AN ENHANCED DISTRIBUTED CHANNEL ACCESS DEFICIT ROUND-ROBIN (EDRR) SCHEME FOR IEEE 802.11E WIRELESS NETWORKS

by

CASEY POWELL

B.Sc., The University of Nevada, Reno, 2002

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical and Computer Engineering)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

November 2004

University of British Columbia
Abstract

AN ENHANCED DISTRIBUTED CHANNEL ACCESS DEFICIT ROUND-ROBIN SCHEME FOR IEEE 802.11E WIRELESS NETWORKS

by Casey Powell

Supervisors: Dr. Hussein Alnuweiri and Dr. Panos Nasiopoulos

Department of Electrical and Computer Engineering

IEEE 802.11 wireless local area network (WLAN) is the most popular and most widely deployed WLAN technology in the world. From large enterprise environments to the everyday home user, 802.11 networks are becoming almost as ubiquitous as wired Ethernet. With the explosion in use of wireless networks comes the increasing demand for them to support the latest technologies. Bandwidth intensive and latency-sensitive applications such as VoIP and streaming video are enjoying huge growth in industry, leading to the requirement that 802.11 WLANs should provide quality of service for these applications.

Development of mechanisms for supporting quality of service at the medium access control layer in IEEE 802.11 WLANs is therefore an issue that is at the forefront of the research community. Task Group E of the 802.11 standards committee is working on a standard that adds quality of service enhancements in the 802.11 MAC layer. They do so by using traffic flows and prioritizing traffic according to what type of application it belongs to. There are, however, some weaknesses in the design that is currently being proposed by the 802.11 standards committee.

In this thesis we propose several enhancements to the IEEE 802.11e draft standard that will provide improved quality of service for all traffic priorities, whether high or low. Through simulation and analytical investigation, we are able to show that we can do so without negatively affecting the performance of any of the wireless stations in the network or the traffic flows that they contain. We then conclude that our proposed scheme provides an attractive for enhancing the IEEE 802.11e standard.

TABLE OF CONTENTS

LIST OF FIGURES

Number *Page*

LIST OF TABLES

LIST OF ABBREVIATIONS

AC – Access Category

ACK – Acknowledgment

AES – Advanced Encryption Standard

AIFS – Arbitration Inter-Frame Space

AP – Access Point

BEB – Binary Exponential Backoff

BI – Backoff Interval

BSS – Basic Service Set

CAP – Controlled Access Period

CFP – Contention-Free Period

CP – Contention Period

CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance

CSMA/CD – Carrier Sense Multiple Access with Collision Detection

CTS – Clear to Send

CW – Contention Window

DC – Deficit Count

DCF – Distributed Coordination Function

DDRR – Distributed Deficit Round-Robin

DIFS – Distributed Coordination Function Inter-Frame Space

DRR – Deficit Round-Robin

DS – Distribution System

DSSS – Direct Sequence Spread Spectrum

EDCA – Enhanced Distributed Channel Access

EDRR – Enhanced Distributed Channel Access Deficit Round-Robin

ESS – Extended Service Set

FH – Frequency Hopping

FSM – Finite State Machine

HC – Hybrid Coordinator

HCCA – Hybrid Coordination Function Controlled Channel Access

HCF – Hybrid Coordination Function

IBSS – Independent Basic Service Set

IFS – Inter-Frame Space

ISM – Industrial, Scientific, and Medical

LLC – Logical Link Control

MAC – Medium Access Control

MPDU – Medium Access Control Protocol Data Unit

NAV – Network Allocation Vector

OFDM – Orthogonal Frequency Division Multiplexing

OPNET – Optimized Network Engineering Tools

OSI – Open System Interconnection

PCF – Point Coordination Function

PDU – Protocol Data Unit

PHY – Physical Layer

PIFS – Point Coordination Function Inter-Frame Space

PLCP – Physical Layer Convergence Procedure

PMD – Physical Medium Dependent

Q – Quantum of Service

QAP – Quality of Service Access Point

QBSS – Quality of Service Basic Service Set

QIBSS – Quality of Service Independent Basic Service Set

QoS – Quality of Service

QSTA – Quality of Service Wireless Station

RTS – Request to Send

SIFS – Short Inter-Frame Space

STA – Wireless Station

TC – Traffic Category

TSPC – Traffic Specification

TXOP – Transmit Opportunity

UP – User Priority

WLAN – Wireless Local Area Network

ACKNOWLEDGMENTS

I would first like to express my sincere gratitude to both of my supervisors Dr. Hussein Alnuweiri and Dr. Panos Nasiopoulos. Without their continued guidance, enlightenment, and persistence, this work would not have been possible.

I would also like to acknowledge the support of Spectrum Signal Processing for their input on our projects, as well as for the financial assistance they provided.

Many thanks go to my family, who have supported me throughout my academic career, regardless of where or what I chose to study. They have been my driving force, and I am forever indebted to them for their support and encouragement.

Last, but definitely not least, I would like to thank all the wonderful friends that I have made while studying in Vancouver. You have made this a truly unforgettable experience, and I am extremely grateful for this.

Chapter 1

INTRODUCTION

This chapter introduces material and motivations for the research that is presented in this thesis. We begin with an overview of IEEE 802.11 wireless networks in Section 1.1. Motivations that are driving research in IEEE 802.11 networks are then discussed in Section 1.2. Related work is discussed in Section 1.3, followed by a summary of the thesis contributions in Section 1.4. Finally, Section 1.5 gives an outline of the remaining chapters of this thesis.

1.1 History of IEEE 802.11 Wireless Local Area Networks

Contrary to what many people may think, wireless local area networks (WLANs) are not a new technology. They have been around for decades. With the ratification of the IEEE 802.11 standard happening in 1997, one might wonder then; why is it that WLANs are only now starting to become so popular [1]? The answers to this question are bandwidth and cost. The new WLAN technologies are capable of providing data rates as high as 54 Mbps at a cost that is more accessible to a larger number of people. With data rates of most Internet connections being far below 54 Mbps, it makes sense that the use of 802.11 networks is on the rise.

The original IEEE 802.11 standard defined both a common Medium Access Control (MAC) mechanism as well as multiple physical access methods (PHYs). Both of the PHY layers supported data rates at 1 and 2 Mbps and operated in the 2.4 GHz industrial, scientific, and medical (ISM) band. Just as the original standard was becoming final, the IEEE was already working on extensions to increase data rates in IEEE 802.11 WLANs. This new work consisted of two initiatives. The first resulted in the IEEE 802.11a [2] PHY for the 5 GHz band; this standard incorporates a coded multicarrier scheme known as orthogonal frequency division multiplexing (OFDM) and yields data rates of up to 54 Mbps. Besides higher data rates, the 802.11a standard also has the advantage of having more channels available, as well as suffering less interference from other devices since it operates in the 5 GHz band. The second initiative produced a standard commonly known as the IEEE 802.11b standard [3]. This standard offers a direct sequence spread spectrum (DSSS) backward compatible transmission definition that added two data rates, 5.5 Mbps and 11 Mbps, as well as two forms of coding [4].

Further research has been done by the IEEE 802.11 standards committee in the areas of security (11i) [5], higher data rates (11g) [6], and quality of service (11e) [7]. The IEEE 802.11g is the most recently approved standard and offers wireless transmission over distances of up to 300 meters at up to 54 Mbps, compared with the 11 Mbps of the IEEE 802.11b standard. Like 802.11b, 802.11g operates in the 2.4 GHz frequency band, and is thus backward compatible. The IEEE 802.11i adds the Advanced Encryption Standard (AES) to the 802.11 standard for wireless LANs. Security has been a primary concern of IT managers reluctant to deploy wireless networks, but AES is a stronger level of security than found in the current Wi-Fi Protected Access security standard [8].

The IEEE 802.11e draft standard is fully backward compatible with the existing 802.11a and 802.11b standards with the addition of quality of service (QoS) and multimedia support. QoS and multimedia support are critical to wireless home networks where voice, video, and data will be delivered. Broadband service providers view QoS and multimedia-capable home networks as an essential ingredient to offering residential customers video on demand, audio on demand, voice over IP and high-speed Internet access [8].

1.2 Motivation

Wireless computing is a rapidly evolving field that provides users with the ability to be connected to networks without the hassle of being tethered to a traditional wired network. Active development in WLANs is being driven by many factors including laptop market penetration, ubiquitous access to the Internet and intranets, being able to provide high bandwidth to users in a small geographical area, and multimedia capabilities such as VoIP and streaming video.

Ideally, users of a WLAN will have the same service levels and capabilities that they are accustomed to with traditional LANs. In order to address these assumptions, the wireless community must overcome some challenges that do not exist with traditional wired networks. Problems such as frequency allocation, interference and reliability, security, power consumption, human safety, mobility, and throughput are some of the most challenging problems facing the wireless LAN community [9]. Active research for enabling the support of real-time traffic in WLANs is also a hot topic. Most of the current research in this area involves improving the MAC layer of the 802.11 standard so that it is able to provide the QoS levels that are required for real-time traffic.

Moreover, the need to give some sort of priority to voice over data is even more of an issue in wireless LANs than in wired LANs for a number of reasons:

- WLANs operate at a lower rate, leading to higher transmission and queuing delays;
- The Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol that is used by IEEE 802.11 WLANS, has worse throughput-delay characteristics compared to the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol, which is used by Ethernet LANs;

- Neighboring cells of a wireless LAN may interfere with each other, thereby decreasing their individual performance, whereas distinct segments of a wired LAN are electromagnetically isolated [10];

In order to address the QoS issues in WLANs, the IEEE 802.11 standards committee formed Task Group E to define QoS enhancements in the 802.11 MAC layer. The 802.11e draft standard introduces a new coordination function, the Hybrid Coordination Function (HCF). The HCF is used only in QoS Basic Service Sets (QBSSs) and has two modes of operation: the Enhanced Distributed Channel Access (EDCA) is a contention-based channel access function that operates concurrently with the HCF Controlled Channel Access (HCCA) based on a polling mechanism, which is controlled by the Hybrid Coordinator (HC). The HC is co-located with the QoS Access Point (QAP) [11]. These new access functions provide QoS enhancements to the distributed coordination function (DCF) and point coordination function (PCF) access functions that were defined in the original 802.11 standard.

Currently, only the mandatory contention-based DCF function is commercially implemented. As a result of this, the widespread success of 802.11 WLANs can be attributed to the existence and development of the DCF access function. Distributed computing environments, which can only use distributed channel access functions, are also becoming more widely used.

In order to ensure that QoS is available in future WLANs, we assert that it is important to research the area of distributed channel access techniques that take into account QoS requirements. Therefore, a natural place to start our research was in the shortcomings of the IEEE 802.11e draft standard. This thesis focuses on MAC layer improvements to the EDCA function of the 802.11e draft standard.

1.3 Related Work

Recent research has been done in the area of providing QoS for traffic differentiation in IEEE 802.11 WLANs. Most of the work attempts to provide QoS by separating traffic into flows with different priorities. The various proposals provide performance differentiation, typically by tuning one or three WLAN parameters: Length of Inter-Frame Space (IFS) [12][13], Length of Contention Window (CW) [14][15], and Length of Backoff Interval (BI) [16].

All of the aforementioned techniques, except for the one described in [12], use static assignment of priorities. This can prove to be problematic if the amount of high priority traffic increases beyond a certain point, lower priority traffic will be choked out. The distributed deficit round-robin (DDRR) method in [12], which uses a more dynamic method of choosing the IFS value, has shortcomings as well. Two of the major problems with the DDRR method are that it does not take into account multiple priorities, and that it deviates from the IEEE 802.11 standard by ignoring the random backoff interval that wireless stations are supposed to use before accessing the channel.

A number of attempts have been made to come up with an analytical model for the IEEE 802.11e EDCA. The number of papers published in this area is small, and the number of useful papers published in this area is even smaller. Two of the more complete papers are [17] and [18], although we found that their results were questionable in their reasoning, as well as their accuracy. The most complete and useful work done for modeling the IEEE 802.11e EDCA is described in [19]. We have used this work as the basis for the analytical model for this thesis.

1.4 Thesis Contributions

The primary objective of our work is to design and implement a distributed scheduler for providing better bandwidth utilization, a "fair" distribution of bandwidth between stations according to the requirements of the application, and QoS guarantees in ad-hoc wireless LANS based on the IEEE 802.11e standard. The significance of our approach is that real-time and multimedia applications will be able to obtain the small delay and jitter values that they need to perform well, and that time-insensitive applications such as file and data transfer will be able to coexist in the network without suffering from starvation. As part of our design, we define the following properties for our scheme:

- The protocol will be fully distributed; due to the ad-hoc nature of our test network, no access point is available to perform centralized scheduling;
- The protocol should be similar to the IEEE 802.11e standard in order to make it an attractive alternative;
- The protocol should be backward compatible with the IEEE 802.11 DCF/ 802.11e EDCA;

This thesis improves on the EDCA that is defined in the IEEE 802.11e draft standard. We implemented an IEEE 802.11e EDCA simulator, which we used for testing multiple distributed MAC layer packet schedulers. Our simulator was implemented according to the IEEE P802.11e/D7.0, January 2004 draft standard [7]. Our proposed scheme improves delay and throughput for high priority traffic while maintaining that a "fair" amount of low priority traffic is still able to be transmitted. Our scheme, which we built on top of the EDCA simulator, is a distributed version of the well-known deficit round-robin queuing technique.

In order to validate our scheme, we designed and implemented an analytical model which simulates both the EDCA and our proposed scheme under saturation conditions. We use this model to verify that our simulator gives the correct results for system throughput under saturation conditions. Having an analytical model provides flexibility, in addition providing verification of the simulator, in that you can use the model to tune parameters for different simulation scenarios.

1.5 Thesis Outline

This thesis is organized as follows. Chapter 2 gives the details of the IEEE 802.11 standard as well as the enhancements that are covered by the 802.11e draft standard. Chapter 3 presents the details and algorithm used for our proposed EDCA Deficit Round Robin scheme. The network models we use for our simulations are also described in this Chapter. We describe the analytical tools used in our research in Chapter 4. We present our simulation scenarios, results and give a discussion of the results in Chapter 5. Finally, Chapter 6 presents a summary of our findings and some suggestions for future work.

Chapter 2

OVERVIEW OF THE IEEE 802.11 STANDARD

The previous chapter gave a short introduction to the IEEE 802.11 standard and some of its history. More detail about the standard is necessary, however, since improving on the standard is the main focus of this thesis. The 802.11 and 802.11e standards are extremely long and overwhelming at first glance, but can be summed up nicely after wading through them a few times. This chapter contains a summary of the IEEE 802.11 standards that we studied in this thesis. Section 2.1 gives an introduction to the standard and the way it is laid out. Both the PHY and MAC layers are discussed, although the focus of our research is on the MAC layer. We then give a discussion of the topological configurations that are allowed in 802.11 wireless LANs. Next, we introduce the IEEE 802.11e draft standard and detail the differences between this standard and the original 802.11 standard. This information, as well as a description of the MAC architecture of the 802.11e are described in Section 2.2.

2.1 Introduction to the IEEE 802.11 Standard

The original IEEE 802.11 standard [20] was introduced in 1997. Since then, use of 802.11 WLANs has been exploding, driving researchers to develop faster and cheaper hardware to support wireless networks. 802.11 is an evolving family of specifications developed by a working group of the IEEE. It consists of a family of several specifications, with new specifications being added as researchers come up with new ways of improving cost, speed, and security. All of the 802.11 standards use CSMA/CA as a means of controlling access to the wireless medium.

The 802.11 standard describes two specifications that make up the architecture of 802.11 WLANs. The PHY and MAC specifications, as they are named by the IEEE committee that defines them, are described in the subsequent sections.

2.1.1 802.11 Protocol Architectures

The architecture of the IEEE 802.11, shown in Figure 2.1, is encompassed by the lower two layers of the Open System Interconnection's (OSIs) seven layer model. The complete separation of the PHY from the MAC layer has allowed both layers to advance at their own pace, while only requiring the interface between the two to be maintained.

The logical link control (LLC) sublayer provides an interface to higher layers and performs flow and error control. The MAC sublayer supplies the functionality required to provide a reliable delivery mechanism for user data over a noisy, unreliable wireless medium. It does so through a number of coordination functions that allow wireless stations (STAs) to gain access to the channel in both contention and contention-free manners.

The Physical Layer Convergence Procedure (PLCP) is essentially a handshaking layer that enables MAC protocol data units (MPDUs) to be transferred between MAC stations over the Physical Medium Dependent (PMD), which is the method of transmitting and receiving data through the wireless medium. In a sense, you can think of the PMD as a wireless transmission service function that is interfaced via the PLCP [1].

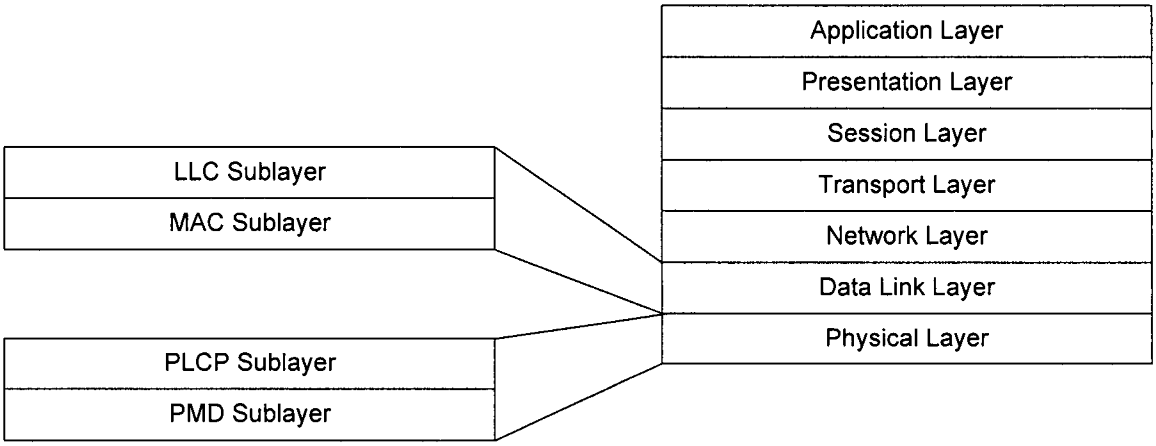

Figure 2.1. OSI Model showing the MAC and PHY layers that are covered by the IEEE 802.11 standard

2.1.2 802.11 Network Topologies

The most basic 802.11 topology is a Basic Service Set (BSS). A BSS consists of two or more wireless STAs, which are involved in communications. In the most basic form, stations communicate directly with each other on a peer-to-peer level sharing a given cell coverage area. This type of network is often formed on a temporary basis, and is commonly referred to as an ad-hoc network, or Independent Basic Service Set (IBSS). An example of an IBSS is shown in Figure 2.2.

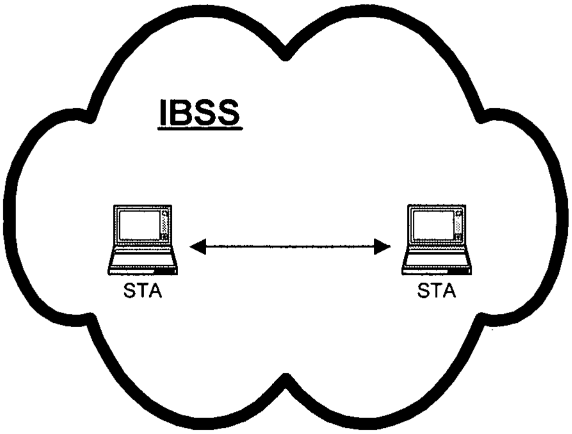

Figure 2.2. An ad-hoc, or Independent Basic Service Set 802.11 network

A BSS may also contain another entity called an Access Point (AP) which can be used to form a bridge to other networks. When an AP is present in an 802.11 network, STAs do not communicate on a peer-to-peer basis, rather they communicate through the AP. AP's are not mobile, and form part of the wired network infrastructure. A BSS in this configuration is said to be operating in infrastructure mode [21].

An Extended Service Set (ESS), shown in Figure 2.3 consists of two or more BSS's connected by means of a Distribution System (DS). Although the DS could be any type of network, it is almost invariably an Ethernet LAN. Mobile nodes can roam between APs and seamless campus-wide coverage is possible [21]. ESSs satisfy the needs for large coverage areas of arbitrary size and complexity.

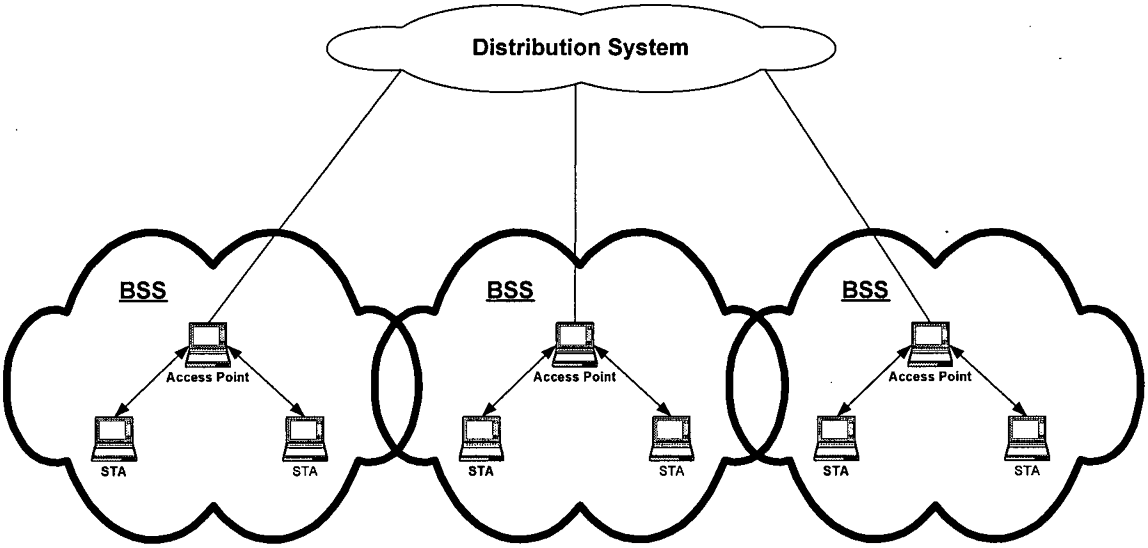

Figure 2.3. An Extended Service Set 802.11 network

2.1.3 IEEE 802.11 PHY Layer

The PHY is the interface between the MAC layer and the wireless media, which transmits and receives data frames over a shared wireless channel. The PHY layer provides a frame exchange between the MAC and PHY under the control of the PLCP sublayer. The PHY uses single carrier and spread spectrum modulation to transmit data frames over the media under the control of the PMD sublayer, as well as providing a carrier sense indication back to the MAC to verify activity on the medium [22].

The original 802.11 PHY defined two RF transmission methods, as well as one Infrared transmission method. The defined RF transmission methods are Frequency Hopping (FH) and Direct Sequence Spread Spectrum (DSSS). Both methods operate in the 2.4 GHz (ISM) frequency band, occupying 83 MHz of bandwidth from 2.400 GHz to 2.483 GHz. DSSS uses differential BPSK (DBPSK) and DQPSK for modulation and FH uses 2 – 4 level Gaussian FSK as the modulation signaling method. FH WLANs support 1 Mbps and 2 Mbps data rates and DSSS WLANs support data rates of 1, 2, 5.5, and 11 Mbps. Further improvements to the PHY have been amended to the 802.11 standard through additions such as the 802.11a, 802.11b, and 802.11g standards. These amendments use different modulation schemes and provide increased data rates over the original PHYs.

The IEEE 802.11a and 802.11g standards introduce a new multiple access technique called Orthogonal Frequency Division Multiplexing (OFDM). 802.11a WLANs operate in the 5 GHz band and support data rates of up to 54 Mbps. One major disadvantage of 802.11a, however, is that due to its operation in the 5 GHz band it is not compatible with other 802.11 WLAN systems. Fortunately, the 802.11g standard alleviates this problem. The 802.11g standard operates in the 2.4GHz (ISM) band and using OFDM, is able to provide data rates of up to 54 Mbps.

The PHY layer, although it is extremely important, is not discussed any further as it does not relate to the research for this thesis. Interested readers are encouraged to read [23][2][3][6] for further information on the PHY layer.

