a. It depends on the little different PHY layer. Generally, they are similar; because both of them use the OFDM construct at the PHY layer.

6.1.2 802.11e EDCF model develop

As the IEEE is yet approve the 802.11e, so the new version OPNET (10.5) did not include the EDCF protocol. We developed an EDCF model by the mechanism that was introduced in chapter 5. EDCF is based on DCF. The main difference from DCF, is classifying the traffic to 4 AC. And sets different defer AIFS and different back-off time for different traffic to access medium with different priorities respectively. The mechanism can reduce the RT-traffic delay.

The developing is based on DCF. OPNET provides two structures for WLAN station. One is from application (see figure 6-11a), another just uses the source block to indicate

the higher layer over the MAC layer (see figure 6-11b). Because our study focuses on MAC layer, so we employ the later.



(a) (b)

Figure 6-11 WLAN Station Model

Then we create a QSTA station in WLAN object palette (is shown in figure 6-12), and add some item in the QSTA interface (see in 6-13). QSTA station must classify the traffic as the EDCF protocol and give RT-traffic high priorities to access medium to reduce the transmission delay.

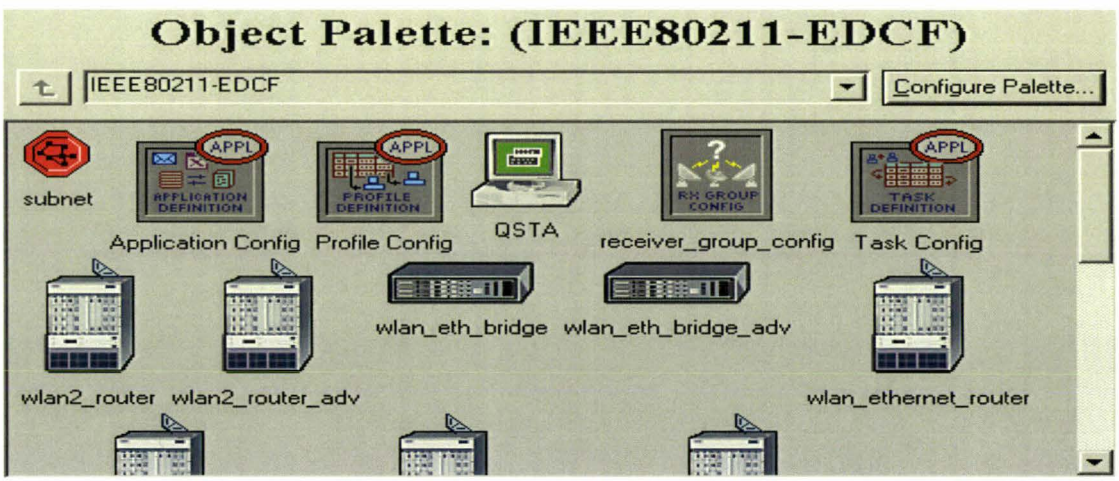


Figure 6-12 QSTA Node Model

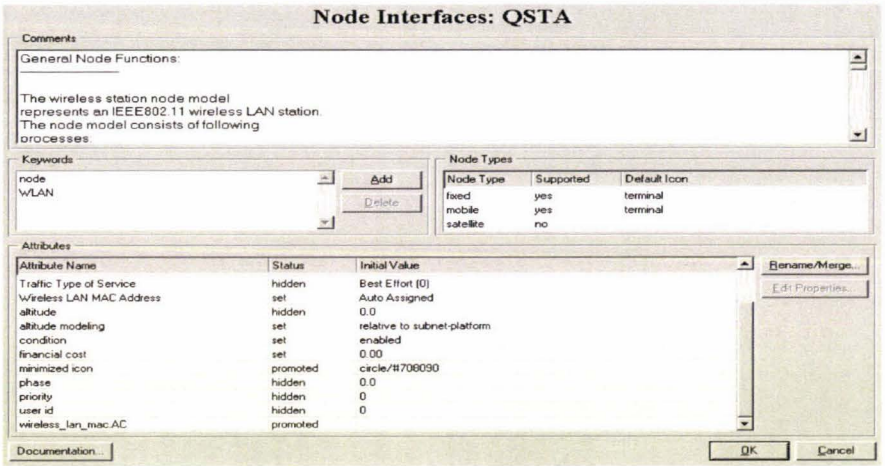


Figure 6-13 QSTA Interface

6.1.2.1 source_qos , sink_qos model and wlan_mac_intf_qos model

The function of the source processor block is generation of bursty packets to simulation applications. The process of the source_qos FSM (Finite State Machine) is show in figure 6-14. The function of source_qos is: generate the packets, specifies the parameters of the traffic pattern that will be generated by this traffic source.

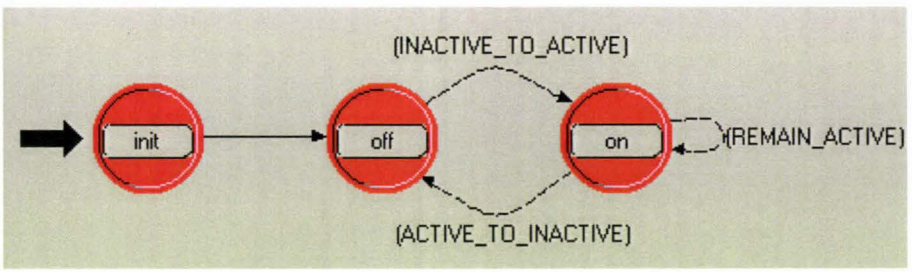


Figure 6-14 the Source_qos Process

The sink_qos FSM process is shown in figure 6-15 The function of the sink_qos model is: The sink process model accepts packets from any number of sources and discards them regardless of their content or format.

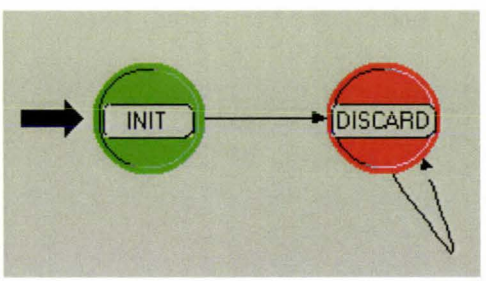


Figure 6-15 the Sink_qos Process

The wlan_mac_intf_qos FSM process is shown in figure 6-16:

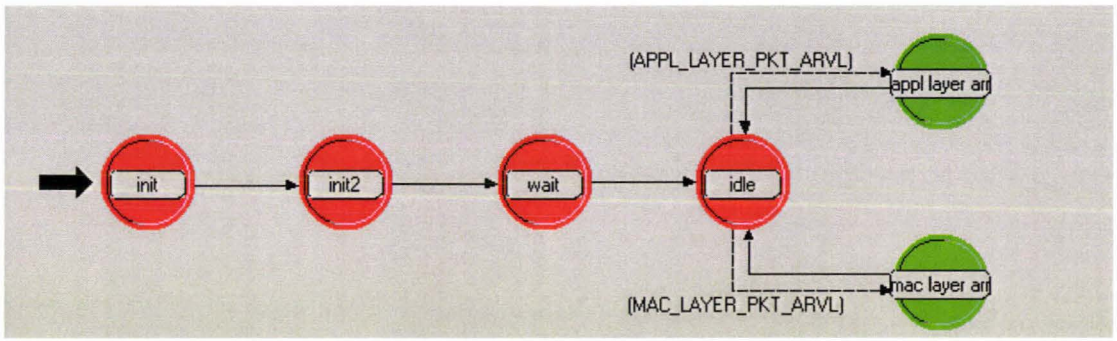


Figure 6-16 the Waln_mac_intf_qos Process

This wlan_mac_intf_qos model is an interface between MAC layer and higher layer. The function of this process is to accept packets from higher layer and generate random destination address for them. This information is then sent to the MAC layer.

6.1.2.2 wlan_mac_qos model

QSTA provides the main priority mechanism based on EDCF protocol is from wlan_mac_qos_model. The process state figure is shown in figure 6-16:

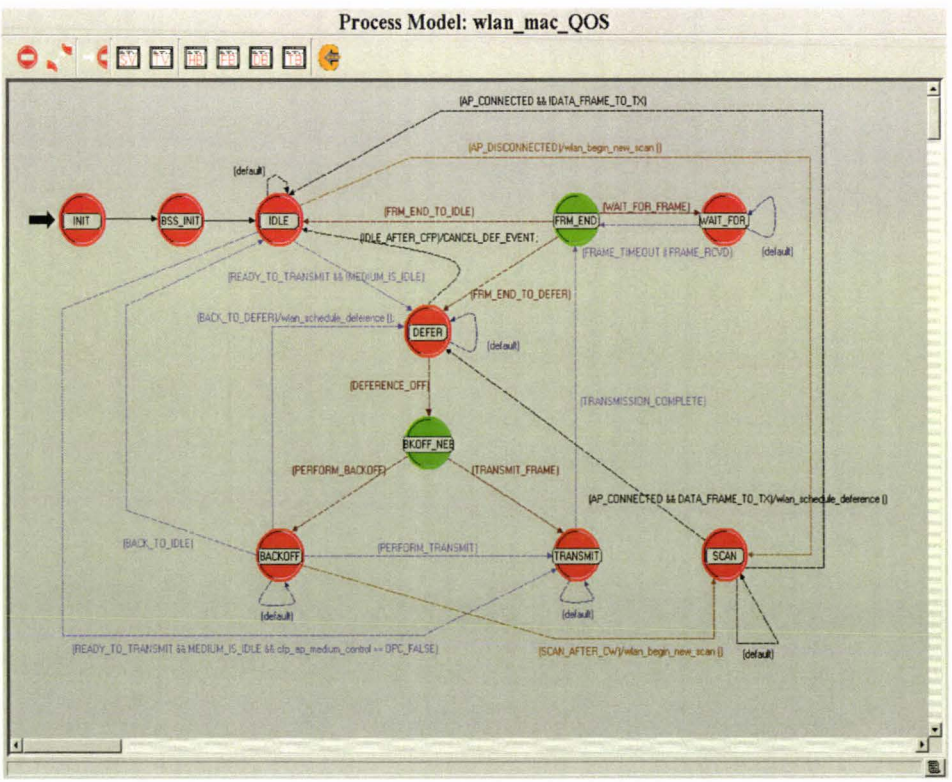


Figure 6-17 the Waln_mac_qos Process

The difference between DCF and EDCF is in this process. In fact, this model runs as an EDCF MAC layer protocol. The role of the wlan_mac_qos is to accept data packets from the higher layer protocols, encapsulate this data into WLAN frames, and to send these frames as their different priorities to the destination station. It provides efficient and fair sharing of bandwidth among all the stations attached to the WLAN. It provides the QoS of the RT-traffic. It provides Collision avoidance and deference is handled by the different traffic.

The state variables code block  determines the state variable by the C language. We determined some state variable for EDCF.

```
/* Read the minimum contention window size from the model attribute.      */
int \cw_min;

/* Read the maximum contention window size from the model attribute.      */
int \cw_max;

/* Read the temporary maximum contention window  size from the model attribute.  */
int \cw_min_tmp;

/* Read the temporary minimum contention window  size from the model attribute.  */
int \cw_max_tmp;

/* DIFS interval is used by the stations to transmit data frames.         */
double  \difs_time;

/* AIFS interval is used by the stations to transmit data frames by the priority. */
double  \aifs_time;

/* Delay in seconds to transmit PLCP Preamble and PLCP Header at lowest  */
/* mandatory data rate of the physical layer for WLAN control frames.     */
double  \plcp_overhead_control;
```

Figure 6-18 SV Block for EDCF

State variants can keep on their value in the whole model, they can use in different process FSM in wlan_mac_qos model of the QSTA node. Figure 6-18 shows a part of the source codes in SV block of the wlan_mac_qos, the part of key C source codes of the EDCF is attached in the Appendix C.

Then we must determine the temporary variables code blocks  . This code block determines the temporary variable. When the state comes to back the kernel Control, the value is lost.

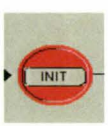
The head code block  determines the global variables; those variables are keeping on in the whole node. Meanwhile, we determine the head file, macro, constant, structure, data type, and function statement.

The function code block  always has very huge codes, because all function the state figure used is determined in this section. In EDCF developing, we write some function from the EDCF protocol mechanism.

```
5420            {
5421            /* EIFS is required. We need to use the larger of rcv_idle_time + EIFS  */
5422            /* and NAV + AIFS since EIFS period starts when the receiver becomes    */
5423            /* idle regardless of the status of virtual carrier sensing (section    */
5424            /* 9.2.3.4).                                                            */
5425            if (rcv_idle_time + eifs_time >= nav_duration + aifs_time)
5426                deference_evh = op_intrpt_schedule_self ((rcv_idle_time + eifs_time), WlanC_Deference_Off);
5427            else
5428                deference_evh = op_intrpt_schedule_self ((nav_duration + aifs_time), WlanC_Deference_Off);
5429
5430            /* After an EIFS period, a backoff is needed.                           */
5431            if (wlan_flags->cw_required == OPC_TRUE)
5432                wlan_flags->perform_cw = OPC_TRUE;
5433            else
5434                wlan_flags->backoff_flag = OPC_TRUE;
5435
5436            /* Reset the EIFS flag.                                                 */
5437            wlan_flags->wait_eifs_dur = OPC_FALSE;
5438            }

5472
5473        else
5474            {
5475            /* If the station needs to transmit or retransmit frame, it will */
5476            /* defer for NAV duration plus  AIFS duration and then backoff   */
5477            deference_evh = op_intrpt_schedule_self ((nav_duration + aifs_time), WlanC_Deference_Off);
5478
5479            /* Before sending data frame or Rts backoff is needed. */
5480            wlan_flags->backoff_flag = OPC_TRUE;
5481            }
5482        }
5483
5484    /* Reset the updated NAV flag, since as of now we scheduled a new   */
5485    /* "end of deference" interrupt after the last update.              */
5486    wlan_flags->nav_updated = OPC_FALSE;
```

Figure 6-19 a Part of the Function Code of EDCF

The diagnostic code block  is for diagnostic the whole program by the determined C/C++ sentences. Those sentences output the diagnostic information to the peripherals. The termination code block  determines some C/C++ sentences top destroy the process and release the memory.

After we had finished those determined by the code writing, the next work is to determine every FSM source code, because we are based on the DCF model, so the INIT FSM  , the BSS_INIT FSM  , the DEEFER FSM  and BACKOFF_NEED FSM  , we need to explore the C/C++ code.

INIT state: Do initialization of the process model and parameter. Loading all the attributes are loaded in this routine, Schedule a self-interrupt to wait for MAC interface to

move to next state after registering. Obtain the process's process handle. Obtain the values assigned to the various attributes Obtain the name of the process. Determine the assigned MAC address, which will be used for address resolution. Determine the assigned access categories.

BSS_INIT state: Does initialization of the process model and parameter. Determine the assigned access categories and assigned MAC address. Schedule a self-interrupt to wait for MAC interface to move to next state after registering. In fact the BSS_INIT stat is the similar as the INIT stat. It is not the redundance, because that in the simulation start (0:00), many models need begsim intrpt to do initialization; however some model must depend on the result of other model initialization. However, the kernel can't assign the reasonable list in the same time, so sequence them by the kind of way.

DEEFER FSM state: This state defers until the medium is available for transmission interrupts that can occur in this state are: Data arrival from application layer; Frame (DATA, ACK, RTS, CTS) rcvd from PHY layer; Busy intrpt stating that frame is being rcvd; Collision intrpt stating that more than one frame is rcvd; Deference timer has expired (self intrpt). For Data arrival from application layer queue the packet. Set back off flag if the station needs to back off after deference because the medium is busy. If the frame is destined for this station then set frame to respond and set a deference timer to SIFS. Set deference timer to SIFS and don't change states. If receiver starts receiving more than one frame then flag the received frame as invalid frame and set deference to EIFS. In fact, we has determined different defer time for differenit traffic aifs_slot for get the reasonable EIFS time in the wlan_schedule_deference () in the function code block.

BACKOFF_NEED state: In this state we determine whether a back off is necessary for the frame we are trying to transmit. It is needed when station preparing to transmit frame discovers that the medium is busy or when the station infers collision. Back off is not needed when the station is responding to the frame. Following a successful packet transmission, again a back-off procedure is performed for a contention window period as stated in 802.11e standard for different traffic as different minimum contention window

period. If back off needed, it assigns different back off period to different traffic as their AC as stated in the 802.11e, then check whether the traffic completed its back off in the last attempt. If not then resume the back off period as the 802.11e state.

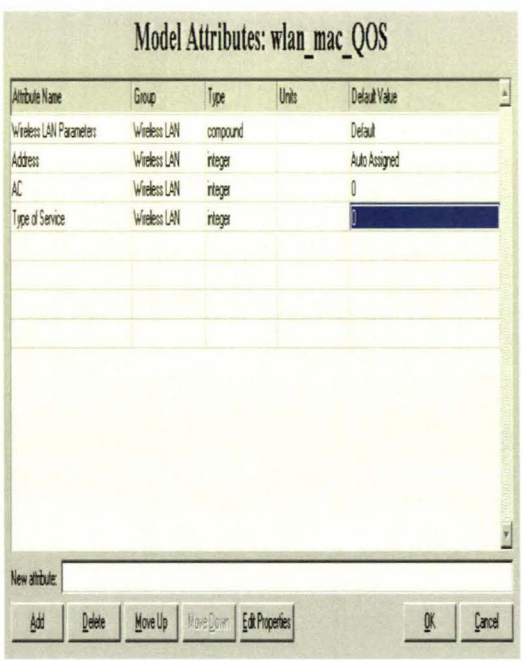
The follows is the part of the BACKOFF_NEED state source code, detail is attached in the Appendix D.

```
/* Checking whether backoff is needed or not.
if (wlan_flags->backoff_flag == OPC_TRUE || wlan_flags->perform_cw == OPC_TRUE)
        {
        if (backoff_slots == BACKOFF_SLOTS_UNSET)
                {
                /* Compute backoff interval using binary exponential process.      */
                /* After a successful transmission we always use cw_min.                   */
                if (retry_count == 0 || wlan_flags->perform_cw == OPC_TRUE)
                 {
                        /* If retry count is set to 0 then set the maximum backoff*/
                        /* slots to min window size.
        */
switch (ac)
                {
                case 0: {
                      max_backoff = ((cw_min+1)/2)-1;
                       min_backoff = ((cw_min+1)/4)-1; break;
                       }
                case 1: {
                       max_backoff = cw_min;
                       min_backoff = ((cw_min+1)/2)-1;  break;
                      }
                case 2: {
                          max_backoff = cw_max;
                         min_backoff = cw_min; break;
                                                  }
                      case 3: {
                           max_backoff = cw_max;
                           min_backoff = cw_min; break;
                                                  }
                         }
                        max_backoff = cw_min;
                        }
                   else
                        {
        /* We are retransmitting. Increase the back-off window size. */
        switch (ac)
             {
        case 0: {
              max_backoff = ((cw_min+1)/2)-1;
             min_backoff = ((cw_min+1)/4)-1 ;break;
                  }
            case 1: {
                  max_backoff = cw_min;
                   min_backoff = ((cw_min+1)/2)-1 ;break;}                    }
                                  case 2: {
                                max_backoff = cw_max;
                                min_backoff = cw_min; break;
                                                  }
                           case 3: {
                                  max_backoff = cw_max;
                                     min_backoff = cw_min; break;
                                        }

                           }
```

```
                    max_backoff = max_backoff + max_backoff_tmp;
                    max_backoff_tmp = max_backoff;
                }
/* The number of possible slots grows exponentially until it exceeds a fixed limit.        */
                if (max_backoff > cw_max)
                    {
                    max_backoff = cw_max;
                    }

/* Obtain a uniformly distributed random integer between 0 and the minimum contention window size. Scale the number of slots
according to the number of retransmissions.*/

backoff_slots =floor (op_dist_outcome (op_dist_load ( "uniform_int ", min_backoff, max_backoff)));
                }


        /* Set a timer for the end of the backoff interval.                    */
        intrpt_time = (current_time + backoff_slots * slot_time);


        /*restore the cw_max  value for next time using max_backoff = cw_max_tmp;

        /* Scheduling self interrupt for backoff.                              */
        if (wlan_flags->perform_cw == OPC_TRUE)
                backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_CW_Elapsed);
        else
backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_Backoff_Elapsed);

        /* Reporting number of backoff slots as a statistic.                                   */
        op_stat_write (backoff_slots_handle, backoff_slots);
        op_stat_write (backoff_max_handle, max_backoff);
        op_stat_write (backoff_min_handle, min_backoff);
        }
```

6.1.2.3 QSTA attribute and node interface

For communication between sources codes we must determined the QSTA node attribute and node interface.



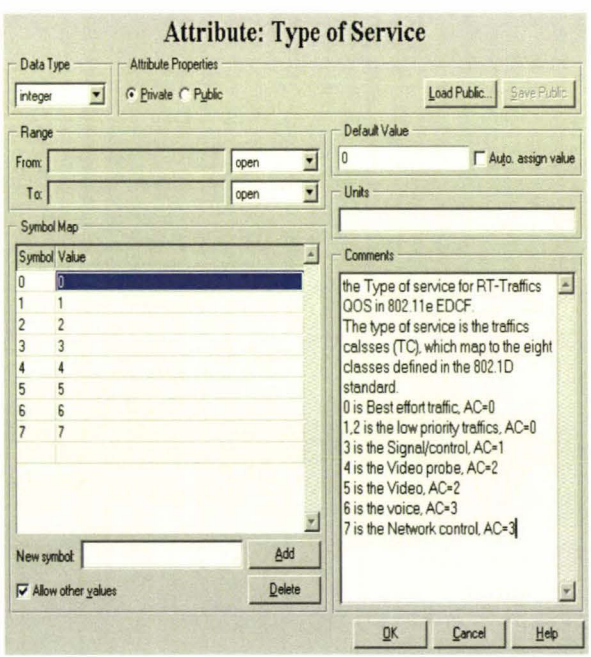


Figure 6-20 Model Attribute **Figure 6-21 Type of Service**

Figure 6-20 shows the QSTA attribute determined; figure 6-21 shows the different traffic type of service determined.