2.1.4 IEEE 802.11 MAC Layer

The 802.11 MAC layer is responsible for channel allocation, protocol data unit (PDU) addressing, frame formatting, and fragmentation/reassembly. The 802.11 MAC is similar to the 802.3 (Wired Ethernet) MAC in that both use a "listen before talk" mechanism to control access to the shared medium. However, additional challenges resulting from the use of a wireless channel require the MAC layer of 802.11 to differ slightly from its wired counterpart. The wireless medium is subject to interference and is inherently less reliable than a wired medium. The medium is also more vulnerable to unwanted intrusion. Wireless networks suffer from the "hidden station" problem: a station transmitting to another station may be interfered with by a third "hidden" station that is within range of the receiver but out of range of the transmitter and therefore does not defer channel access [24]. Finally, mobile stations cannot monitor for the occurrence of collisions while they are transmitting, as can be done in wired Ethernet. The 802.11 MAC must address all of these issues while still providing a robust means of accessing the wireless medium.

The 802.11 MAC allows the transmission medium to operate in two different modes depending on the type of access that is required. In the Distributed Coordination Function (DCF) mode, otherwise known as the contention period (CP), all mobile stations compete for access to the channel by applying a CSMA/CA protocol. If the channel is busy, the station will go into a back-off period and wait for the channel to become available. DCF provides best-effort, asynchronous data transfer and is mandatory in all 802.11 wireless stations. Medium access control in ad-hoc networks is by means of DCF only. A second, optional medium access method is the Point Coordination Function (PCF), or the contention-free period (CFP). The PCF mode is a centralized, polling-based access method and is controlled by an access point, thereby relieving the mobile stations of the task of contending for access to the channel. The medium access method in a network can use the DCF alone or a combination of PCF and DCF access. The best configuration can be decided depending on the profile of the traffic that will be flowing across the network. Figure 2.4 shows the IEEE 802.11 MAC Architecture.

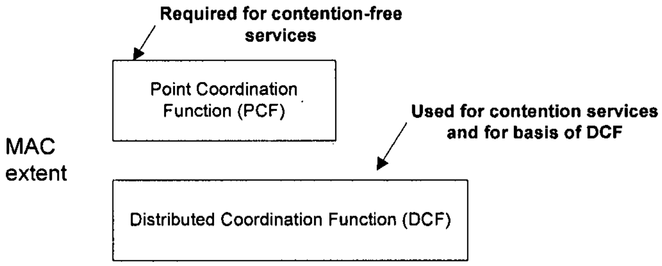
Required for contention-free services
Point Coordination Function (PCF)
Used for contention services and for basis of DCF
MAC extent
Distributed Coordination Function (DCF)

Figure 2.4. IEEE 802.11 MAC Architecture [23]

IEEE 802.11 supports three types of frames: control, data, and management. Control frames are used for handshaking during the CP, for positive acknowledgements during the CP, power-save polling, and to end the CFP. Data frames are used for the transmission of data during the CP and CFP, and can be combined with polling and acknowledgements during the CFP. Management frames are used for station association and disassociation with access points, timing and synchronization, and authentication and deauthentication [9].

Priority to obtain control of the transmission medium is done using interframe spaces (IFS). Three IFS intervals are defined by IEEE 802.11:

- The shortest, and therefore highest priority interval, is called the short IFS (SIFS);
- The point coordination function IFS (PIFS) is the next longest interval and has the second highest priority;
- The longest and lowest priority interval is called the distributed coordination function IFS (DIFS);

DCF is the fundamental access mode of the 802.11 MAC protocol and all stations are required to implement it. As can be seen in Figure 2.4, the DCF builds the foundation for the PCF. A discussion of the DCF mechanism follows. For completeness, we discuss all functionality of the DCF, even though we do not use all of its features in our simulations.

Before attempting to transmit a packet, a station may perform virtual carrier sensing by sending an MPDU duration information in the header of a request to send (RTS), clear to send (CTS), and data frames. An MPDU is a MAC data unit that is passed from the MAC layer to the physical layer. An MPDU consists of header information, payload, and 32 bits of CRC. The duration field of the MPDU tells other stations in the network the amount of time (in microseconds) that the transmitting station needs to complete the transmission of its frame. This information can be used by stations to update their network allocation vector (NAV), telling them when they should next attempt to access the channel.

When the DCF is being used for channel access, prior to transmitting, a mobile station must first sense the channel to determine if another station is currently transmitting. If no other station is found to be transmitting, the station waits a DIFS time period and begins transmission if the channel is still free after the DIFS period. If a second station is transmitting, or if one starts to transmit during the DIFS period, the first station will wait for the channel to become free for a DIFS period. After the channel has been idle for a DIFS period, a random backoff timer is started. If the channel becomes busy while the timer is running, the timer will pause. The timer will then restart after the channel has been free for a DIFS period. When the timer expires, the station will begin transmission of its data. The timer is selected using a slotted binary exponential backoff (BEB) technique. The time following the idle DIFS is slotted and stations can only transmit at the beginning of a slot. The backoff time is uniformly chosen in the interval [0, CW-1] where CW is the Contention

Window. At the first transmission attempt, $CW=CW_{min}$ and it is doubled at each retransmission up to $CW_{max.}$ The values of slot time, CW_{min} and CW_{max} are 20μs, 32 and 1024 respectively [25]. The prior values are used when Direct Sequence Spread Spectrum is used for the PHY layer and differ with different physical layer implementations. All of our simulations use the DSSS PHY.

Upon successful reception of a frame, the receiver waits an SIFS interval and then transmits an acknowledgment (ACK) for the received frame. When the ACK is received, the station must wait a random backoff time before attempting to access the channel again. If the ACK is not received within a timeout period, the station must contend for channel access again and try a retransmission of the frame. Seven retransmission attempts are allowed, at which time the station must drop the frame. Figure 2.5 gives a timing diagram of a simple frame transmission in DCF mode.

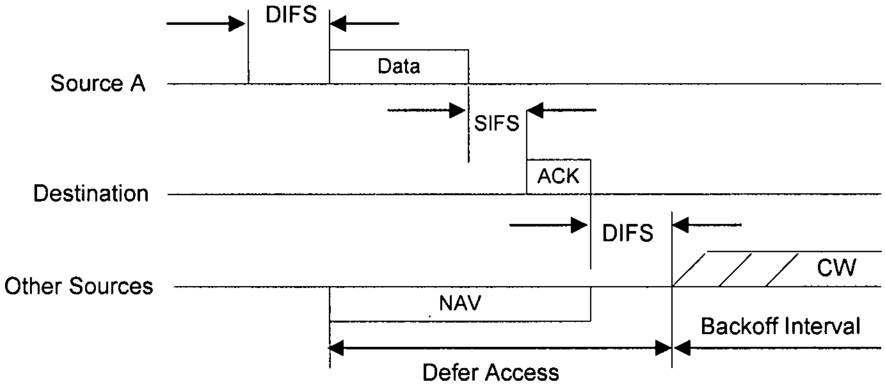

Figure 2.5. IEEE 802.11 MPDU transmission in DCF mode [9]

Optionally, stations can choose to reduce the possibility of collision due to the "hidden" station problem mentioned above. This can be done by implementing a handshaking mechanism called the RTS/CTS access control mode. Figure 2.6 illustrates this scheme. The top portion of the figure depicts the "hidden" station problem. The bottom half is a timeline showing an RTS/CTS scenario timeline. The rectangles represent the NAV windows for Stations C and D. When station A wants to send a frame to station B, it first sends an RTS frame. Upon receiving the RTS frame, B sends a CTS frame back to A (after an SIFS period) to indicate that it is okay for A to transmit. The duration field is updated in each of the RTS/CTS frames, therefore allowing nodes in range of both the transmitter and receiver to update their NAV fields. This helps to combat the "hidden" station problem. From this point, the data is transmitted normally as described above. The duration field of the RTS/CTS MPDU ensures that other stations in the same BSS will not interfere with station A's transmission. In addition, the RTS/CTS scheme is also useful due to the fact that the actual RTS/CTS frames are small in size (20 octets for RTS and 14 octets for CTS), when compared to the maximum MPDU frame size (2346 octets) and hence, only minimally affect the throughput if there is a collision when sending them. However, for a lightly loaded channel, additional delay is added with the use of RTS/CTS frames. A configurable parameter called RTS_Threshold is used to decide which frames will be transmitted using the RTS/CTS access control mode, and which frames will be transmitted without RTS/CTS.

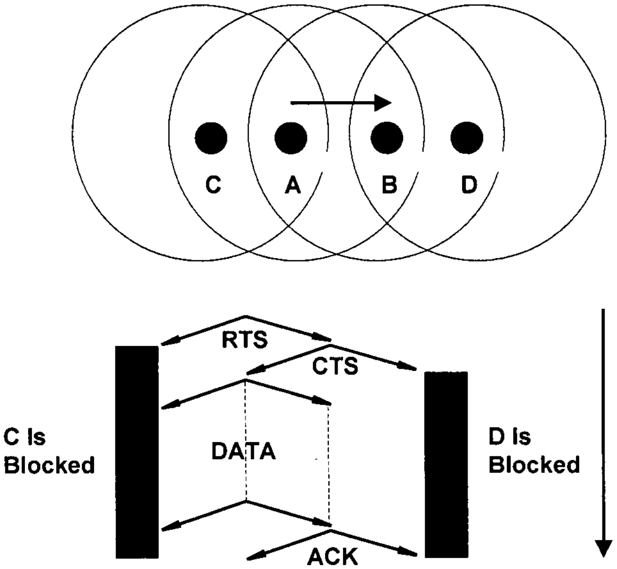

Figure 2.6. RTS/CTS access control mode scheme

The PCF mode of the IEEE 802.11 MAC layer provides contention-free medium access. The contention-free service is achieved by the AP gaining control of the medium and acting as a point coordinator. At the beginning of the CFP, the AP waits a PIFS period and then takes control of the medium. During the CFP, the AP will poll all mobile stations that are CF-aware and wait for a response. The polling sequence is implementation specific but most simple implementations use a round-robin scheme. When a mobile station receives a CF-Poll from the AP, it will begin transmission of its data after a SIFS period. In order to preserve the ability of stations to transmit asynchronous traffic as well as synchronous traffic, the CFP and CP must be able to coexist. The combination of a CFP and a CP is called a superframe. In certain cases a mobile station may begin transmission of a frame just prior to the end of a superframe, causing the superframe to be "stretched." Not surprisingly, this is referred to as "superframe stretching." The effect of stretching a superframe is to make the current superframe longer, while making the CFP of the next superframe shorter. Superframe "stretching" is detailed by a timing diagram in Figure 2.7.

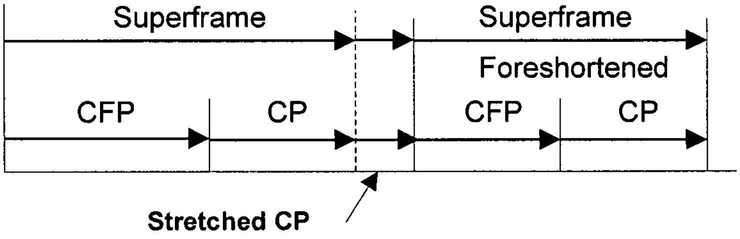

Figure 2.7. IEEE 802.11 Superframe Stretching [26]

To understand the effect of stretching on the CFP, one should allow for an MPDU of maximum size (which is 2304 bytes) to be sent right before the end of a superframe. If the fragmentation threshold is not equal to this maximum size, then additional overhead will be incurred due to the fragmentation of the MPDU. All fragments of an MPDU are sent a SIFS interval apart, which means that the AP, in waiting for a PIFS interval to initiate the CFP, cannot acquire the medium in between fragment transmissions. Thus, in the worst case, the stretching period could be as large as is needed to send a 2304 byte payload with fragmentation [26].

Within a superframe, CFP repetition interval(s) exist. The CFP repetition interval is used to determine the frequency with which the PCF controls the medium. A repetition interval is made up of some time for contention-free traffic, with the remainder of the time being used for contention based traffic. In order to start a CFP repetition interval, an AP transmits a beacon frame for synchronization and timing purposes. The duration of the CFP interval is configurable and is always an integral number of beacon frames. The maximum size of the CFP is determined by the manageable parameter CFP_Max_Duration. The minimum value of CFP_Max_Duration is the time required to transmit two maximum-size MPDUs, including overhead, the initial beacon frame, and a CF-End frame. The maximum value of CFP_Max_Duration is the CFP repetition interval minus the time required to successfully transmit a maximum-size MPDU during the CP (which includes the time for RTS/CTS handshaking and the ACK). Therefore, time must be allotted for at least one MPDU to be transmitted during the CP [9]. The CFP interval may vary from repetition interval to repetition interval depending on the amount of CF-Enabled traffic and whether or not stretching occurs.

The timing diagram for MPDU transmission when both DCF and PCF modes are used is illustrated in Figure 2.8. The detailed portion of the timing information shown is for PCF mode. When a new CFP repetition interval begins, all mobile stations update their NAV window to match the CFP_Max_Duration parameter. This ensures that they will not be able to transmit during the CFP without being polled, unless they are transmitting an ACK to a received packet. To begin a CFP, the AP must first wait a PIFS interval before sending the first beacon frame. If the AP is indeed able to get access to the medium, it will wait a SIFS interval and send its first CF-Poll frame. The AP may also piggyback data in the frame if it has data queued for the station that is being polled. Upon receiving a CF-Poll, a station may respond that it has no data to send (CF-ACK only), or it may acknowledge the poll and send back some data (CF-ACK and data) after a SIFS interval. If the AP receives data back from the mobile station, it can acknowledge the received data, poll another station, and send data to that station all in the same frame (CF-ACK and CF-Poll and data). This piggybacking of data, ACKs, and polling improves the efficiency of the polling scheme. An AP only waits a PIFS interval for an ACK to the data frame to arrive. After this time, the data frame is assumed lost and the AP polls the next station in its list. If at any time during the CFP there is no traffic queued at the AP or if all CF-aware stations have no data to send, the AP can end the CFP with a CF-End frame. At this point, the medium will be open for stations to transmit in DCF mode.

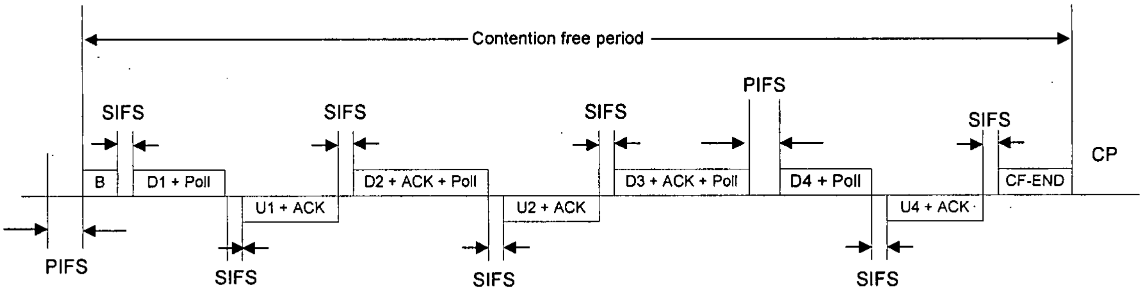

Figure 2.8. IEEE 802.11 MPDU transmission with alternation between PCF and DCF modes [9]

2.2 Introduction to the IEEE 802.11e Draft Standard

With the popularity of WLANs on the rise, the expectation that WLANs will support the same applications as their wired counterparts is also on the rise. The inability of the original 802.11 MAC layer to support end-to-end QoS for latency-sensitive applications is one of the major bottlenecks that prevents this. With that said, development of a QoS enhancement to the 802.11 standard is currently in the works. Task Group E of the 802.11 standards committee is working on a new standard for provisioning QoS in the MAC layer of 802.11 WLANs.

The proposed IEEE 802.11e standard provides a means of prioritizing the radio channel access within a BSS by defining up to 8 User Priorities (UPs) that allow separation of traffic into different priority flows. The values a UP may take are the integer values from 0 to 7. The UPs, which are shown in Table 2.1, map directly to the eight priority tags that are defined in the 802.1D (MAC Bridges) standard. An MPDU with a particular UP is said to belong to a Traffic Category (TC) with that UP. The UP is provided with each MPDU at the medium access control service access point, either directly, in the UP parameter, or indirectly, in a traffic specification (TSPC) designated by the UP parameter [7]. When a QoS station (QSTA) uses contention-based access to transmit a frame, the access is based on the Access Category (AC) of the MPDU that is to be transmitted. The AC is derived from the UP's as show in Table 2.1.

Priority Level	User Priority	Designation	Category (AC)	(Informative)
Lowest ↓	1	BK	AC_BK	Background
	2	-	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

Table 2.1. User Priority to Access Category mappings [7]

The 802.11e draft standard introduces a new coordination function called HCF, which is designed to add QoS capabilities and improve the channel utilization of the 802.11 MAC. The HCF operates in two different modes: a contention period that is controlled by the EDCA function and a contention-free period that is controlled by the HCCA function. The EDCA allows prioritization of data traffic in a contention-based channel access environment which is similar to the DCF of the original 802.11 standard. On the other hand, the HCCA allows the HC to start polling based contention-free Controlled Access Periods (CAPs) at any time during the contention period, after the medium remains idle for at least a PIFS interval, as needed to conform to the QoS parameterization [27]. As in the PCF, a superframe is divided into a CFP and a CP. The EDCA is used for medium access during the CP, except during CAPs. With the addition of the HCF, the PCF is no longer useful, although it remains in the standard for backward compatibility. The MAC frames used in HCF are identical to those used by PCF except that QoS is added to the front of the names (i.e. QoS NULL, QoS DATA, QoS CF-ACK, QoS CF-Poll, and QoS CF-End). The frames may be "piggybacked" on one another just as in the case of the PCF. A CAP is a sequence of transmit opportunities (TXOPs) initiated by the HC and begins with either a QoS CF-Poll or QoS CF-DATA frame. Our research focuses on improving the EDCA function to provide shorter MAC delay, less jitter, and better overall system throughput in 802.11 ad-hoc wireless networks. The HCCA is also a hot area for research and we therefore leave it to the reader to further investigate this area if it is of interest.

2.2.1 IEEE 802.11e MAC Architecture

The MAC architecture of 802.11e can be described, as shown in Figure 2.9, as providing the PCF and HCF through the services of the DCF [7]. All QSTAs must implement both the DCF and HCF functions. The PCF remains optional in all STAs. The EDCA function is described in detail in the following section.

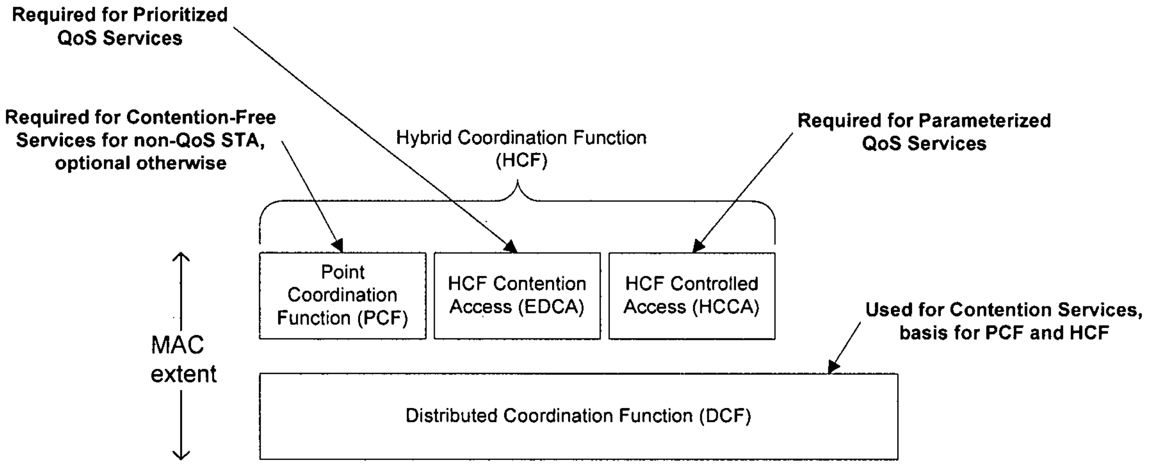
Required for Prioritized QoS Services
Required for Contention-Free Services for non-QoS STA, optional otherwise
Hybrid Coordination Function (HCF)
Required for Parameterized QoS Services
Point Coordination Function (PCF)
HCF Contention Access (EDCA)
HCF Controlled Access (HCCA)
Used for Contention Services, basis for PCF and HCF
MAC extent
Distributed Coordination Function (DCF)

Figure 2.9. MAC Architecture of the IEEE 802.11e [7]

2.2.2 The IEEE 802.11e EDCA

The DCF of the 802.11 standard lacks any prioritization mechanisms and is therefore unsuited for real-time traffic. In order to address this problem, an enhanced version of the DCF is proposed in the IEEE 802.11e draft standard. The EDCA introduces prioritization enhancement based on ACs, and functions similar to the DCF with some elements of the MAC parameterized per-AC. The 802.11e enhancement supports up to eight priority TCs (4 ACs).

The backoff algorithm of the EDCA works the same as in DCF but each TC has an Arbitration Interframe Space (AIFS), which is at least as large as the DIFS and is chosen individually for each TC. The AIFS [TC] value is used instead of DIFS when deferring for the channel to be idle. A QSTA using the EDCA shall obtain a TXOP for an AC if the QSTA's carrier sense mechanism determines that the medium is idle at the AIFS[AC] slot boundary, after a correctly-received frame, and the backoff time for that AC has expired [7]. A TXOP is a bounded-duration time interval in which a station may transmit a sequence of frames which are spaced at SIFS time intervals. The recommended values for arbitration inter-frame spaces in IEEE 802.11e WLANs are shown in Table 2.2 and some IFS relationships are illustrated in Figure 2.10. The minimum initial value of the CW, denoted by CW_{min}, is selected on a per-TC basis. As collisions occur, the subsequent CW is doubled (binary exponential back-off), thus providing a probabilistic priority mechanism between the TCs. The CW_{max} value sets the maximum possible value for the CW and is also selected on a per-TC basis. Each station maintains a separate queue for transmission of frames within each TC. The recommended values for maximum and minimum contention window parameters in an IEEE 802.11e wireless station are shown in Table 2.3.

TC	AIFS[TC]
0	SIFS + 7 * slot_time
1	SIFS + 7 * slot_time
2	SIFS + 3 * slot_time
3	SIFS + 3 * slot_time
4	SIFS + 2 * slot_time
5	SIFS + 2 * slot_time
6	SIFS + 2 * slot_time
7	SIFS + 2 * slot_time

Table 2.2: Recommended values for arbitration inter-frame spaces in an IEEE 802.11e wireless station [7]

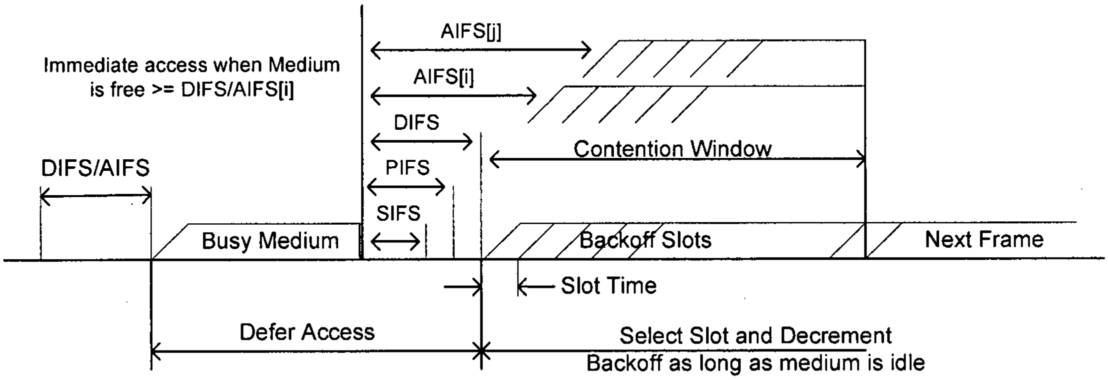
AIFS[j]
Immediate access when Medium is free >= DIFS/AIFS[i]
AIFS[i]
DIFS
Contention Window
DIFS/AIFS
PIFS
SIFS
Busy Medium
Backoff Slots
Next Frame
Slot Time
Defer Access
Select Slot and Decrement Backoff as long as medium is idle

Figure 2.10. IFS relationships for the IEEE 802.11e [7]

TC	Cwmin[TC]	Cwmax[TC]
0	31	1023
1	31	1023
2	31	1023
3	31	1023
4	15	31
5	15	31
6	7	15
7	7	15

Table 2.3: Recommended values for maximum and minimum contention window parameters in an IEEE 802.11e wireless station [7]

In the event that the backoff timer expires at the same time for two or more TCs, a TXOP will be granted to the TC with the highest priority. The other colliding TC(s) will enter a backoff mode and increase its contention window just as it would have if there was an actual collision on the medium. 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 limit (See Table 2.4). A TXOP duration of zero implies that only one frame can be transmitted at a time.

	TXOP Limit		
AC	DS-CCK	Extended Rate/OFDM	Other PHYs
0	0	0	0
1	0	0	0
2	6.016 ms	3.008 ms	0
3	3.264 ms	1.504 ms	0

Table 2.4. Default TXOP limits for all PHYs of the 802.11e [7]

Chapter 3

THE EDCA DEFICIT ROUND-ROBIN SCHEME

The previous chapter introduced the IEEE 802.11 standard as well as the proposed IEEE 802.11e draft standard. Although the EDCA method of the IEEE 802.11e attempts to provide service differentiation for different traffic priorities, it does so without any consideration of the effects that result when the number of high priority traffic sources becomes too high and begins to choke out lower priority traffic. We now propose a new scheme called EDRR that maintains the desirable attributes of the EDCA, while taking into consideration the fact that low priority traffic must still be served with some level of priority. The first section in this chapter describes the deficit round-robin technique that we use to build the base for our EDRR scheme. Section 3.1 introduces deficit round-robin scheduling, which is the basis for our proposed scheme. The objectives and reasoning behind the proposed scheme are introduced in Section 3.2 and the EDRR algorithm is presented in Section 3.3. In Section 3.4, the OPNET IEEE 802.11 standard model and the EDRR extensions are described.

3.1 Deficit Round-Robin

Deficit round-robin (DRR) is a queuing technique that attempts to provide a "fair" share of bandwidth to competing queues, independent of the packet sizes that originate from the queues. DRR is a centralized queuing technique that can be implemented with particularly low complexity. Two parameters serve as the means of providing this fair queuing: a deficit counter and a quantum of service. The deficit counter specifies the total number of bytes that the queue is permitted to transmit each time it is visited by the scheduler, and the quantum of service is a number of bytes that is added to the deficit count after each scheduling round. To service the queues, round-robin servicing is used with a quantum of service assigned to each queue. The only difference from traditional round-robin is that if a queue was not able to send a packet in the previous round because its head of line packet size was too large, the remainder from the previous quantum (the deficit counter) is added to the quantum for the next round. Thus, deficits are kept track of; queues that were shortchanged in a round are compensated in the next round [28].

As stated in [29], a major drawback of DRR is that its latency bound is highly dependent on the reserved rate of the flow and the reserved rates of the other flows. In other words, the latency bound of DRR varies as the difference of the reserved rate between two flows varies. Thus, flows with lower reserved rates may have a high latency bound when compared to flows with high reserved rates. Despite this, DRR is a highly valuable scheme due to its low complexity and its ability to provide fair distribution of bandwidth when servicing queues that contain variable-length packets.

Figure 3.1 depicts a typical round in a DRR queuing environment. In this example, we assume that the deficit counters for all queues are zero before this scheduling round. Queue 1 is allocated 50% of the bandwidth, while queues 2 and 3 are allocated 25% of the bandwidth each. At the beginning of the round, the quantum of service is added to the deficit counters, giving queue 1 a deficit count of 1000 bytes, queue 2 a deficit count of 500 bytes, and queue 3 a deficit count of 500 bytes. Because the 500 byte packet at the head of queue 1 is less than the value of the deficit count for queue 1, the first packet in the queue is transmitted. The deficit counter for queue 1 is then decremented to 500 bytes. Next, since the deficit count for queue 1 is still equal to the size of the next packet in the queue, this packet is also transmitted. The deficit count for queue 1 is now zero and the scheduler moves on to the next queue in the round. Queue 2 has a deficit count of 500, which is less than the size of the packet at the head of the queue. As a result of this queue 2 is not able to transmit in this round and the scheduler passes to queue 3. Queue 3 finds that the 1000 byte packet at the head of line is larger than the 500 bytes of deficit that it has to transmit. Therefore, as was the case with queue 2, the scheduler does not allow queue 3 to transmit in this round. In the next round, all of the deficit counters are incremented by the corresponding quantum of service, giving queue 1 1000 bytes of deficit, queue 2 1000 bytes of deficit, and queue 3 1000 bytes of deficit. Now, queue 1 is able to transmit two packets, exhausting its deficit count. Queue 2 finds that its 1000 bytes of deficit is greater than the 800 byte packet at the head of line and transmits the packet. Queue 2 now has 200 bytes of deficit, which is not enough to transmit its next 800 byte packet. The scheduler passes to queue 3, which transmits its 1000 byte packet and decrements its deficit counter to zero. This is the end of round two and the scheduler returns to queue 1. This process will continue as long as there are packets to be transmitted.

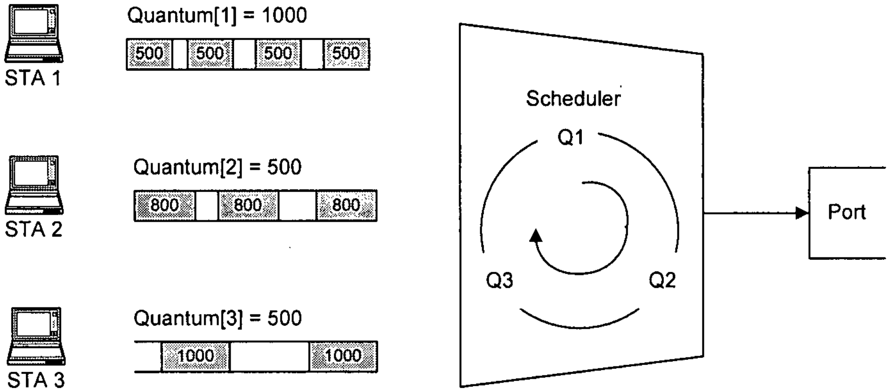

Figure 3.1. Deficit round-robin queuing [30]

3.2 The EDRR Scheme

We propose a contention-based distributed scheduling scheme for ad-hoc multiple access wireless networks. The proposed scheme attempts to improve channel utilization by approximating a DRR system, while at the same time providing QoS. The QoS can be provided by supporting fair distribution of bandwidth to stations in an ad-hoc network in a distributed manner. The proposed scheme, EDCA Deficit Round Robin (EDRR), can be built with minor modifications to the way in which two of the wireless LAN parameters are determined, namely the interframe spaces and the backoff interval. The EDRR method fits within the proposed IEEE 802.11e standard, therefore making it a viable option for implementation.

The proposed EDRR scheme is based on the IEEE 802.11e MAC and DRR in the following ways:

- The EDRR scheme borrows on DRR's idea of visiting stations in a round-robin manner but only allowing the station to transmit if it has enough "credits" to do so. This is accomplished by using the *deficit counter* and the *quantum* of service that are defined in DRR.
- A distributed approach for determining the DRR parameters is employed. Once these parameters are determined, we use the IFS and backoff interval mechanisms from the IEEE 802.11e MAC. The idea behind this approach is to choose IFS and backoff intervals that are inversely proportional to the deficit counter parameter.
- The 802.11e MAC approach of using multiple IFS and multiple contention windows is used as a means of supporting QoS when there is more than one type of traffic in the network (i.e. VoIP and data that are both transmitted in the same wireless network).

3.3 The EDRR Algorithm

We now discuss the proposed EDRR scheme. As defined in the IEEE 802.11e draft standard, we have defined the protocol to work with up to 8 traffic categories (flows) per station. Each queue in a remote station needs to set and maintain the following parameters:

- A *deficit counter* that represents the amount of "credits" that a given queue has. The IFS and backoff intervals for a queue are inversely proportional to the deficit counter, resulting in shorter IFS and backoff intervals for a queue that has been waiting a long period of time to transmit.
- A *quantum* of service that is expressed in terms of bytes. In DRR, the *deficit counter* for a queue is incremented by the quantum each time that queue determines that a round of scheduling is finished. Since there is no central coordinator in an ad-hoc network, there is no way of determining that a scheduling round is finished. For this reason, the EDRR protocol makes an attempt to follow a round-robin ordering but does not guarantee that this will be the case in every "round." The *quantum* of service (*Q*) is added to the deficit counter proportional to the user throughput requirements every T_i seconds and a station attempts to transmit when the deficit counter becomes greater than the size of the head of line packet.

The deficit counter is maintained using the following equations:

- Calculate Deficit Counter for Queue *i* in Station *j* at time t:

$$DC_i^j(t) = DC_i^j(t') + \frac{Q}{T_i} * (t - t') \qquad (1)$$

- Update Deficit Counter for Queue *i* in Station *j* after a successful packet transmission:

$$DC_i^j(t) = DC_i^j(t') - Frame\_Size(t) \qquad (2)$$

Where *t* is the current system time and *t'* is the last time the deficit count was updated. Equations (1) and (2) represent the functionality needed to update the deficit counter at the beginning of a new round, and after transmitting a packet, respectively.

Once the station has accumulated enough "credits" to transmit, the next step is to choose the IFS and backoff intervals such that the queue with the highest deficit counter will ideally be assigned the smallest IFS and backoff intervals. For simplicity purposes, we use liner functions to calculate these values. The IFS and backoff intervals are calculated as follows:

- Calculate IFS Interval for Queue *i*:

$$IFS_i^j(t) = AIFS[i] - \alpha_i * \frac{DC_i^j(t)}{Q} \qquad (3)$$

The values for AIFS[*i*] are the recommended values from the IEEE 802.11e draft standard and α_i are scaling factors that are used to translate the ratio between Q and $DC_i^j(t)$ into an appropriate value that maintains the priority levels for different traffic categories. The recommended AIFS values were given in Table 2.2 and are repeated again in Table 3.1 for convenience. The value of $DC_i^j(t)$ is bounded between zero and the value shown in (4), depending on the AC, in order to maintain the IFS priority levels that are defined in the IEEE 802.11e draft standard.

- Make Sure that $DC_i^j(t)$ does not exceed its maximum value:

$$\begin{cases} \dfrac{AIFS[AC_i] - AIFS[AC_{i+1}]}{\alpha_i} * Q & 0 \le i \le 3 \\ \dfrac{AIFS[AC_i] - PIFS}{\alpha_i} * Q & 4 \le i \le 7 \end{cases} \qquad (4)$$

TC	AIFS[TC]
0	SIFS + 7 * slot_time
1	SIFS + 7 * slot_time
2	SIFS + 3 * slot_time
3	SIFS + 3 * slot_time
4	SIFS + 2 * slot_time
5	SIFS + 2 * slot_time
6	SIFS + 2 * slot_time
7	SIFS + 2 * slot_time

Table 3.1: Recommended values for arbitration inter-frame spaces in an IEEE 802.11e wireless station [7]

- Calculate Backoff Interval for Queue *i*:

$$B_i^j = \left\lfloor \beta_i * \frac{1}{DC_i^j(t)} \right\rfloor \qquad (5)$$

The values of β_i are scaling factors that are used to choose backoff intervals that maintain the priority levels for different traffic categories as defined by the recommended values for the minimum and maximum contention window parameters in the IEEE 802.11e draft standard. These values were given in Table 2.3 and are repeated again in Table 3.2 for convenience.

TC	Cwmin[TC]	Cwmax[TC]
0	31	1023
1	31	1023
2	31	1023
3	31	1023
4	15	31
5	15	31
6	7	15
7	7	15

Table 3.2: Recommended values for maximum and minimum contention window parameters in an IEEE 802.11e wireless station [7]

To reduce the probability of collisions, B_i is chosen as a uniformly distributed random number between 1 and the value of B_i calculated in (5).

- Randomize B_i:

$$B_i^j = \left\lfloor uniform(1, B_i^j) \right\rfloor \qquad (6)$$

When collisions occur, the contention window is doubled as in the IEEE 802.11 standard. This reduces the probability of another collision, while at the same time maintaining a relatively small contention window, since the retransmitted packet should be transmitted with high priority. As in the IEEE 802.11, the contention window is not allowed to increase above the maximum value that is specified in the IEEE 802.11e draft standard Table 3.2. A flowchart depicting the EDRR algorithm is given in Figure 3.2.

The EDRR process begins with a packet arrival from the higher layer, or the receipt of an ACK packet where packets still remain in one of the higher layer queues. If an ACK is received, the deficit count is decremented to reflect that a packet has been transmitted, and the deficit counter is updated. Otherwise, the deficit count is updated to include the deficit since the last time the queue was able to transmit. Now, the deficit counter is compared to the size of the head of line packet. If the packet is larger than the deficit counter, the STA must keep updating the deficit counter until the deficit counter is greater than or equal to the packet size. When the deficit counter meets or exceeds the packet size, it is compared to the maximum allowable deficit count. Should it be larger than this value, the deficit counter is set to the maximum deficit counter value. Next, the IFS is calculated using the deficit counter. The STA must now check whether the channel has been idle for the calculated IFS. If it has not, the deficit counter is updated again and the algorithm proceeds to calculate a new IFS. When the STA finds that the channel has been idle for at least IFS, a backoff interval is calculated using the deficit counter. This backoff value is randomized to reduce the possibility of a collision between STAS having the same deficit counter value, and the STA proceeds to the backoff process.

3.4 OPNET IEEE 802.11 Model

Now that our method has been introduced, we will describe the OPNET simulation model that we use to evaluate its performance. For readers that are unfamiliar with OPNET, a brief introduction to the OPNET modeler program as well as how it can be used to create networks and emulate new protocols is given in Appendix A.

The standard model that we use for our research is the IEEE 802.11b model. The model includes basic parameters for making changes in the physical layer of the IEEE 802.11 as well as extensive options for making changes to the MAC layer. We use the base MAC layer as a starting point for our research. We next implement the enhancements for the IEEE 802.11e, as well as the DDRR [12] enhancements. This new model becomes our baseline MAC that we use for comparing with the proposed EDRR scheme. The next couple of sections describe the OPNET models of the IEEE 802.11 PHY and MAC layers.

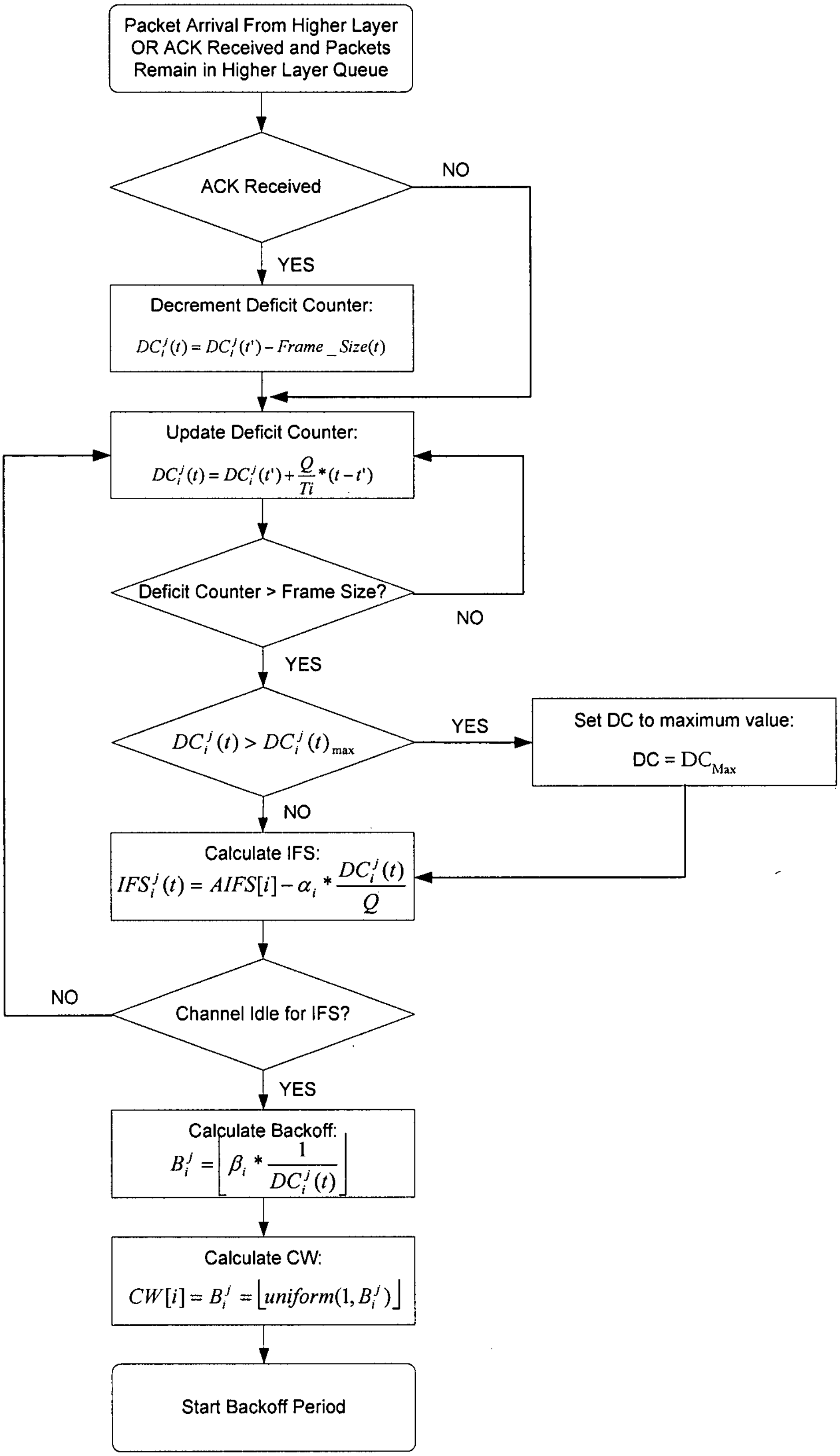
Packet Arrival From Higher Layer OR ACK Received and Packets Remain in Higher Layer Queue
ACK Received
NO
YES
Decrement Deficit Counter:
$DC_i^j(t) = DC_i^j(t') - Frame\_Size(t)$
Update Deficit Counter:
$DC_i^j(t) = DC_i^j(t') + \frac{Q}{Ti} * (t - t')$
Deficit Counter > Frame Size?
NO
YES
$DC_i^j(t) > DC_i^j(t)_{\max}$
YES
Set DC to maximum value:
$DC = DC_{Max}$
NO
Calculate IFS:
$IFS_i^j(t) = AIFS[i] - \alpha_i * \frac{DC_i^j(t)}{Q}$
NO
Channel Idle for IFS?
YES
Calculate Backoff:
$B_i^j = \left\lceil \beta_i * \frac{1}{DC_i^j(t)} \right\rceil$
Calculate CW:
$CW[i] = B_i^j = \lfloor uniform(1, B_i^j) \rfloor$
Start Backoff Period

Figure 3.2. Flowchart describing the EDRR scheme

3.4.1 IEEE 802.11 PHY Model

Details of the physical layer-dependent parameters are provided in (Table 3.3).

Model Feature	Description
Data Rate	Data rates supported by the WLAN protocol are: 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps.
Physical Layer	The following physical layer technologies from the IEEE 802.11 specification are modeled: -- Frequency hopping spread spectrum (FHSS) -- Direct sequence spread spectrum (DSSS) -- Infrared The radio pipeline stages have the following modifications: -- Radio Receiver - The receiver group of the station is restricted to its subnet and the transmitting station does not receive its transmitted packets. -- Channel Match - The transmitter and receiver data rate should match to successfully transmit a packet. -- Power Stage - Packets with a received signal power below the threshold (which is configurable) do not make the receiver busy and the receiver treats such packets as noise packets. This pipeline stage computes the received signal power and considers the results of the channel match stage when determining which packets are considered noise. -- Error Correction - If the receiving packet contains more errors than permitted by the error threshold, the pipeline stage marks it as a corrupted packet and the MAC layer discards it.
Communication Distance and Spatial Reuse	The maximum communication distance between two WLAN nodes is a function of three parameters: the transmission power of the sending node, the path-loss propagation model, and the reception power threshold (receiver sensitivity) of the receiving node. Based on the configured values of these parameters, you can model WLAN networks in which the communication distance is more than 300 meters. The IEEE 802.11 standard limits the distance between WLAN nodes to 300 meters. Therefore, WLAN networks that extend beyond 300 meters might incur a performance degradation in the WLAN MAC algorithm. The receiver sensitivity concept that is implemented through the reception power threshold attribute enables spatial reuse modeling with WLAN models. Packets with a reception power that is lower than the threshold cannot make the receiver lock onto their signal and will be treated as noise packets. When the signal of these packets is very weak, the receiver can simultaneously receive another packet with a strong signal from a nearby neighbor. This means that if two sets of WLAN nodes are far away from each other, they can act as two different LANs but still use the same BSS ID and the frequency band (and therefore double the total available bandwidth).

Table 3.3. OPNET WLAN PHY model features [31]

3.4.2 IEEE 802.11 MAC Models

The IEEE 802.11 MAC layer model that is included with OPNET Modeler 10.0 is an extensive implementation of the MAC portion of the standard and is based on the information contained in [23] and [3]. The model is written in a modular manner, such that it facilitates easy editing for researching new methods of improving on the standard and evaluating their performance. There are many configurable features that are easily changed through the node model interface. The node model interface contains an extensive library of IEEE 802.11 nodes such as workstations, servers, routers, and access points. Important features of the OPNET WLAN MAC model are included in Table 3.4.

Model Feature	Description
Access Mechanism	Carrier sense multiple access with collision avoidance (CSMA/CA) DCF access scheme as defined in the standard. The PCF access scheme, which can be used in infrastructure network configurations, is also supported.
Frame Exchange Sequence	Data and Acknowledgment frame exchange to ensure the reliability of data transfer. Optional RTS/CTS frame exchange for media reservation.
Deference and Backoff	Interframe spacing: DIFS, SIFS, EIFS for DCF, and PIFS for PCF implementation. The values of the intervals are selected based on the physical layer characteristics. Binary Exponential Backoff
Data Rate	Data rates supported by the WLAN protocol are: 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps.
Recovery Mechanisms	Retransmission mechanism for data frames used when the acknowledgment frame is not received. Short and long retry counters are defined in the standard.
Fragmentation and Reassembly	Optional data frame fragmentation based on the size of the data packet received from the higher layer. The fragments are reassembled at the destination station.
Duplicate Packet Detection	Tuple cache to store the information of the received packet so that duplicate packets are discarded by the MAC layer.
Access Point Functionality	A station can be configured as an access point in an infrastructure BSS network. All stations are capable of being an AP, however, only WLAN bridges, switches, or routers can connect a BSS to the DS - use these nodes when you are configuring an ESS.
Buffer Size	Data that the WLAN MAC received from a higher layer is stored in a buffer. The buffer size is limited to the maximum value set by the user. Higher layer packets are dropped after the maximum buffer size is reached.

Table 3.4. OPNET WLAN MAC model features [31]

Before we could implement our proposed EDRR scheme, we needed a baseline MAC with which to compare our results. Therefore, the first issue that needed to be addressed was the implementation of the EDCA functionality. In order to provide per flow QoS in the MAC layer, stations generate traffic that is classified by assigning different priorities to different types of traffic. In the case of our test network, the highest priority will be real-time voice traffic that represents a VoIP call. Other types of traffic that we use are video streams and data transfer. When traffic is generated by the "traffic_source," it passes through the "wlan_mac_intf," as depicted in Figure 3.3. If the packet has a destination address, it is passed down to the "wireless_lan_mac" and transmitted across the network. If there is no destination address, a random address is assigned and the packet is passed down to the "wireless_lan_mac" and transmitted.

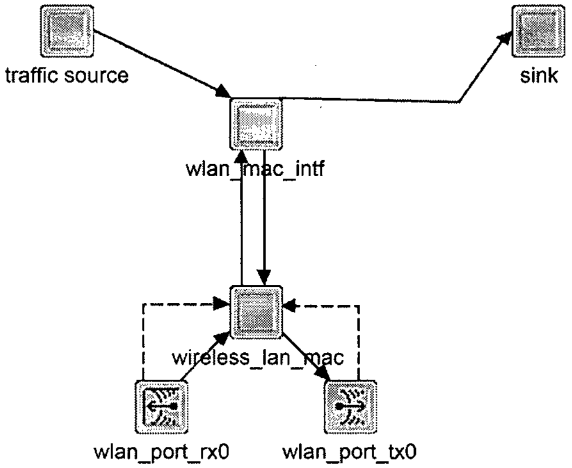

Figure 3.3. IEEE 802.11 workstation OPNET node model

Once a packet reaches the MAC layer, it is placed into the queue that corresponds to its TC. Packets are removed from the queues in a similar fashion to the DCF model that comes with OPNET. A couple of slight modifications are made to make the model fit the

EDCA of the 802.11e. The changes that are necessary to change the model from 802.11b to 802.11e are as follows:

- Provide 8 queues for different priorities of traffic
- Maintain a contention window for each of the queues (there will be up to 8 values of both cw_{min} and cw_{max}
- Maintain a backoff count and a backoff event handle for each queue
- Keep track of a retry count for each of the traffic categories for retransmission purposes
- Provide different deference intervals for each traffic category (AIFS [TC]) and keep an event handle for each of these intervals
- Display statistics on a per-flow basis (i.e. per-flow MAC delay, backoff slots, queue sizes etc.)