For study of the fairness of the EDCF, we determine some statistic handle in the local and the global statistic; it is shown in figure 6-22.

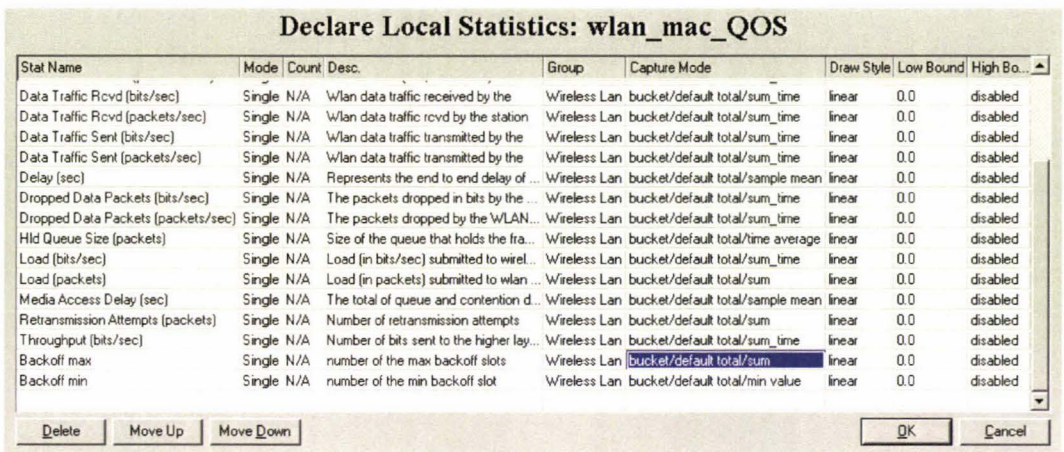


Figure 6-22 the Local Statistics Determined

When all is finished we can compile all model source code, the EDCF model developing is down. It is shown in figure 6-23

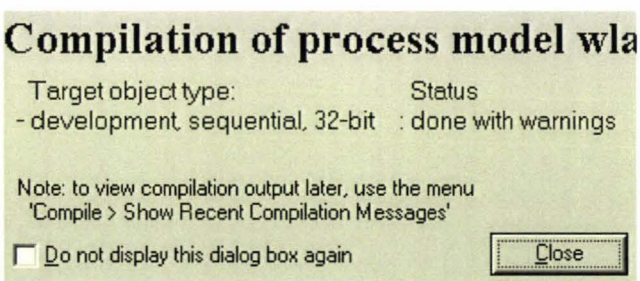


Figure 6-23 Compilation Process

6.1.3 Performance of RT-traffic over EDCF

From the simulation for the DCF based on 802.11b,a and g, we can understand clearly the performance of RT-traffic based on the original 802.11 MAC protocol. In the light loading, in the highest speed rate (11Mpbs) based on 802.11b, the delay is the same; because of no priority there. And the delay time results are acceptable. Actually, there are rarely so light loading that exists in the real world. In the heavy loading, because there is

no priority in DCF, all delay is the same, so the result is unacceptable. The situation is the same in the 802.11a and g. however; there is difference in detail due to difference in the mechanisms for OFDM respectively.

At this stage, we need no simulation based on every PHY layer. Because the situation should be similar, we focus on EDCF protocol based on 802.11b. The experimental parameter is shown in table 6-1

Table 6-1 the Parameter for EDCF

AslotTime	20us			
aFragmentation Threshold	1024Byte			
SIFS	20us			
DIFS	60us			
AIFS	60+aslotTime*20(us)			
ACWmin	**Voice**	**Video**	**BF**	**BK**
	7	15	15	15
ACWmax	**Voice**	**Video**	**BF**	**BK**
	15	31	1023	1023
Dot11DaultCPTXOPLimit	5040us			
MAC header	38bytes			
PLCP header length	24bytes			
ACK size	14bytes			
PHY rate	11Mbps			
Bite Error Rate	BER11=1.3E-5,BER1=0			

We also employ four kinds of traffic as the AC category: Voice (AC=3), Video (AC=2), Best effort traffic (AC=1), Background traffic (AC=0). We divided the simulation to 10 scenarios as the percent of the total loading, from 10% to 100% of the 11Mpbs as 802.11b. In all scenarios, the voice traffic are the same, it is the G729 encoded speech, it is about 64kbps (8kByte/second). The Video traffic is different, in scenarios 1(10%) to 2 (20%), it is the low definition type Videoconference; it is the 128*120 pixels/ frame and the 10 frames /second. The loading throughput is the 153.6Kbps (19.2Kbyte/second). In scenarios 3(30%) to5 (50%), it is the general definition type Videoconference; it is the 128*240 pixels/ frame and the 15frames /second. The loading throughput is the 460.8Kbps (57.6Kbyte/second). In scenarios 6(60%) to10 (100%), it is the high definition type Videoconference; it is the 352*240 pixels/ frame and the 30frames /second. The loading throughput is the 2534.4Kbps (316.8Kbyte/second). The BE and BK increase as

the scenarios, it is shown in table 6-2. Typical result for scenarios 1, 3 and 10 will be presented next.

Table 6-2 Traffic Loading for EDCF Simulation

Traffic loading (Kpbs)					
Scenarios	Voice	Video	BE	BK	Total
0	0	0	0	0	0
1	64	153.6	320	512	1049.6
2	64	153.6	880	1100	2197.6
3	64	460.8	1280	1500	3304.8
4	64	460.8	1800	2000	4324.8
5	64	460.8	2100	2800	5424.8
6	64	2534.4	1800	2200	6598.4
7	64	2534.4	2300	2800	7694.8
8	64	2534.4	2800	3400	8798.4
9	64	2534.4	3200	4156	9890.4
10	64	2534.4	3700	4732	11030.4

6.1.3.1 Delays over EDCF

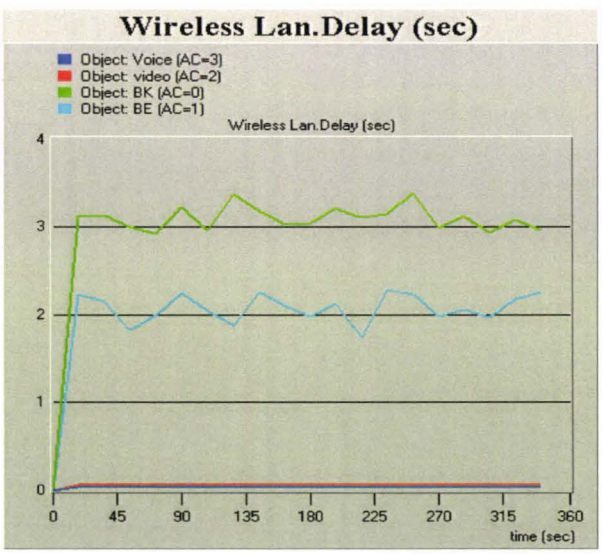


Figure 6-24 Traffic delay in the 10% loading

Analysis: Different from DCF, in the light loading, the traffic delay is different. We can find that the delay time of Voice just about 3.5ms. The delay of the Video traffic is just in round 6ms. It is the acceptable result. We can notice the delay time of BK is around 2.1s

and BE is around 2.9s. It looks more than in DCF. So we can know, when the loading is light, the QoS performance of RT-traffic can be guaranteed over the EDCF.

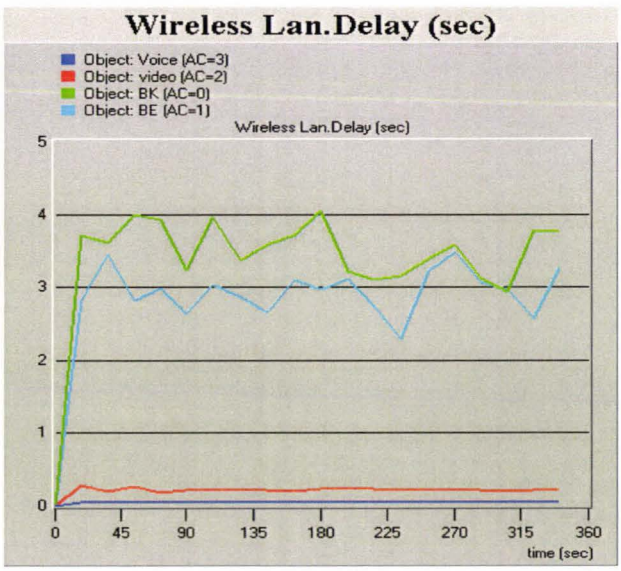


Figure 6-25 Traffic Delay in Scenario 3

Analysis: in the 30% throughput loading, we can find that the delay time of Voice also about 3.5ms. The delay of the Video traffic is increase to round 7ms. The results of the RT-traffic delay are also acceptable. The delay time of BK is around 2.8s and BE is around 3.7s. It means that in the 30% loading base on 802.11b, the QoS performance of RT-traffic can be guaranteed over the EDCF.

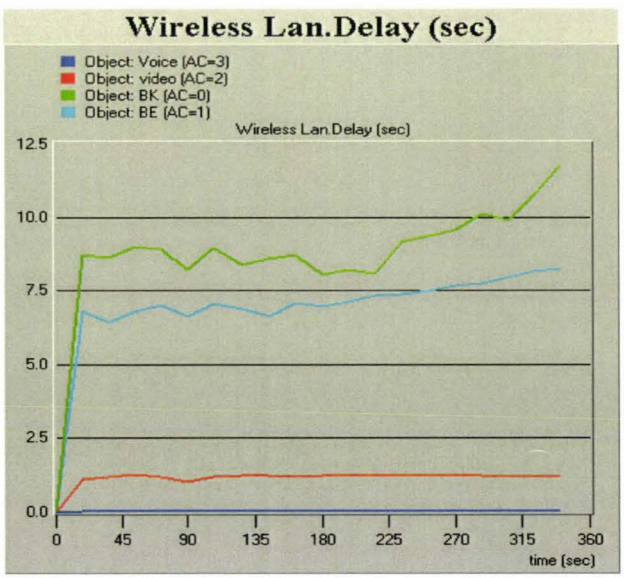


Figure 6-26 Traffic Delay in the in Scenario 9

Analysis: in the 90% of 11Mpbs throughput loading, the situation is different; we can find that although the delay time of Voice didn't change. The delay of the Video traffic is increase quickly to round 1.1s. It is the unacceptable result. Because, we know, the AC3's standard must be below the 350 ms. In fact, for some VBR traffic, the required value should be shorter than 200ms. The delay time of BK is around 6.3s and BE is around 8.5s. So we can know when the loading is in 30% base on 802.11b, the QoS performance of RT-traffic can be guaranteed over the EDCF.

6.1.3.2 Loading Throughputs over EDCF

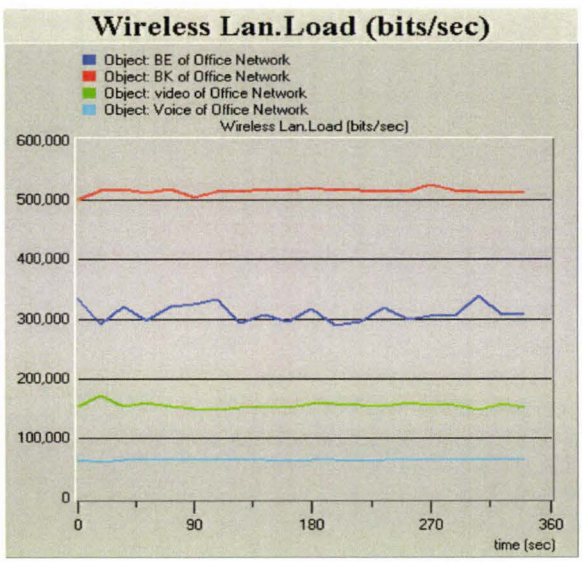


Figure 6-27 Traffic Loading Throughputs in scenario 1

Analysis: in the 10% total loading, all traffic loading throughput is full. We can find the original loading from table 6-2, the Voice loading throughput is 64Kbps, and the video loading is 153.6kbps, the BE traffic loading is 320kpbs, the BK traffic loading is 512Kpbs. From the figure, we can see the loading throughput is around the design. The curve dither is reason why the exponential distribute.

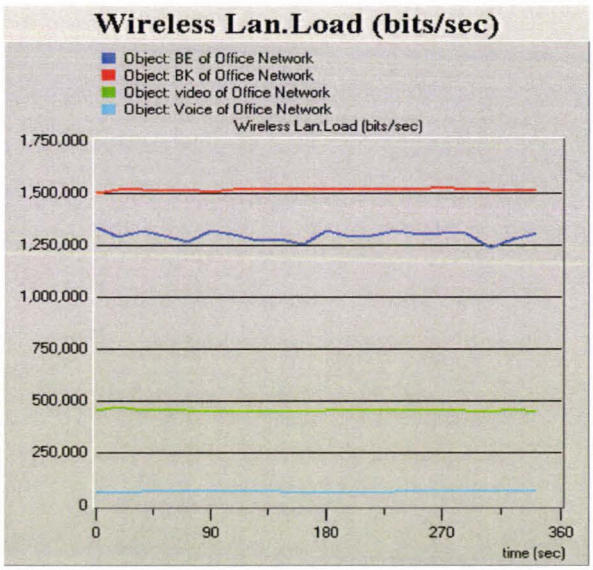


Figure 6-28 Traffic Loading Throughputs in Scenario 3

Analysis: from the figure, we can see that the every traffic loading throughput didn't changed as the priority in the 30% loading. The BK loading throughput is the 1500Kbps. The BE loading throughput is the 1280Kpbs. The video is the 460.8Kpbs. The voice is the 64Kpbs. Comparing to table 6-2; we can find they remain loading as the design.

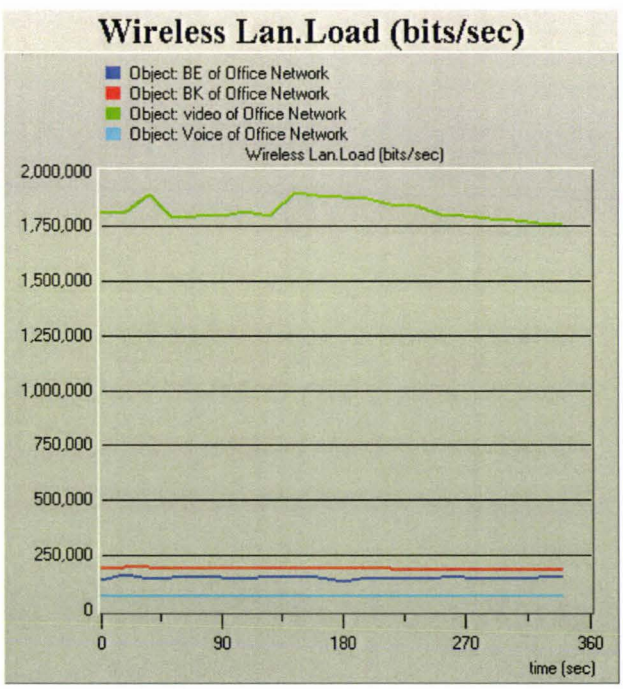


Figure 6-29 Traffic Loading Throughput in Scenario 10

Analysis: from the figure, we can find that in the heavy loading 90% of the total, the every category traffic can not get the designed loading throughput. They reduce as the priority, the BK reduce most quickly from 4156Kpbs to 188Kpbs. Its AC is 0. The BE is from 3200Kpbs to 145Kpbs. The Video traffic is from 2534.4Kpbs to 1584.8Kpbs. In other words, the video cannot guarantee the QoS performance from the loading throughput.

From the delay data and throughput data, we can find the EDCF has pro and con clearly: the advantage of the EDCF is that it can give the RT-traffic high priority to guarantee the RT-traffic's delay is in the round. In the other words, EDCF guarantee the RT-traffic performance over the WLAN. However, when the whole loading of the network is very heavy, it cannot guarantee the RT-traffic performance.

6.2 Simulation with NS-2

NS (Network Simulation) derives from REAL network simulator that was built in 1989. In 1995, NS developing group obtains the support from DARPA (Defense Advanced Research Projects Agency): it is approved by VINT (Virtual Internet Test-bed) project. NS is an open structure system with easy extension. NS-2 has a great deal of the existed wireless model that is provided source codes by UCB Daedelus [68], CMU Monarch plan and SUM Corporation.

6.2.1 Explore HCF model by NS2 simulator

6.2.1.1 Discrete events machine (DEM)

The mechanism of NS-2 is the similar as the OPNET, which is a discrete events machine. DEM is a one of systems that used to be employed for the simulation. The mechanism of NS-2 is that: events change the system states. The system state just is changed while the events happening. In the HCF simulation system, typical events include packets arrive; over time etc. the simulation time is promoted by the time of the events happening. An

event executive should trigger the followed event starting. As like this, the DEM is executed events one by one until out of the events.

The kernel of the NS-2 is the discrete events simulation engineer. There is a schedule in NS, which in charge to record the current time, and schedule the queue of the events list.

6.2.1.2 Abundance object library

In fact, NS-2 can simulate any system, not only the WLAN. We compile the source codes to execute the events for HCF, which can get the wanted result for followed studying. In fact, there are many object models for WLAN 802.11 in NS-2, which is explored by other researcher. In the other words, there is abundance of object library in the NS-2.

6.2.1.3 Dividual object model

For exploring a HCF model, the difference against the OPNET is that the NS-2 compiles the model by the dividual object model. There should use two kinds of languages: C++ and the OTCL. OTCL is the object toolkit command language. It is explored by MIT. OTCL language is flexible and alternate. For HCF, we always achieve the HCF behavior as in the pervious chapter mechanism based on C++ compile, meanwhile category object correspond C++ by the OTCL language. In short, OTCL is the basic language for deploying object group and describe the simulation process. OTCL always explain behavior of the HCF for the whole system. In fact, this is the dividual object model.

Figure 6-30 shows the HCF exploring process:

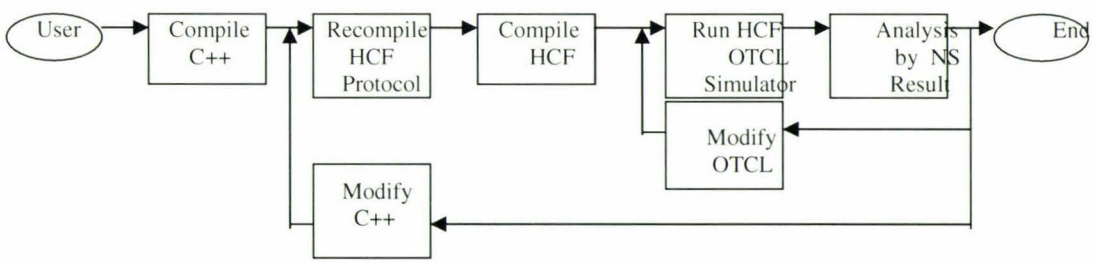


Figure 6- 30 the HCF Exploring Process by NS-2

The advantage of the dividual object model is obviously that make the whole system flexible. C++ compiles the codes for process function. On the other hand, OTCL is the

explained the execution. We can deploy the parameter without recompile the whole model.

6.2.1.4 Open source code

For NS2 is the free software, its source code is open. In fact, in this section we had downloaded some source code for HCF simulation from reference [69]. This situation makes the network simulation result easy to share. The part of key source file is shown below; the detail is attached in appendix D.

```
class queueTimer : public TimerHandler {                                    //Timer attached to
public:                                                                                    //droptail queue
    queueTimer(DropTail *d) : TimerHandler(), d_(d) { }
protected:
    virtual void expire(Event *);
    DropTail *d_;
};

void queueTimer::expire(Event *e)                    // Timer expired – increase priority
{
    printf("\nTimer expired!");
    int pr = d_->get_prio();
    printf("\nPkt priority = %d",pr);
    if (d_->length() != 0) {
        Packet *p = d_->deque();                    //remove packet from old queue
        printf("\nPacket dequed from old queue");
        PriQueue priq;
        priq.incr_prio(p,pr);
    }
    else printf("\nQueue empty!!");
}

void DropTail::enque(Packet* p)
{

    PacketQueue* q_ = pq_;
    save_prio(p);
    if (pqlow_ && (HDR_IP(p)->prio() > 0)) {
        if (pq_->length() >= pqlim_) {
            q_ = pqlow_;
        } else {
            assert(!pqlow_->length());
        }
    }

     if (q_->head() == 0)                           //first packet – set timer
        qTimer.resched((double)timeout);
     q_->enque(p);


    if (q_->length() >= limit()) {
        if (drop_front_) { /* remove from head of queue */
            Packet *pp = q_->deque();
            drop(pp, DROP_IFQ_QFULL);
        } else {
            q_->remove(p);
            drop(p, DROP_IFQ_QFULL);
 }
    }
}

void DropTail::enque_pkt(Packet* p)                    //enque packet in new queue
{
    if ( length() == 0)
        qTimer.resched((double)timeout);
    enque(p);
```

```
    printf("\nPacket enqued in new queue!!");
    if(!blocked())
      resume();
}

Packet* DropTail::deque()
{
    Packet* p = pq_->deque();
    qTimer.resched((double)delay);          // packet removed from queue – reset
    if (!pqlow_) {                                  // timer for queue
        return p;
    }

    if (!p) {
        assert(!pqlim_ || !pqlow_->length());
        return pqlow_->deque();
    }

    if ((pq_->length() < pqlim_) && pqlow_->length()) {
        pq_->enque(pqlow_->deque());
    }

    return p;
}
```

6.2.1.5 The parameter for Simulation of HCF by NS-2

From the exploring of the HCF, we can get a QSTA node base on HCF protocol. Now we can simulate the performance of RT-traffic based on HCF. We determine four traffic in the WLAN for each QSTA; they are voice (AC3), video (AC2), BF (AC1) and BK (AC0). Because we had studied the performance of the RT-traffic in 802.11a, b and g based on DCF and EDCF, so in this stage, we just employ the 802.11a in 36Mpbs as the PHY layer. This is enough data to study the performance of RT-traffic QoS based on the HCF. For the WLAN, we deployed some parameters as the real 802.11a WLAN, it is shown in table 6-3.