The eight queues are implemented using the standard list data structure that is provided by OPNET. All of the queues are first-come, first-served with queue 7 being the highest priority and queue 0 being the lowest priority. A contention window interval is maintained by scheduling interrupts to signify the end of a contention window. The minimum and maximum contention window values are set statically at the beginning of the simulation. The backoff count for each queue is an integer that is used to schedule interrupts to signify that a specific queue has finished backing off. When a backoff interval is paused due to a busy channel, the residual value is held in the backoff count variable. The number of times that a frame transmission is attempted is maintained on a per flow basis so that the frame can be dropped after the maximum number of retries. In order to give priority to the higher numbered queues, the amount of time that stations defer before attempting to transmit on an idle channel must be varied. The higher priority traffic will have a smaller deference interval

(AIFS interval) than the lower priority traffic. The final implementation issue is to provide some statistics for comparing the performance metrics that we are measuring in this thesis. Our simulation model collects statistics for MAC delay, delay jitter, throughput, and packet drop rate to be used for measuring the performance of our method with other proposed methods. We also collect a number of other statistics that were used for debugging the model.

After the 802.11e model was implemented, we had a baseline model with which to compare new models with. The next step in our process was to implement the functionality that would allow our model to emulate a distributed DRR scheduler. A number of changes had to be made to the 802.11e model in order to facilitate this and are as follows:

- Define the quantum of service for each type of traffic
- Maintain a per-flow deficit count for every queue in each QSTA
- Define scaling factors for the IFS and Backoff intervals so that the intervals map to the values that are recommended in the IEEE 802.11e draft standard
- Implement a function to calculate an appropriate IFS time from the deficit count
- Maintain $\text{Deficit\_Count}_{min}$ and $\text{Deficit\_Count}_{max}$ variables
- Implement a function to calculate an appropriate contention window from the deficit count

Figure 3.4 shows the EDRR MAC finite state machine. The FSM consists of 12 states with numerous transitions that define the semantics of our proposed scheme. We have implemented our scheme such that the states and transitions remain the same as for the 802.11b and 802.11e OPNET models. The interrupts and the executive code inside of the states, however, have been modified to reflect our changes to the algorithm. Each of the states in the EDRR finite state machine represents a logical state in which an EDRR station might be in. The use of each state in the FSM is now described.

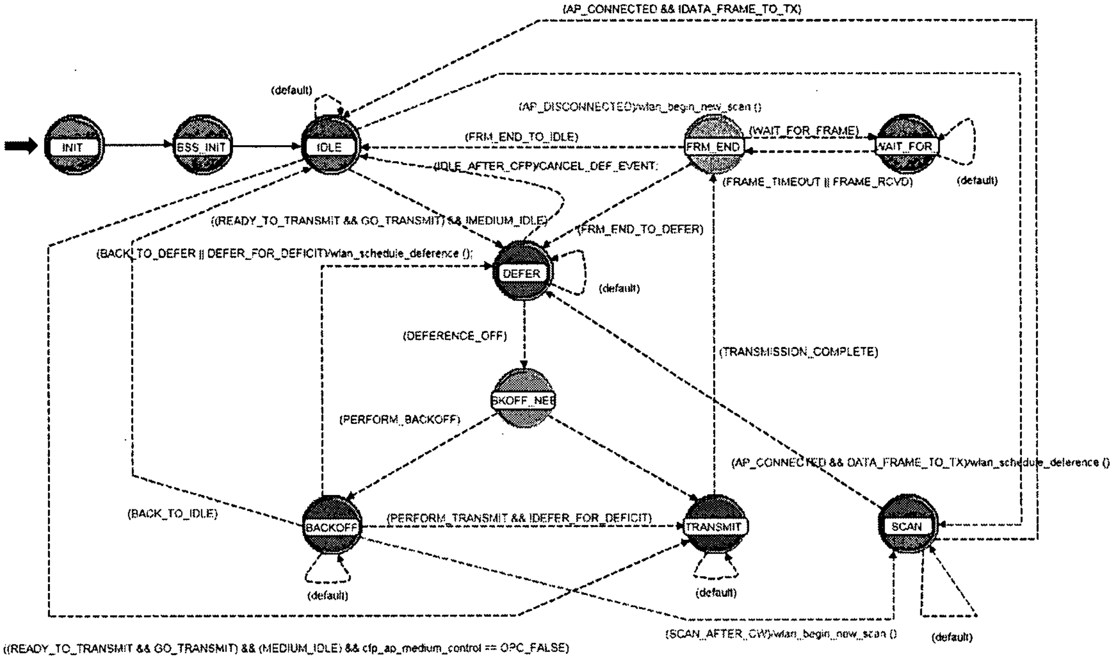
(AP_CONNECTED && !DATA_FRAME_TO_TX)
(default)
(AP_DISCONNECTED)/wlan_begin_new_scan ()
INIT
BSS_INIT
IDLE
(FRM_END_TO_IDLE)
FRM_END
(WAIT_FOR_FRAME)
WAIT_FOR_
(IDLE_AFTER_CFP)/CANCEL_DEF_EVENT;
(FRAME_TIMEOUT || FRAME_RCVD)
(default)
((READY_TO_TRANSMIT && GO_TRANSMIT) && !MEDIUM_IDLE)
(FRM_END_TO_DEFER)
(BACK_TO_DEFER || DEFER_FOR_DEFICIT)/wlan_schedule_deference ();
DEFER
(default)
(DEFERENCE_OFF)
(TRANSMISSION_COMPLETE)
BKOFF_NEE
(PERFORM_BACKOFF)
(AP_CONNECTED && DATA_FRAME_TO_TX)/wlan_schedule_deference ()
(BACK_TO_IDLE)
BACKOFF
(PERFORM_TRANSMIT && !DEFER_FOR_DEFICIT)
TRANSMIT
SCAN
(default)
(default)
(SCAN_AFTER_CW)/wlan_begin_new_scan ()
(default)
((READY_TO_TRANSMIT && GO_TRANSMIT) && (MEDIUM_IDLE) && cfp_ap_medium_control == OPC_FALSE)

Figure 3.4. The Finite State Machine for the EDRR MAC

The first two states in the FSM are used for initialization and are only used once per simulation. The INIT state initializes the process model, performs auto-addressing of the BSS and registers the node as an AP if the AP functionality is enabled. The BSS_INIT state gathers information about the other stations in the BSS and initializes variables that keep track of the other stations.

When a station has no packets to transmit, it waits in the IDLE, state for a packet to arrive from either the higher or lower layer. If data arrives from the higher layer, the station will either transmit immediately (move to the TRANSMIT state) if the medium is idle and the station has enough "credits" to transmit, or it will move to the DEFER state until the channel is idle and it has enough "credits" to transmit. If a frame arrives that is destined for this station, move to the DEFER state and wait SIFS before proceeding. In the case that the AP becomes disconnected, the station will go to the SCAN state.

The DEFER state is where a station waits if it has any packet to transmit and the medium is busy, the station lacks "credits" to transmit, or both. The station will remain in DEFER until the medium has been idle for SIFS in the case of transmitting a response packet, or the calculated IFS time for any queue that has data packets to transmit.

The BKOFF_NEEDED state is no longer used; the station will go to the BACKOFF state if it needs to backoff, or the TRANSMIT state if backoff is not needed.

When backoff is necessary, stations wait in the BACKOFF state until the contention window expires. Should the station receive a frame while it is backing off, the backoff is paused and the station returns to the DEFER state. If the station is backing off after the successful transmission of a packet and it has no more packets to send, the station will return to the IDLE state after the contention window expires. Otherwise, if the contention window expires and the station has a data frame to transmit, the station will transition to the TRANSMIT state.

A station will remain in the TRANSMIT state until it has finished transmitting a frame and immediately moves to the FRM_END state when the transmission is complete. When reaching the FRM_END state after transmitting a frame, a station will transition to the WAIT_FOR_RESPONSE state. In this state, the station will either receive the response within a certain time interval or a timeout will occur. In either case, the station will go back to the FRM_END state. At this point, the station will either go to the DEFER state if it has more packets to send or it will go back to the IDLE state.

3.4.3 Traffic Source Models

Our simulations consider three types of traffic that would all be common in real-life ad-hoc network situations. All of our simulations model scenarios with a mix of voice, video, and data traffic. All of the voice streams are assumed to be bi-directional, as are the data streams. The video stream is assumed to be a high-quality stream that is being pulled from one station down to another (i.e. client-server architecture). The traffic parameters are selected to closely emulate actual voice, video, and data sources. The priorities of the data sources are as follows: voice packets receive the highest priority, followed by video packets, and finally data packets. This ordering of priorities is a common one for multimedia environments due to the sensitivity to latency of the voice and video steams.

In order to accurately model the voice traffic, we use a bursty traffic model as the traffic source in our voice stations. In general, bursty traffic sources can be modeled as an alternating-state Markov process with two states of ON (active) and OFF (silent). Voice packets are generated when in the ON state and no packets are generated when the source is in the OFF state. The ON duration and the OFF duration both follow exponential distributions with a mean ON duration of 1 second and a mean OFF duration of 1.35 seconds [32].

For voice traffic, we assume the G.711 codec is used, which generates voice packets at a bit rate of 64 Kbps when the source is in the ON state. Our source generates a 260 byte packet every 30 ms when the source is in the ON state. G.711 is a good choice for a voice codec because it is robust against packet losses and it provides high quality voice.

In our simulations, we model a single video source that is streamed to one of the stations in the network. In order to show that we can transmit medium priority traffic at a high bit rate without harming the performance of other sources of both higher and lower priorities, we model a 1.6 Mbps MPEG video source. We assume the source is an MPEG-1 source that transmits at a constant bit rate. In order to achieve this, the source generates an 800 byte packet every 4 ms.

We assume that each of the data stations generate asynchronous data traffic at a constant bit rate of 100 Kbps. This traffic could represent any number of data sources such as receiving data from sensors, sharing music files, or transferring appointment information from one wireless device to another. The average length of a data MSDU is taken to be 1000 bytes and packets are generated every 80 ms.

Chapter 4

PERFORMANCE ANALYSIS OF EDCA AND EDRR

The previous chapter introduced our proposed EDRR scheme and the simulation models that we developed to evaluate the scheme. In order to validate our model, we devise and implement an analytical model for the EDCA and EDRR schemes. In Section 4.1, we describe existing analytical models for IEEE 802.11 networks, as well as the analytical models that we have developed. Sections 4.1.1 and 4.1.2 detail the analytical model for the IEEE 802.11 DCF and the saturation throughput for the DCF, respectively. Section 4.1.3 describes the performance analysis of the IEEE 802.11e EDCA. Next, in Section 4.1.4, we examine the saturation throughput for both the EDCA and EDRR methods. Finally, we validate our analytical model in Section 4.1.5.

4.1 Analytical Models for IEEE 802.11 Networks

We build an analytical model for determining the saturation throughput of IEEE 802.11e systems based on the existing models presented in [33] and [19]. The model presented in [33] is widely regarded as the definitive model for the IEEE 802.11 DCF, giving the simplest and most accurate results. In [19], the DCF model is extended to account for the differential IFS and CW parameters of the IEEE 802.11 EDCA. We begin our discussion of 802.11 analytical models with a description of the model proposed in [33].

4.1.1 Performance Analysis of the IEEE 802.11 DCF

The model proposed by Bianchi provides a simple and accurate analytical model for computing the IEEE 802.11 DCF throughput under saturation conditions, assuming ideal channel conditions, and a finite number of stations. Saturation throughput can be computed for both the RTS/CTS and basic access mechanisms that are provided in the DCF.

The model uses two discrete-time stochastic processes to track the backoff and retransmission mechanisms. The first process, *b(t)*, represents the backoff time counter of a given station. A discrete and integer time scale is adopted: *t* and *t* +1 correspond to the beginning of two consecutive slot times, and the backoff time counter of each station decrements at the beginning of each slot time [33]. A second process, *s(t)*, is defined to represent the backoff stage of a station at time *t*. For convenience, we define $W=CW_{min}$ and let $W_i=2^iW$. *S(t)* ranges from 0 to a maximum backoff stage *m*, which represents the maximum contention window value such that $CW_{max} = 2^mW$. When *b(t)* reaches zero, the station will attempt to transmit. Should the transmission attempt be successful, *s(t)* will be reset to zero and *b(t)* will be chosen uniformly between zero and CW_{max}. On the other hand, if the transmission attempt fails, *s(t)* will be incremented, provided that it is not already in the *m*th stage, and *b(t)* will be chosen uniformly between zero and 2^iW (where *i* represents the stage that *s(t)* resides in).

An important fact to note is that the discrete timescale for the backoff processes, *b(t)* and *s(t)*, do not correspond to system time. In fact, the backoff time decrement is stopped when the channel is sensed busy, and thus the time interval between two consecutive time slot beginnings may be longer than the slot time size σ, as the interval may include a packet transmission [33].

The key assumption to Bianchi's model is that for each transmission attempt, a packet collides with constant and independent probability *p*, regardless of the transmission history. *p* is referred to as the *conditional collision probability*, and represents the probability of a packet "seeing" a collision on the channel. The bi-dimensional process {*s(t)*, *b(t)*} with the discrete-time Markov chain shown in Figure 4.1 represents the model for the IEEE 802.11 DCF. The processes are governed by the following non-null one-step transition probabilities:

$$\begin{cases} P\{i,k \mid i,k+1\} = 1 & k \in [0, W_i - 2] \quad i \in [0,m] \\ P\{0,k \mid i,0\} = \dfrac{(1-p)}{W_0} & k \in [0, W_0 - 1] \quad i \in [0,m] \\ P\{i,k \mid i-1,0\} = \dfrac{p}{W_i} & k \in [0, W_i - 1] \quad i \in [1,m] \\ P\{m,k \mid m,0\} = \dfrac{p}{W_m} & k \in [0, W_m - 1] \end{cases} \tag{1}$$

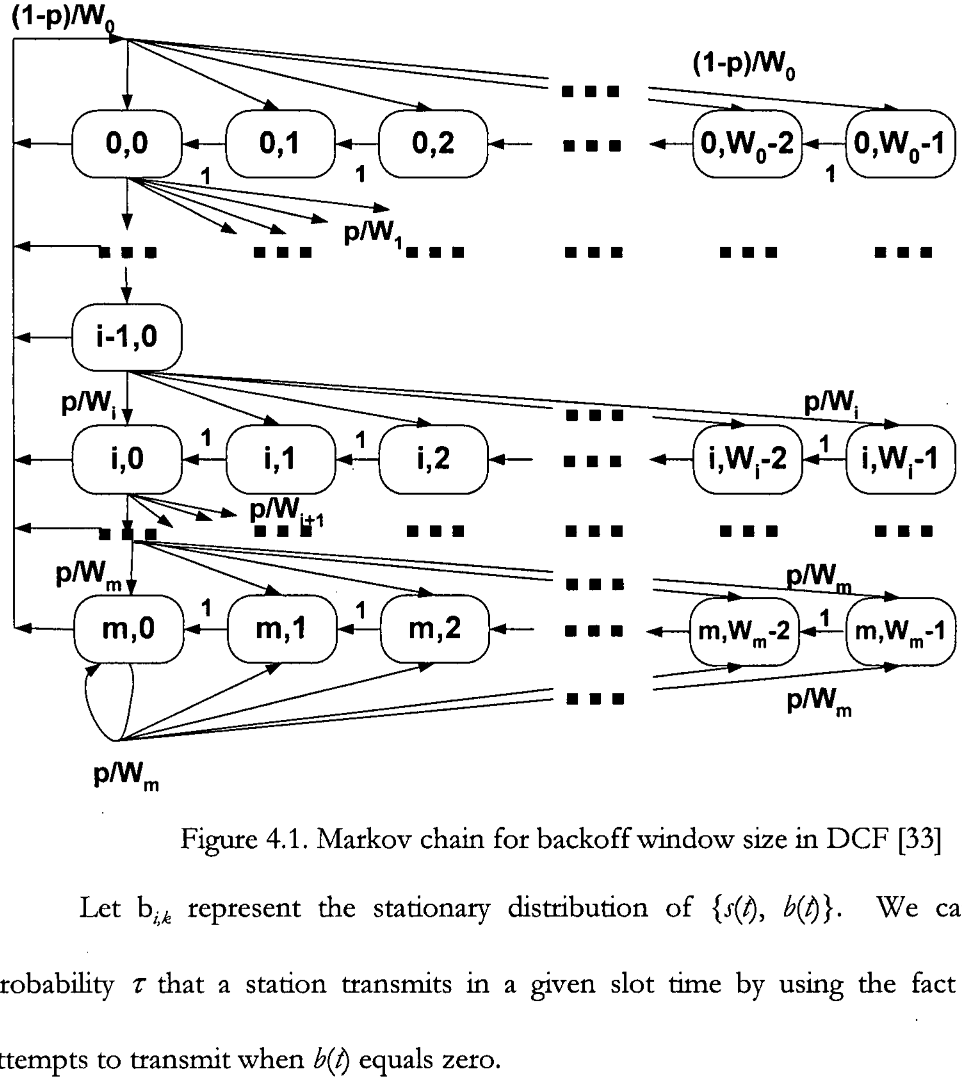

Figure 4.1. Markov chain for backoff window size in DCF [33]

Let $b_{i,k}$ represent the stationary distribution of $\{s(t), b(t)\}$. We can express the probability τ that a station transmits in a given slot time by using the fact that a station attempts to transmit when $b(t)$ equals zero.

Thus, τ is given by:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \qquad (2)$$

We now note that τ is given in terms of p, which has yet to be determined. In order to find p we use our assumption of fundamental independence, which implies that a transmitted packet "sees" a collision when one of the n-1 remaining stations transmits at the same time. At steady state, each of the remaining n-1 stations transmits with probability τ. Therefore, the conditional collision probability is given by:

$$p = 1-(1-\tau)^{n-1} \qquad (3)$$

Equations (2) and (3) represent a two equation non-linear system with unknowns τ and p, which can be solved using numerical techniques. Using τ, we can then calculate the normalized system throughput. This is done by defining two additional equations, the probability that a station transmits (P_{tr}), and the probability of a successful transmission (P_s), both of which are in terms of τ. These equations are defined as follows:

$$P_{tr} = 1-(1-\tau)^n \qquad (4)$$

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} \qquad (5)$$

4.1.2 DCF Saturation Throughput

We can now express the throughput, S, as the ratio:

$$S = \frac{E[Payload \quad Transmitted \quad Per \quad Timeslot]}{E[Length \quad of \quad Slot]} = \frac{P_s P_{tr} E[P]}{(1-P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}(1-P_s)T_c} \qquad (6)$$

Where *E[P]* is the average packet payload size, σ is the duration of an empty slot time, T_s is the average time the channel is sensed busy because of successful transmission, and T_c is the average time the channel is sensed busy due to collision. The numerator of (6) results from the fact that a successful transmission occurs in a slot time with probability $P_{tr}P_s$. The denominator is obtained by considering that with probability 1- P_{tr}, the slot time is empty, with probability $P_{tr}P_s$ a successful transmission occurs, and a collision happens with probability $P_{tr}(1-P_s)$. The values of T_s and T_c can be changed to accommodate both the RTS/CTS and basic access mechanisms of the DCF.

Using the above described model, and adjusting the CW and AIFS parameters in our EDCA simulator to match those of the DCF's DIFS and CW parameters, we were able to verify that our simulator gives results that are extremely close to the theoretical values presented in [33]. We forego presenting these results here, as we feel that the validity of our simulator is proven later by our results from the EDCA and EDRR simulators.

4.1.3 Performance Analysis of the IEEE 802.11e EDCA

The work from [33] is extended in [19] to accommodate the QoS features of the EDCA. This new model shares the approach of using p and τ to form the basis for a two-dimensional Markov process that represents the backoff counter and backoff stage of an IEEE 802.11e STA. In [19], the following conjectures are used to encompass the model:

- Each queue in the system is modeled by a Markov process specific to the AC associated with the queue.
- The probability that any transmission experiences a collision (p) is constant within a contention zone, regardless of the number of retransmissions suffered.

- The probability that a station initiates a transmission in a given backoff slot (τ) is constant across all of its backoff slots.
- The Markov process does not observe post-collision timeslots prior to the expiry of the EIFS; backoff and transmission during post-collision contention periods are treated outside the Markov process [19].

Using these conjectures, the authors of [19] are able to obtain highly accurate results for the saturation throughput of stations functioning under the EDCA. For our research purposes, this model gives us an excellent starting point for developing a model to analyze our EDCA implementation and our EDRR scheme under saturation conditions. We use the model developed in [19], with a few modifications, in order to make it fit our simulator for the IEEE 802.11 EDCA and EDRR.

The first modification we make is to remove the fourth conjecture used in [19] from consideration in the model. The reason for this relates to the fact that all collisions in 802.11e networks where all stations are within radio range of one another occur due to simultaneous transmissions. As we have made the assumption that all stations are within radio range of one another in our simulations, we are able to assume that all stations will be able to infer a collision has happened when there are simultaneous transmissions, therefore eliminating the possibility of the post-collision success defined in [19].

We use the remaining three conjectures to develop a simple, yet accurate model, for analyzing the throughput performance of the EDCA in saturation.

The 802.11e standard details that each AC in the EDCA should have a different set of parameters which consist of a CW_{min}, CW_{max}, and AIFS Tuple. The values of these parameters are varied in order to provide the different priority levels that enable QoS in the EDCA. The first conjecture is a result of this fact.

The second conjecture parallels Bianchi's assumption of constant and independent collision probability. In the EDCA, different contention zones exist, causing the number of stations competing for transmission attempts to vary. This is a result of the higher priority AC's having a shorter AIFS value, which expires earlier than for the low priority AC's and gives a marked advantage to high priority traffic. Since our model does not track progress within a transmission period, we use an average value for *p*, defined in terms of contention zone specific values for the probability of collision and the distribution of transmission attempts across contention zones [19]. The methodology behind this concept is discussed further below.

This modification requires some changes to the transitions and transition probabilities for the Markov chain that represents the EDCA backoff. Figure 4.2, the Markov chain model for the EDCA backoff procedure, reflects these modifications. For this Markov chain, the only non-null one-step transition probabilities are:

$$\begin{cases} P\{i,k \mid i,k+1\} = 1 & k \in [1, W_i - 1] \quad i \in [0,m] \\ P\{0,k \mid i,1\} = \dfrac{(1-p)}{W_0} & k \in [1, W_0] \quad i \in [0,m) \\ P\{i,k \mid i-1,1\} = \dfrac{p}{W_i} & k \in [1, W_i] \quad i \in [0,m] \\ P\{0,k \mid m,1\} = \dfrac{1}{W_0} & k \in [1, W_0] \end{cases} \qquad (7)$$

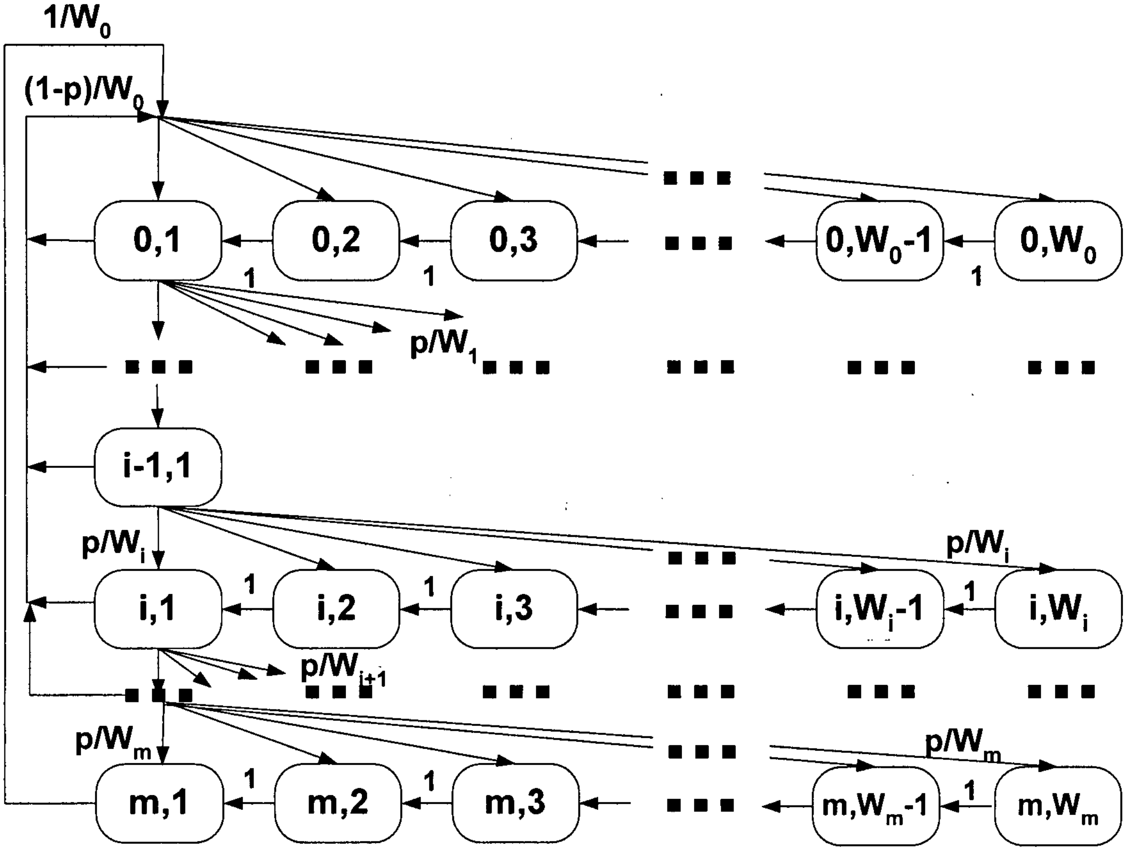

Figure 4.2. Markov chain for backoff window size in EDCA and EDRR

Once again, we let $b_{i,k}$ represent the stationary distribution of $\{s(t), b(t)\}$. We can note the following relationships between the backoff stages of the Markov chain.

$$b_{i,1} = pb_{i-1,1} = p^i b_{0,1} \tag{8}$$

Relationships for the stationary distributions of neighboring backoff states are:

$$b_{0,k} = \left(\frac{W_0 + 1 - k}{W_0}\right)\left[(1-p)\sum_{i=0}^{m-1}(b_{i,1}) + b_{m,1}\right] = \left(\frac{W_0 + 1 - k}{W_0}\right)b_{0,1} \tag{9}$$

Using (9) and the fact that $\sum_{i=0}^{m-1} b_{i,1} = b_{0,1}\left[\frac{1-p^m}{1-p}\right]$, we can see that the following holds:

$$b_{i,k} = \left(\frac{W_i + 1 - k}{W_i}\right)b_{i,1} \tag{10}$$

Thus, by relations (8) and (10), all the values of $b_{i,k}$ are expressed as functions of $b_{i,1}$ and p. We find $b_{i,1}$ in terms of p by imposing the normalization condition. We do not give a closed-form solution, however, due to its lack of visual appeal.

$$1 = \sum_{i=0}^{m}\sum_{k=1}^{W_i} b_{i,k} = \sum_{i=0}^{m}\sum_{k=1}^{W_i}\left(\frac{W_i + 1 - k}{W_i}\right) b_{i,1} \qquad (11)$$

Recalling our definition that a STA will attempt to transmit when its backoff counter process b(t) reaches zero, we can express the probability τ that a station attempts to transmit in a randomly chosen slot time as:

$$\tau = \sum_{i=0}^{m} b_{i,1} \qquad (12)$$

Once again, we leave the equation in this form, as there is no simple closed-form solution for τ. We note that there will actually be a value of τ for each AC in the system (2 in our case), which is dependent on the average collision probability for each AC.

Following the approach taken in [19], subsequent development is based on a hypothetical environment which consists of STAs that contain traffic from one of two ACs (A and B), with a finite number of stations and distinct values for AIFS, CW$_{min}$, and CW$_{max}$ for each AC. Such an environment consists of two contention zones: the first zone consists of only high priority stations (A) attempting to transmit, and the second zone contains stations of both high and low priority attempting to transmit (A and B). The model can easily be extended to accommodate more ACs if necessary; however, the inclusion of more than two ACs does not add anything to the analysis.

Since the probability of a collision depends on the number of stations that can attempt to transmit, we must define a zone-specific conditional collision probability to account for the varying number of stations across adjacent time zones. As was the approach in [33], we define the zone-specific conditional collision probability p_{zone} as the probability that in a time slot, at least one of the other stations in the network transmits. The value of p_{zone} can be calculated by:

$$\begin{aligned} p_{A:zone=1} &= 1-(1-\tau_A)^{n_A-1} \\ p_{A:zone=2} &= 1-(1-\tau_A)^{n_A-1}(1-\tau_B)^{n_B} \\ p_{B:zone=2} &= 1-(1-\tau_A)^{n_A}(1-\tau_B)^{n_B-1} \end{aligned} \tag{13}$$

In order to find the average conditional probability, which is needed to find τ, we weight the zone-specific conditional collision probabilities according to the long-term occupancy of the contention zones. Since a particular backoff slot can only be reached when no transmissions have taken place prior to it occurring in the current transmission period, and the probability of passing through a timeslot is constant, we can model the occupancy of the backoff slots as a Markov process. This process is governed by the probability of at least one transmission, p^{tr}_{zone}, occurring in a series of backoff slots and is shown in Figure 4.3. p^{tr}_{zone}, is given by the following expressions:

$$\begin{aligned} p^{tr}{}_{zone=1} &= 1-(1-\tau_A)^{n_A} \\ p^{tr}{}_{zone=2} &= 1-(1-\tau_A)^{n_A}(1-\tau_B)^{n_B} \end{aligned} \tag{14}$$

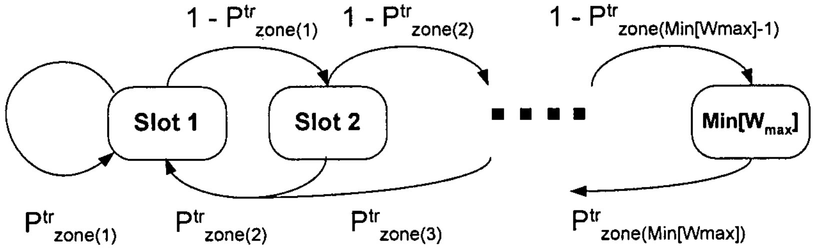

Figure 4.3. Relationship between adjacent timeslots in a transmission period [19]

From Figure 4.3, we can see note the relationship between adjacent backoff slots as:

$$b_i = (1-p^{tr}{}_{zone(i-1)})b_{i-1} \tag{15}$$

where b_i represents the occupancy of the i^{th} backoff slot and $p^{tr}_{zone(i-1)}$ represents the probability of a transmission occurring in the zone that the i-1^{th} backoff slot resides in.

The maximum number of terms in the stationary distribution for this Markov chain is bounded by the smallest contention window in the system. As the smallest contention window value varies over time due to collisions, the number of terms in the stationary distribution also varies. Therefore, we approximate the size of the stationary distribution by taking the smallest CW_{max} value as the number of states in the Markov chain. This is a valid approximation since as a stations progresses through the states in the Markov chain, the probability of transmission becomes much less, and hence has only a small effect on the occupancy of the distribution.

Working on this assumption, the stationary distribution is given by:

$$b_1 = \frac{1}{1 + \sum_{i=1}^{Min[cw_{\max}]-1} \prod_{j=1}^{i} (1 - p^{tr}{}_{zone(j)})} \tag{16}$$

Using (14) – (16) we can now calculate the average conditional collision probability by summing the zone-specific collision probabilities, weighted according to the backoff slot occupancy. The average collision probability is calculated per AC and is given by:

$$\begin{aligned} p_A &= \sum_{i=1}^{Min[cw_{\max}]} p_{A:zone(i)} \cdot b_i \\ p_B &= \frac{\sum_{i=1+AIFS_B-AIFS_A}^{Min[cw_{\max}]} p_{B:zone(i)} \cdot b_i}{\sum_{i=1+AIFS_B-AIFS_A}^{Min[cw_{\max}]} b_i} = p_{B:zone=2} \end{aligned} \tag{17}$$

Equations (12) and (17) form a non-linear system in the four unknown's τ_a, τ_b, p_a, p_b, which can be solved using numerical techniques. We implemented this analytical model using Mathematica 5.0 [34].

4.1.4 EDCA/EDRR Saturation Throughput Analysis

We define *S*, the system throughput, as the fraction of time the channel is used to successfully transmit payload bits. In order to compute *S*, we first analyze the event possibilities for a given transmission period. We use a transmission period, rather than a single slot time, so that we can account for the different contention zones that occur in an EDCA system. There are a number of possible events that can occur in a transmission period, including deferral, empty slots, successful transmissions, and collisions.

The probability of a successful transmission, $p^s_{zone(i)}$, is both zone and AC specific. We define $p^s_{zone(i)}$ as the probability that exactly one station transmits, conditioned on the fact that at least one station transmits, as in [33]:

$$\begin{aligned}
p^s{}_{A:zone=1} &= \frac{n_A\tau_A(1-\tau_A)^{n_A-1}}{P^{tr}{}_{zone=1}} = \frac{n_A\tau_A(1-\tau_A)^{n_A-1}}{1-(1-\tau_A)^{n_A}} \\
p^s{}_{B:zone=1} &= 0 \\
p^s{}_{A:zone=2} &= \frac{n_A\tau_A(1-\tau_A)^{n_A-1}(1-\tau_B)^{n_B}}{P^{tr}{}_{zone=2}} = \frac{n_A\tau_A(1-\tau_A)^{n_A-1}(1-\tau_B)^{n_B}}{1-(1-\tau_A)^{n_A}(1-\tau_B)^{n_B}} \\
p^s{}_{B:zone=2} &= \frac{n_B\tau_B(1-\tau_A)^{n_A}(1-\tau_B)^{n_B-1}}{P^{tr}{}_{zone=2}} = \frac{n_B\tau_B(1-\tau_A)^{n_A}(1-\tau_B)^{n_B-1}}{1-(1-\tau_A)^{n_A}(1-\tau_B)^{n_B}}
\end{aligned} \tag{18}$$

Now, using (14) and (18), we can compute S as the ratio:

$$S = \frac{E[Payload \quad Per \quad Transmission \quad Period]}{E[Length \quad of \quad Transmission \quad Period]} \tag{19}$$

In order to analyze the saturation throughput, once again, we let *E[P]* be the average packet payload size, σ is the duration of an empty slot time, T^s is the average time the channel is sensed busy because of successful transmission, and T^c is the average time the channel is sensed busy due to collision. The numerator of (19) results from the fact that a successful transmission occurs in a transmission period with probability $p^{tr}_{zone(i)}p^s$. We must also weight the payload, using (15), to reflect the steady state probability of reaching a particular time slot. From this, we define the expected payload transmitted per transmission period as (20). The denominator of (19) is obtained by considering that with probability 1- $p^{tr}_{zone(i)}$, the slot time is empty, with probability $p^{tr}_{zone(i)}p^s$ a successful transmission occurs, and a collision happens with probability $p^{tr}_{zone(i)}$ (1-p^s). Summing the events that are possible in a transmission period, and weighting them according to the long-term occupancy of the backoff slots by using (15), we get the expected length of a transmission period as (21).

$$\begin{aligned}
&\text{E[A Payload Per Transmission Period]} = n_a \sum_{i=1}^{\text{Min}[CW_{max}]} [b_i (p^s{}_{A:zone(i)})(p^{tr}{}_{zone(i)}) E[P_A]] \\
&\text{E[B Payload Per Transmission Period]} = n_b \sum_{i=1}^{\text{Min}[CW_{max}]} [b_i (p^s{}_{B:zone(i)})(p^{tr}{}_{zone(i)}) E[P_B]]
\end{aligned} \tag{20}$$

E[Length of transmission Period]

$$= \sum_{i=1}^{\text{Min}[CW_{max}]} \begin{aligned} &b_i [\sigma(1 - p^{tr}{}_{zone(i)}) + (p^{tr}{}_{zone(i)} p^s{}_{A:zone(i)} T^s{}_A) + (p^{tr}{}_{zone(i)} p^s{}_{B:zone(i)} T^s{}_B) \\ &+ (p^{tr}{}_{zone(i)})(1 - p^s{}_{A:zone(i)} + p^s{}_{B:zone(i)}) T^c] \end{aligned} \tag{21}$$

As in [33], modeling the throughput in this way allows the flexibility to specify either the RTS/CTS or basic access scheme. The scheme is determined by the definition of T^s and T^c, which are given in (22). We let the header size, $H = PHY_{hdr} + MAC_{hdr}$, and δ be the propagation delay. In the IEEE 802.11e standard, EIFS is defined as *EIFS=SIFS+ACK+AIFS[AC]* from the last time the medium was busy. Since this definition is different for each AC, and since the expiration of EIFS does not necessarily correspond to a slot boundary, we define *EIFS'* as (23).

$$\begin{aligned} T^{s}{}_{basic:j} &= H + E[P] + \delta + SIFS + ACK + \delta + AIFS_{\min} \\ T^{c}{}_{basic:j} &= H + E[P] + EIFS' \\ T^{s}{}_{RTS:j} &= RTS + \delta + SIFS + CTS + \delta + SIFS + T^{s}{}_{basic} \\ T^{c}{}_{RTS} &= RTS + EIFS' \end{aligned} \tag{22}$$

$$EIFS' = \left\lceil \frac{ACKTime - SlotTime}{SlotTime} \right\rceil + AIFS_{\min} \tag{23}$$

4.1.5 EDCA/EDRR Model Validation

We verify our analytical model using the EDCA and EDRR simulators that we have developed in OPNET. The actual OPNET models are described in the next chapter, but it suffices here to say that we have developed these models.

All of our validation simulations consist of an increasing number of STAs competing to transmit a fixed size UDP packet in a single-hop WLAN. All STAs use the RTS/CTS access mechanism, and all maintain the same EDCA parameters. The PHY dependent parameters are configured as the pertaining values for the DSSS PHY layer. The parameters used for our EDCA simulations are listed in Table 4.1. Our simulations for the EDRR scheme use the same parameters as for the EDCA, except that the AIFS value is reduced to the value of the next lowest AC (i.e. AIFS[4]->AIFS[3]). We maintain saturation by increasing the arrival rate of packets beyond a level that STAs can successfully attempt to transmit packets.

Frame Payload	8000 bits
MAC Header	224 bits
PHY Header	192 bits
ACK	112 bits + PHY Header
RTS	160 bits + PHY Header
CTS	112 bits + PHY Header
Channel Bit Rate	1 Mbit/s
Propagation Delay	1μs
Slot Time	20μs
SIFS	10μs
Retry Limit	6
AIFS[4]	SIFS + 2 x Slot Time
AIFS[3]	SIFS + 2 x Slot Time
AIFS[2]	SIFS + 3 x Slot Time
AIFS[1]	SIFS + 7 x Slot Time
CWmin[4]	7
CWmin[3]	15
CWmin[2]	31
CWmin[1]	31
CWmax[4]	15
CWmax[3]	31
CWmax[2]	1023
CWmax[1]	1023

Table 4.1. EDCA and PHY parameters for simulations using a DSSS PHY

Figures 4.4 – 4.8 present our analytical and simulation results for a number of different scenarios operating under the EDCA scheme. All of the Figures plot throughput versus the number of stations per AC in order to show the effect of increasing traffic load on the system throughput. We also plot the total system throughput versus the number of stations per access category. It can be seen in the Figures that our analytical model gives an accurate estimation of the values that result from the simulations for a variety of AC combinations.

Figure 4.4 shows the throughput of AC[4] (high priority) and AC[3] (low priority) under increasing load. In this scenario, the two priorities are differentiated by the contention window parameters only. We can see from the Figure that although service differentiation is provided, the low priority traffic is still able to achieve a substantial throughput, relative to the total system throughput. As the number of stations increases, the throughput approaches zero due to the small contention window parameters in the stations. This is because the number of colliding stations is large compared with the number of slots that the stations can choose as a backoff counter, resulting inevitably in more collisions.

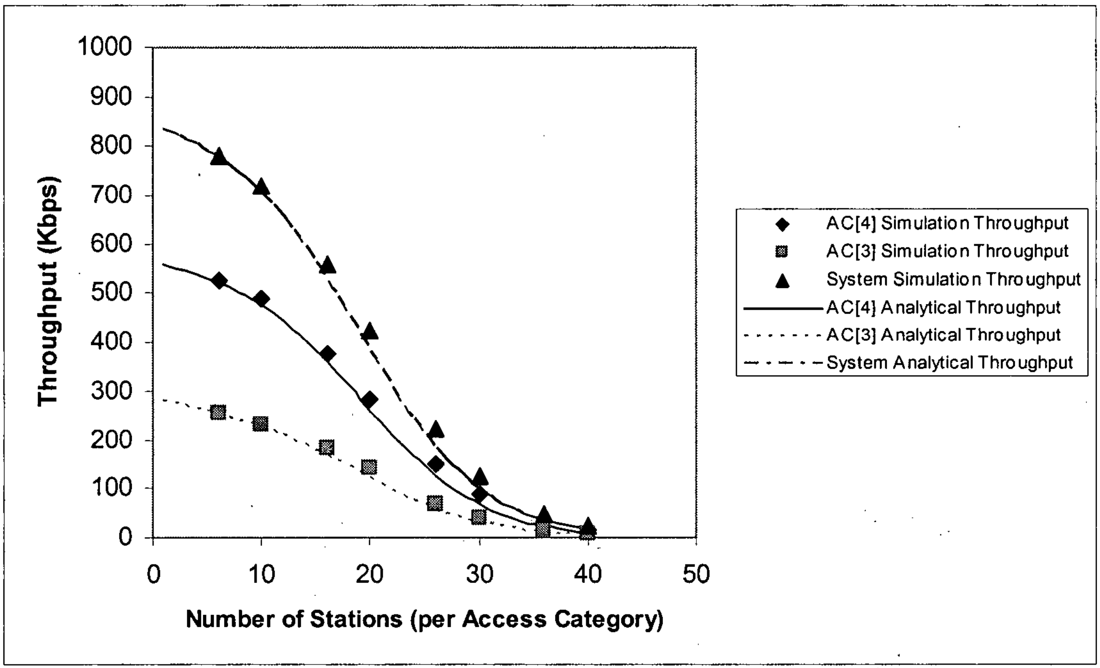

Figure 4.4. Simulation and analysis results for AC[4] and AC[3] throughput with increasing load under EDCA

Figure 4.5 shows the throughput of AC[3] (high priority) and AC[2] (low priority) under increasing load. In this scenario, the two priorities are differentiated by the contention window parameters and the AIFS parameters. We notice a big difference in Figure 4.5 compared to Figure 4.4, in that the low priority traffic receives a much lower percentage of the bandwidth than it did when the ACs were differentiated by the contention window parameters only. When the number of stations in the network is small, the number of collisions is correspondingly small. This allows some of the low priority packets to be transmitted when their backoff counter expires. As the load increases, however, collisions become more frequent and the low priority traffic is choked out due to the large contention window values. As this happens, the high priority traffic eats up the bandwidth that is given up by the low priority traffic, resulting in a temporary increase in the throughput for high priority traffic.

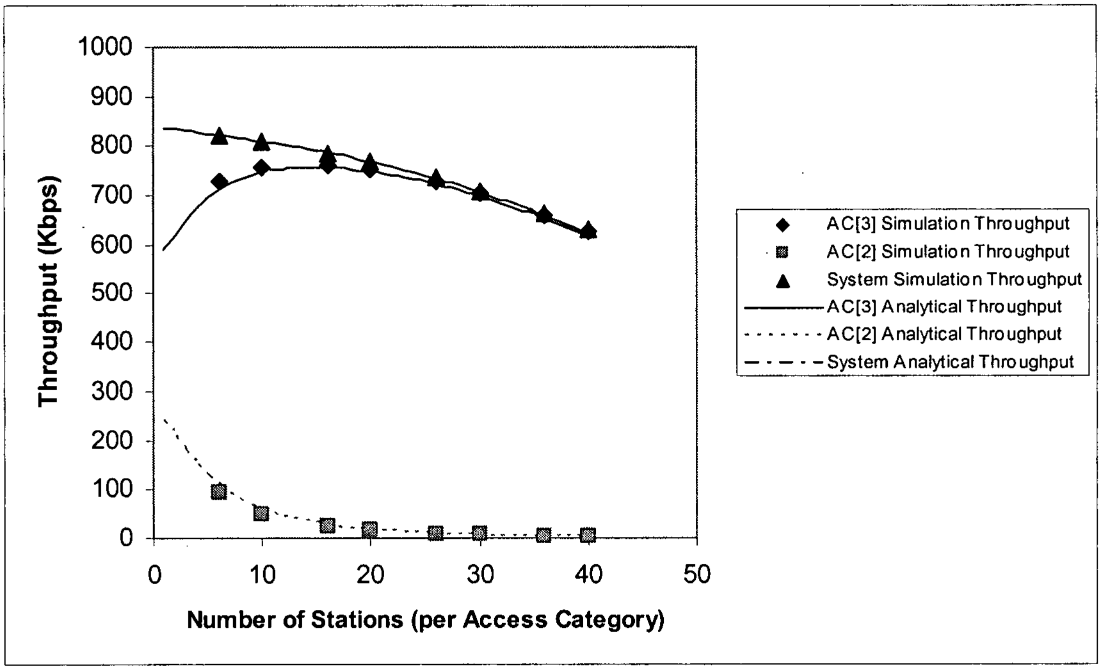

Figure 4.5. Simulation and analysis results for AC[3] and AC[2] throughput increasing load under EDCA

Figure 4.6 shows the throughput of AC[2] (high priority) and AC[1] (low priority) under increasing load. In this scenario, the two priorities are differentiated by the AIFS parameters only. Comparing Figure 4.6 to Figure 4.5, we notice that although the curves are shaped nearly the same, the low priority traffic receives a slightly higher percentage of the bandwidth than it did when the ACs were differentiated by the contention window and AIFS parameters. Our reasoning for describing the shape of the throughput curve is the same as for the previous Figure. As the low priority stations surrender transmission attempts due to collisions and large contention windows, the high priority traffic takes advantage of the available bandwidth. When there are around 16 stations per AC, the channel saturates and the throughput begins to decrease with the increasing load.

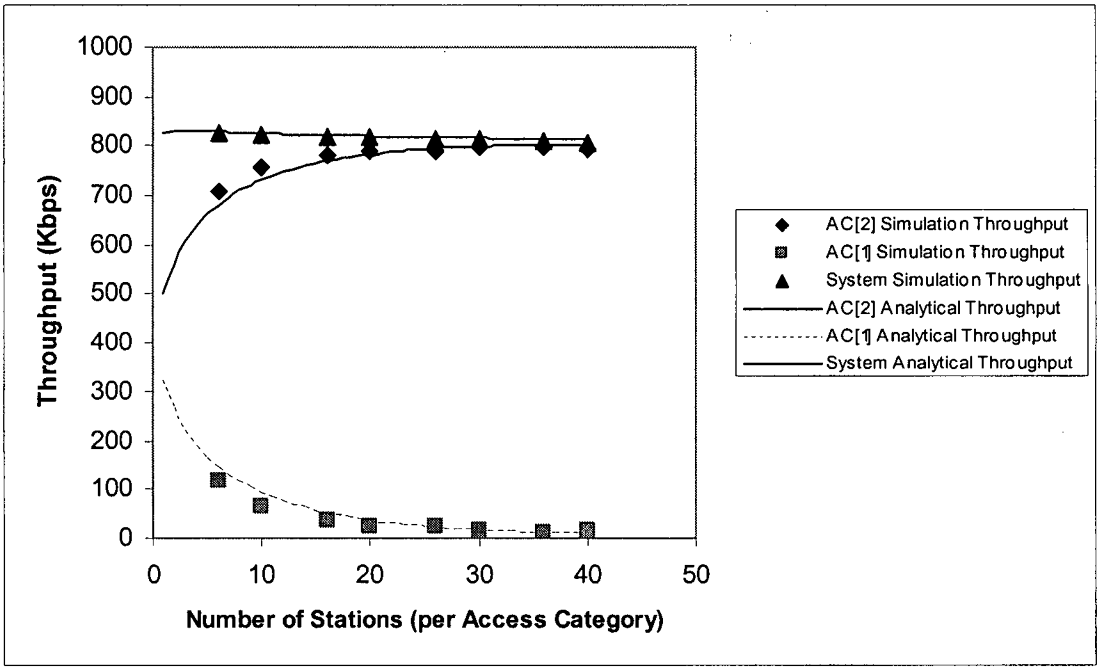

Figure 4.6. Simulation and analysis results for AC[2] and AC[1] throughput with increasing load under EDCA

Figures 4.7 and 4.8 are for the purpose of comparing the throughput between two ACs when one of the ACs is of the highest priority. Such a scenario may exist when VoIP stations exist in the network. Figure 4.4, which we have already discussed, also falls into this category. We notice that the general shape of Figures 4.7 and 4.8 is similar to that of Figure 4.4. The difference is that in Figures 4.7 and 4.8, the low priority traffic achieves a much lower throughput than in the low priority traffic in Figure 4.4. We note in Figures 4.7 and 4.8 that the high priority traffic uses most of the available bandwidth as collisions become more frequent with increasing load. Figure 4.7 shows that the low priority traffic is choked out after the number of stations per AC increases beyond twelve. Due to the longer AIFS of the low priority traffic in Figure 4.8, the low priority traffic is starved after only four stations per AC exist in the network. In addition, the low priority stations in Figure 4.8 achieve a lower bandwidth than the low priority stations in Figure 4.7, even when the number of stations per AC is one or two.