Table 6-3 the Parameters of HCF Simulation

AslotTime	9us			
aFragmentation Threshold	1024Byte			
CCA time	4us			
SIFS	16us			
DIFS	25us			
AIFS)	16+aslotTime*15(us)			
ACWmin	**Voice**	**Video**	**BF**	**BK**
	7	15	15	15
ACWmax	**Voice**	**Video**	**BF**	**BK**
	15	31	255	255
Dot11CAPrate	21us			
DotCAPMAX	5040us			
MAC header	38bytes			
PLCP header length	0.5bytes			
ACK size	14bytes			
CAP timer update time	5120us			
PHY rate	36Mbps			
Bite Error Rate	BER11=1.3E-5,BER1=0			

6.2.2 The performance of RT-traffic over HCF

We chose traffic: voice is G7.32 speech : packet-size is 160bytes , packet interval is 20ms, so the data rate is 64kbites/second. The access category (AC) is 3. The Video is the general videoconference: packet-size is 1280bytes , packet interval is 10ms, so the data rate is 1024kbites/second. The access category (AC) is 2. The Best-Effort traffic is: packet-size is 1500bytes , packet interval is during in 12.5ms, so the data rate is 960kbites/second. The access category (AC) is 1. The Back-Ground traffic is: 1500bytes , packet interval is during is 10ms, so the data rate is 1200kbites/second. The access category (AC) is 0. The total data rate is 3248kpbs. For study of the performance, we increase the QSTA numbers to increase the loading and obtain different performance data of the traffic under different loading. In this chapter, we divided two scenarios. One is in the 10% (1 stations, around 3.3Mpbs) light loading, another is the in 70% (7 stations around 23.1Mbps) heavy loading.

6.2.2.1 Delay of RT-traffic over HCF

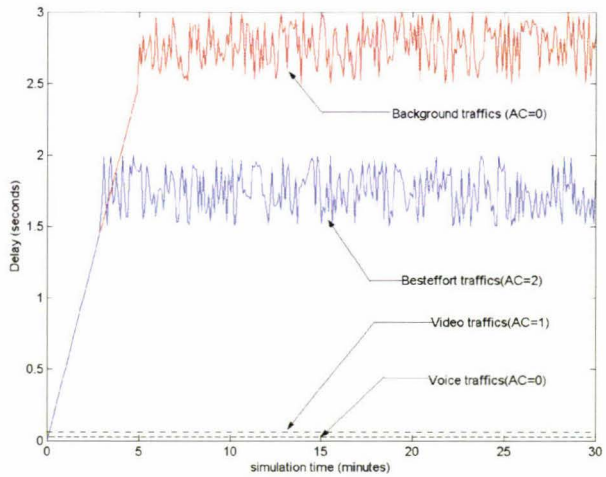


Figure 6- 31the Delay in the Light Loading (10%)

Analysis: from the figure we can find the Voice delay is round the 25ms, it is accepted by the AC3 below the 150ms. The Video is around 60ms; it should be acceptable as the regular rule that below the 200ms. In the light-loading situation, the BF delay about 1.75s, and the BK is the 2.75s. It is just similar as the EDCF. The performance of RT-traffic over QoS over the HCF under the light loading is can guarantee.

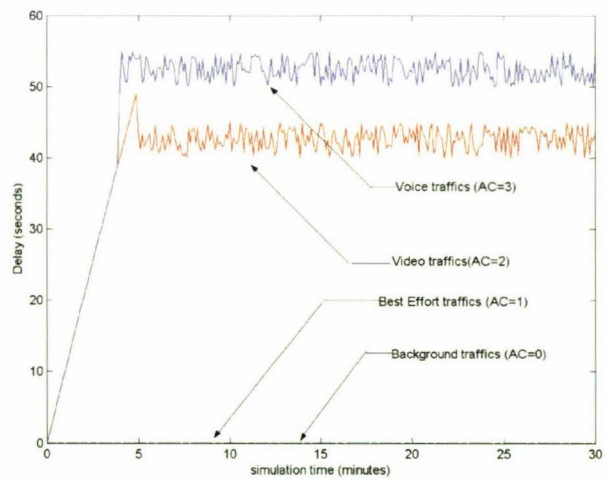


Figure 6-32 the Delay in the Heavy Loading (70%)

Analysis: from the delay we can find the voice delay is round 70ms, video delay is round 121ms, although the delay is the little bit increasing, it is the acceptable result with a bad grace for the QoS performance of the RT-traffic. In the other words, the HCF can

guarantee the RT-traffic performance under the heavy network loading. It is the better than EDCF. While we notice, the low priority traffic delay increase quickly.

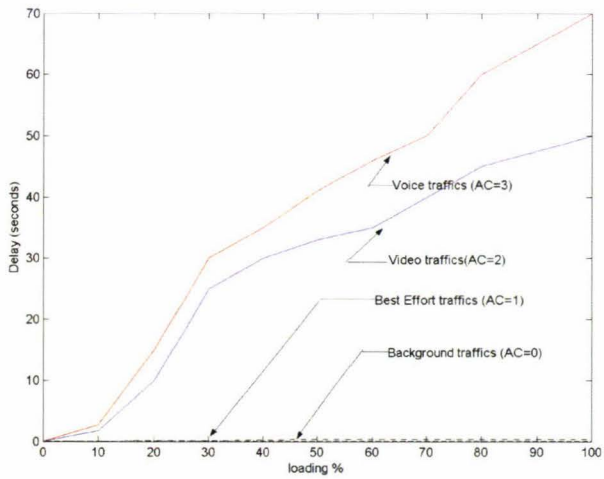


Figure 6-33 the Total Delay of HCF

Analysis: in every loading percent, from 10% loading to 100%, the voice delay just from 25ms increase to 75ms; the video from 60ms increase to 200ms. Meanwhile, the curve is the stable in the end. It is means the HCF protocol can better support the QoS performance of the RT-traffic over the WLAN. But the delay of the low priority traffic BK and BE, increase very quick after 30% it is result the low priority traffic starve. This should result bad fairness in the network; we should discuss it in the next chapter base on those data.

6.2.2.2 Throughput of RT-traffic over HCF

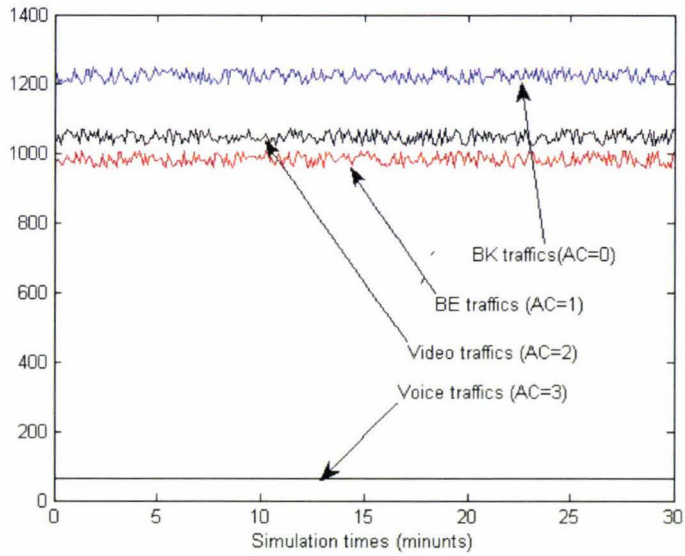


Figure 6- 34 the Throughputs in the Light Loading (10%)

Analysis: all traffic throughput did not reduce under the light loading. From the throughput we know that performance of the RT-traffic can guarantee over the HCF.

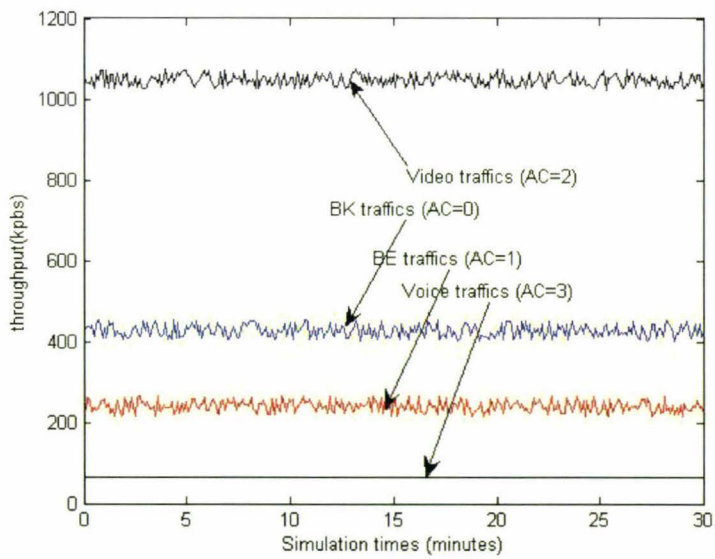


Figure 6- 35 the Throughputs in the Heavy Loading (70%)

Analysis: from this figure, we can find that the throughput of the low priority reduces very quickly. The BE traffic (AC0) reduce to round 400kpbs from 1200kpbs. The BF

from 960 reduces to round 200. It should let the low priority traffic starve. This should result bad fairness in the network; we should discuss it in the next chapter base on those data.

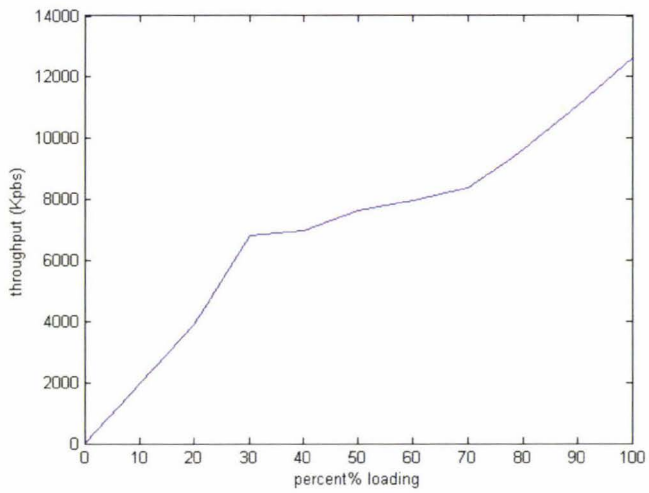


Figure 6- 36 the Total Throughput

Analysis: from the total throughput figure, we can find that the actually throughput is about around the 60% of the total throughputs. The reason why is the HCF must organizing the traffic as the determined AP, and many command packets, say ACK, beacon, and PCLP header etc. Those consumes some network source, in fact too mach overhead is the problems of those protocols.

Summary

This chapter used OPNET and NS-2 to simulate the QoS performance of the RT-traffic based on the WLAN 802.11 protocols. Those protocols includes the 802.11a, b and g in the PHY layer, based on PHY layer we simulated the RT-traffic in MAC layer original protocol and upcoming 802.11e. By those simulation and analysis, we can understand the QoS performance of the RT-traffic over WLAN clearly. By those simulations, we also get data that we can study other related problems about the QoS performance of the RT-traffic, and predict the QoS performance of the RT-traffic in the future. That will be discussed in the next chapter.

CHAPTER 7

SIMULATION DATA ANALYSIS AND RESULTS

From the results of simulation in chapter 7; we can understand the QoS performance of RT-traffic based on the 802.11 protocol. As discussed in chapter 1, the main performance metrics for RT-traffic is traffic delay, jitter and packet loss. Because the jitter and packet loss are changed with the traffic delay, the traffic delay is the most important factor for the performance of the RT-traffic.

For DCF, the RT-traffic delay time is the same due to no priority. From the 6.1.1.1 simulations, base on 802.11b PHY, the RT-traffic performance are not acceptable in both for light and heavy loading. From the 7.1.1.2 and 7.1.1.3 simulations, the performance is better based on 802.11a PHY. In the light loading, the voice and video delay just around 0.15ms, it is acceptable performance. In the heavy loading, the delay of all categories of traffic is over 500ms and unstable. It cannot be acceptable. Based on 802.11g, the performance of RT-traffic is similar as for the 802.11a; it is due to the PHY based on OFDM mapping.

For EDCF, different categories of traffic delay times are different. It is due to the fact that EDCF provides the priority mechanism for guaranteed the RT-traffic (time-sensitive) transmission in the round shorter delay times. From the 7.1.3.1 simulation, we can find the delay times in the light loading and general loading, the voice delay time is just the in 3.5 ms; the video delay time is just within 6ms and 7ms respectively. It is the acceptable performance. In other words, the EDCF guarantees the performance of RT-traffic QoS over the WLAN, in the general situation. However, for heavy loading, the video delay time increase very quickly. In the 90% loading, the delay time is out of the 1000ms, which cannot be acceptable at all.

For HCF, because of the priority mechanism, different categories of traffic delay times are different. From the 7.2.1.1 simulations, we can find that the voice delay time is increase from 25ms to 75ms as the loading increase providing acceptable performance. However, it is larger than EDCF. The video delay time increase from 60ms to 200ms.

Although the delay time is increased, different from the EDCF, in the heavy loading, the HCF can guarantee the RT-traffic QoS performance.

From the simulation of the chapter 7, we can concluded easily that the original 802.11 MAC protocols do not provide the priority, and can't guarantee the RT-traffic QoS performance except the bandwidth is big enough in 802.11a and 802.11g PHY layer. For enough bandwidth, the conclusion is the same as we discussed in chapter 2. The upcoming 802.11e provide the priority mechanism; it can guarantee the QoS performance of the RT-traffic for WLAN.

7.1 SIMULATION DATA ANALYSIS

In this section, we study the Voice (AC3) and Video (AC2) performance in terms of delay, jitter, packet loss ratio and throughput. Finally, we study the fairness for every category of traffic. In fact, although, fairness is not one of the QoS performance metric factors of RT-traffic, the fairness is related to the optimization of the WLAN and enhancing the QoS performance of the RT-traffic.

From the data of the simulation, we can get the mean delay; mean jitter and packet loss ratio figures, it is shown figure 7.1 and 7.2:

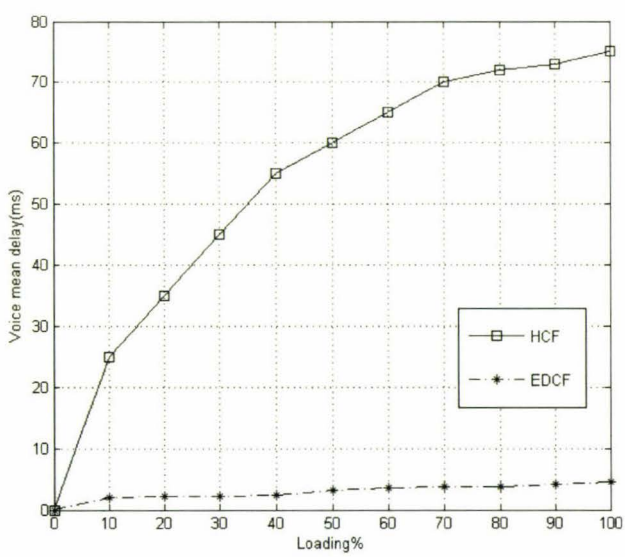


Figure 7-1 Voice Mean Delay

The main performances for voice traffic are delay, jitter and packet loss. As for delay, we collected the delay data, and calculated its mean delay in every loading throughput. From the study, we can easily find that the voice gets significantly less little time in the EDCF than HCF. In fact, from this study we can find, the delay of RT-traffic in the EDCF is not reasonable; it means EDCF gives much network bandwidth for the AC3 traffic. In other words, the low priority traffic have too less bandwidth. So in the next section, study the fairness index for these protocols. For the HCF, although the highest voices delay in the 100% loading, the delay time just around 70ms, it's also the acceptable performance for the voice. However, the EDCF is just around 3ms, it look so large distance between two protocols, it also mean the HCF has higher delay. It is due to the polling-induced overheads, unlike the contention based EDCF scheme. In other words, the HCF has too much overhead time waste.

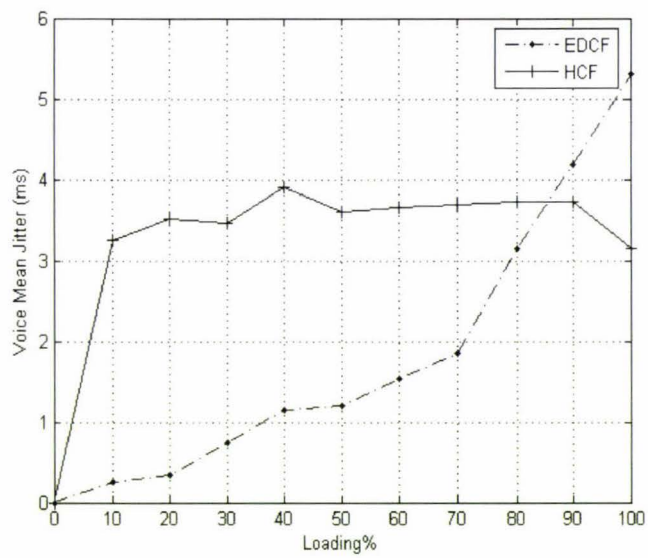


Figure 7-2 Voice Mean Jitter

Just by the unreasonable distribution of the bandwidth, although the voice gets the less delay time, the jitter between the packets is unstable. We can see: in the figure, the jitter is increased in the EDCF at the 40% loading throughput. To 70% loading, it increases very quickly. It means the jitter become unpredictable. It is the big problem in the EDCF protocol. For HCF, the jitter is even during the heavy loading. The Jitter time is acceptable, it looks like is a little bit bigger. We have analyzed the reason in the delay section.

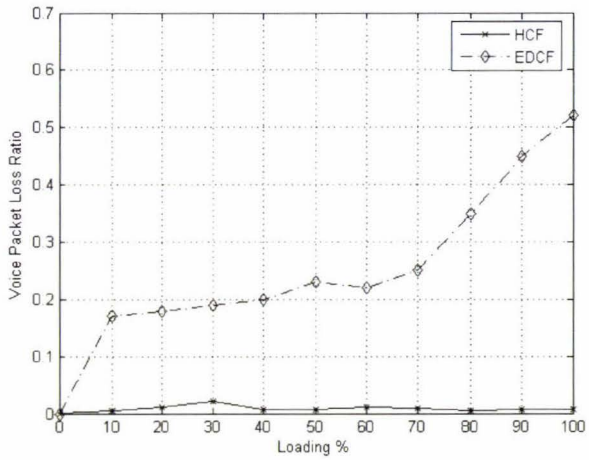


Figure 7-3 Voice Packet Loss Ratio

The HCF has better packet loss performance for voice. The reason is that the AP in HCF collects from all stations required send packets and calculate the total beacon interval, than divides them to select the reasonable SI. The SI duration is the highest sub multiple value of the 802.11e beacon interval. So when the beacon interval is divided into SI, it should be not enough for the total SIs. So in fact, the HCF calculates the packets time that want to be sent, and assigns them time by TXOP. It did not loss any packet indeed. However, in the EDCF, the situation is different, all packets must take part in the contention, so more collisions happened. Farther more, when the limit waiting time is expired, the packets are discarded.

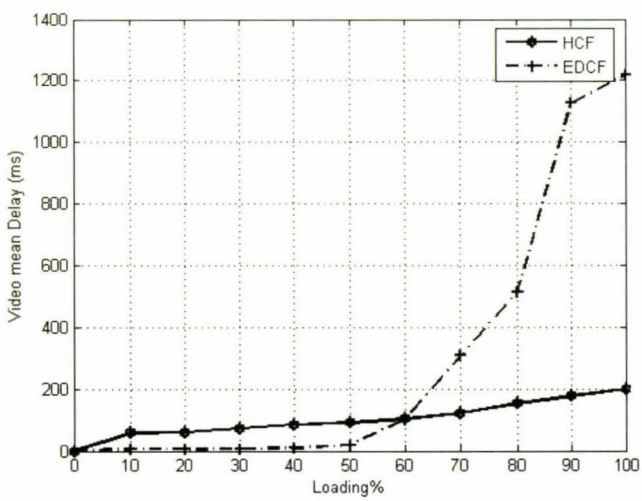


Figure 7-4 Video Mean Delays

The main performances for video traffic also are delay, jitter and packet loss.

For the mean delay time, the video traffic are different to the voice traffic. In the light-loading throughput, the situation is similar as happened to voice.

From the study, we can easily find that the voice gets the more little delay time in the EDCF than HCF.

However, when the loading throughput increases more than 50% of the total designed throughput, the delay time increased quickly. At 70%, it can get more than 300ms, and for more than 80% loading, the delay time lost control, it becomes unacceptable. From the study, we can find the delay of RT-traffic in the EDCF is not reasonable. The reason why the delay is unacceptable is that every packets in the EDCF must defer and back off. While the collisions in the channel are limited, it should be blocked. Every packet has the maximum back off time and AIFS time, resulting into bandwidth narrow. For the HCF, although the highest voice delays in the 100% loading, the delay time is just around 70ms, also an acceptable performance of the video. However, the longest delay gets 200ms; it also means the HCF faces higher delay.

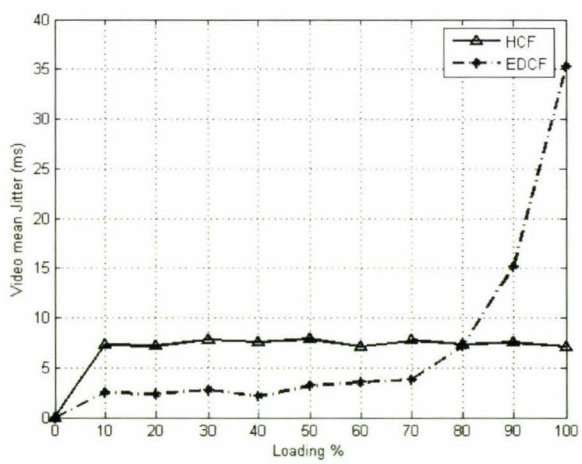


Figure 7-5 Video Mean Jitter

For jitter, the situation is similar as happened to the voice traffic. From the figure, we remind that for different categories of traffic the jitter time is in similar situation and value under the 70% loading. When the loading over the 70%, the jitter increases quickly, it corresponds with the delay. For the HCF, the situation is the same as we have discussed above.