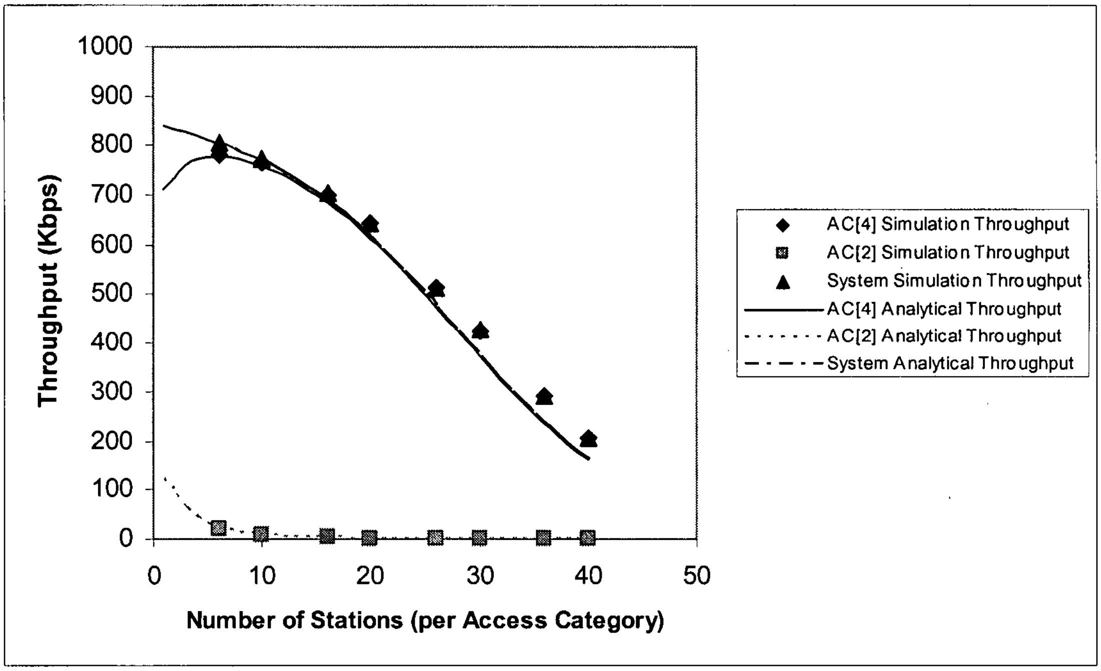

Figure 4.7. Simulation and analysis results for AC[4] and AC[2] throughput with increasing load under EDCA

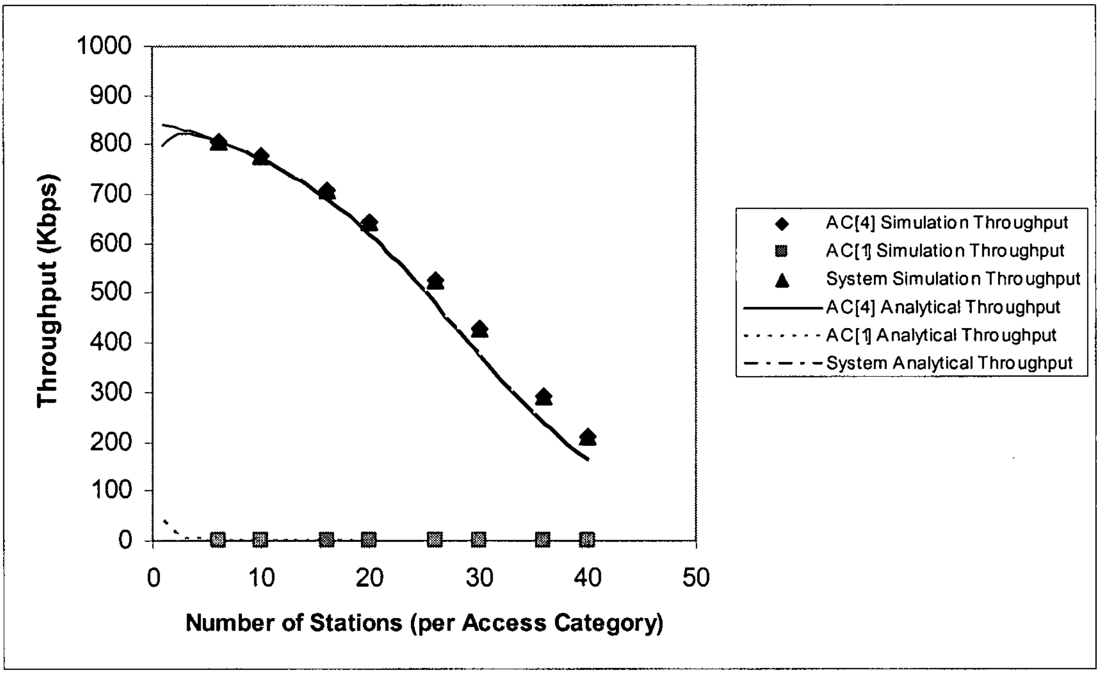

Figure 4.8. Simulation and analysis results for AC[4] and AC[1] throughput with increasing load under EDCA

Figures 4.9 – 4.13 show the simulation and analytical results under the EDRR scheme for the same set of scenarios that were presented in the previous five Figures. Once again, we can see that the analytical and simulation results are close to one another, therefore giving proof that our analytical and simulation models are both correct.

Figures 4.9 and 4.10 correspond almost identically with Figures 4.4 and 4.5. This is because the EDRR contention window parameters for AC[4] and AC[3] are the same as the contention window parameters for AC[4] and AC[3] under the EDCA. The only difference between this simulation and the EDCA simulation for the same ACs is that the AIFS parameter is reduced by 1 slot time for both the high and low priority traffic. The general shapes of the curves remain the same because the high and low priority traffic are still differentiated by the contention window parameters only.

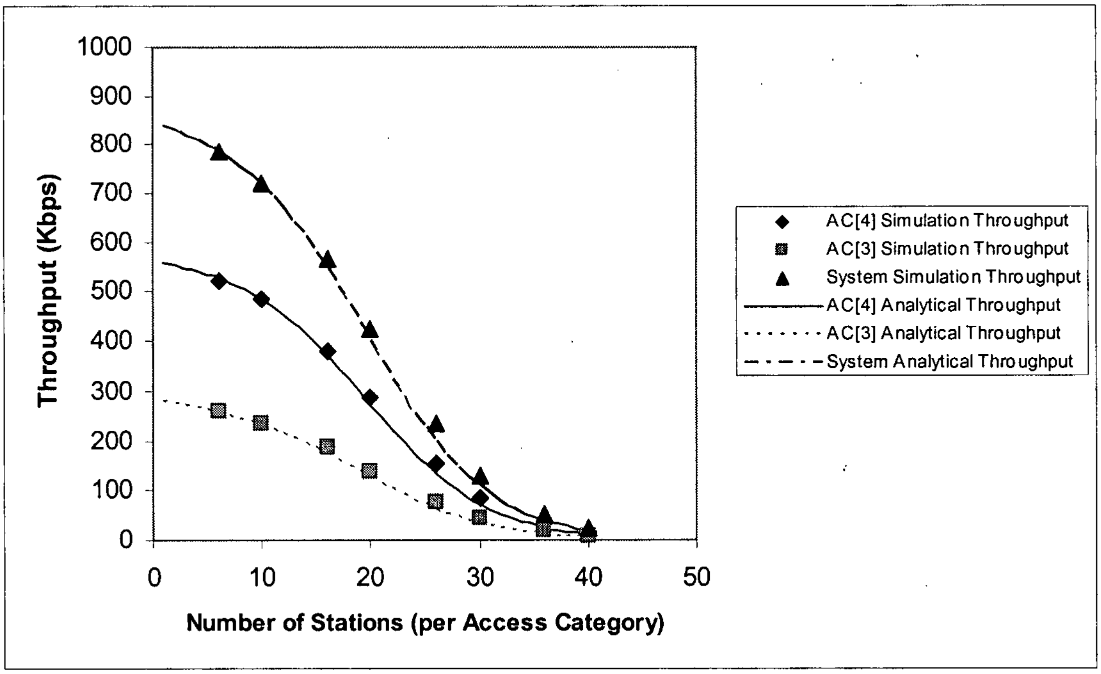

Figure 4.9. Simulation and analysis results for AC[4] and AC[3] throughput with increasing load under EDRR

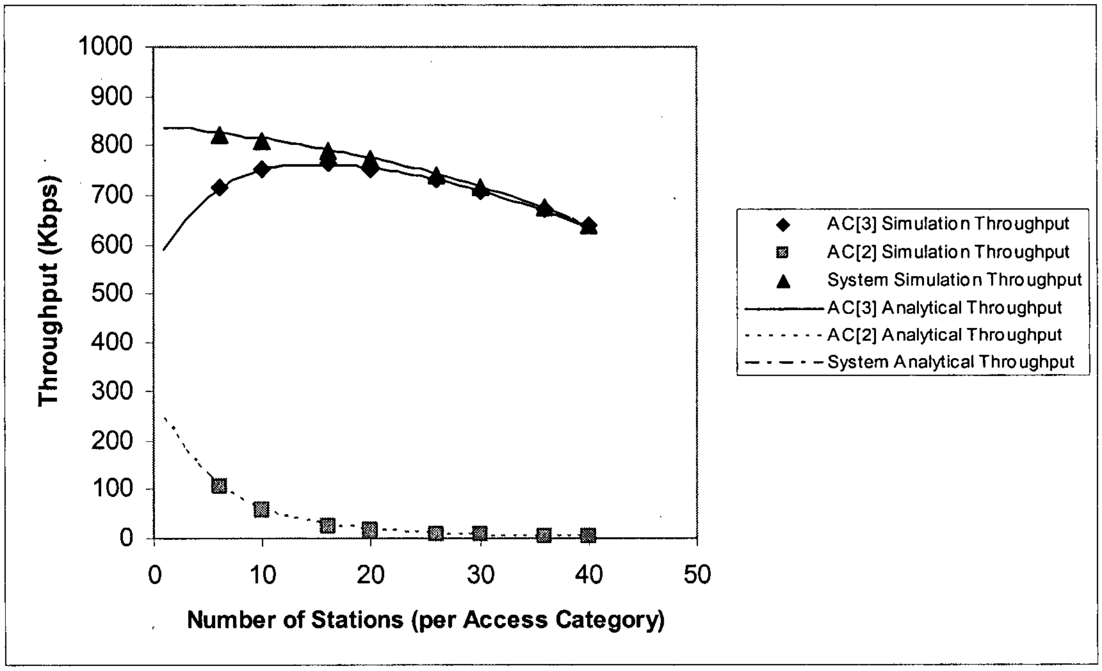

Figure 4.10. Simulation and analysis results for AC[3] and AC[2] throughput with increasing load under EDRR

Figure 4.11 shows the throughput of AC[2] (high priority) and AC[1] (low priority) for EDRR stations under increasing load. In this scenario, the two priorities are differentiated by the AIFS parameters only, as was the case in the EDCA simulations. Comparing Figure 4.11 to Figure 4.6, we notice a substantial difference in the shapes of the curves. The reason for this is due to the differentiation that is provided by the AIFS parameter in the EDRR simulations. In the EDCA simulations, high and low priority traffic AIFS parameters were differentiated by four slot times. For our EDRR simulation, this difference is reduced to one slot time. Therefore, as a result of the lower difference in AIFS values, the low priority traffic is able to get a higher throughput result. We note, though, that the general trend of the curves remains the same as it was in the EDCA simulations. The trend is that as the low priority stations surrender transmission attempts due to collisions and large contention windows, the high priority traffic takes advantage of the available bandwidth, increasing the total throughput for the high priority stations.

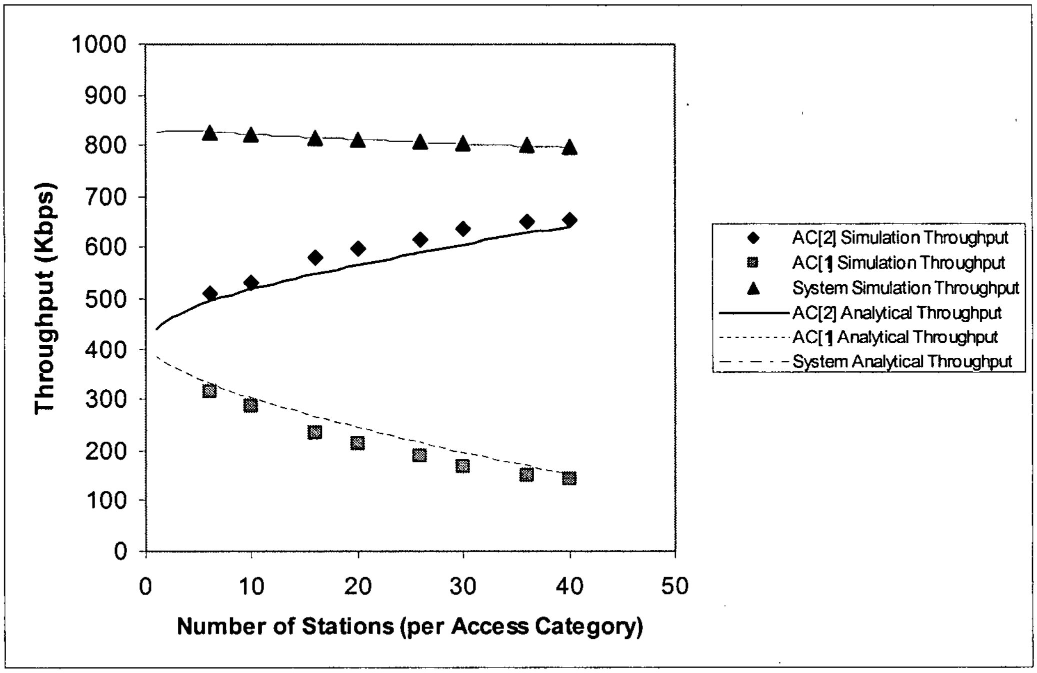

Figure 4.11. Simulation and analysis results for AC[2] and AC[1] throughput with increasing load under EDRR

Figures 4.12 and 4.13 are for the purpose of comparing the throughput between two ACs when one of the ACs is of the highest priority. These simulations correspond to the simulations that we ran under the EDCA for networks that might contain a mix of VoIP and data traffic. Figure 4.9, which we have already discussed, also falls into this category. We notice that the general shape of Figures 4.12 and 4.13 is similar to that of Figure 4.9. As was the case in the EDCA simulations, the difference is that in Figures 4.12 and 4.13, the low priority traffic achieves a much lower throughput than in the low priority traffic in Figure 4.9. Once again, we note in Figures 4.12 and 4.13 that the high priority traffic uses most of the available bandwidth as collisions become more frequent with increasing load. Figure 4.12 shows that the low priority traffic is choked out after the number of stations per AC increases beyond twelve. Due to the longer AIFS of the low priority traffic in Figure 4.13, the low priority traffic is starved after only ten stations per AC exist in the network. As a result of the smaller differentiation of AIFS values between high and low priority traffic, the EDRR scheme allows low priority traffic to achieve a slightly higher throughput than in the EDCA scheme.

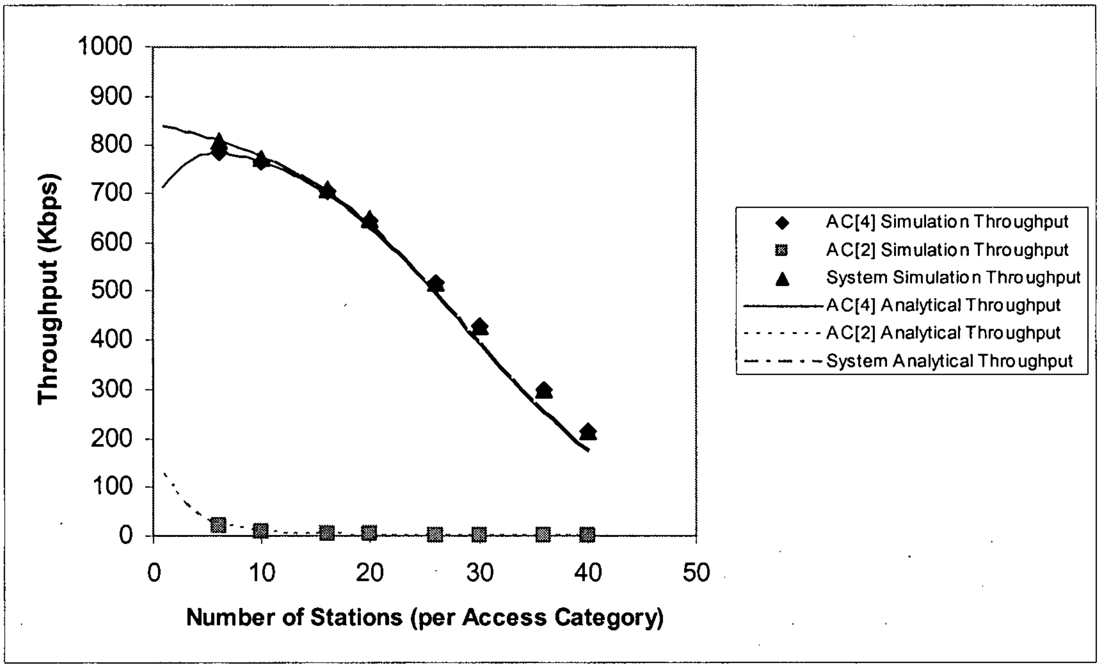

Figure 4.12. Simulation and analysis results for AC[4] and AC[2] throughput with increasing load under EDRR

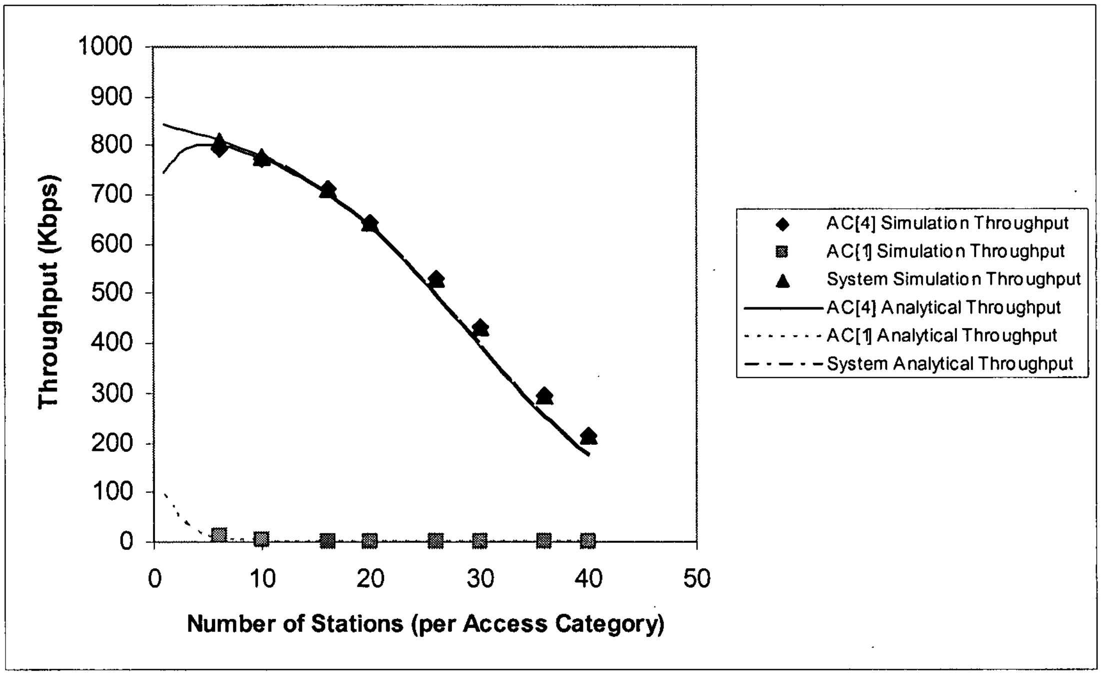

Figure 4.13. Simulation and analysis results for AC[4] and AC[1] throughput with increasing load under EDRR

Chapter 5

SIMULATION SCENARIOS AND PERFORMANCE RESULTS

The previous two chapters introduced OPNET, the 802.11 models, our EDRR model, and the analytical models that we used in our simulations. The performance of our EDRR scheme will be evaluated using these models and compared with other similar schemes in order to show the performance gains that can be achieved by using EDRR. This chapter details our simulation scenarios, settings and results for ad-hoc networks with a number of different traffic types and under different load stresses.

All of our network simulation scenarios are QoS IBSSs (QIBSSs) and we assume that all QSTAs are operating in the same QIBSS and are within radio range of all other QSTAs; therefore the hidden-node problem is excluded. We use all ad-hoc networks in order to exemplify the fact that the EDRR scheme is fully distributed. Fully distributed schedulers are important in Wireless LANS since contention-based channel access is the primary coordination function used in industrial implementations of the IEEE 802.11. Our scheme can also be easily extended to include a QBSS under the control of an access point that is operating in the contention period of a superframe.

Section 5.1 details the usefulness of QIBSS networks and Section 5.2 describes performance metrics that are used to evaluate QoS in such networks. Section 5.3 describes a network scenario with a mix of voice and data traffic and Section 5.4 describes a QIBSS network with voice, video, and data traffic being sent. A discussion of the results from each simulation and a discussion of the settings used in the simulation is provided following the scenario description. Section 5.5 gives conclusions that are drawn taking into account the simulations for each of the network scenarios. Each of the scenarios is simulated with 5 different seed values for a period of 5 minutes, and the average of the results from all stations is used for generating the plots. The data rate for all simulations is set to 5.5 Mbps and HCF functionality is disabled. The scaling factor values that we used for our simulations are shown in Tables 5.1 and 5.2. The scaling factor values are chosen statically, and are dependent on the AC, the packet size, and the application throughput requirements. The scaling factors are chosen such that the IFS and contention window values that they multiply should fit within the recommend values for AIFS and contention windows in [7].

AC	Scaling Factor (α)
2	7.50E-10
3	8.00E-10
4	1.70E-09

Table 5.1. IFS scaling factors (α) used for simulations

AC	Scaling Factor (β)
2	5.50E+05
3	2.50E+05
4	7.50E+04

Table 5.2. Backoff scaling factors (β) used for simulations

5.1 QIBSS Ad-Hoc Network Topologies

The rapid increase in the use of mobile devices and the accompanied maturity in wireless technologies have contributed to wireless networks becoming almost as ubiquitous as their wired counterpart [35]. As a result of this, ad-hoc wireless networks are becoming a viable solution for setting up networks quickly, easily and without any required infrastructure. Some of the possible applications and scenarios for ad-hoc networks are as follows:

- Task oriented collaborative computing in emergency relief stations or in a battlefield.

- The ability to share entertainment sources with others, such as music, games and videos. This capability would be extremely useful while waiting for a flight in an airport lounge or similar situation of idle time in a public place [36].
- Any computing situations in an outdoor environment, where the availability of infrastructure is not feasible. Such a situation might be a research field that is located in a wilderness area.

For this research, we have chosen a QIBSS topology as our simulation environment.

5.2 Performance Metrics

In order to measure the effectiveness of EDRR, we must define some performance metrics that will enable us to see that EDRR not only performs well, but that it performs better than some of the other proposed schemes. We must also be able to see that acceptable QoS is provided to high priority traffic while not choking out lower priority traffic. The most important metrics for multimedia applications are delay, jitter, loss, and throughput. Definitions of the performance metrics are defined as follows:

- Medium Access Delay – The time from when a packet arrives from the higher layer until the time that the station first attempts to transmit the packet
- Jitter – The statistical distribution of variation in the arrival time of a packet to its destination, or the delay variation
- Packet Loss – The percentage of packets that do not successfully reach their destination
- Throughput – The amount of data successfully transmitted and received per unit time

5.3 Scenario 1 – Mixed Voice and Data Traffic in a QIBSS

This Section presents a scenario in which voice and data traffic compete for bandwidth in an ad-hoc network. Section 5.3.1 describes the setup of the simulated network topology and the results of the simulation are presented in section 5.3.2. Finally, we discuss the results in Section 5.3.3.