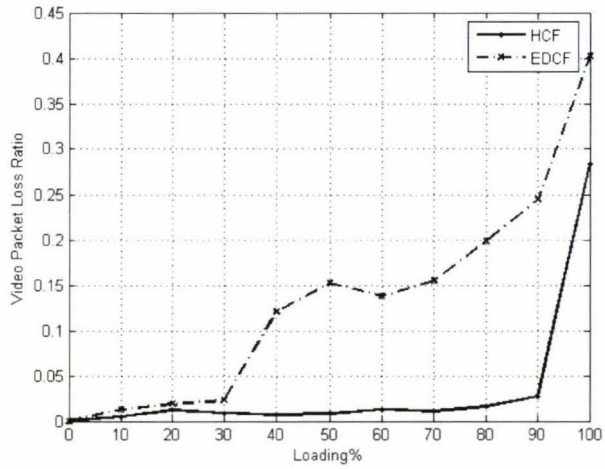


Figure 7-6 Video Packet Loss Ratio

For packet loss analysis, from the figure 8-6, we can find the value of the packet loss ratio similar as for the voice. However, the priority is different between them. It means that between the different categories of traffic the change of the packet loss is the similar. This reason why each categories traffic, has the equal the collision change as the EDCF mechanism. For the HCF, the situation is changed comparing the voice traffic. From the figure, we can see when traffic loading over the 90%, the packet loss ratio increase quickly. The reason why is that: when the AP calculates the TXOP, every RI was determined the minimum value of the entire maximum RSIs required by the different traffic streams. It results the whole beacon interval is little less than the sum of all RSI. The designed method is for the reasonable bandwidth utilization. When the whole time is not enough, the HCF always assigned the little time for the low priority packets. In other words, HCF gives the high priority packet discard rate to the low priority.

7.2 THROUGHPUT AND FAIRNESS STUDY

Because of the high BER and transmission medium, the WLAN has not enough bandwidth now. Further more, from the 6.3.1-simulation results, we can see that the fairness is the important factor relative to the RT-traffic QoS performance. If the fairness index is too low, the low priority traffic should starve. As a perfect network for the future market, WLAN must achieve the traffic fairness. From the simulation data, we can study

the throughput and fairness. For studying the total throughput, we also employ the 802.11a PHY layer in 36Mbps based on the different MAC protocols. The relative figure is shown in figure 7-7.

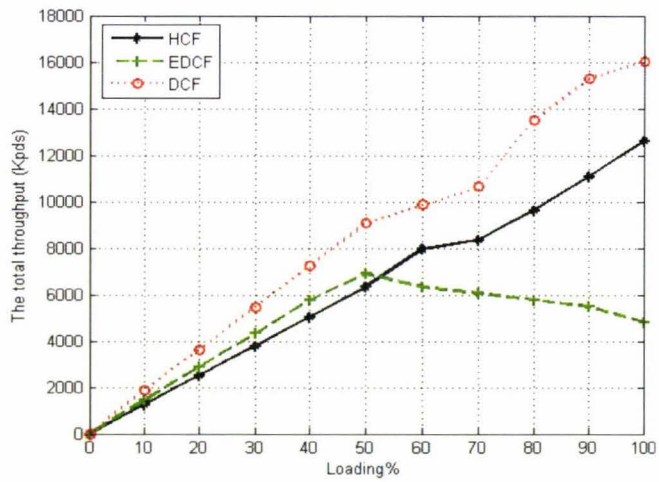


Figure 7-7 the Total Throughputs

From the figure, the shortcomings of the strict priority regime of delay-sensitive traffic against the delay-insensitive traffic are highlighted. Both for EDCF and HCF, the total throughput is so poor. For EDCF with more than the 50% total loading, because of a great deal of collisions happened in the contention process, the throughput decreased very quickly. From the figure; we can easily see the curve goes down after 50% loading. It means the utilization of the channel bandwidth is poor and unpredictable. From the figure, we can calculate the utilization of the channel bandwidth around 25% over the 50% loading. For the HCF, although the total throughput increases monotonously, the total utilization of the channel resource just is round 35%. It's too poor. Comparing the upcoming 802.11e based on the strict priority regime, the DCF protocol can provide higher utilization for all traffic. The reason why is that either EDCF or HCF, have so huge overhead, this wastes too much network resource. Especially, the HCF is the polling-induced overheads. In fact, the degradation may cause many serious problems for the WLAN. In some case, it should be effective with the QoS performance of the RT-traffic. We have to say that the throughput problems are the big limitation in the upcoming 802.11e

Corresponding to throughput, the fairness is an important issue when accessing a sharing wireless channel. Now we can study the fairness in the WLAN over the upcoming 802.11e. We employ the famous Jain's fairness index formula to calculate the fairness index:

$$J = \frac{(\sum_{i=1}^{n} g_i)^2}{n\sum_{i=1}^{n}(g_i)^2} \quad \text{.....(8-1)}$$

In formula (8-1):

The n denotes the total number of the flows with the same priority traffic

The g_i denotes the througput of flow i.

In this formula, the J equal to 1, it means all g_i are equal to access the channel. In other words, the fairness is the best. The calculated result is shown below:

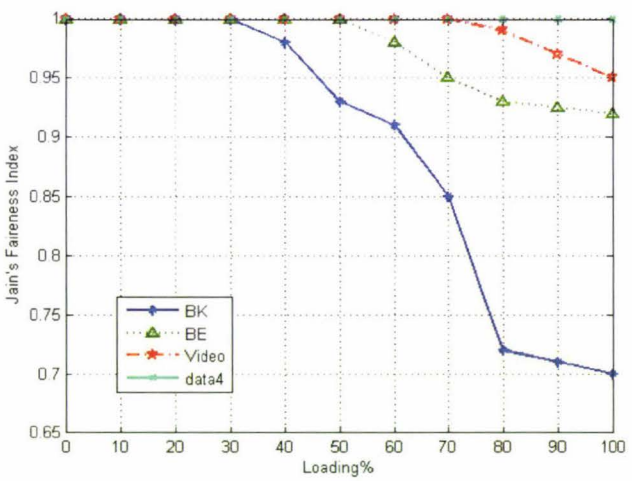


Figure 7-8 the Fairness Index of EDCF

From the figure we find that the fairness is not satisfied in the heavy loading. In the 5.1.2.3, we understand the DCF was the poor performance for the fair on the short-time scales. In fact, in DCF, different flows sharing a wireless channel can be unpredictable, but it is better during the total process. However, we can see in the EDCF, the BK (AC0) is very poor preference. It means that the BK traffic have less change to transmission in long-time high loading WLAN. This is the reason why the high packet loss rate and whole throughput low utilization over the upcoming 802.11e.

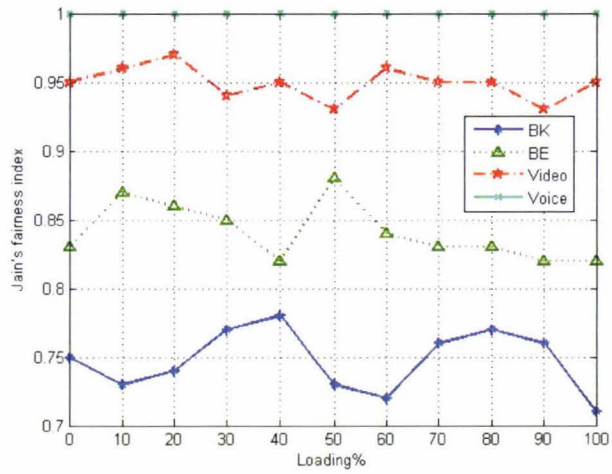


Figure 7-9 the Fairness Index of HCF

From the figure, the fairness is another limitation for the 802.11e. The fairness index in HCF is even worst than EDCF. The reason is the HCCA protocol, assigned every categories traffic as their AC to ensure the RT-time traffic has the less delay. This regime results the fairness index as less for the low priority traffic. This is the reason why the throughput utilization of the whole channel is poor. In fact, in the real circumstance the AC2 traffic are always the VBR. Because of the throughput of different traffic, the AP cannot calculate the RI and RSI correctly. Meanwhile, the HCF has not the mechanism to absorb traffic fluctuation; it causes the fairness become worst. The channel utilization becomes more decreased.

7.3 THE FUTURE OF QOS OVER WALN

7.3.1 Development of the PHY layer

For RT-traffic QoS performance, the increase of the bandwidth is the most significant topic in the future. We can develop transmission medium instead the old one. On the other hand, developing effective transmission mode is the other approache to increase the transmission rate.

As the frequency spectrum and infrared inherent characteristic, WLAN recently have high BER [70]. This is due to environment or geographical conditions. For this reason, to increase the reliability, WLAN deploys the DSSS and FHSS technologies etc. Those waste some frequency spectrum resource. Furthermore, from the safe consideration,

government always just allows limited frequency band license for WLAN. So the scope of developing bandwidth of the WLAN is very narrow in the future.
The space optics is a developing new transmission medium. FSO (free space optics) is one of the technologies that are based on space optics. In fact, FSO transmits signals by air not by the optical fiber. It can get higher data rate than T1 or T3. FSO can provide more than 1Gbps rate. Although, FSO has many limitations now, it would be a mature technology in the future.

For developing transmission mode, signal mapping and transmission can be considered. From 802.11 to 802.11g, signal mapping and modulation is developed step by step. From QPSK, GFSK, and CCK to OFDM, the max data rate increase more and more. Now the MIMO-OFDM is developing for 802.11n, the new max data rate can get 320Mbps. In the future, the new symbol mapping and transmission technologies will be developed to increase the WLAN bandwidth to guarantee the RT-traffic QoS performance.

7.3.2 Development of the MAC layer

Undoubtedly, the MAC layer development is one the most important research topics for the RT-traffic performance over the WLAN. From the simulations in chapter 7 and discussions above in this chapter, we find that upcoming 802.11e can mitigate the high latency and variable jitter values. It can guarantee the QoS performance of RT-traffic in some cases. However, there are some limitations in the upcoming 802.11e. Firstly, the throughput over EDCF and HCF is so lower. Secondly, the delay and jitter cannot be guaranteed under the heavy loading in EDCF. Finally, the fairness is not good in EDCF and HCF. These limitations results the instability of the upcoming 802.11e. There are stops to further research for enhancing the MAC performance.

There are many proposals in recent research papers to enhance the 802.11e [71] [72][73][74]. However, these proposals have some limitations. Furthermore, these are not suitable for general situation; they are better it used in some special case. In addition, IEEE is not yet to approve the 802.11e. In this work, we explored an EDCF model by

OPNET, and a HCF model by NS-2. However, simulation results are different for the different parameters. Say, the AIFS value, in different 802.11e draft version, it is different. In fact, from the mathematic analysis we find that the AIFS value varying has a large effect on the throughput value.

The 802.11e is not perfect for every scenario for RT-traffic QoS performance over the WLAN. For example, in the heavy loading based on EDCF, the RT-traffic QoS performance is even poorer than on DCF. For HCF, it collects the QSTA transmission requirements, and calculates the total TXOPs and allocates it to every QSTA. However, in some cases the transmission rate actually is always varying. For example, the VBR H.261 is the varying data rate RT-traffic. As a result the RIs that is calculated by HC are not correct for every time interval to result error in TXOPs. This results into more collisions and low throughput in 802.11e. In fact, the RT-traffic QoS performance for 802.11e is not a fixed scenario; it is a dynamic scenario for different situations. So we should evaluate IEEE802.11e with different QoS requirements under different scenarios. This is the huge job for the evaluation. Firstly, we must collect all the evaluation of the RT-traffic possible arranges in pairs or group, analyses the QoS requirements. If we finished this work, the resolution is not difficult; we can set some parameters in the future 802.11e to response to different scenarios.
In the future, the 802.11e will not be the total model for all AC traffic; it should add some parameters for the adaptive-self protocol to different scenarios.

Followed, in the future, 802.11e should optimize the tradeoff between channel efficiency, priority and fairness [75][76]. One of the 802.11e limitations is that the low network throughput utilization. From the above analysis, we can see it is due to the huge overhead, priority mechanism and the bad fairness index. How to optimize them should be the developing topic in the future. In our opinion, from the structure of the frame and transmission model we can optimize it. For example, every new technology created should add on field in the frame. In a common 802.11, the overhead size gets 120bits (PLCP header72bits, PLCP 48bits). In 802.11e, one TSPEC frame overhead gets 336bits. The TSPEC is just a little size frame for the HCF admission control centers on the

transmission specification IE. In fact, every field in those frames was some relation. We can optimize them possibly. However, it is a huge job, because if the frame changed, all structure of the protocols and facilities should be changed. In the future, it should be a solution. Except the frame structure, the transmission mode should be optimized, for example, in HCF; the entire frames must be transmitted by the HC. We can just use command frame to arrange the transmission. Of course, it is just an ideal; it must be studied by the simulations in the future. Corresponding with the different parameters that we can use the explored algorithm to optimized the tradeoff, and obtain the better fairness index.
In the future, the enhanced throughput utilization, good fairness, less tradeoff means the mature technology for guaranteeing the RT-traffic QoS performance.

Finally, one possibility should be reconsidered in WLAN MAC layer for RT-traffic QoS performance. It is throwing away the priority mechanism. From the introduction and simulation, we can know most of QoS protocol in MAC is based on the priority mechanism. However, the limitation of the priority is clear [77]. Although this mechanism guarantee less delay and jitter for RT-traffic QoS performance, it lost the fairness and high overhead for it. It's the inherent characteristic. We can optimize the tradeoff and fairness. However, it just can get the limited target. It depends on the complicated algorithm, priority evenly difficult and the varying loading is differently predicted. So maybe, we can use another idea. If we don't use the priority mechanism, we can also categories those traffic and allocations fixed different bandwidth for different AC traffic. Say 20% for AC3, 50% for AC2, 10% for AC1 and 20% for AC0. It is simple to say, we can consider different situation to assign the bandwidth. It can ensure the fairness for the WLAN. Another idea is that we can divide HCF beacon interval to RT-traffic time interval and non-RT-traffic interval. It can increase the throughput utilization. Of course, these just are ideas suitable for some special cases.

The original 802.11 MAC protocols do not provide the priority; they can not guarantee the RT-traffic QoS performance except the bandwidth is big enough in 802.11a and

802.11g PHY layer. Generally, the upcoming 802.11e provides the priority mechanism; it can guarantee the performance of the RT-traffic QoS over the WLAN.

Based on this conclusion, we do further study from the simulation data. It studied every AC RT-traffic detail from delay, jitter and loss ratio. We calculate every total data in every loading, and get the average value. Then we did the analyses.

After detail analyses of the simulation results, we find the upcoming 802.11e is not perfect for RT-traffic QoS performance. The findinds are summarized below:

1) The HCF total delay of the voice is far more then the EDCF. Of course, it is acceptable. It is due to the polling-induced overheads, unlike the contention based EDCF scheme.

2) The jitter of voice becomes unpredictable from the 40% loading throughput, over the EDCF, to 70% it increase very quickly. This happens because of unreasonable distribution of the bandwidth by the EDCF, the jitter between the packets is unstable.

3) HCF has better packet loss performance of voice. This is due to the algorithm of the HCF. In the EDCF, packet loss is not bad generally; however in the heavy loading it is not good. This is because all packets must take part in the contention, so more collisions happened. Farther more, when the limit waiting time is expired, the packets are discarded.

4) The delay of the video is acceptable under 50% of the total designed throughput. From 50%, the delay increases. To70%-100%, the delay time increases quickly until lost control. The reason for this is that every packet in the EDCF must defer and back off, so when the collisions in the channel get the limit, it should be blocked. Every packet, have the maximum back off time and AIFS time, result the bandwidth narrow.

5) The packet loss rate of video based on EDCF is more than voice. It is due to the priority. The packet loss rate of video based on HCF increases quickly under the 90% loading. The reason is that when the AP calculates the TXOP, every RI determines the minimum value of the entire maximum RSIs required by the different traffic streams. It's a result the whole beacon interval is little less than the sum of all RSI.

From the analysis of the result, we find the delay and jitter are closely related with the throughput and fairness index. From the study we find: for EDCF and HCF, the total

throughput is poor. For the EDCF, because of a great deal of collisions happens in the contention process, the throughput decreases very quickly to more than the 50% total loading. EDCF or HCF has so huge overhead that wastes too much network resource. Especially, the HCF is the polling-induced overheads

We studied the fairness of them; find that the fairness of low priority of voice is not satisfied in the heavy loading. The high packet loss rate and whole throughput low utilization over the upcoming 802.11e results the poor fairness.

We predict that the QoS will always play the important role in future WLAN. The new medium, new symbol mapping and transmission technologies will by developed to increase the WLAN bandwidth to guarantee the RT-traffic QoS performance.

For MAC layer, we predict that the 802.11e would not be the ultimatel model for all AC traffic; it should add some parameters for the self-adaptive protocol to different scenarios. The enhanced throughput utilization, good fairness, less tradeoff means the mature technology for guaranteeing the RT-traffic QoS performance.

CHARPTER 8

CONCLUSIONS

8.1 Research Conclusion

From chapter 1 through chapter 7, we studied the topic step by step. Firstly, from the network technology, we studied what were the real-time traffic and what were the special requirements for QoS from the real-time traffic; how to measure the performance of real time traffic for QoS. Secondly, we studied the WLAN and 802.11 protocols, their characteristics relative with the real-time traffic QoS performance. Then, we found that the QoS performances of RT-traffic were based on the MAC layer and PHY layer. Then, we studied the PHY layer detail at first. We illustrated how to increase the QoS performance of RT-traffic by exploring the PHY layer technologies. We did some analysis for the traffic over 802.11b, a, g, n base on PHY layer. The thesis focuses on MAC layer. From chapter 4, we studied the existing MAC layer protocols mechanism and 802.11e protocols that is being standard for WLAN QoS. We explored EDCF and HCF models by the OPNET and NS-2 to simulate 802.11e performance. We found some limitations in the 802.11e. We predicted and gave some proposals for future developing. The research conclusions are the following:

Real-time traffic are time-sensitive traffic. The QoS requirements over network is minimal delay (latency) and delay variation (jitter) between the sender and receiver. Generally, real-time traffic includes interactive applications and responsive applications. As another categorizing, RT-traffic means the CBR traffic and VBR traffic. In IEEE 802.1D, the real-time traffic are the Video and Voice at the TC5 and TC6 respectively. QoS refers to the ability of a network to provide better service to selected network traffic over various technologies. Different traffic have the different QoS requirements. The main measure of the QoS performance is the network availability, bandwidth, delay, jitter, and packet loss rate.

The QoS requirements of real-time traffic are a little bit different in the wired networks. For the high BER, the main performance measure of RT-traffic with QoS is mainly the delay, jitter and packet loss rate. For the IEEE protocols, these measures are just considered in PHY layer and MAC layer. Meanwhile, these factors are relative with the WLAN topology. Traditional IEEE802.11 is the one of the first generation standards of the wireless LAN. IEEE 802.11 defines two topologies for the WLAN: infrastructure and ad hoc mode. In PHY layer, the standard of RF transmission is DSSS and FHSS. In MAC layer, The CSMA/CD is the original technology for access mechanism. There are many other standards for RT traffic over the WLAN. There are Blue tooth, HomeRF, 802.16 etc. For increasing the capacity of the channel, there are many 802.11 protocols developed by IEEE. These are 802.11, 802.11b, 802.11a, 802.11g, 802.11j, 802.11n focused on the PHY layer. The 802.11e is for the MAC layer. In fact, those protocols for PHY layer just try to increase the transmission data rate (bandwidth). Actually, it also can achieve the RT-traffic QoS requirements indeed. 802.11e has been developed specially for the QoS performance of RT-traffic for MAC layer.

Enhancing the MAC layer protocols is the focus topic for guaranteeing the QoS performance for RT-traffic. In fact, the MAC layer mechanism research is the medium access technologies study. The original medium access technologies were based on CSMA/CA. The target reduces the medium transmission collisions. There are two kinds of access mechanisms in IEEE802.11 MAC sub-layer. They are DCF and PCF. The DCF is almost identical to the basic CSMA/CA but incorporates ideas from earlier wireless multiple access protocols MACA and MACAW. The transmitting station notices other stations that want to send the packet by the NAV (Network Allocation Vector). NAV is a virtual signal. Then, other stations can 'sense' the carries. PCF uses a central-controlled polling method to support synchronous data transmission. The PCF is an optional protocol; only can be applied in wireless network with access point. Different against in the ad hoc mode, an AP is needed to connect all stations information to a DS, and each station can communicate with other through AP in the infrastructure mode. For real-time traffic, PCF provides contention-free frame delivery from AP. Because of the heavy

overhead in PCF mechanism, PCF is not wide deployed. For the QoS, DCF is just designed for best-effort service. It does not support QoS for RT-traffic. PCF has been designed to try to support RT-Traffic. However, the performance of the RT-traffic under this mechanism is very poor. So DCF and PCF cannot guarantee the performance of the RT-traffic over QoS for their mechanisms.

In MAC layer research areas, many schemes were proposed for the MAC enhance. Most of them are based on DCF, few bases on PCF. DFS is a perfect schedule for the fairness allocation. Upcoming 802.11e provides the QoS guarantee for the RT-traffic. It combines with two parts EDCF and HCF. EDCF is run in the contention time interval of the HCF. EDCF provides this priority mechanism for different traffic. For the priority mechanism, EDCF use static parameter setting for differentiating among traffic categories as 802.1d. Those parameters include AIFS and CWmax and CWmin. In fact, EDCF gives the RT-traffic less deferrable time and back off time in the contention time interval. From the access occurring, HCF is different against EDCF. EDCF just occurs during contention free period (CFP), however HCF can occur during both of the contention period (CF) and CFP. It always coexists with EDCF in CP well. HCF uses the traffic specification (TSPEC) as negotiation between the QAP and the QSTAs. Then, collect the QSTA RSIs required, selecting Service Interval, polled accordingly during each elected SI, calculate the different TXOP values allocated to the different traffic streams reasonable for different QSTAs.

8.2 Research Contribution

We explored the EDCF model by OPNET, and explored HCF model by NS-2. By simulating the models, we validated some performance of RT-traffic QoS based on WLAN. For studying the performance of the RT-traffic QoS based on 802.11, we had three methods to study, Mathematical analysis method, Experimental method and Simulation method. The availability and precision of mathematic analysis method is limited by the effect of the assumptions. The experimental cost is too expensive and inflexible. So we use OPNET and NS-2 simulation to study the MAC protocols for RT-traffic QoS performance. From simulations of DCF, we validate that: 1) the 802.11b

cannot guarantee the RT-traffic QoS performance over DCF; 2) the 802.11a,g cannot guarantee it under heavy loading; 3) if the bandwidth is big enough, the RT-traffic QoS can be guarantee; 4) there is no priority mechanism in DCF; 5) DCF just provides the same delay and throughput. Followed, we developed the EDCF model by OPNET. By the simulation of the EDCF, we validate that: 1) EDCF provides the priority for different traffic. 2) Generally, EDCF can guarantee the performance of the RT-traffic over QoS. We explored the HCF protocol by NS-2. We validate that: HCF provides the priority for different traffic and can guarantee the performance of the RT-traffic QoS as an acceptable level. Meanwhile, the limitations of EDCF and HCF are found: under the heavy loading, the EDCF cannot guarantee the QoS performance for RT-traffic. The throughput of the HCF is too low.

We found and analyzed some limitations of the 802.11e for RT-traffic QoS performance over WLAN. From further study of the protocol detail from the data of simulation, the 802.11e is not perfect for RT-traffic QoS performance. The HCF total delay of the voice is far more then the EDCF due to the polling-induced overheads. The jitter of voice becomes unpredictable above the 40% loading. Especially, when the loading gets 70% of the whole, it increases very quickly. As we analyzed, it is the reason why the unreasonable distribution of the bandwidth by the EDCF. It results the jitter between the packets is unstable. For the packet loss, HCF has better performance of voice due to the allocation algorithm of the HCF. In the EDCF, packet loss is not bad generally, however in the heavy loading is not good. It is the reasons why all packets must take part in the contention and more and more collisions should have happened there. Farther more, when the rounded waiting time is expired, the packets are discarded. The delay of the video is good increase under 50% of the total designed throughput. At the 50% loading, the delay starts to increase. During 70%-100%, the delay time increases quickly until lost control. It is due to every packet in the EDCF must defer and back off. It results to more collisions. However, the allowed collisions times in the channel are limited. It should be blocked. In other words, every packet, which has the maximum back off time and AIFS time, should make the bandwidth narrow. The packet loss rate of video based on EDCF is more than voice. It is due to the priority. The packet loss rate of video based on HCF

increases quickly above the 90% loading. The reason is: when the AP calculates the TXOP, every RI determines the minimum value of the entire maximum RSIs required by the different traffic streams. As a result the whole beacon interval is little less than the sum of all RSI.

The delay and jitter of RT-traffic are related with the throughput and fairness index. Both of EDCF and HCF, the total throughput is so poor. For the EDCF, because of a great deal of collisions happen in the contention process, the throughput decreases very quickly more than the 50% total loading. The reason is that either EDCF or HCF has so huge overhead, this waste too much network resource. Especially, the HCF is the polling-induced overheads. The high packet loss rate and the throughput low utilization over the upcoming 802.11e contribute to the poor fairness index. The fairness low priority of traffic is not satisfied in the heavy loading case.

8.3 FUTURE RESEARCH DIRECTIONS

Dynamic self-adaptive 802.11e: from this thesis, we found that 802.11e is not perfect. For those limitations, we can do further study from EDCF and HCF mechanisms. 1) Because of the un-prediction of the EDCF in the heavy loading, we can add some parameters to the traffic loading and channel condition efficiently. What kind of parameters we can employ? How many of them? It must depend on the study and analysis. After this work, we see possibilities that the channel utilization can be increased and collision time can be reduced for RT-traffic over EDCF. These parameters can include loading rate, collision rate and total throughput saturation. Further research can buildup a mathematical model to analyse those relations. By simulation, we should find the perfect value. 2) Because of the huge polling-induced overheads, HCF has the unsatisfied tradeoff. This leads to poor fairness and poor throughput. Undoubtedly, how to optimize is the valuable work in the future. It depends on developing enhanced HCF. We can enhance the RI polling interval and TXOP allocation mechanism to get the better fairness index and channel utilization. 3) By evaluating IEEE 802.11e with different QoS

performance in different scenarios, we can try to find the different QoS requirement types. From the simulation, we noticed that the traffic deployment could affect the total QoS performance more. We can explore whether the classification of traffic deployments to different categories is a good idea. For example, we can classify the heavy loading, 50% RT-traffic as one category. With different categories, the RT-traffic QoS may be enhanced be by different bandwidth allocation Strategy.

References

[1] Prentice Hall (2000). *Data and Computer Communications*. Stallings W6th edition, ISBN: 0130843709,.

[2] Ho H. J., Rawles M. S., Vrijkorte M., Fei L. (2002). RF Challenges for 2.4 and 5GHz *WLAN Deployment and Design.* IEEE Wireless Communications andNetworking Conference (WCNC 2002), Volume 2, Orlando,Florida, USA, March 17-21,

[3] IEEE Std 802.3 (2002) *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local andmetropolitan area networks- Specific requirements*. Edition Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications, March 2002.

[4] Behrouz A. Forouzan (DeAnza College) & Catherine Coombs and Sophia Chung Fegan (2001). *Data Communications and Networking,* ISBN 0-07-282294-5

[5] IEEE 802.11g/D4.0(2002). *Further Higher-Speed Physical Layer Extension in the 2.4GHz Band.*

[6] Banchs A., Radimirsch M., Perez X (2002),*Assured and Expedited Forwarding Extensions for IEEE 802.11 Wireless LAN* Miami, USA, May 15-17,2002.

[7] Ala-Laurila J., Mikkonen J., Rinnemaa J(2001).*Wireless LAN Access Network Architecture for Mobile Operators* IEEE Communications Magazine, Volume 39.

[8] IEEE Draft P802.15.1/D0.9.2(2001) *Information technology - Telecommunicationsand Information Exchange Between Systems - LAN/MAN SpecificRequirements*. Draft Standard for Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for WirelessPersonal Area Networks (WPAN).

[9] ETSI TR (1999). *Technical Report, Broadband Radio Access Networks (BRAN); High Performance Radio Local Area Network (HIPERLAN).* Type 2; Requirements and architectures for wireless broadband access..

[10] Bray J., Sturman C(2002) *Bluetooth: Connect Without Cables.*, 2nd edition, Prentice Hall, ISBN: 0130661066.

[11] Lansford J., Stephens A.& Nevo R(2001) *Wi-Fi (802.11b) and Bluetooth: Enabling Coexistence*, IEEE Network, Volume 15, Issue 5.

[12] Bluetooth SIG, Core (2001) *Specification of the Bluetooth System.*Version 1.1.

[13] IEEE 802.11b(1999) *Higher-Speed Physical Layer Extension in the 2.4GHz Band,*

[14] ISO/IEC 7498-1(1994).*Information Technology - Open Systems Interconnections - Basic Reference Model: The Basic Model*

[15] Nan Si Shi (2004). *Wireless communication and mobile commerce.* ISBN 1-59140-212-3 (s/c)

[16] Almes, G.T & Lazowska, E.D. (1979) *The Behaviour of Ethernet-Like computer Communications Networks*

[17] Karthikeyan Sundaresan & Raghupathy Sivakumar(2004). *A unified MAC layer framework for ad-hoc networks with smart antennas.* Proceedings of the 5th ACM internation symposium on mobil ad hoc networking and computing, pp. 244-255.

[18] Jing Deng, Ben Liang, Pramod K. Varshney. (2004).*Tuning the Carrier Sensing Range of IEEE802.11 MAC,*

[19] Andrew Campbell, Cristina Aurrecoechea and Linda Hauw.(1996). *A Review of QoS Architectures.*

[20] M. Vouk, Z. Ortiz, A. Rindos, S. Woolet, D. Cosby, J. Sents and M. Aydemir.(1997).*Throughput and video performance of emerging LAN technologies: Switched Ethernet and LAN emulation over ATM.* pp.131-134, Blacksburg VA.

[21] J. Postel. (1980) *User Datagram Protocol.* RFC 768.

[22] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. (1996). *RTP: A Transport Protocol for Real-Time Applications.* RFC 1889.

[23] David L. Mills1(1992) *Network Time Protocol (Version 3) Specification, Implementation and Analysis.*RFC 305.

[24] *IEEE Std 802.11b*, September 1999, ISBN 0-7381-1811-7

[25] *CCK Modulation Delivers 11Mbps for High Rate IEEE 802.11 Extension,*

[26] Carl Andren and Mark Webster (1999) *Complementary Code Keying Demodulation System,*

[27] *IEEE 802.11g*/D3.1

[28] Yunli Chen, Qing-An Zeng and Dharama P.Agrawal(2004) *Performance evalution for IEEE 802.11e enhance districuted coordination function.*Wireless communications and Mobile Computing, Vol.4, pp.639-653.

[29] A. K. Parekh and R.G. Gallager(1993). *A generalized processor sharing approach to flow control in integrated services networks: the single-node case,* IEEE/ACM Transactions on Networking.

[30] Qiang Ni, Lamia Romdhani, Thierry Turletti (2004).*A survey of QOS Enhancements for IEEE 802.11 WLAN,*

[31] IEEE/ACM Trans. Network (2000). *Dynamic Tuning of the IEEE 802.11 protocol to Achieve a Theoretical Throughput Limit,*

[32] Alaa Muqattash and Marwan Krunz (2003).*CDMA-based MAC protocol for wireless ad hoc networks,*

[33] Bharghavan V (1994). *MACAW: A Media Access Protocol For Wireless LANs .*

[34] IEEE Stds (1996) *Wireless LAN.* Dept. D3.

[35] Weinmiller J., Woesner H., Wolisz A. (1996) *Analyzing and Improving the IEEE 802.11-MAC Protocol For Wireless LANS*

[36] A.Veres, A.T.Campbell, M.Barry, and L.H.Sun (2001) *Supporting service differentiation in wireless packet networks using distributed control.* IEEE JSAC special Issue on mobility and Resource Management in Next-Generation Wireless Systems, pp2094-2104.

[37] A.Lindgen, A. Almquist and O. Schelen. (2001) *Evaluation of quality of service schemes for IEEE 802.11 wireless LANs.* Proc. Of the 26th Annual IEEE Conference on local computer networks, Florida, USA, pp. 348-351.

[38] S.Mangold, S.Choi, P.May, O.Klein, G.Hiertz and L.Stibor (2002). *IEEE 802.11e wireless LAN for quality of service,*

[39] M.A.Visser and M.E.Zarki (1995).*Voice and data transmission over an 802.11 wireless network,*

[40] Reference number ISO/IEC 8802-11(e)(1999).*International Standard information Techology.*

[41] N. H. Vaidya, P. Bahl, and S. Gupa.(2000).*Distributed fair scheduling in a wireless LAN.* Proc. of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, USA.

[42] A. Demers, S, Keshav, and S. Shenker (1995). *Analysis and simulation of a fair queueing algorithm.* in proc. SIGCOMM.

[43] I. Aad and C. Castelluccia. (2001). *Differentiation mechanisms for IEEE 802.11*, Proc. Of IEEE infocom.

[44] J.C.R.Bennett and H.Zhang. (1996).*Wf2q: Worst-case fair weighted fair queueong*,

[45] P.Goyal, H.M.Vin and H.Cheng (1997). *Start-Time fair queuing: A scheduling algorithm for intergrated services packet switching networks.* IEEE/ACM transactions on the Networking, October

[46] S. J. Golestani (1994). *A self-clcoked fair queueing scheme for broadband applications,* Proc. Of Infocom

[47] J.Deng and Z. Haas (1998).*Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks,*

[48] J.Deng, and R. S. Chang (1999). *A priority scheme for IEEE802.11 DCF access method.* IEICE Trans in Com.

[49] Sobrinho JL and Krishnakumarm (1996). *Real-time traffics over the IEEE 802.11 medium access control layer.* AS. Bell Labs Technical Journal.

[50] Verse A, Campbell AT, Barry M, and Sun LH. (2001). *Supporting service differentiation in wireless packet networks using disturbed control.* IEEE Journal of selected Area in Communications (JSAC), special Issue on Mobility and Resource Management in Next-Generation Wireless Systems.

[51] A. Ganz, A . Phonphoem, and Z. Ganz,(2001). *Robust SuperPoll with Chaining Protocol for IEEE 802.11 WLAN in Support of Multimedia Applications.*

[52] J. Y. Yen & C. Chen (2002). *Support of multimedia services with the IEEE 802.11 MAC protocol.* Proc. of IEEE ICC, May 2002.

[53] D. A. Eckhardt and P. Steenkiste (2000) *Effort – limited fair (ELF) scheduling for wireless network*, Proc. of IEEE ICCS.

[54] *IEEE Std 802.11e, Medium Access Control (MAC) Enhancements for Quality of Service (QOS),* D4.4, June 2003

[55] G. Anastasi and L. Lenzini (2000). *QOS provided by the IEEE 802.11 wireless LAN to advanced data application: a simulation analysis.* Wireless network (6):99-108,2000

[56] D.-Y. Chen, S. Garg, M. Kappes, and K. S. Trivedi.(2002). *Supporting VBR VOIP traffic with IEEE 802.11 WLAN in PCF mode*. In proceedings of OPNET Work 2002, Washing D.C.

[57] M. Veeraraghavan, N, Cocker, and T. Moors.(2001).*Support of voice services in IEEE 802.11 wireless LANs.*

[58] Dongyan Chen, Daqing Gu and Jinyun Zhang, February (2004) *Supporting Real-time Traffic with QOS in IEEE 802.11e Based Home network.*

[59] ANSI/IEEE Std. 802.1 D (2002) *IEEE 802.11 WG, IEEE 802.11d, Part 3:MAC bridges.*

[60] S.Mangold, S. Choi, P.May, O.Klein, G, Hiertz, and L. Stibor (2003). *IEEE 802.11e wireless LAN for QOS.*

[61] A. Banchs and X.perez (2002). *Providibg throughput guarantees in IEEE 802.11 WLAN*, , WCNC, vol 1.

[62] Anders Lindgren, Andreas and Olov Schelen (2001). *Evaluation of QOS schemes for IEEE 802.11 WLAN*,

[63] Jie Hui and Michacel (2004). *Performance Analysis of IEEE 802.11e EDCA by a Unified Model*, North Carolina state University.

[64] *OPNET Modeler Version 10.5 tutorial,* May, 2004

[65] Gavin Holland, Nitin Vaidya, and paramvir Bahl (2004). *A Rate-adaptive MAC protocol for low-power ultra-wide ad-hoc network*, 3rd International Conference on AD-HOC networks and wireless,

[66] Rob Flickenger (2003).*Wireless Hack.* ISBN: 0-596-00559-8

[67] Regis J. (Bud) Bates & Donald W. Gregory (2000).*Voice & Data Communications Handbook.* ISBN: 0-07-212276-5.

[68] *NS2 Simulator,* http//www.isi.edu/nsnam/ns/

[69] Q.Ni, L. Romdhani, and T. Turletti (2004) *A Survey of QOS enhancements for IEEE 802.11 wireless LAN*,

[70] Luciano Bononi, Luca Budrisesi, Danilo Blasi, Vincezo Cacace, Luca Casone (2004).*A differentiated distributed corrdination function MAC protocol for cluster-based wireless ad hoc networks*,

[71] Martin Nilsson, Lars Staalhagen (2005). *Performance Modeling of IEEE 802.111e using OPNET Modeler, Fredrik Clementson.*

[72] Balasubramanian Appiah Venkatakrishnan, S.Selvakennedy (2004). *An Enhance HCF for IEEE 802.11e wireless Networks.*

[73] Sedhu Karthick Tanveer, Shwetha B. Ramesh and Siddesh K. Shivakumaraswamy (2004). *A QOS Enhancement Scheme for IEEE 802.11 WLAN.*

[74] Aravind Velayutham and J. Morries Chang (2003).*An Enhance Alternative to the IEEE802.11e MAC Scheme.*

[75] Y. G and J. Hou, (2003). *An Analysis Model for Service Differentiation in IEEE 802.11.* In Proc.ICC

[76] J. Deng and R.S.Chang, IEICE Trans (1999). *A priority Scheme for IEEE 802.11 DCF Access Method .*

[77] IEEE 802.11 WG (2003). *Draft Supplement to standard for telecommunication and information exchange between systems-LAN/Man specific requirements.*

Appendix A: The relative data

----- PHY LAYER DATA -----

FHSS Data	
AslotTime	50us
SIFS	28us
ACWmin	15
ACWmax	1023
Bite Error Rate	BER11=1.3E-5,BER1=0

DSSS Data	
AslotTime	20us
SIFS	18us
ACWmin	31
ACWmax	1023
Bite Error Rate	BER11=1.3E-5,BER1=0

WlanC_Infra_Red Data	
AslotTime	8us
SIFS	10us
ACWmin	63
ACWmax	1023
Bite Error Rate	BER11=1.3E-5,BER1=0

802.11b PHY layer Data				
AslotTime	20us			
aFragmentation Threshold	1024Byte			
SIFS	20us			
DIFS	60us			
AIFS	60+aslotTime*20(us)			
ACWmin	**Voice**	**Video**	**BF**	**BK**
	7	15	15	15
ACWmax	**Voice**	**Video**	**BF**	**BK**
	15	31	1023	1023
Dot11DaultCPTXOPLimit	5040us			
MAC header	38bytes			
PLCP header length	24bytes			
ACK size	14bytes			
PHY rate	11Mbps			
Bite Error Rate	BER11=1.3E-5,BER1=0			

802.11a PHY layer Data				
AslotTime	9us			
aFragmentation Threshold	1024Byte			
CCA time	4us			
SIFS	16us			
DIFS	25us			
AIFS	16+aslotTime*15(us)			
ACWmin	**Voice**	**Video**	**BF**	**BK**
	7	15	15	15
ACWmax	**Voice**	**Video**	**BF**	**BK**
	15	31	255	255
Dot11CAPrate	21us			
DotCAPMAX	5040us			
MAC header	38bytes			
PLCP header length	0.5bytes			
ACK size	14bytes			
CAP timer update time	5120us			
PHY rate	36Mbps			
Bite Error Rate	BER11=1.3E-5,BER1=0			

802.11g PHY layer Data				
AslotTime	9us			
aFragmentation Threshold	1024Byte			
CCA time	4us			
SIFS	10us			
DIFS	25us			
AIFS)	10+aslotTime*15(us)			
ACWmin	**Voice**	**Video**	**BF**	**BK**
	7	15	15	15
ACWmax	**Voice**	**Video**	**BF**	**BK**
	15	31	255	255
Dot11CAPrate	21us			
DotCAPMAX	5040us			
MAC header	38bytes			
PLCP header length	0.5bytes			
ACK size	14bytes			
CAP timer update time	5120us			
PHY rate	36Mbps			
Bite Error Rate	BER11=1.3E-5,BER1=0			

------ CALCULATED RESULT DATA -------

EDCF Average Delay (ms)				
%loading	Voice	Video	BE	BK
0	0	0	0	0
10	0.0035	0.006	2.1	2.9
20	0.0035	0.006	2.3	3.4
30	0.0035	0.007	2.8	3.7
40	0.0035	0.010	3.5	4.6
50	0.0035	0.019	3.9	5.0
60	0.0035	0.10	4.5	6.1
70	0.0035	0.30	5.9	6.9
80	0.0035	0.50	6.8	8.1
90	0.0035	1.1	7.3	8.9
100	0.0035	1.2	8.1	9.6

EDCF Every Traffic Average and Total Throughput (Kpbs) Actual						
%loading	Voice	Video	BE	BK	Total	In fact
0	0	0	0	0	0	0
10	64	153.6	320	512	1049.6	734.72
20	64	153.6	880	1100	2197.6	1538.32
30	64	460.8	1280	1500	3304.8	2313.36
40	64	460.8	1800	2000	4324.8	3027.36
50	64	460.8	2100	2800	5424.8	3797.36
60	64	2534.4	1046	1077	6598.4	4618.88
70	64	2317.8	805	602	7694.8	4932.11
80	64	2038.9	328	436	8798.4	5105.76
90	64	1808.6	145	188	9890.4	5300.87
100	64	1584.8	80	90	11030.4	5582.88

HCF Average Delay (s)				
%loading	Voice	Video	BE	BK
0	0	0	0	0
10	0.025	0.060	1.750	2.758
20	0.034	0.715	7.321	11.950
30	0.041	0.818	13.685	28.881
40	0.047	0.931	20.633	29.310
50	0.052	0.108	27.818	31.582
60	0.065	0.113	38.653	44.683
70	0.070	0.121	42.001	53.007
80	0.071	0.137	44.812	59.234
90	0.074	0.168	47.016	65.113
100	0.075	0.200	50.301	70.501

Comparing the Throughput (Kpbs) Actually						
%1oading	DCF	EDCF	HCF			
0	0	0	0			
10	1876	1443	1261			
20	3612	2885	2524			
30	5465	4323	3786			
40	7231	5764	5038			
50	9073	6903	6333			
60	9858	6320	7965			
70	10638	6078	8359			
80	13485	5763	9620			
90	15277	5484	11088			
100	16032	4832	12630			

EDCF throughput loading fairness index						
%1oading	Voice	Video	BE	BK		
0	1	1	1	1		
10	1	1	1	1		
20	1	1	1	1		
30	1	1	1	1		
40	1	1	1	0.98		
50	1	1	1	0.93		
60	1	1	0.98	0.91		
70	1	1	0.95	0.85		
80	1	0.99	0.93	0.72		
90	1	0.97	0.925	0.71		
100	1	0.95	0.92	0.70		

HCF throughput loading fairness index						
%1oading	Voice	Video	BE	BK		
0	1	0.95	0.83	0.75		
10	1	0.96	0.87	0.73		
20	1	0.97	0.86	0.74		
30	1	0.94	0.85	0.77		
40	1	0.95	0.82	0.78		
50	1	0.93	0.88	0.73		
60	1	0.96	0.84	0.72		
70	1	0.95	0.83	0.76		
80	1	0.95	0.83	0.77		
90	1	0.93	0.82	0.76		
100	1	0.95	0.82	0.71		

Video Mean Delay comparing (ms)				
%Loading	EDCF	HCF		
0	0	0		
10	6	59		
20	6	60		
30	7	73		
40	10	84		
50	19	92		
60	101	103		
70	308	121		
80	512	152		
90	1126	177		
100	1218	200		