5.3.1 Network Topology for Scenario 1

As mentioned above, one possible application of ad-hoc networks is task oriented collaborative computing in emergency relief situations. In such a case, voice services may be used to let emergency workers communicate from wireless devices to the command center. As with any network, data will also likely need to be transported in emergency relief situations. In the case of a wildfire disaster, information such as the number of acres burned or the movement patterns of fires might need to be monitored and could be transmitted to the command center via a wireless connection.

Figure 5.1 shows the topology of a QIBSS that could be representative of an emergency relief situation. The network consists of 8 EDRR stations sending voice traffic and *n* number of data stations, where *n* is varied in order to show the response of the performance metrics.

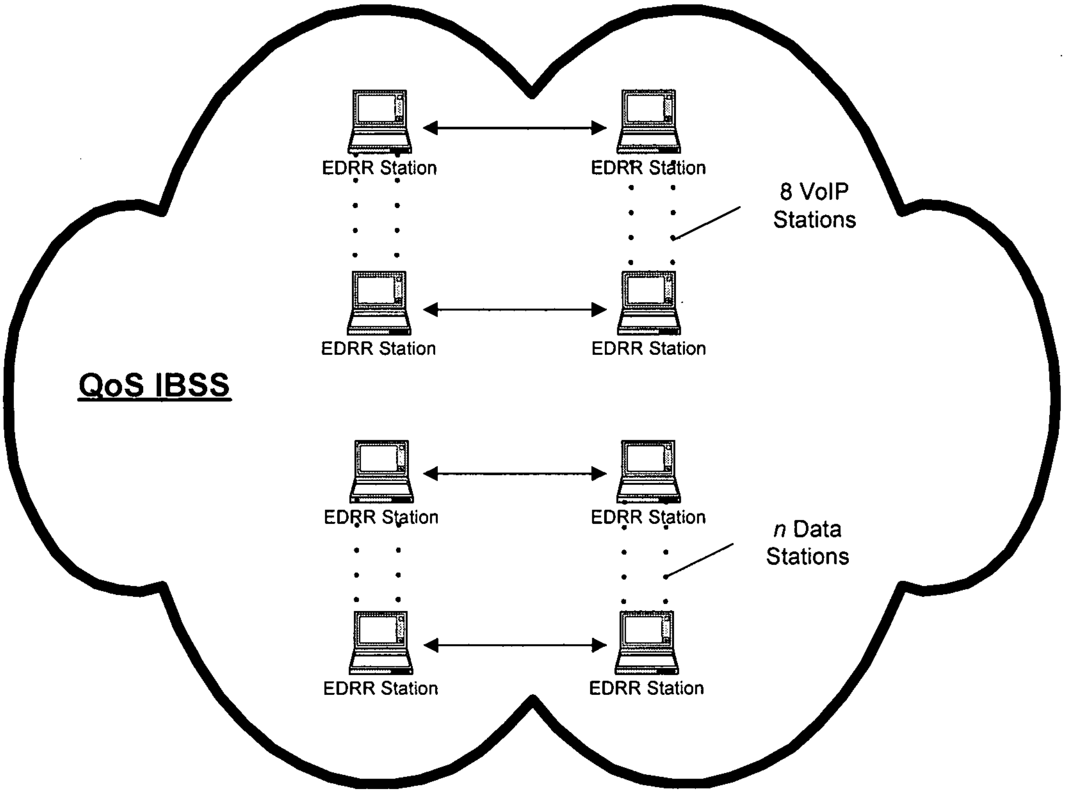

Figure 5.1. Mixed Voice and Data Traffic in a QIBSS

5.3.2 Results from Scenario 1 Simulations

This section presents the results from our ad-hoc network with mixed voice and data traffic. Figures 5.2, 5.3, and 5.4 show the average MAC delay, average jitter, and average throughput, for voice packets, respectively. A discussion of the results from Scenario 1 is provided in Section 5.3.3.

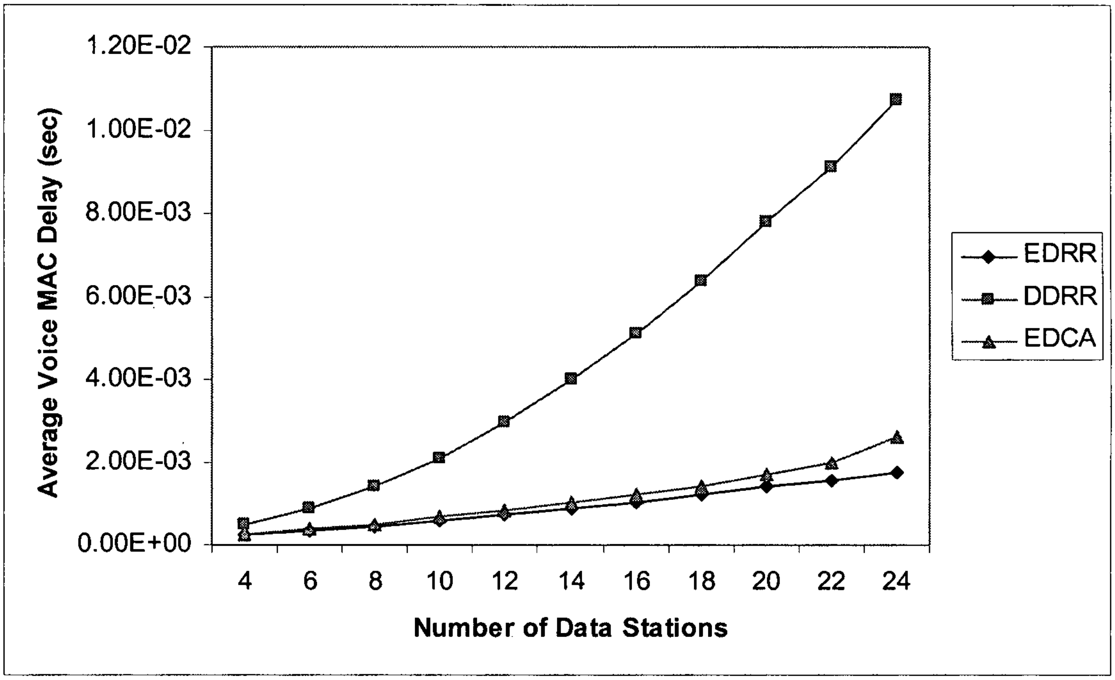

Figure 5.2. Average MAC Delay for Voice Packets under EDRR, DDRR, and EDCA

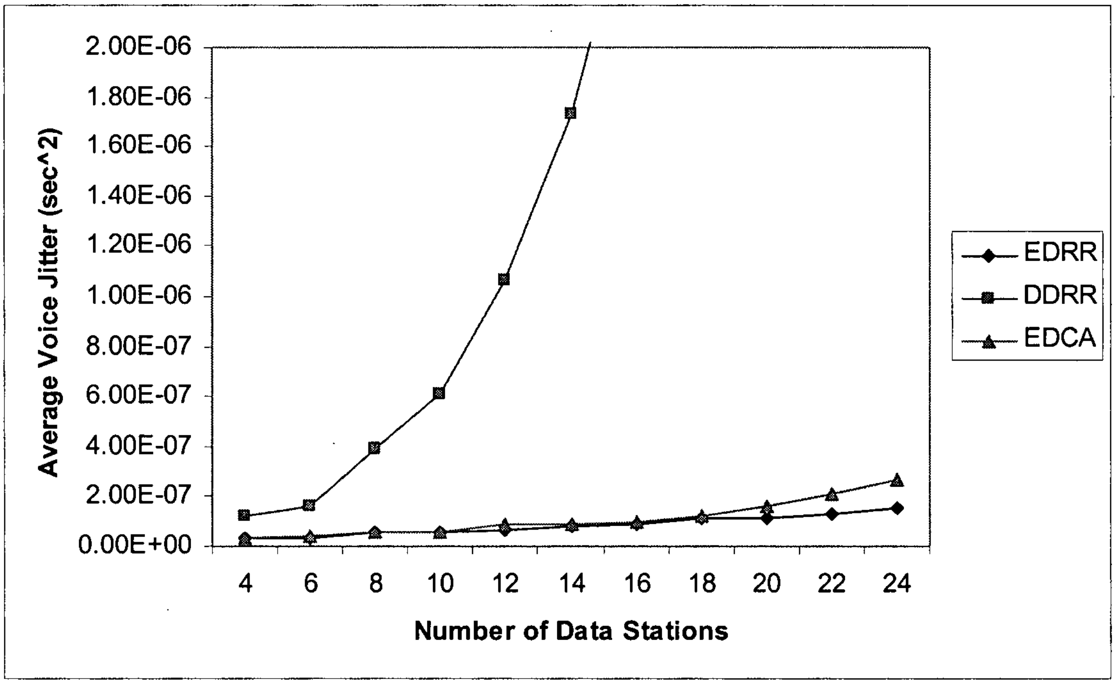

Figure 5.3. Average Jitter for Voice Packets under EDRR, DDRR, and EDCA

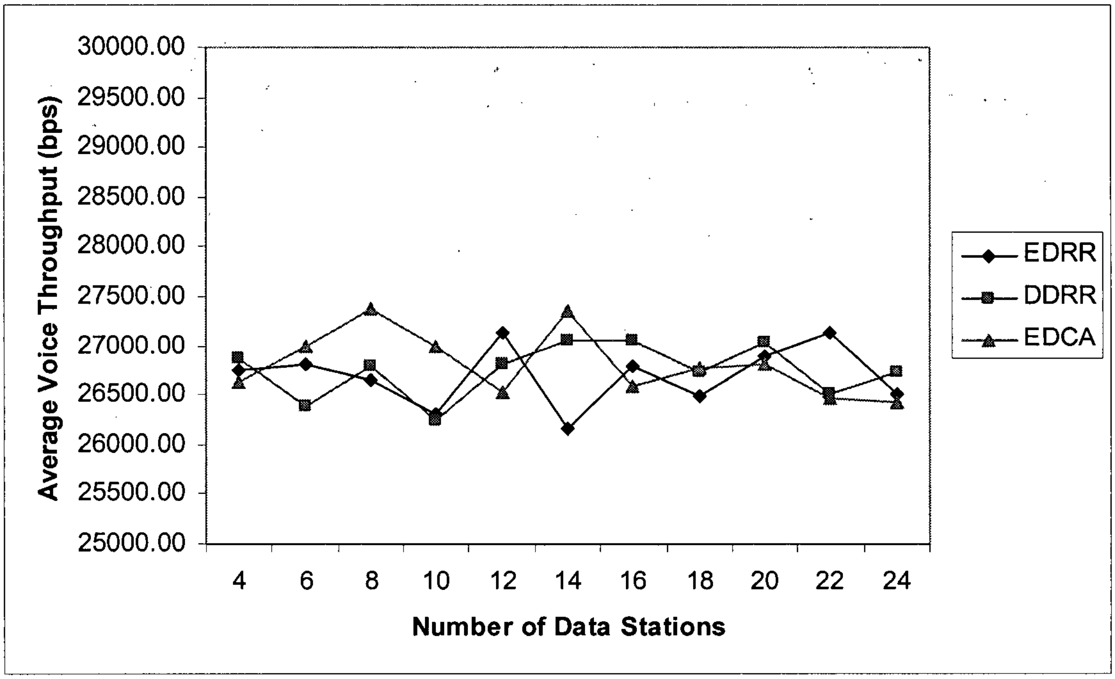

Figure 5.4. Average Throughput for Voice Packets under EDRR, DDRR, and EDCA

Figures 5.5, 5.6, and 5.7 show the average MAC delay, average jitter, and average throughput for data packets, respectively.

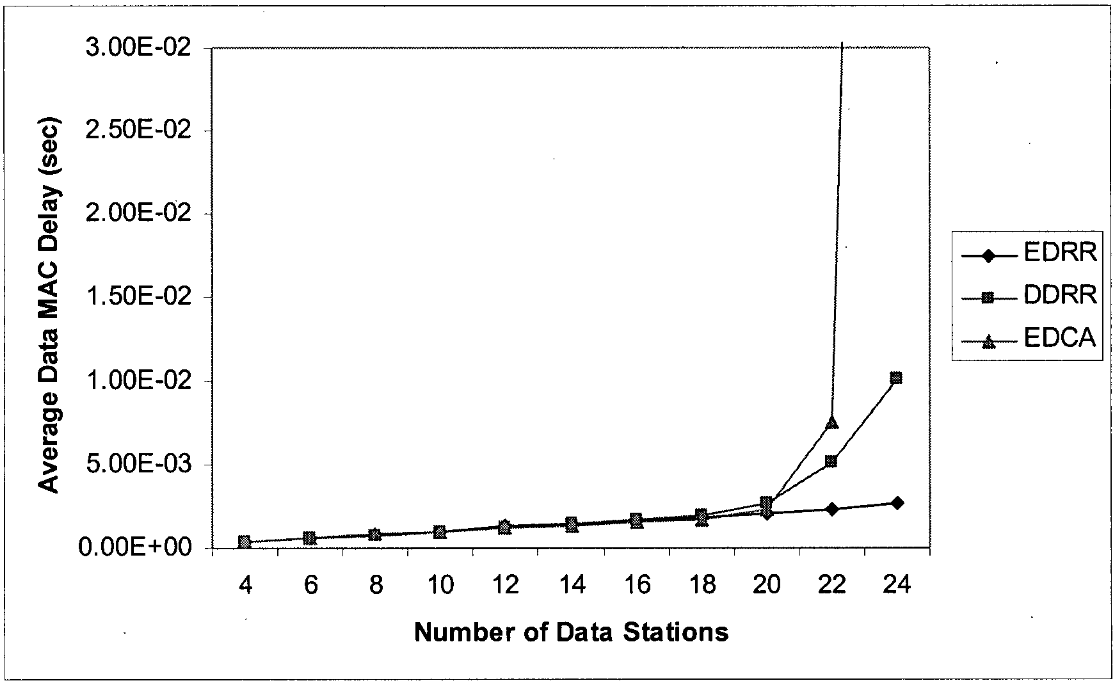

Figure 5.5. Average MAC Delay for Data Packets under EDRR, DDRR, and EDCA

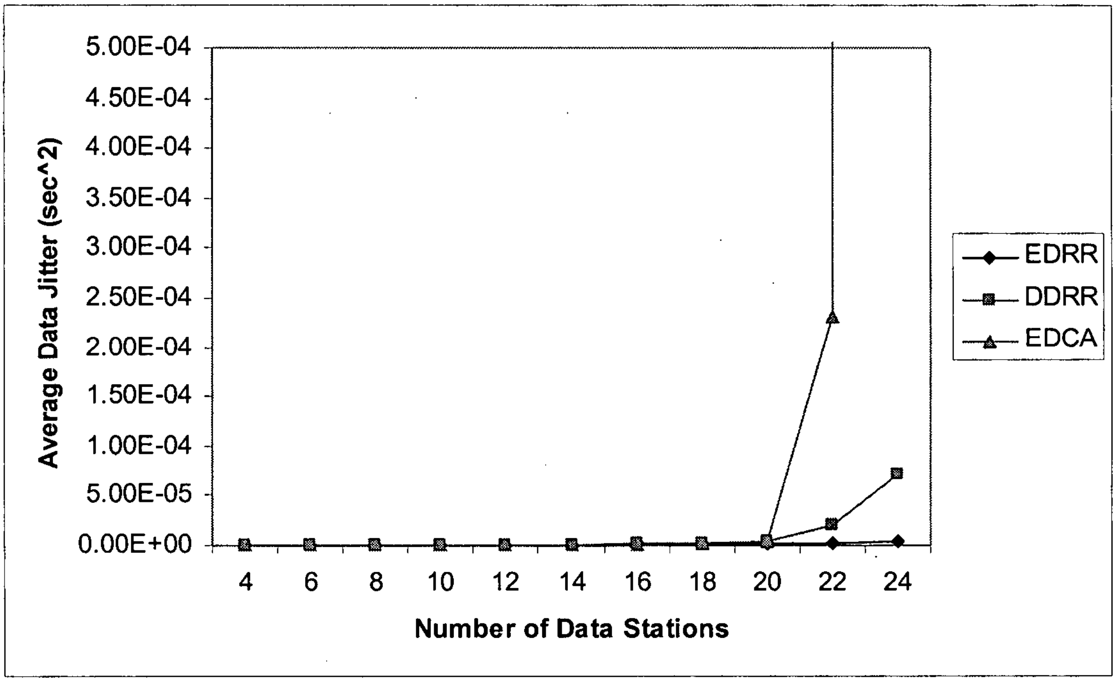

Figure 5.6. Average Jitter for Data Packets under EDRR, DDRR, and EDCA

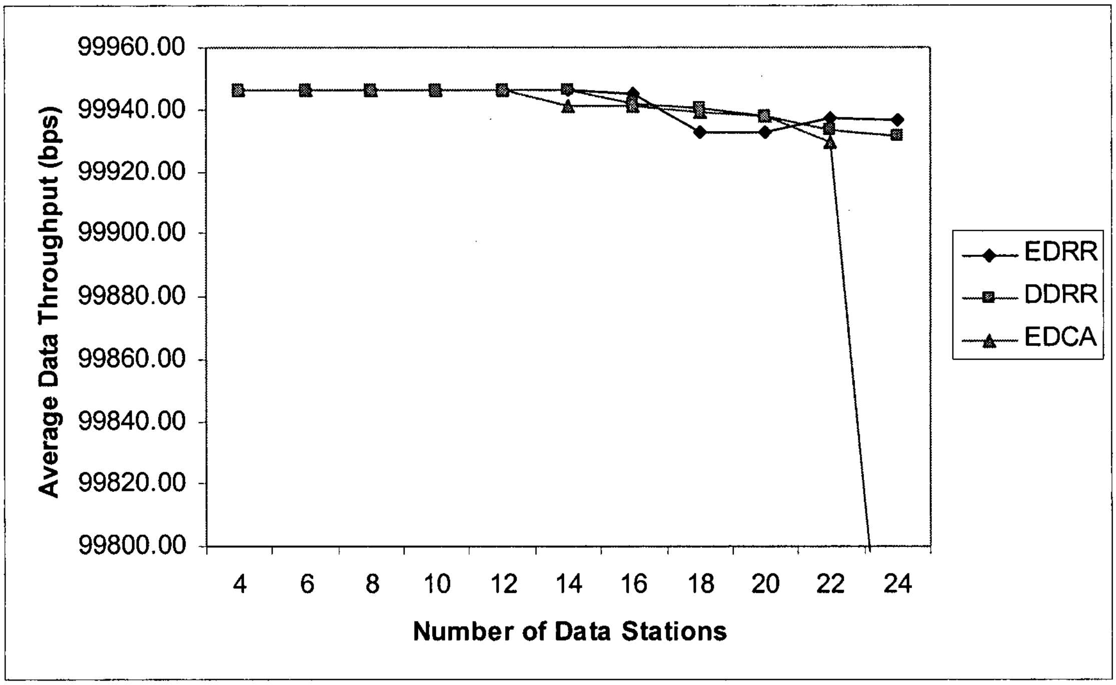

Figure 5.7. Average Throughput for Data Packets under EDRR, DDRR, and EDCA

5.3.3 Discussion of Results From Scenario 1 Simulations

The results of our simulations show that the proposed EDRR scheme equals or outperforms DDRR and EDCA in relation to all of the measured performance metrics.

Figures 5.2 and 5.3 show that due to the lack of support for traffic flow priorities, the MAC delay and jitter values suffer for voice packets under DDRR. Using the flow priority methodology provided under the EDCA, EDRR is able to achieve the same MAC delay and jitter values as the EDCA for up to 20 data stations, and performs better than the EDCA when there are more than 20 data stations.

It can be seen from Figure 5.4 that the EDCA gives the highest throughput for voice packets under a lightly loaded network. This is due to the lack of overhead incurred in deference and backoff time in the EDCA methodology when there are few collisions. When the network becomes more loaded, EDRR and EDCA give the same results for average throughput of voice packets. On the other hand, the average throughput for DDRR under light to medium network load is slightly less than that of EDRR. After the load increases above 16 stations, the throughput of voice packets suffer once again, due to the lack of support for flow priorities in DDRR.

Due to the fact that this scenario does not provide high enough levels of traffic to over stress the network, the average drop rate for voice packets was approximately zero for all three methods.

Figures 5.5 and 5.6 show that under the EDCA, the MAC delay and jitter values are much higher, on average, than they are under EDRR and DDRR. This results from the strict prioritization of the traffic flows under the EDCA. Since voice packets are always given shorter IFS and backoff times than data packets, the data packets will suffer in the long run.

DDRR improves on this flaw by mapping the deficit count of the voice packets to the IFS time and EDRR does the same with the addition of a backoff time that is also related to the deficit count. Because of this, the voice packets still achieve a low MAC delay and jitter without choking out the data traffic. EDRR performs slightly better than DDRR due to the reduced number of collisions. This is because EDRR uses binary exponential backoff when a collision occurs, where DDRR does not.

Figure 5.7 shows that all three methods give a high throughput for data packets under light and medium load. As the number of stations increases, however, throughput under the EDCA begins to suffer due to the large contention windows that result from having more collisions. EDRR and DDRR are able to maintain a high throughput for the data packets because there are fewer collisions due to their round-robin nature.

As was the case with voice packets, the average packet drop rate for data packets is approximately zero.

5.4 Scenario 2 – Voice, Video, and Data in a QIBSS Network

This section presents a scenario in which voice, video and data traffic coexist in a QIBSS ad-hoc network. Section 5.4.1 describes the setup of the simulated network topology and the results of the simulation are presented in Section 5.4.2. Finally, we discuss the results from Scenario 2 in Section 5.4.3.

5.4.1 Network Topology for Scenario 2

Another possible application of an ad-hoc network is in a wilderness environment, where it is not possible to setup an infrastructure network. In this situation, biologists might mount a camera to a tree and point it at a bird's nest in order to study the behavior of the mother bird or it's young. The camera could be equipped with a wireless NIC so that an ad-hoc network can be established with it, allowing the biologists to study the birds from a distance, without disturbing them. Accompanying the video might be a number of microphones for listening to the sounds made by the chicks or the mother bird. Temperature and precipitation sensors may also be in the general area for monitoring the weather and how it effects the development of the young birds.

Figure 5.8 shows the topology of a QIBSS that could be representative of a wilderness research environment. The network consists of 4 EDRR stations sending voice traffic, 1 EDRR station sending a video stream, and *n* number of data stations, where *n* is varied in order to show the response of the performance metrics.

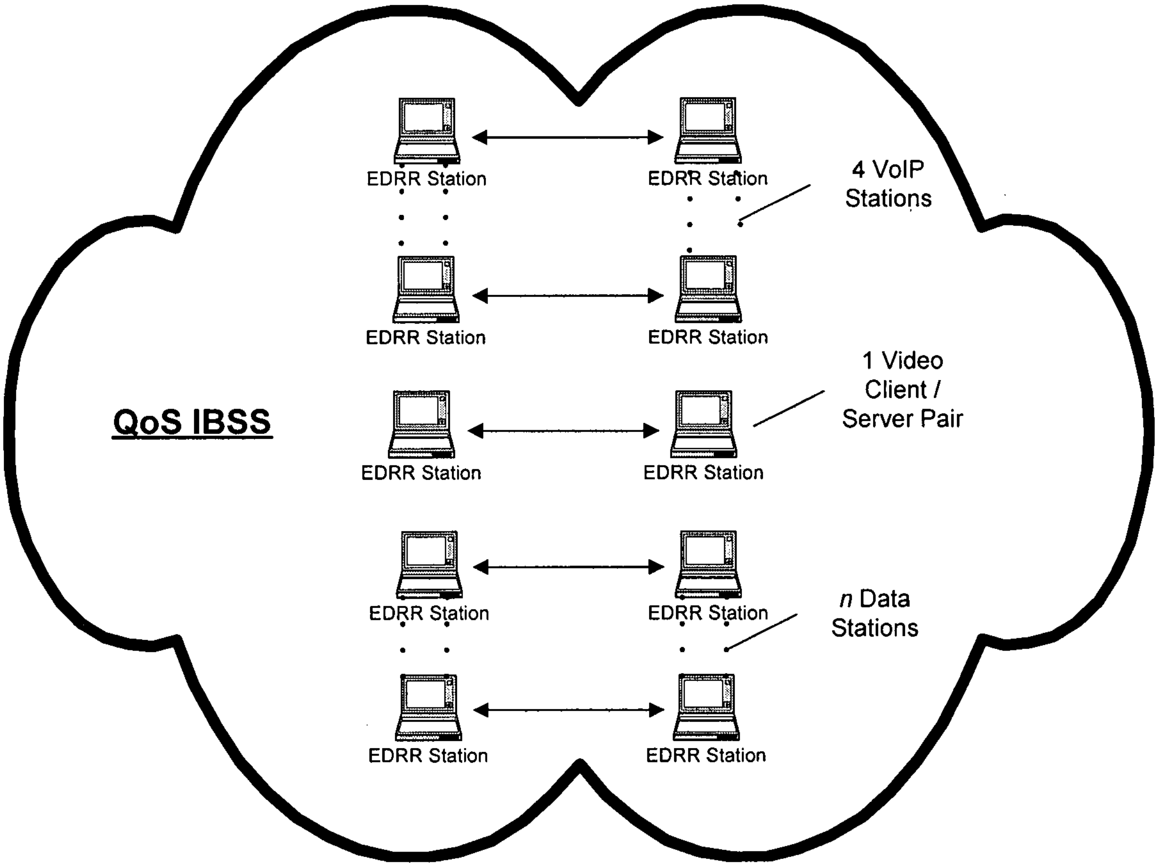

Figure 5.8. Mixed Voice, Video, and Data Traffic in a QIBSS

5.4.2 Results From Scenario 2 Simulations

This section presents the results from our ad-hoc network with mixed voice, video and data traffic. Figures 5.9, 5.10, and 5.11 show the average MAC delay, average jitter, and average throughput for voice packets, respectively. A discussion of the results is provided in Section 5.4.3.