Video Jitter comparing (ms)				
%Loading	EDCF	HCF		
0	0	0		
10	2.51	7.27		
20	2.35	7.15		
30	2.75	7.82		
40	2.15	7.55		
50	3.21	7.84		
60	3.53	7.12		
70	3.85	7.72		
80	7.15	7.32		
90	15.18	7.53		
100	35.30	7.13		

Video Packet loss Ratio				
%Loading	EDCF	HCF		
0	0	0		
10	0.013	0.005		
20	0.019	0.012		
30	0.023	0.009		
40	0.121	0.007		
50	0.152	0.008		
60	0.138	0.013		
70	0.155	0.011		
80	0.198	0.016		
90	0.245	0.027		
100	0.402	0.283		

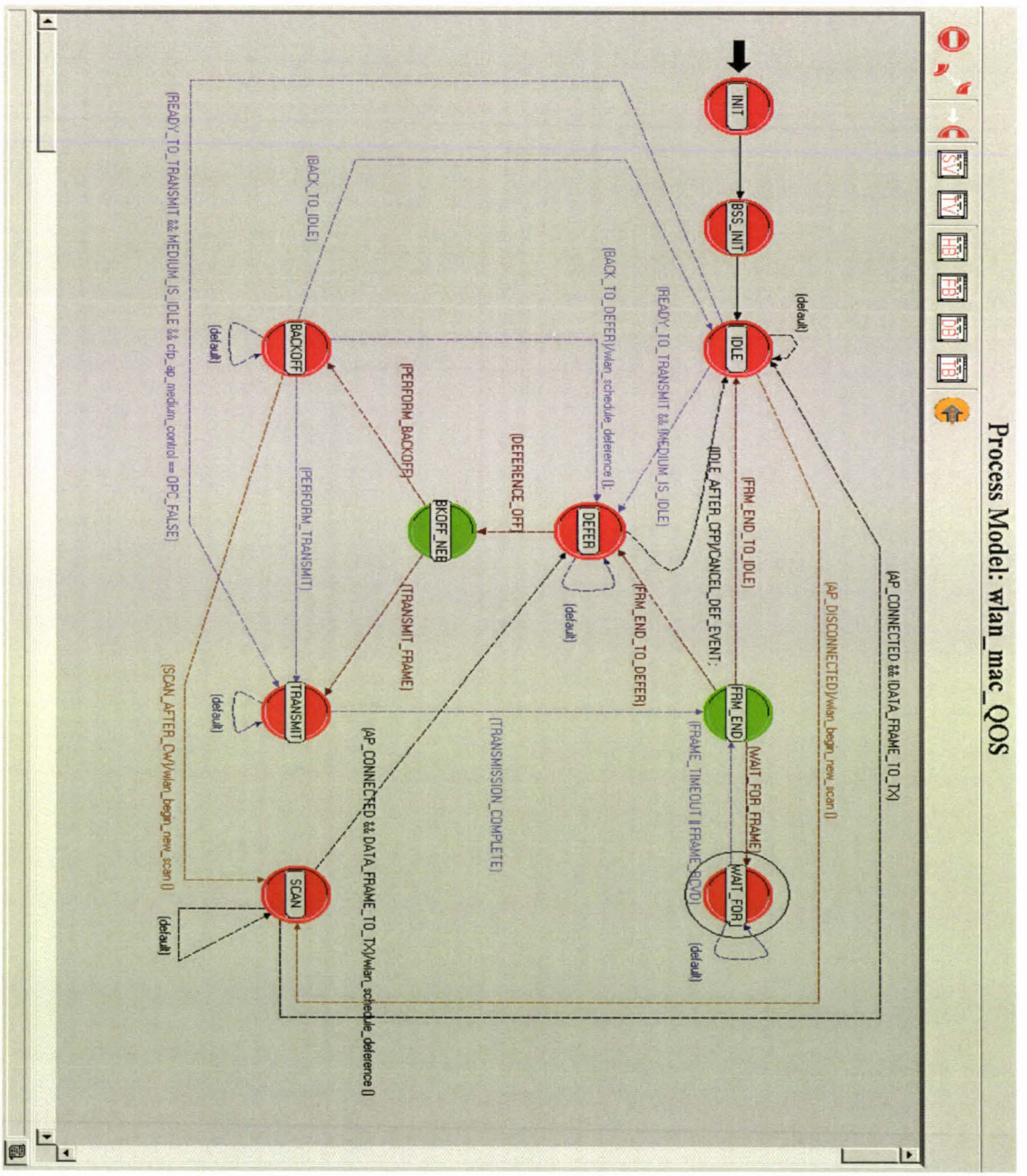
Voice Mean Delay comparing (ms)				
%Loading	EDCF	HCF		
0	0	0		
10	2.10	25		
20	2.15	35		
30	2.20	45		
40	2.35	55		
50	3.15	60		
60	3.60	65		
70	3.70	70		
80	3.80	72		
90	4.10	73		
100	4.50	75		

Voice Jitter comparing (ms)				
%Loading	EDCF	HCF		
0	0	0		
10	0.25	3.25		
20	0.35	3.51		
30	0.75	3.47		
40	1.15	3.91		
50	1.21	3.60		
60	1.53	3.65		
70	1.85	3.70		
80	3.15	3.72		
90	4.18	3.73		
100	5.30	3.15		

Voice Packet loss Ratio				
%Loading	EDCF	HCF		
0	0	0		
10	0.17	0.005		
20	0.18	0.012		
30	0.19	0.021		
40	0.20	0.007		
50	0.23	0.008		
60	0.22	0.012		
70	0.25	0.009		
80	0.35	0.006		
90	0.45	0.007		
100	0.52	0.008		

Appendix B: The detail Figure

Figure 7-17 the Waln_mac_qos Process



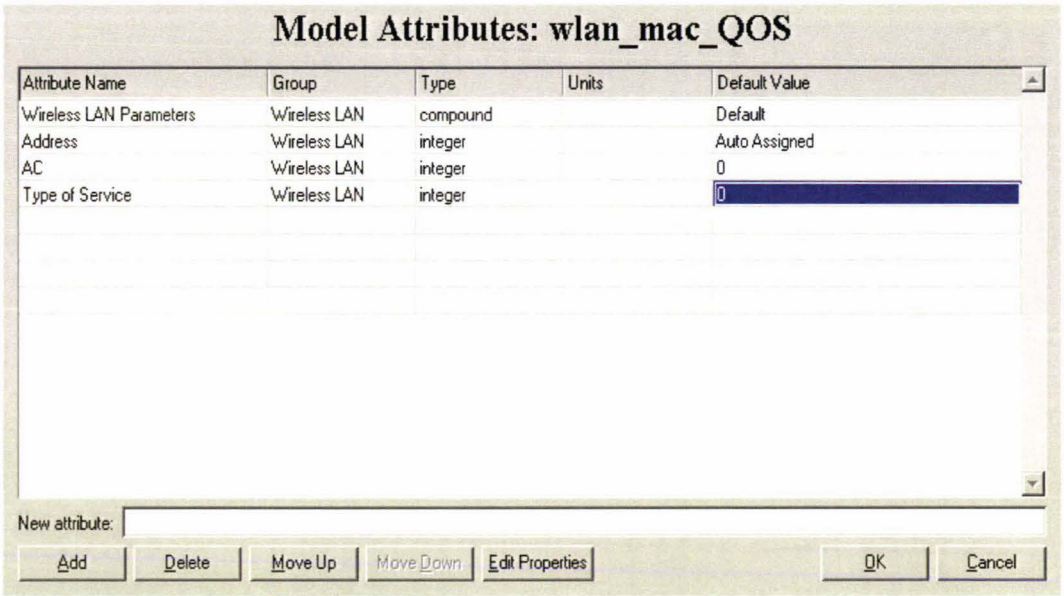


Figure 7-20 Model Attribute

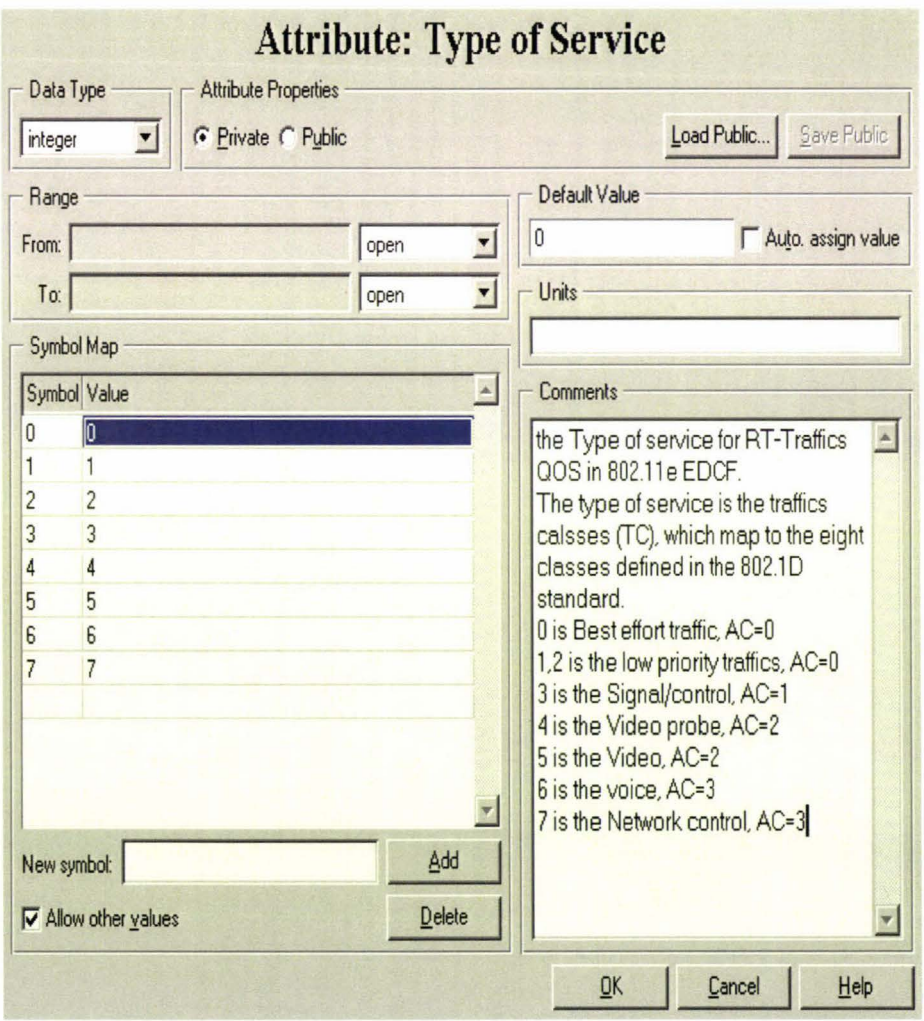


Figure 7-21 Type of Service

Appendix C: Key C/C++ source code

----- OPNET SOURCE CODE ------

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan_mac_QOS.pr.c

```
#define WLANC_CH_STEP_FOR_NO_OVERLAP    ((int) ceil (channel_bandwidth / channel_spacing))


/** State Transitions **/

/** The data frame send flag is set whenever there is a data to be send by  **/
/** the higher layer or the response frame needs to be sent. However, in    **/
/** either case the flag will not be set if the receiver is busy            **/
/** Frames cannot be transmitted until medium is idle. Once, the medium     **/
/** is available then the station is eligible to transmit provided there    **/
/** is a need for backoff. Once the transmission is complete then the       **/
/** station will wait for the response provided the frame transmitted       **/
/** requires a response (such as RTS and Data frames). If response          **/
/** is not needed then the station will defer to transmit next packet       **/

/* After receiving a stream interrupt, we need to switch states from    */
/* idle to defer or transmit if there is a frame to transmit and the    */
/* receiver is not busy                                                 */
/* If a frame is received indicating that the STA should scan, all bets */
/* are off, and the STA moves into the scan state to look for other APs */
#define READY_TO_TRANSMIT       (((intrpt_type == OPC_INTRPT_STRM && wlan_flags->data_frame_to_send == OPC_TRUE && \
                                   (pcf_flag == OPC_BOOLINT_DISABLED || (wlan_flags->pcf_active == OPC_FALSE && \
                                    (ap_flag == OPC_BOOLINT_ENABLED || cfp_ap_medium_control == OPC_FALSE)))) || \
                                  fresp_to_send != WlanC_None || \
                                  wlan_flags->polled == OPC_TRUE || \
                                  wlan_flags->tx_beacon == OPC_TRUE || \
                                  (wlan_flags->pcf_active == OPC_TRUE && ap_flag == OPC_BOOLINT_ENABLED)) && \
                                 !roam_state_ptr->scan_mode)

/* When we have a frame to transmit, we move to transmit state if the   */
/* medium was idle for at least a AIFS time, otherwise we go to defer   */
/* state.                                                               */
#define MEDIUM_IS_IDLE          (((current_time - nav_duration + PRECISION_RECOVERY >= aifs_time) && \
                                  wlan_flags->receiver_busy == OPC_FALSE && \
                                  (current_time - rcv_idle_time + PRECISION_RECOVERY >= aifs_time) && \
                                  wlan_flags->pcf_active == OPC_FALSE) || \
                                 wlan_flags->forced_bk_end == OPC_TRUE)

/* Change state to Defer from Frm_End, if the input buffers are not empty or a frame needs  */
/* to be retransmitted or the station has to respond to some frame.                         */
#define FRAME_TO_TRANSMIT       (wlan_flags->data_frame_to_send == OPC_TRUE || fresp_to_send != WlanC_None || \
                                 retry_count != 0 || wlan_flags->tx_beacon == OPC_TRUE || \
                                 wlan_flags->cw_required == OPC_TRUE)

/* After deferring for either collision avoidance or interframe gap     */
/* the channel will be available for transmission.                      */
#define DEFERENCE_OFF           (intrpt_type == OPC_INTRPT_SELF && \
                                 intrpt_code == WlanC_Deference_Off && \
                                 wlan_flags->receiver_busy == OPC_FALSE)

/* Issue a transmission complete stat once the packet has successfully  */
/* been transmitted from the source station                             */
#define TRANSMISSION_COMPLETE   (intrpt_type == OPC_INTRPT_STAT && \
                                 op_intrpt_stat () == TRANSMITTER_BUSY_INSTAT)

/* Backoff is performed based on the value of the backoff flag.         */
#define PERFORM_BACKOFF         (wlan_flags->backoff_flag == OPC_TRUE || wlan_flags->perform_cw == OPC_TRUE)

/* Need to start transmitting frame once the backoff (self intrpt)      */
/* completed                                                            */
#define BACKOFF_COMPLETED       (intrpt_type == OPC_INTRPT_SELF && intrpt_code == WlanC_Backoff_Elapsed && \
                                 (wlan_flags->receiver_busy == OPC_FALSE || wlan_flags->forced_bk_end == OPC_TRUE))

/* Contention Window period, which follows a successful packet          */
/* transmission, is completed.                                          */
#define CW_COMPLETED            (intrpt_type == OPC_INTRPT_SELF && intrpt_code == WlanC_CW_Elapsed && \
                                 (wlan_flags->receiver_busy == OPC_FALSE || wlan_flags->forced_bk_end == OPC_TRUE))

/* After transmission the station will wait for a frame response for    */
/* Data and Rts frames.                                                 */
```

```
441 #define SCAN_TIMEOUT            (intrpt_type == OPC_INTRPT_SELF && intrpt_code == WlanC_Scan_Timeout)
442
443 #define SCAN_AFTER_CW           (CW_COMPLETED && AP_DISCONNECTED)
444
445
446 /** Function Prototypes **/
447 static void         wlan_mac_sv_init (void);
448 static void         wlan_transceiver_channel_init (void);
449 static void         wlan_rxgroup_reduce (void);
450 static void         wlan_higher_layer_data_arrival (void);
451 static void         wlan_hl_packet_drop (Packet* hld_pkptr, OpT_Packet_Size data_size);
452 static void         wlan_hlpk_enqueue (Packet* hld_pkptr, int dest_addr, Boolean polling);
453 static void         wlan_frame_transmit (void);
454 static double       wlan_non_11b_plcp_overhead_compute (OpT_Packet_Size mpdu_length, double data_rate);
455 static double       wlan_plcp_overhead_ofdm_compute (OpT_Packet_Size mpdu_length, double data_rate);
456 static Boolean      wlan_dest_is_11g_enabled (int dest_mac_addr);
457 static void         wlan_prepare_frame_to_send (WlanT_Mac_Frame_Type frame_type);
458 static double       wlan_ctrl_response_drate_determine (double rcvd_data_rate);
459 static void         wlan_slot_time_set (double new_slot_time);
460 static void         wlan_frame_tx_phy_info_set (Packet* frame_ptr, double tx_data_rate);
461 static void         wlan_interrupts_process (void);
462 static void         wlan_physical_layer_data_arrival (void);
463 static Boolean      wlan_tuple_find (int sta_addr, int seq_id, int frag_num, int dest_addr);
464 static void         wlan_data_process (Packet* seg_pkptr, int dest_addr, int sta_addr, int final_dest_addr,
465 static void         wlan_accepted_frame_stats_update (Packet* seg_pkptr);
466 static void         wlan_schedule_deference (void);
467 static void         wlan_frame_discard (void);
468 static void         wlan_pcf_frame_discard (void);
469 static void         wlan_mac_rcv_channel_status_update (int channel_id);
470 static void         wlan_mac_error (const char* msg1, const char* msg2, const char* msg3);
471 static Boolean      wlan_poll_list_member_find (int dest_addr);
472
473 static void                     wlan_frame_type_conv (WlanT_Mac_Frame_Type frame_type, char* frame_type_name
474 static int                      wlan_bss_id_list_manage (int bssid, const char* operation);
475 static PrgT_Mapping_Handle      wlan_bss_mapping_get (void);
476 static int                      wlan_get_ap_sta_addr (int bss_idx);
477 static WlanT_Sta_Mapping_Info*  wlan_sta_addr_register (int bss_idx, int sta_addr, int sta_is_ap, Objid sta_
478 static WlanT_Bss_Mapping_Info*  wlan_bss_info_get (int bssid);
479 static WlanT_Sta_Mapping_Info*  wlan_sta_info_get (int sta_addr, Boolean serialize);
480 static double                   wlan_min_freq_for_chan (int chan_num);
481 static Boolean                  wlanc_11a_channel_is_regular (double frequency, int* channel_num_ptr);
482 static void                     wlan_begin_new_scan (void);
483 static void                     wlan_set_transceiver_channel (int chan_num);
484 static void                     wlan_ap_switch (void);
485 static void                     wlan_sta_addr_deregister (int bss_idx, int sta_addr);
486 static void                     wlan_reset_sv (void);
487 static void                     wlan_ap_position_publish (void);
488 static void                     wlan_ap_eval_virtual (void);
489 static double                   wlan_ap_signal_strength_calc (double prop_distance, WlanT_AP_Position_Info *
490 static void                     wlan_find_new_ap_virtual (void);
491
492 /* Callback functions       */
493 #if defined (__cplusplus)
494 extern "C" {
495 #endif
496
497 static int                      wlan_hld_list_elem_add_comp (const void* list_elem_ptr1, const void* list_el
498 static void*                    wlan_bss_info_get_key (void *value_ptr);
499 static int                      wlan_mapping_int_key_compare (void *key_a_ptr, void *key_b_ptr);
500 static void                     wlan_bss_info_free (void *value_ptr);
501 static void*                    wlan_sta_info_get_key (void *value_ptr);
502 static void                     wlan_sta_info_free (void *value_ptr);
503 static void*                    wlan_dup_info_get_key (void *value_ptr);
504 static void                     wlan_dup_info_free (void *value_ptr);
505
506 #if defined (__cplusplus)
507 } /* end of 'extern "C" {' */
508 #endif
509
/* End of Header Block */
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
579
580     double                          backoff_slots
581
582     Stathandle                      packet_load_handle
583
584     double                          intrpt_time
585     Packet *                        wlan_transmit_frame_copy_ptr
586     Stathandle                      backoff_slots_handle
587     int                             instrm_from_mac_if
588     int                             outstrm_to_mac_if
589     int                             num_fragments
590
591     OpT_Packet_Size                 remainder_size
592     List*                           defragmentation_list_ptr
593
594
595
596
597
598     WlanT_Mac_Flags*                wlan_flags
599
600
601
602     OmsT_Aa_Address_Handle          oms_aa_handle
603     double                          current_time
604     double                          rcv_idle_time
605     Pmohandle                       hld_pmh
606
607
608
609
610
611
612
613
614     int                             max_backoff
615     int                             min_backoff
616     char                            current_state_name [32]
617     Stathandle                      hl_packets_rcvd
618     Stathandle                      media_access_delay
619
620     Stathandle                      ete_delay_handle
621
622     Stathandle                      global_ete_delay_handle
623     Stathandle                      global_throughput_handle
624     Stathandle                      global_load_handle
625     Stathandle                      global_dropped_data_handle
626     Stathandle                      global_mac_delay_handle
627     Stathandle                      ctrl_traffic_rcvd_handle_inbits
628     Stathandle                      ctrl_traffic_sent_handle_inbits
629     Stathandle                      ctrl_traffic_rcvd_handle
630     Stathandle                      ctrl_traffic_sent_handle
631     Stathandle                      data_traffic_rcvd_handle_inbits
632     Stathandle                      data_traffic_sent_handle_inbits
633     Stathandle                      data_traffic_rcvd_handle
634     Stathandle                      data_traffic_sent_handle
635     double                          sifs_time
636     double                          slot_time
637     int                             cw_min
638     int                             cw_max
639     int                             cw_min_tmp
640     int                             cw_max_tmp
641     double                          difs_time
642     double                          aifs_time
643     double                          plcp_overhead_control
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
644
645     double                          plcp_overhead_data
646
647     Stathandle                      channel_reserv_handle
648     Stathandle                      retrans_handle
649
650     Stathandle                      throughput_handle
651     int                             long_retry_limit
652
653     int                             short_retry_limit
654
655     int                             retry_limit
656
657     WlanT_Mac_Frame_Type            last_frametx_type
658
659
660     Evhandle                        deference_evh
661     Evhandle                        backoff_elapsed_evh
662     Evhandle                        frame_timeout_evh
663
664     double                          eifs_time
665
666     int                             i_strm
667     Boolean                         wlan_trace_active
668     OpT_Packet_Id                   pkt_in_service
669     Stathandle                      bits_load_handle
670     int                             ap_flag
671
672     Boolean                         bss_flag
673     int                             ap_mac_address
674     int                             hld_max_size
675
676     double                          max_receive_lifetime
677
678     int                             accept_large_packets
679     WlanT_Phy_Char_Code             phy_char_flag
680     WlanT_Phy_Char_Code             ap_phy_char_flag
681
682
683     WlanT_Phy_Type                  phy_type
684
685
686     OpT_Packet_Size                 total_hlpk_size
687     Stathandle                      drop_packet_handle
688
689     Stathandle                      drop_packet_handle_inbits
690
691     Log_Handle                      drop_pkt_log_handle
692
693     Log_Handle                      config_log_handle
694     int                             drop_pkt_entry_log_flag
695     int                             packet_size
696     double                          receive_time
697     Ici*                            llc_iciptr
698     double                          rx_power_threshold
699     int                             bss_id
700
701
702
703
704     int                             pcf_retry_count
705     int                             poll_fail_count
706
707     int                             max_poll_fails
708
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
878  #define drop_packet_handle                  pr_state_ptr->drop_packet_handle
879  #define drop_packet_handle_inbits           pr_state_ptr->drop_packet_handle_inbits
880  #define drop_pkt_log_handle                 pr_state_ptr->drop_pkt_log_handle
881  #define config_log_handle                   pr_state_ptr->config_log_handle
882  #define drop_pkt_entry_log_flag             pr_state_ptr->drop_pkt_entry_log_flag
883  #define packet_size                         pr_state_ptr->packet_size
884  #define receive_time                        pr_state_ptr->receive_time
885  #define llc_iciptr                          pr_state_ptr->llc_iciptr
886  #define rx_power_threshold                  pr_state_ptr->rx_power_threshold
887  #define bss_id                              pr_state_ptr->bss_id
888  #define pcf_retry_count                     pr_state_ptr->pcf_retry_count
889  #define poll_fail_count                     pr_state_ptr->poll_fail_count
890  #define max_poll_fails                      pr_state_ptr->max_poll_fails
891  #define cfpd_list_ptr                       pr_state_ptr->cfpd_list_ptr
892  #define pcf_queue_offset                    pr_state_ptr->pcf_queue_offset
893  #define beacon_int                          pr_state_ptr->beacon_int
894  #define pcf_frag_buffer_ptr                 pr_state_ptr->pcf_frag_buffer_ptr
895  #define wlan_pcf_transmit_frame_copy_ptr        pr_state_ptr->wlan_pcf_transmit_fr
896  #define pcf_num_fragments                   pr_state_ptr->pcf_num_fragments
897  #define pcf_remainder_size                  pr_state_ptr->pcf_remainder_size
898  #define polling_list                        pr_state_ptr->polling_list
899  #define poll_list_size                      pr_state_ptr->poll_list_size
900  #define poll_index                          pr_state_ptr->poll_index
901  #define pifs_time                           pr_state_ptr->pifs_time
902  #define beacon_evh                          pr_state_ptr->beacon_evh
903  #define cfp_end_evh                         pr_state_ptr->cfp_end_evh
904  #define pcf_pkt_in_service                  pr_state_ptr->pcf_pkt_in_service
905  #define pcf_flag                            pr_state_ptr->pcf_flag
906  #define active_pc                           pr_state_ptr->active_pc
907  #define cfp_prd                             pr_state_ptr->cfp_prd
908  #define cfp_offset                          pr_state_ptr->cfp_offset
909  #define cfp_length                          pr_state_ptr->cfp_length
910  #define ap_relay                            pr_state_ptr->ap_relay
911  #define packet_size_dcf                     pr_state_ptr->packet_size_dcf
912  #define packet_size_pcf                     pr_state_ptr->packet_size_pcf
913  #define receive_time_dcf                    pr_state_ptr->receive_time_dcf
914  #define receive_time_pcf                    pr_state_ptr->receive_time_pcf
915  #define cfp_ap_medium_control               pr_state_ptr->cfp_ap_medium_control
916  #define pcf_network                         pr_state_ptr->pcf_network
917  #define beacon_eff_mode                     pr_state_ptr->beacon_eff_mode
918  #define channel_count                       pr_state_ptr->channel_count
919  #define channel_num                         pr_state_ptr->channel_num
920  #define first_chan_min_freq                 pr_state_ptr->first_chan_min_freq
921  #define channel_bandwidth                   pr_state_ptr->channel_bandwidth
922  #define channel_spacing                     pr_state_ptr->channel_spacing
923  #define eval_bss_id                         pr_state_ptr->eval_bss_id
924  #define roam_state_ptr                      pr_state_ptr->roam_state_ptr
925  #define rx_state_info_ptr                   pr_state_ptr->rx_state_info_ptr
926  #define ap_connectivity_check_interval      pr_state_ptr->ap_connectivity_check_ir
927  #define ap_connectivity_check_time          pr_state_ptr->ap_connectivity_check_time
928  #define ap_connectivity_check_evhndl        pr_state_ptr->ap_connectivity_check_ev
929  #define conn_ap_pos_info_ptr                pr_state_ptr->conn_ap_pos_info_ptr
930  #define my_sta_info_ptr                     pr_state_ptr->my_sta_info_ptr
931  #define my_bss_info_ptr                     pr_state_ptr->my_bss_info_ptr
932  #define mapping_info_mutex                  pr_state_ptr->mapping_info_mutex
933  #define ac                                  pr_state_ptr->ac
934  #define aift_ac                             pr_state_ptr->aift_ac
935  #define type_of_service                     pr_state_ptr->type_of_service
936  #define backoff_slot_ptr                    pr_state_ptr->backoff_slot_ptr
937  #define max_backoff_tmp                     pr_state_ptr->max_backoff_tmp
938  #define backoff_max_handle                  pr_state_ptr->backoff_max_handle
939  #define backoff_min_handle                  pr_state_ptr->backoff_min_handle
940  #define ac_slot                             pr_state_ptr->ac_slot
941
942  /* These macro definitions will define a local variable called  */
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
9769  #undef cfp_end_evh
9770  #undef pcf_pkt_in_service
9771  #undef pcf_flag
9772  #undef active_pc
9773  #undef cfp_prd
9774  #undef cfp_offset
9775  #undef cfp_length
9776  #undef ap_relay
9777  #undef packet_size_dcf
9778  #undef packet_size_pcf
9779  #undef receive_time_dcf
9780  #undef receive_time_pcf
9781  #undef cfp_ap_medium_control
9782  #undef pcf_network
9783  #undef beacon_eff_mode
9784  #undef channel_count
9785  #undef channel_num
9786  #undef first_chan_min_freq
9787  #undef channel_bandwidth
9788  #undef channel_spacing
9789  #undef eval_bss_id
9790  #undef roam_state_ptr
9791  #undef rx_state_info_ptr
9792  #undef ap_connectivity_check_interval
9793  #undef ap_connectivity_check_time
9794  #undef ap_connectivity_check_evhndl
9795  #undef conn_ap_pos_info_ptr
9796  #undef my_sta_info_ptr
9797  #undef my_bss_info_ptr
9798  #undef mapping_info_mutex
9799  #undef ac
9800  #undef aift_ac
9801  #undef type_of_service
9802  #undef backoff_slot_ptr
9803  #undef max_backoff_tmp
9804  #undef backoff_max_handle
9805  #undef backoff_min_handle
9806  #undef ac_slot
9807
9808  #undef FIN_PREAMBLE_DEC
9809  #undef FIN_PREAMBLE_CODE
9810
9811  #define FIN_PREAMBLE_DEC
9812  #define FIN_PREAMBLE_CODE
9813
9814  VosT_Obtype
9815  wlan_mac_QOS_init (int * init_block_ptr)
9816      {
9817
9818  #if !defined (VOSD_NO_FIN)
9819      int _op_block_origin = 0;
9820  #endif
9821      VosT_Obtype obtype = OPC_NIL;
9822      FIN_MT (wlan_mac_QOS_init (init_block_ptr))
9823
9824      obtype = Vos_Define_Object_Prstate ("proc state vars (wlan_mac_QOS)",
9825          sizeof (wlan_mac_QOS_state));
9826      *init_block_ptr = 0;
9827
9828      FRET (obtype)
9829      }
9830
9831  VosT_Address
9832  wlan_mac_QOS_alloc (VOS_THREAD_INDEX_ARG_COMMA VosT_Obtype obtype, int init_block
9833      {
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
10602           }
10603       if (strcmp ("ap_connectivity_check_evhndl" , var_name) == 0)
10604           {
10605           *var_p_ptr = (void *) (&prs_ptr->ap_connectivity_check_evhndl);
10606           FOUT
10607           }
10608       if (strcmp ("conn_ap_pos_info_ptr" , var_name) == 0)
10609           {
10610           *var_p_ptr = (void *) (&prs_ptr->conn_ap_pos_info_ptr);
10611           FOUT
10612           }
10613       if (strcmp ("my_sta_info_ptr" , var_name) == 0)
10614           {
10615           *var_p_ptr = (void *) (&prs_ptr->my_sta_info_ptr);
10616           FOUT
10617           }
10618       if (strcmp ("my_bss_info_ptr" , var_name) == 0)
10619           {
10620           *var_p_ptr = (void *) (&prs_ptr->my_bss_info_ptr);
10621           FOUT
10622           }
10623       if (strcmp ("mapping_info_mutex" , var_name) == 0)
10624           {
10625           *var_p_ptr = (void *) (&prs_ptr->mapping_info_mutex);
10626           FOUT
10627           }
10628       if (strcmp ("ac" , var_name) == 0)
10629           {
10630           *var_p_ptr = (void *) (&prs_ptr->ac);
10631           FOUT
10632           }
10633       if (strcmp ("aift_ac" , var_name) == 0)
10634           {
10635           *var_p_ptr = (void *) (&prs_ptr->aift_ac);
10636           FOUT
10637           }
10638       if (strcmp ("type_of_service" , var_name) == 0)
10639           {
10640           *var_p_ptr = (void *) (&prs_ptr->type_of_service);
10641           FOUT
10642           }
10643       if (strcmp ("backoff_slot_ptr" , var_name) == 0)
10644           {
10645           *var_p_ptr = (void *) (&prs_ptr->backoff_slot_ptr);
10646           FOUT
10647           }
10648       if (strcmp ("max_backoff_tmp" , var_name) == 0)
10649           {
10650           *var_p_ptr = (void *) (&prs_ptr->max_backoff_tmp);
10651           FOUT
10652           }
10653       if (strcmp ("backoff_max_handle" , var_name) == 0)
10654           {
10655           *var_p_ptr = (void *) (&prs_ptr->backoff_max_handle);
10656           FOUT
10657           }
10658       if (strcmp ("backoff_min_handle" , var_name) == 0)
10659           {
10660           *var_p_ptr = (void *) (&prs_ptr->backoff_min_handle);
10661           FOUT
10662           }
10663       if (strcmp ("ac_slot" , var_name) == 0)
10664           {
10665           *var_p_ptr = (void *) (&prs_ptr->ac_slot);
10666           FOUT
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
739     OpT_Packet_Size                 packet_size_dcf
740     OpT_Packet_Size                 packet_size_pcf
741     double                          receive_time_dcf
742     double                          receive_time_pcf
743     Boolean                         cfp_ap_medium_control
744     int                             pcf_network
745     int                             beacon_eff_mode
746     int                             channel_count
747
748
749     int                             channel_num
750     double                          first_chan_min_freq
751
752     double                          channel_bandwidth
753
754     double                          channel_spacing
755
756     int                             eval_bss_id
757     WlanT_Roam_State_Info*          roam_state_ptr
758     WlanT_Rx_State_Info*            rx_state_info_ptr
759     double                          ap_connectivity_check_interval
760     double                          ap_connectivity_check_time
761     Evhandle                        ap_connectivity_check_evhndl
762     WlanT_AP_Position_Info*         conn_ap_pos_info_ptr
763     WlanT_Sta_Mapping_Info*         my_sta_info_ptr
764     WlanT_Bss_Mapping_Info*         my_bss_info_ptr
765
766     PrgT_Mutex*                     mapping_info_mutex
767
768
769     int                             ac
770     int                             aift_ac
771     int                             type_of_service
772     Distribution *                  backoff_slot_ptr
773     int                             max_backoff_tmp
774     Stathandle                      backoff_max_handle
775     Stathandle                      backoff_min_handle
776     int                             ac_slot
777     } wlan_mac_QOS_state;
778
779 #define pr_state_ptr                    ((wlan_mac_QOS_state*) (OP_SIM_CONTEXT_PTR
780 #define retry_count                     pr_state_ptr->retry_count
781 #define intrpt_type                     pr_state_ptr->intrpt_type
782 #define intrpt_code                     pr_state_ptr->intrpt_code
783 #define my_address                      pr_state_ptr->my_address
784 #define my_objid                        pr_state_ptr->my_objid
785 #define my_node_objid                   pr_state_ptr->my_node_objid
786 #define my_subnet_objid                 pr_state_ptr->my_subnet_objid
787 #define tx_objid                        pr_state_ptr->tx_objid
788 #define txch_objid                      pr_state_ptr->txch_objid
789 #define rx_objid                        pr_state_ptr->rx_objid
790 #define rxch_objid                      pr_state_ptr->rxch_objid
791 #define own_process_record_handle       pr_state_ptr->own_process_record_handle
792 #define hld_list_ptr                    pr_state_ptr->hld_list_ptr
793 #define data_tx_rate                    pr_state_ptr->data_tx_rate
794 #define operational_speed               pr_state_ptr->operational_speed
795 #define control_data_rate               pr_state_ptr->control_data_rate
796 #define rcvd_frame_drate                pr_state_ptr->rcvd_frame_drate
797 #define frag_threshold                  pr_state_ptr->frag_threshold
798 #define packet_seq_number               pr_state_ptr->packet_seq_number
799 #define packet_frag_number              pr_state_ptr->packet_frag_number
800 #define destination_addr                pr_state_ptr->destination_addr
801 #define fragmentation_buffer_ptr        pr_state_ptr->fragmentation_buffer_ptr
802 #define common_rsmbuf_ptr               pr_state_ptr->common_rsmbuf_ptr
803 #define fresp_to_send                   pr_state_ptr->fresp_to_send
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
775     Stathandle                      backoff_min_handle
776     int                             ac_slot
777     } wlan_mac_QOS_state;
778
779 #define pr_state_ptr                    ((wlan_mac_QOS_state*) (OP_SIM_CONTEXT_PTR
780 #define retry_count                     pr_state_ptr->retry_count
781 #define intrpt_type                     pr_state_ptr->intrpt_type
782 #define intrpt_code                     pr_state_ptr->intrpt_code
783 #define my_address                      pr_state_ptr->my_address
784 #define my_objid                        pr_state_ptr->my_objid
785 #define my_node_objid                   pr_state_ptr->my_node_objid
786 #define my_subnet_objid                 pr_state_ptr->my_subnet_objid
787 #define tx_objid                        pr_state_ptr->tx_objid
788 #define txch_objid                      pr_state_ptr->txch_objid
789 #define rx_objid                        pr_state_ptr->rx_objid
790 #define rxch_objid                      pr_state_ptr->rxch_objid
791 #define own_process_record_handle       pr_state_ptr->own_process_record_handle
792 #define hld_list_ptr                    pr_state_ptr->hld_list_ptr
793 #define data_tx_rate                    pr_state_ptr->data_tx_rate
794 #define operational_speed               pr_state_ptr->operational_speed
795 #define control_data_rate               pr_state_ptr->control_data_rate
796 #define rcvd_frame_drate                pr_state_ptr->rcvd_frame_drate
797 #define frag_threshold                  pr_state_ptr->frag_threshold
798 #define packet_seq_number               pr_state_ptr->packet_seq_number
799 #define packet_frag_number              pr_state_ptr->packet_frag_number
800 #define destination_addr                pr_state_ptr->destination_addr
801 #define fragmentation_buffer_ptr        pr_state_ptr->fragmentation_buffer_ptr
802 #define common_rsmbuf_ptr               pr_state_ptr->common_rsmbuf_ptr
803 #define fresp_to_send                   pr_state_ptr->fresp_to_send
804 #define nav_duration                    pr_state_ptr->nav_duration
805 #define rts_threshold                   pr_state_ptr->rts_threshold
806 #define duplicate_entry                 pr_state_ptr->duplicate_entry
807 #define expected_frame_type             pr_state_ptr->expected_frame_type
808 #define remote_sta_addr                 pr_state_ptr->remote_sta_addr
809 #define backoff_slots                   pr_state_ptr->backoff_slots
810 #define packet_load_handle              pr_state_ptr->packet_load_handle
811 #define intrpt_time                     pr_state_ptr->intrpt_time
812 #define wlan_transmit_frame_copy_ptr        pr_state_ptr->wlan_transmit_frame_copy
813 #define backoff_slots_handle            pr_state_ptr->backoff_slots_handle
814 #define instrm_from_mac_if              pr_state_ptr->instrm_from_mac_if
815 #define outstrm_to_mac_if               pr_state_ptr->outstrm_to_mac_if
816 #define num_fragments                   pr_state_ptr->num_fragments
817 #define remainder_size                  pr_state_ptr->remainder_size
818 #define defragmentation_list_ptr        pr_state_ptr->defragmentation_list_ptr
819 #define wlan_flags                      pr_state_ptr->wlan_flags
820 #define oms_aa_handle                   pr_state_ptr->oms_aa_handle
821 #define current_time                    pr_state_ptr->current_time
822 #define rcv_idle_time                   pr_state_ptr->rcv_idle_time
823 #define hld_pmh                         pr_state_ptr->hld_pmh
824 #define max_backoff                     pr_state_ptr->max_backoff
825 #define min_backoff                     pr_state_ptr->min_backoff
826 #define current_state_name              pr_state_ptr->current_state_name
827 #define hl_packets_rcvd                 pr_state_ptr->hl_packets_rcvd
828 #define media_access_delay              pr_state_ptr->media_access_delay
829 #define ete_delay_handle                pr_state_ptr->ete_delay_handle
830 #define global_ete_delay_handle         pr_state_ptr->global_ete_delay_handle
831 #define global_throughput_handle        pr_state_ptr->global_throughput_handle
832 #define global_load_handle              pr_state_ptr->global_load_handle
833 #define global_dropped_data_handle      pr_state_ptr->global_dropped_data_handle
834 #define global_mac_delay_handle         pr_state_ptr->global_mac_delay_handle
835 #define ctrl_traffic_rcvd_handle_inbits     pr_state_ptr->ctrl_traffic_rcvd_handle
836 #define ctrl_traffic_sent_handle_inbits     pr_state_ptr->ctrl_traffic_sent_handle
837 #define ctrl_traffic_rcvd_handle        pr_state_ptr->ctrl_traffic_rcvd_handle
838 #define ctrl_traffic_sent_handle        pr_state_ptr->ctrl_traffic_sent_handle
839 #define data_traffic_rcvd_handle_inbits     pr_state_ptr->data_traffic_rcvd_handle
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan

```
814  #define instrm_from_mac_if              pr_state_ptr->instrm_from_mac_if
815  #define outstrm_to_mac_if               pr_state_ptr->outstrm_to_mac_if
816  #define num_fragments                   pr_state_ptr->num_fragments
817  #define remainder_size                  pr_state_ptr->remainder_size
818  #define defragmentation_list_ptr        pr_state_ptr->defragmentation_list_ptr
819  #define wlan_flags                      pr_state_ptr->wlan_flags
820  #define oms_aa_handle                   pr_state_ptr->oms_aa_handle
821  #define current_time                    pr_state_ptr->current_time
822  #define rcv_idle_time                   pr_state_ptr->rcv_idle_time
823  #define hld_pmh                         pr_state_ptr->hld_pmh
824  #define max_backoff                     pr_state_ptr->max_backoff
825  #define min_backoff                     pr_state_ptr->min_backoff
826  #define current_state_name              pr_state_ptr->current_state_name
827  #define hl_packets_rcvd                 pr_state_ptr->hl_packets_rcvd
828  #define media_access_delay              pr_state_ptr->media_access_delay
829  #define ete_delay_handle                pr_state_ptr->ete_delay_handle
830  #define global_ete_delay_handle         pr_state_ptr->global_ete_delay_handle
831  #define global_throughput_handle        pr_state_ptr->global_throughput_handle
832  #define global_load_handle              pr_state_ptr->global_load_handle
833  #define global_dropped_data_handle      pr_state_ptr->global_dropped_data_handle
834  #define global_mac_delay_handle         pr_state_ptr->global_mac_delay_handle
835  #define ctrl_traffic_rcvd_handle_inbits    pr_state_ptr->ctrl_traffic_rcvd_handle
836  #define ctrl_traffic_sent_handle_inbits    pr_state_ptr->ctrl_traffic_sent_handle
837  #define ctrl_traffic_rcvd_handle        pr_state_ptr->ctrl_traffic_rcvd_handle
838  #define ctrl_traffic_sent_handle        pr_state_ptr->ctrl_traffic_sent_handle
839  #define data_traffic_rcvd_handle_inbits    pr_state_ptr->data_traffic_rcvd_handle
840  #define data_traffic_sent_handle_inbits    pr_state_ptr->data_traffic_sent_handle
841  #define data_traffic_rcvd_handle        pr_state_ptr->data_traffic_rcvd_handle
842  #define data_traffic_sent_handle        pr_state_ptr->data_traffic_sent_handle
843  #define sifs_time                       pr_state_ptr->sifs_time
844  #define slot_time                       pr_state_ptr->slot_time
845  #define cw_min                          pr_state_ptr->cw_min
846  #define cw_max                          pr_state_ptr->cw_max
847  #define cw_min_tmp                      pr_state_ptr->cw_min_tmp
848  #define cw_max_tmp                      pr_state_ptr->cw_max_tmp
849  #define difs_time                       pr_state_ptr->difs_time
850  #define aifs_time                       pr_state_ptr->aifs_time
851  #define plcp_overhead_control           pr_state_ptr->plcp_overhead_control
852  #define plcp_overhead_data              pr_state_ptr->plcp_overhead_data
853  #define channel_reserv_handle           pr_state_ptr->channel_reserv_handle
854  #define retrans_handle                  pr_state_ptr->retrans_handle
855  #define throughput_handle               pr_state_ptr->throughput_handle
856  #define long_retry_limit                pr_state_ptr->long_retry_limit
857  #define short_retry_limit               pr_state_ptr->short_retry_limit
858  #define retry_limit                     pr_state_ptr->retry_limit
859  #define last_frametx_type               pr_state_ptr->last_frametx_type
860  #define deference_evh                   pr_state_ptr->deference_evh
861  #define backoff_elapsed_evh             pr_state_ptr->backoff_elapsed_evh
862  #define frame_timeout_evh               pr_state_ptr->frame_timeout_evh
863  #define eifs_time                       pr_state_ptr->eifs_time
864  #define i_strm                          pr_state_ptr->i_strm
865  #define wlan_trace_active               pr_state_ptr->wlan_trace_active
866  #define pkt_in_service                  pr_state_ptr->pkt_in_service
867  #define bits_load_handle                pr_state_ptr->bits_load_handle
868  #define ap_flag                         pr_state_ptr->ap_flag
869  #define bss_flag                        pr_state_ptr->bss_flag
870  #define ap_mac_address                  pr_state_ptr->ap_mac_address
871  #define hld_max_size                    pr_state_ptr->hld_max_size
872  #define max_receive_lifetime            pr_state_ptr->max_receive_lifetime
873  #define accept_large_packets            pr_state_ptr->accept_large_packets
874  #define phy_char_flag                   pr_state_ptr->phy_char_flag
875  #define ap_phy_char_flag                pr_state_ptr->ap_phy_char_flag
876  #define phy_type                        pr_state_ptr->phy_type
877  #define total_hlpk_size                 pr_state_ptr->total_hlpk_size
878  #define drop_packet_handle              pr_state_ptr->drop_packet_handle
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan_mac_

```
8219
8220            if (wlan_flags->backoff_flag == OPC_TRUE || wlan_flags->perform_cw == OPC_TRUE
8221                {
8222                if (backoff_slots == BACKOFF_SLOTS_UNSET)
8223                    {
8224                    /* Compute backoff interval using binary exponential process.   */
8225                    /* After a successful transmission we always use cw_min.        */
8226                    if (retry_count == 0 || wlan_flags->perform_cw == OPC_TRUE)
8227                      {
8228                        /* If retry count is set to 0 then set the maximum backoff  */
8229                        /* slots to min window size.                                */
8230                        switch (ac)
8231                          {
8232                          case 0: {
8233                                  /* if (cw_max > 3) { */ max_backoff = ((cw_min+1)/2)-1;/
8234                                   /*if (cw_min > 7) { */ min_backoff = ((cw_min+1)/4)-1;/
8235                                   }
8236                          case 1: {
8237                                   max_backoff = cw_min;
8238                                  /*if (cw_min > 3) {*/ min_backoff = ((cw_min+1)/2)-1; /*
8239                                   }
8240                          case 2: {
8241                                  max_backoff = cw_max;
8242                                  min_backoff = cw_min; break;
8243                                  }
8244                          case 3: {
8245                                  max_backoff = cw_max;
8246                                  min_backoff = cw_min; break;
8247                                  }
8248                          }
8249
8250
8251                        max_backoff = cw_min;
8252                        }
8253                    else
8254                        {
8255                        /* We are retransmitting. Increase the back-off window      */
8256                        /* size.                                                    */
8257                        switch (ac)
8258                            {
8259                            case 0: {
8260                                    /*if (cw_max > 3) { */max_backoff = ((cw_min+1)/2)-1;
8261                                    /*if (cw_min > 7) { */min_backoff = ((cw_min+1)/4)-1;
8262                                    }
8263                             case 1: {
8264                                     max_backoff = cw_min;
8265                                    /* if (cw_min > 3) {*/ min_backoff = ((cw_min+1)/2)-1
8266                                     }
8267                             case 2: {
8268                                     max_backoff = cw_max;
8269                                     min_backoff = cw_min;
8270                                     max_backoff = max_backoff + max_backoff_tmp;
8271                                     max_backoff_tmp = max_backoff;break;
8272                                     }
8273
8274                             case 3: {
8275                                     max_backoff = cw_max;
8276                                     min_backoff = cw_min;
8277                                     max_backoff = max_backoff + max_backoff_tmp;
8278                                     max_backoff_tmp = max_backoff;break;
8279                                     }
8280                            }
8281
8282                        }
8283                    /* The number of possible slots grows exponentially until it    */
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan_mac_

```
8284                    /* exceeds a fixed limit.                                   */
8285                    if (max_backoff > cw_max)
8286                        {
8287                        max_backoff = cw_max;
8288                        }
8289
8290                    /* Obtain a uniformly distributed random integer between 0 and  */
8291                    /* the minimum contention window size. Scale the number of      */
8292                    /* slots according to the number of retransmissions.            */
8293
8294                    backoff_slots = floor (op_dist_uniform (max_backoff + 1));
8295                    }
8296
8297
8298                /* Set a timer for the end of the backoff interval.                 */
8299                intrpt_time = (current_time + backoff_slots * slot_time);
8300
8301
8302                /*restore the cw_max  value for next time using
8303                max_backoff = cw_max_tmp;
8304
8305                restore the cw_min value for next time using
8306                min_backoff = cw_min_tmp;*/
8307
8308                /* Scheduling self interrupt for backoff.                           */
8309                if (wlan_flags->perform_cw == OPC_TRUE)
8310                    backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_CW_E
8311                else
8312                    backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_Back
8313
8314                /* Reporting number of backoff slots as a statistic.                */
8315                op_stat_write (backoff_slots_handle, backoff_slots);
8316                op_stat_write (backoff_max_handle, max_backoff);
8317                op_stat_write (backoff_min_handle, min_backoff);
8318                }
8319            }
8320            FSM_PROFILE_SECTION_OUT (state3_enter_exec)
8321
8322        /** state (BKOFF_NEEDED) exit executives **/
8323        FSM_STATE_EXIT_FORCED (3, "BKOFF_NEEDED", "wlan_mac_QOS [BKOFF_NEEDED exit execs]"
8324
8325
8326        /** state (BKOFF_NEEDED) transition processing **/
8327        FSM_PROFILE_SECTION_IN ("wlan_mac_QOS [BKOFF_NEEDED trans conditions]", state3_tra
8328        FSM_INIT_COND (TRANSMIT_FRAME)
8329        FSM_TEST_COND (PERFORM_BACKOFF)
8330        FSM_TEST_LOGIC ("BKOFF_NEEDED")
8331        FSM_PROFILE_SECTION_OUT (state3_trans_conds)
8332
8333        FSM_TRANSIT_SWITCH
8334            {
8335            FSM_CASE_TRANSIT (0, 4, state4_enter_exec, ;, "TRANSMIT_FRAME", "", "BKOFF_NEE
8336            FSM_CASE_TRANSIT (1, 5, state5_enter_exec, ;, "PERFORM_BACKOFF", "", "BKOFF_NE
8337            }
8338            /*---------------------------------------------------------*/
8339
8340
8341
8342        /** state (TRANSMIT) enter executives **/
8343        FSM_STATE_ENTER_UNFORCED (4, "TRANSMIT", state4_enter_exec, "wlan_mac_QOS [TRANSMI
8344            FSM_PROFILE_SECTION_IN ("wlan_mac_QOS [TRANSMIT enter execs]", state4_enter_ex
8345            {
8346            /** In this state following intrpts can occur:                  **/
8347            /** 1. Data arrival from application layer.                     **/
8348            /** 2. Frame (DATA,ACK,RTS,CTS) rcvd from PHY layer.            **/
```

```
8414                strcpy (current_state_name, "transmit");
8415                }
8416
8417            /* Unlock the mutex that serializes accessing the       */
8418            /* roaming related information of this MAC.             */
8419            op_prg_mt_mutex_unlock (roam_state_ptr->roam_info_mutex);
8420            }
8421            FSM_PROFILE_SECTION_OUT (state4_enter_exec)
8422
8423        /** blocking after enter executives of unforced state. **/
8424        FSM_EXIT (9,"wlan_mac_QOS")
8425
8426
8427        /** state (TRANSMIT) exit executives **/
8428        FSM_STATE_EXIT_UNFORCED (4, "TRANSMIT", "wlan_mac_QOS [TRANSMIT exit execs]")
8429            FSM_PROFILE_SECTION_IN ("wlan_mac_QOS [TRANSMIT exit execs]", state4_exit_exec
8430            {
8431            /* Lock the mutex that serializes accessing the roaming     */
8432            /* related information of this MAC.                         */
8433            op_prg_mt_mutex_lock (roam_state_ptr->roam_info_mutex, 0);
8434
8435            /* Check the interrupt type.                                */
8436            if (op_intrpt_type () == OPC_INTRPT_STAT)
8437                {
8438                /* If the packet is received while the the station is   */
8439                /* transmitting then mark the received packet as bad.   */
8440                intrpt_code = (WlanT_Mac_Intrpt_Code) op_intrpt_stat ();
8441                if (intrpt_code < TRANSMITTER_BUSY_INSTAT && op_stat_local_read (intrpt_co
8442                    {
8443                    wlan_flags->rcvd_bad_packet = OPC_TRUE;
8444
8445                    /* If we are transmitting a CTS-to-self, then mark  */
8446                    /* it as bad, too.                                  */
8447                    if (last_frametx_type == WlanC_Cts && expected_frame_type == WlanC_Cts
8448                        wlan_flags->rcvd_bad_cts = OPC_TRUE;
8449                    }
8450
8451                /* If we completed the transmission then reset the      */
8452                /* transmitter flag.                                    */
8453                else if (intrpt_code == TRANSMITTER_BUSY_INSTAT)
8454                    {
8455                    wlan_flags->transmitter_busy = OPC_FALSE;
8456
8457                    /* Also update the receiver idle time, since with   */
8458                    /* the end of our transmission, the medium may      */
8459                    /* become idle again.                               */
8460                    rcv_idle_time = op_sim_time ();
8461                    }
8462                }
8463
8464            else if ((op_intrpt_type () == OPC_INTRPT_STRM) && (op_intrpt_strm () != instr
8465                {
8466                /* While transmitting, we received a packet from        */
8467                /* physical layer. Mark the packet as bad.              */
8468                wlan_flags->rcvd_bad_packet = OPC_TRUE;
8469
8470                /* If we are transmitting a CTS-to-self, then mark it   */
8471                /* as bad, too.                                         */
8472                if (last_frametx_type == WlanC_Cts && expected_frame_type == WlanC_Cts)
8473                    wlan_flags->rcvd_bad_cts = OPC_TRUE;
8474                }
8475
8476            /* Call the interrupt processing routine for each interrupt.*/
8477            wlan_interrupts_process ();
8478            }
```

```
8834                    /* not unnecessarily delayed.                          */
8835                    nav_duration = current_time;
8836
8837                    /* Reset the rts_sent flag in case we didn't receive an */
8838                    /* ACK for our data tranmission in spite of a           */
8839                    /* successful RTS/CTS frame exchange.                   */
8840                    wlan_flags->rts_sent = OPC_FALSE;
8841
8842                    /* Check whether further retries are possible or the    */
8843                    /* data frame needs to be discarded.                    */
8844                    wlan_frame_discard ();
8845                    }
8846                }
8847
8848            }
8849            FSM_PROFILE_SECTION_OUT (state7_exit_exec)
8850
8851
8852        /** state (WAIT_FOR_RESPONSE) transition processing **/
8853        FSM_PROFILE_SECTION_IN ("wlan_mac_QOS [WAIT_FOR_RESPONSE trans conditions]", state
8854        FSM_INIT_COND (FRAME_TIMEOUT || FRAME_RCVD)
8855        FSM_DFLT_COND
8856        FSM_TEST_LOGIC ("WAIT_FOR_RESPONSE")
8857        FSM_PROFILE_SECTION_OUT (state7_trans_conds)
8858
8859        FSM_TRANSIT_SWITCH
8860            {
8861            FSM_CASE_TRANSIT (0, 6, state6_enter_exec, ;, "FRAME_TIMEOUT || FRAME_RCVD", "
8862            FSM_CASE_TRANSIT (1, 7, state7_enter_exec, ;, "default", "", "WAIT_FOR_RESPONS
8863            }
8864            /*---------------------------------------------------------*/
8865
8866
8867
8868        /** state (BSS_INIT) enter executives **/
8869        FSM_STATE_ENTER_UNFORCED (8, "BSS_INIT", state8_enter_exec, "wlan_mac_QOS [BSS_INI
8870            FSM_PROFILE_SECTION_IN ("wlan_mac_QOS [BSS_INIT enter execs]", state8_enter_ex
8871            {
8872            /* Schedule a self interrupt to wait for mac interface  */
8873            /* to move to next state after registering              */
8874            op_intrpt_schedule_self (op_sim_time (), 0);
8875            }
8876            FSM_PROFILE_SECTION_OUT (state8_enter_exec)
8877
8878        /** blocking after enter executives of unforced state. **/
8879        FSM_EXIT (17,"wlan_mac_QOS")
8880
8881
8882        /** state (BSS_INIT) exit executives **/
8883        FSM_STATE_EXIT_UNFORCED (8, "BSS_INIT", "wlan_mac_QOS [BSS_INIT exit execs]")
8884            FSM_PROFILE_SECTION_IN ("wlan_mac_QOS [BSS_INIT exit execs]", state8_exit_exec
8885            {
8886            /* Obtain the values assigned to the various attributes             */
8887            op_ima_obj_attr_get (my_objid, "Wireless LAN Parameters", &wlan_params_comp_at
8888            params_attr_objid = op_topo_child (wlan_params_comp_attr_objid, OPC_OBJTYPE_GE
8889
8890            /* Determining the final MAC address after address resolution.      */
8891            op_ima_obj_attr_get (my_objid, "Address", &my_address);
8892
8893            /* Determine the assinged access categories */
8894
8895            op_ima_obj_attr_get (my_objid,"AC", &ac);
8896
8897            /* Update our own process registery record with the final address   */
8898            /* information.                                                     */
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan_mac_

```
9466  FIN_MT (wlan_mac_QOS_diag ())
9467
9468  if (1)
9469      {
9470      /* Temporary Variables */
9471      /* variables used for registering and discovering process models */
9472      OmsT_Pr_Handle          process_record_handle;
9473      List*                   proc_record_handle_list_ptr;
9474      int                     record_handle_list_size;
9475      int                     ap_count;
9476      double                  sta_addr;
9477      double                  statype ;
9478      Objid                   mac_objid;
9479      Objid                   mac_if_module_objid;
9480      Objid                   parent_subnet_objid;
9481      char                    name_str [128];
9482      Objid                   params_attr_objid;
9483      Objid                   wlan_params_comp_attr_objid;
9484      int                     i_cnt, j_cnt, k_cnt;
9485      int                     addr_index;
9486      WlanT_Hld_List_Elem*    hld_ptr;
9487      Prohandle               own_prohandle;
9488      double                  timer_duration;
9489      char                    msg1 [256];
9490      WlanT_Phy_Char_Code     sta_phy_char_flag;
9491      Boolean                 bad_packet_rcvd = OPC_FALSE;
9492      Boolean                 bad_cts_to_self_rcvd;
9493      Boolean                 pre_rx_status;
9494      double                  pcf_active;
9495      int                     address;
9496      int                     pcf_enabled_stations;
9497      Boolean                 pcf_enabled_on_AP;
9498      double                  tx_power;
9499      double                  x_pos, y_pos, z_pos;
9500      /* End of Temporary Variables */
9501
9502      /* Diagnostic Block */
9503
9504
9505      BINIT
9506      {
9507      /* Print information about this process.    */
9508      if (wlan_trace_active == OPC_TRUE)
9509          {
9510          printf ("Current state name:%s\t", current_state_name);
9511          }
9512
9513      printf ("Station MAC Address: %d\n\n", my_address);
9514
9515      printf ("Total buffer usage: " OPC_PACKET_SIZE_FMT "/%d bits.\n\n", total_hlpk_size, h
9516
9517      /* Print the contents of the DCF queue.     */
9518      printf ("Printing the higher layer DCF queue contents (packet ids):\n");
9519      for (i_cnt = 0; i_cnt < op_prg_list_size (hld_list_ptr);)
9520          {
9521          hld_ptr = (WlanT_Hld_List_Elem*) op_prg_list_access (hld_list_ptr, i_cnt);
9522          printf ("" OPC_PACKET_ID_FMT "\t", op_pk_id (hld_ptr->pkptr));
9523          if ((++i_cnt % 4) == 0)
9524              printf ("\n");
9525          }
9526      printf ("\n");
9527
9528      /* Print the contents of the PCF queue if   */
9529      /* we are a PCF-enabled access point.       */
9530      if (active_pc)
```

C:\Program Files\OPNET\10.5.A\models\std\wireless_lan\wlan_mac_

```
9578
9579
9580
9581
9582 void
9583 wlan_mac_QOS_terminate (OP_SIM_CONTEXT_ARG_OPT)
9584     {
9585
9586 #if !defined (VOSD_NO_FIN)
9587     int _op_block_origin = __LINE__;
9588 #endif
9589
9590     FIN_MT (wlan_mac_QOS_terminate ())
9591
9592     if (1)
9593         {
9594         /* Temporary Variables */
9595         /* variables used for registering and discovering process models */
9596         OmsT_Pr_Handle          process_record_handle;
9597         List*                   proc_record_handle_list_ptr;
9598         int                     record_handle_list_size;
9599         int                     ap_count;
9600         double                  sta_addr;
9601         double                  statype ;
9602         Objid                   mac_objid;
9603         Objid                   mac_if_module_objid;
9604         Objid                   parent_subnet_objid;
9605         char                    name_str [128];
9606         Objid                   params_attr_objid;
9607         Objid                   wlan_params_comp_attr_objid;
9608         int                     i_cnt, j_cnt, k_cnt;
9609         int                     addr_index;
9610         WlanT_Hld_List_Elem*    hld_ptr;
9611         Prohandle               own_prohandle;
9612         double                  timer_duration;
9613         char                    msg1 [256];
9614         WlanT_Phy_Char_Code     sta_phy_char_flag;
9615         Boolean                 bad_packet_rcvd = OPC_FALSE;
9616         Boolean                 bad_cts_to_self_rcvd;
9617         Boolean                 pre_rx_status;
9618         double                  pcf_active;
9619         int                     address;
9620         int                     pcf_enabled_stations;
9621         Boolean                 pcf_enabled_on_AP;
9622         double                  tx_power;
9623         double                  x_pos, y_pos, z_pos;
9624         /* End of Temporary Variables */
9625
9626         /* Termination Block */
9627
9628
9629         BINIT
9630         {
9631
9632         }
9633
9634         /* End of Termination Block */
9635
9636         }
9637     Vos_Poolmem_Dealloc_MT (OP_SIM_CONTEXT_THREAD_INDEX_COMMA pr_state_ptr);
9638
9639     FOUT
9640     }
9641
9642
```

```
class queueTimer : public TimerHandler {                    //Timer attached to
public:                                                     //droptail queue
    queueTimer(DropTail *d) : TimerHandler(), d_(d) { }
protected:
    virtual void expire(Event *);
    DropTail *d_;
};

void queueTimer::expire(Event *e)                  // Timer expired – increase priority
{
    printf("\nTimer expired!");
    int pr = d_->get_prio();
    printf("\nPkt priority = %d",pr);
    if (d_->length() != 0) {
        Packet *p = d_->deque();                    //remove packet from old queue
        printf("\nPacket dequed from old queue");
        PriQueue priq;
        priq.incr_prio(p,pr);
    }
    else printf("\nQueue empty!!");
}

void DropTail::enque(Packet* p)
{

    PacketQueue* q_ = pq_;
    save_prio(p);
    if (pqlow_ && (HDR_IP(p)->prio() > 0)) {
        if (pq_->length() >= pqlim_) {
            q_ = pqlow_;
        } else {
            assert(!pqlow_->length());
        }
    }

    if (q_->head() == 0)                    //first packet – set timer
        qTimer.resched((double)timeout);
    q_->enque(p);


    if (q_->length() >= limit()) {
        if (drop_front_) { /* remove from head of queue */
            Packet *pp = q_->deque();
            drop(pp, DROP_IFQ_QFULL);
```

```
            } else {
                q_->remove(p);
                drop(p, DROP_IFQ_QFULL);
  }
        }
}

void DropTail::enque_pkt(Packet* p)                    //enque packet in new queue
{
      if ( length() == 0)
            qTimer.resched((double)timeout);
      enque(p);
      printf("\nPacket enqued in new queue!!");
      if(!blocked())
         resume();
}

Packet* DropTail::deque()
{
      Packet* p = pq_->deque();
      qTimer.resched((double)delay);              // packet removed from queue – reset
      if (!pqlow_) {                              // timer for queue
            return p;
      }

      if (!p) {
            assert(!pqlim_ || !pqlow_->length());
            return pqlow_->deque();
      }

      if ((pq_->length() < pqlim_) && pqlow_->length()) {
            pq_->enque(pqlow_->deque());
      }

      return p;
}

void PriQueue::incr_prio(Packet *p,int pri)          // called after timer expires for queue
{
      int flag = 0;
      int old_prio = pri;
      printf("\nOld Priorityyy = %d",pri);
      if(pri==0)
      {
            printf("\nPriority cannot be increased");
      }
```

```
        else
        {
                pri--;                                  //goto higher level queues
                int pr = map_plevel(pri);

               for ( int i = pri ; (i >= 0 && !flag) ; i-- ) {
                        if ( priq_[(i)].length() >= priq_[(i)].qlim_ ) {
                                printf("\nQueue %d FULL!!",pri);
                               continue;
                        }
                        else {
                                flag = 1;
                                pri = HDR_IP(p)->changePrio(p,pri);
                                printf("\nPriority changed to %d ", HDR_IP(p)->prio());
                                priq_[(i)].enque(p);
                                printf("Packet enqued in new queue!!");
                        }
                }
        }

        if (!flag) {
                printf("\nPacket priority cannot be changed:( ");
                priq_[map_plevel(old_prio)].enqueHead(p);
                printf("\nPacket enqued in old queue %d",old_prio);
        }
}
```