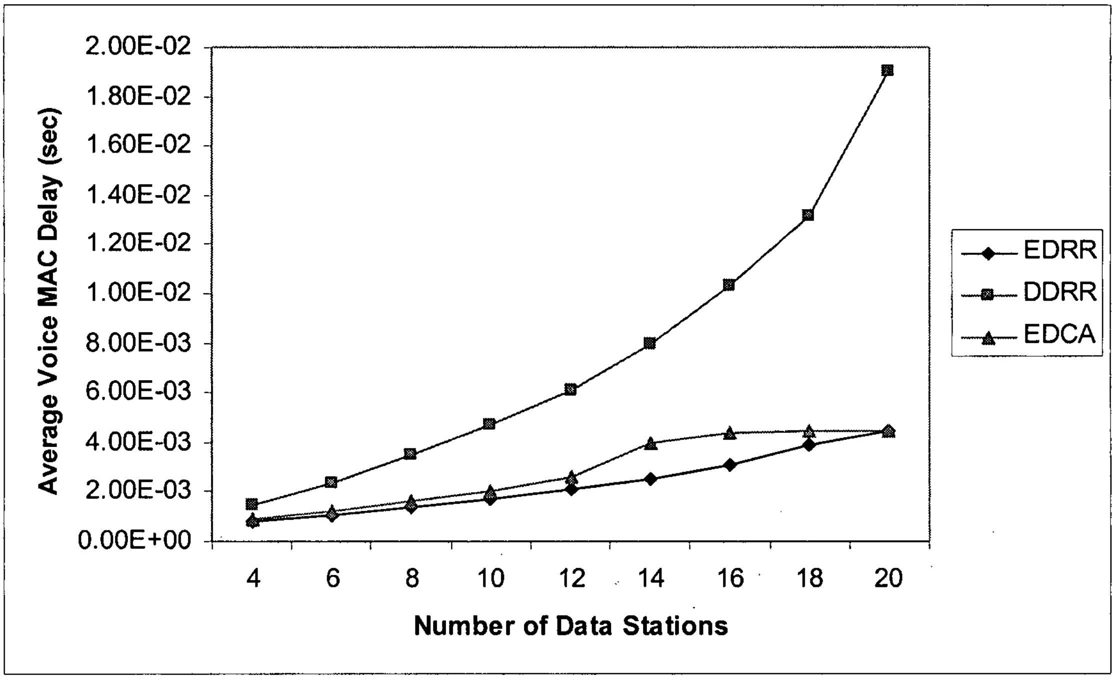

Figure 5.9. Average MAC Delay for Voice Packets under EDRR, DDRR, and EDCA

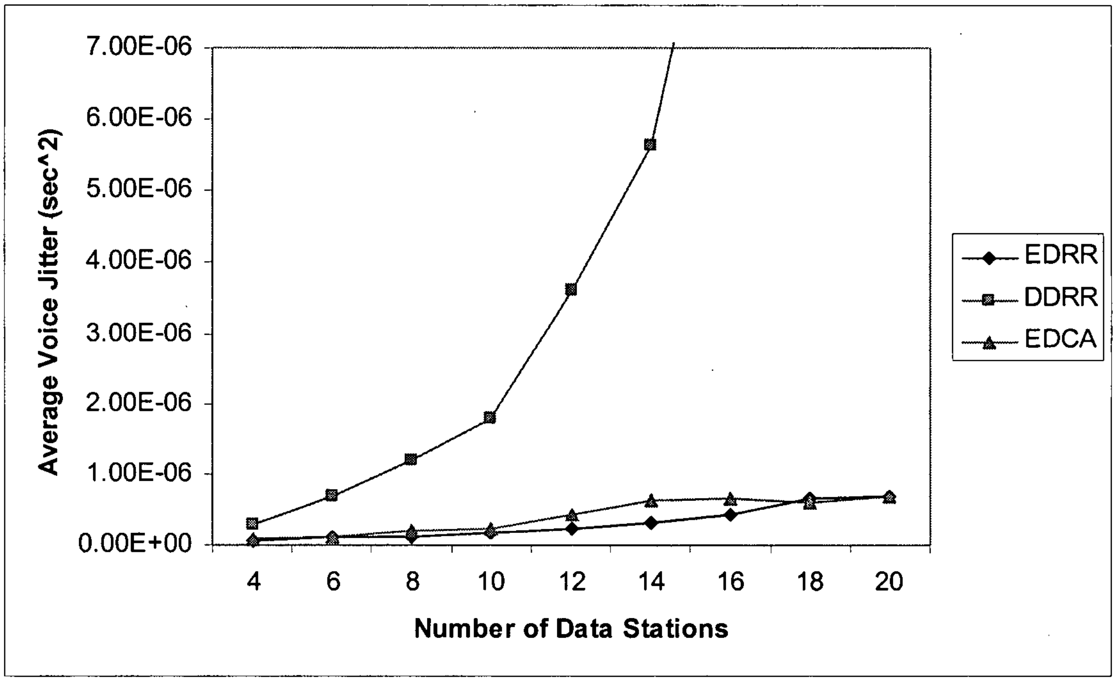

Figure 5.10. Average Jitter for Voice Packets under EDRR, DDRR, and EDCA

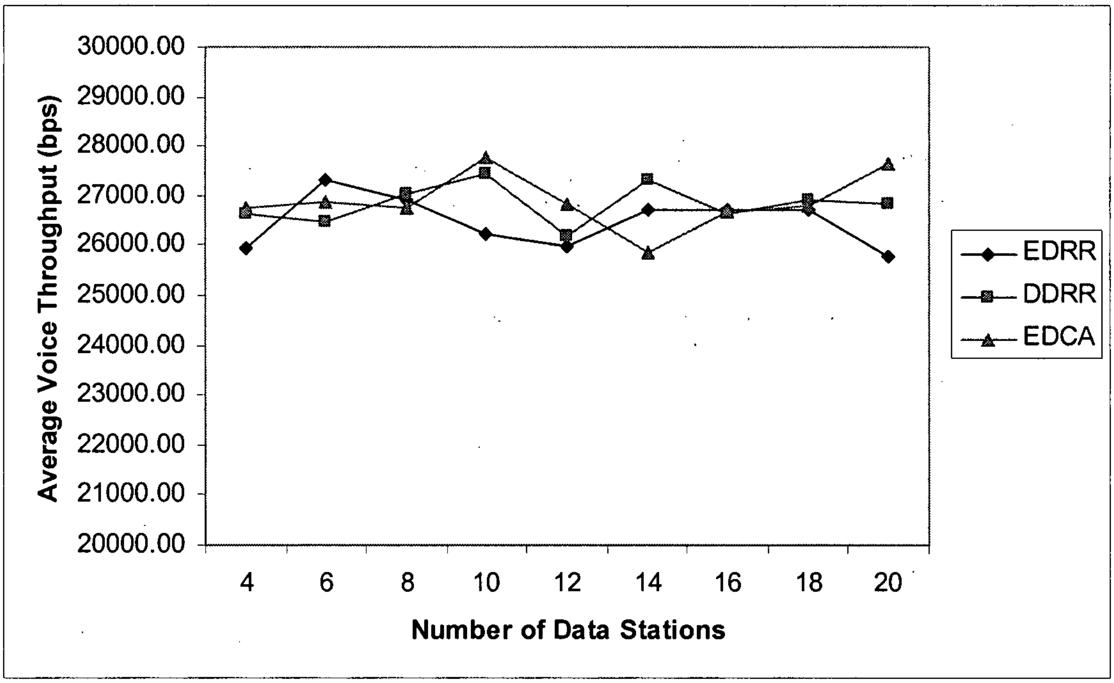

Figure 5.11. Average Throughput for Voice Packets under EDRR, DDRR, and EDCA

Figures 5.12, 5.13, and 5.14, and 5.15 show the average MAC delay, average jitter, average throughput, and packet drop rate for video packets, respectively.

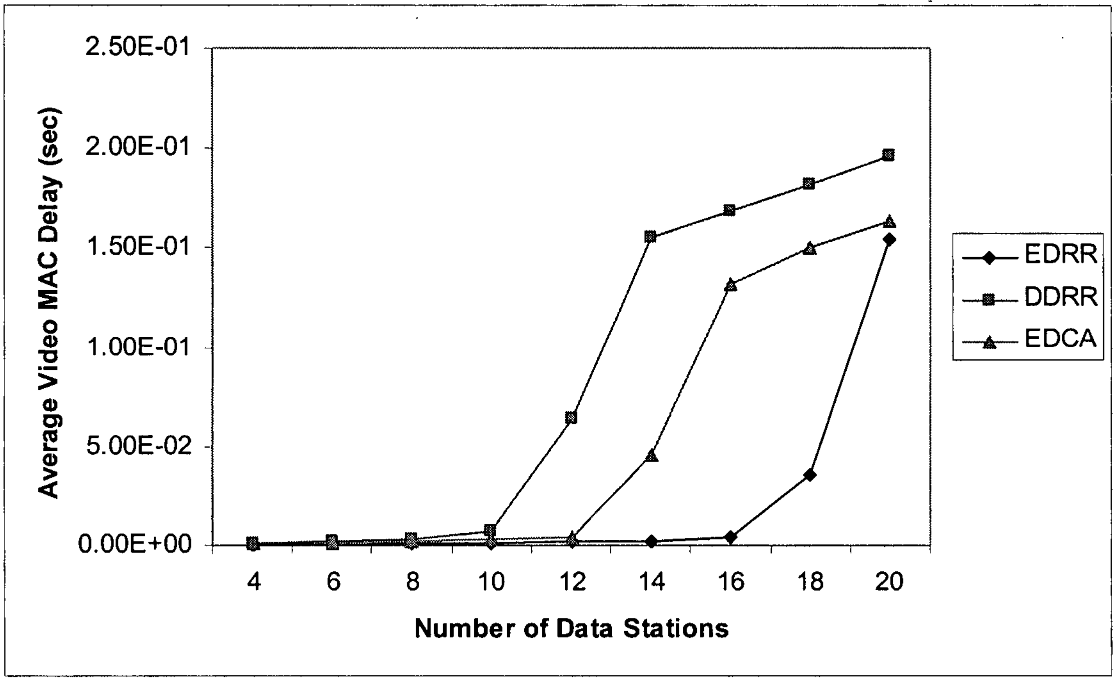

Figure 5.12. Average MAC Delay for Video Packets under EDRR, DDRR, and EDCA

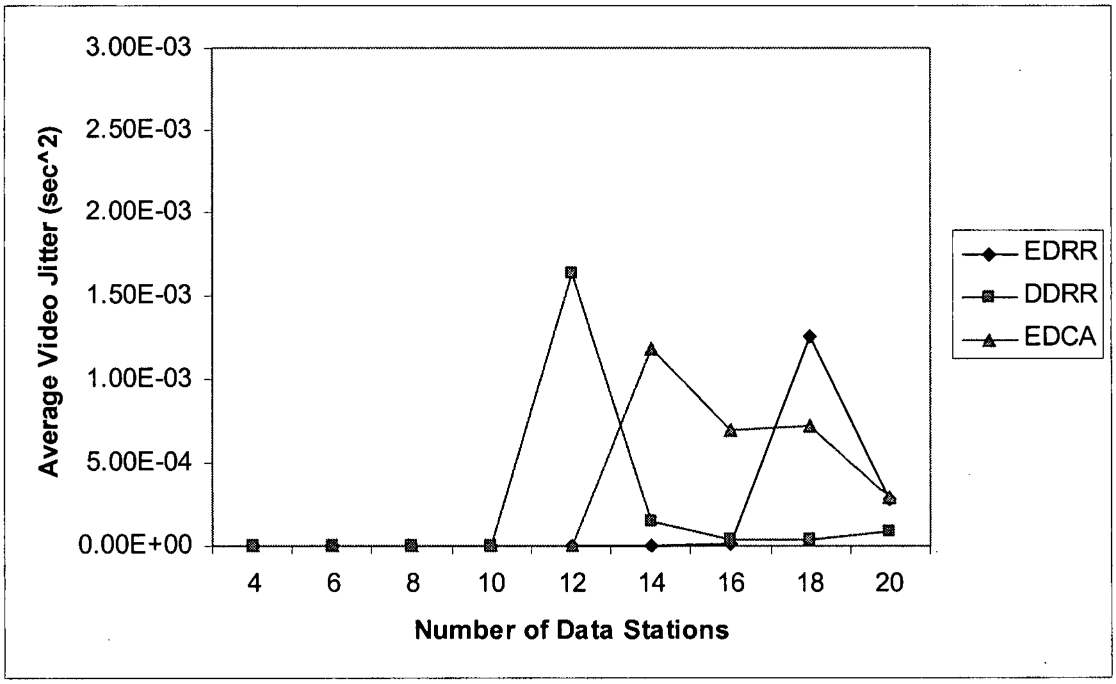

Figure 5.13. Average Jitter for Video Packets under EDRR, DDRR, and EDCA

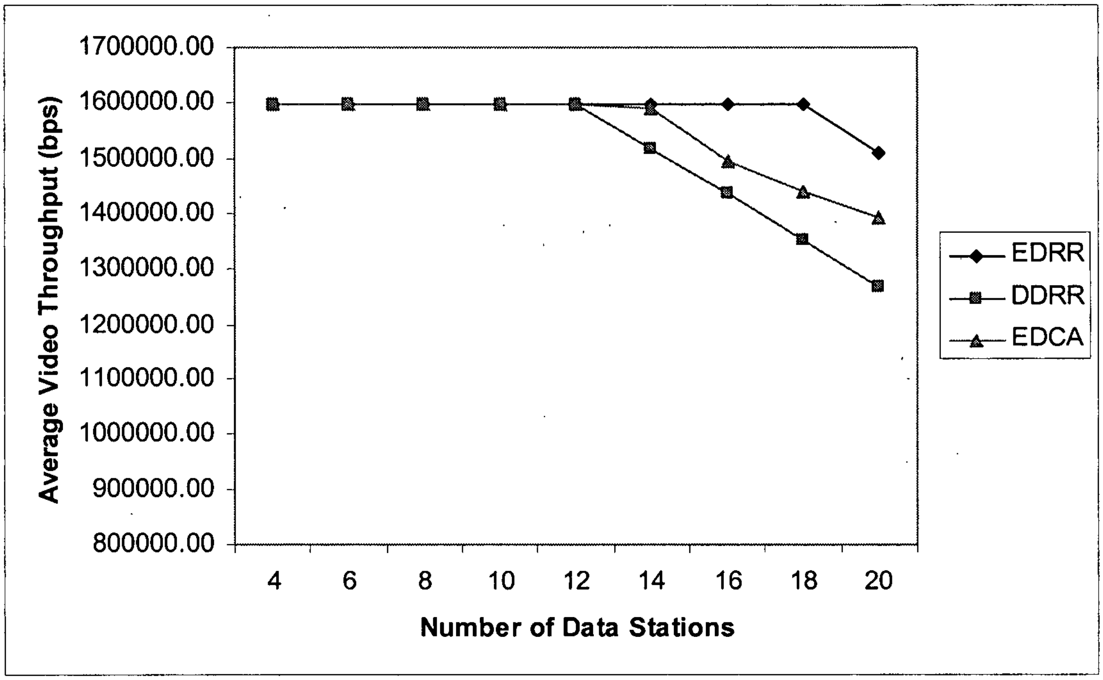

Figure 5.14. Average Throughput for Video Packets under EDRR, DDRR, and EDCA

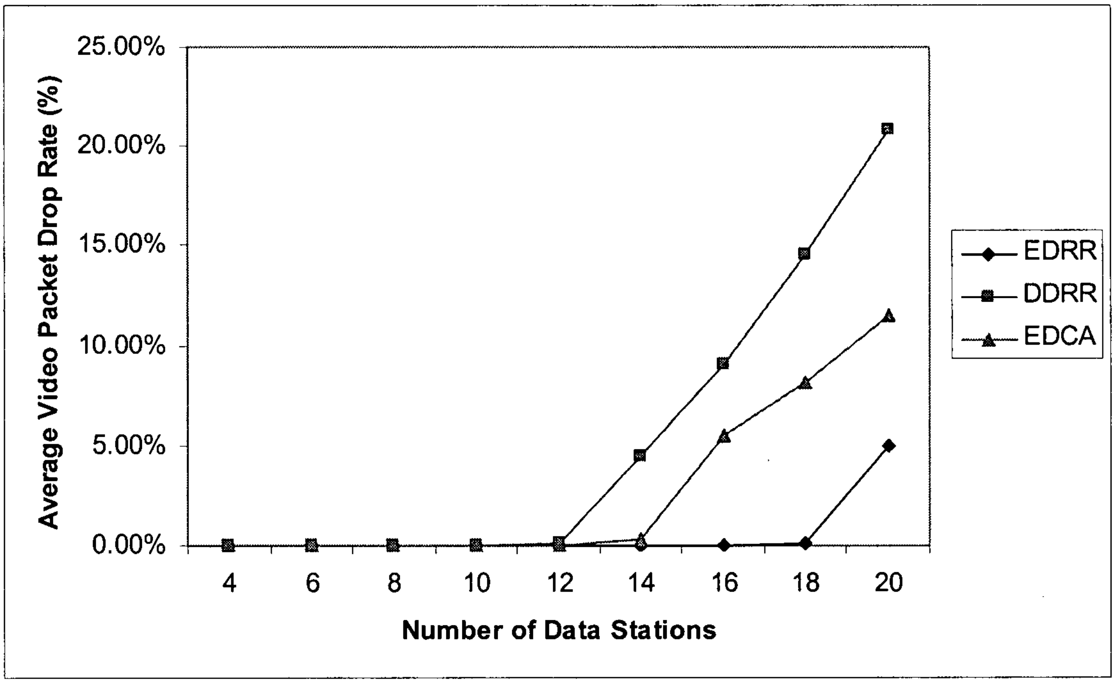

Figure 5.15. Packet Drop Rate for Video Packets under EDRR, DDRR, and EDCA

Figures 5.16, 5.17, 5.18, and 5.19 show the average MAC delay, average jitter, average throughput, and packet drop rate for data packets, respectively.

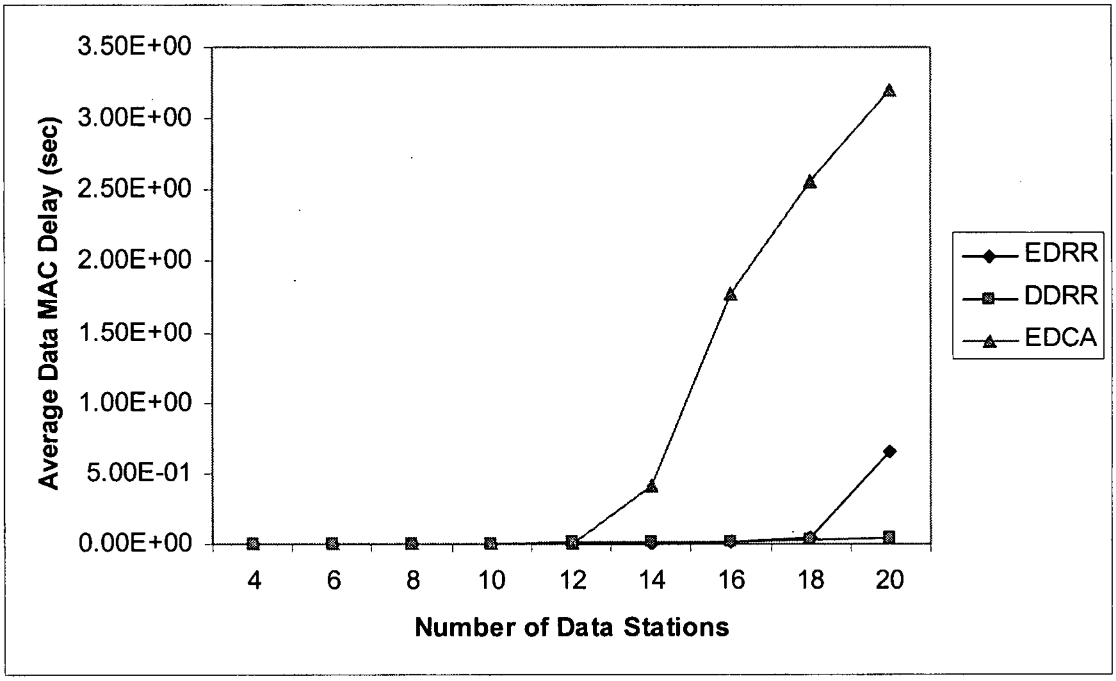

Figure 5.16. Average MAC Delay for Data Packets under EDRR, DDRR, and EDCA

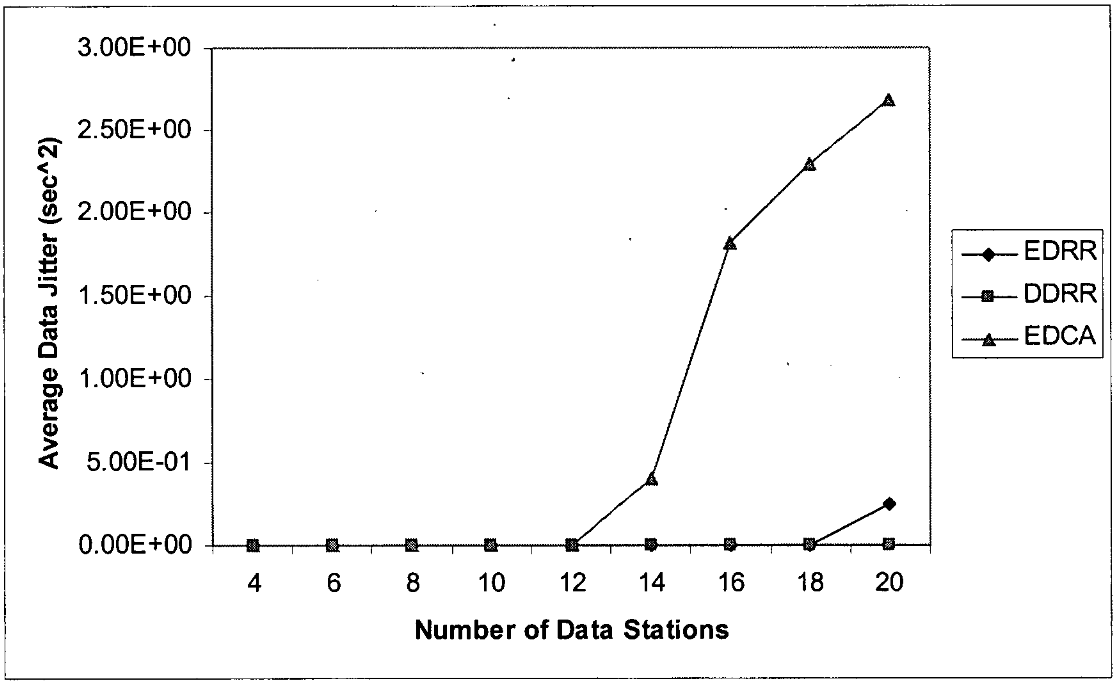

Figure 5.17. Average Jitter for Data Packets under EDRR, DDRR, and EDCA

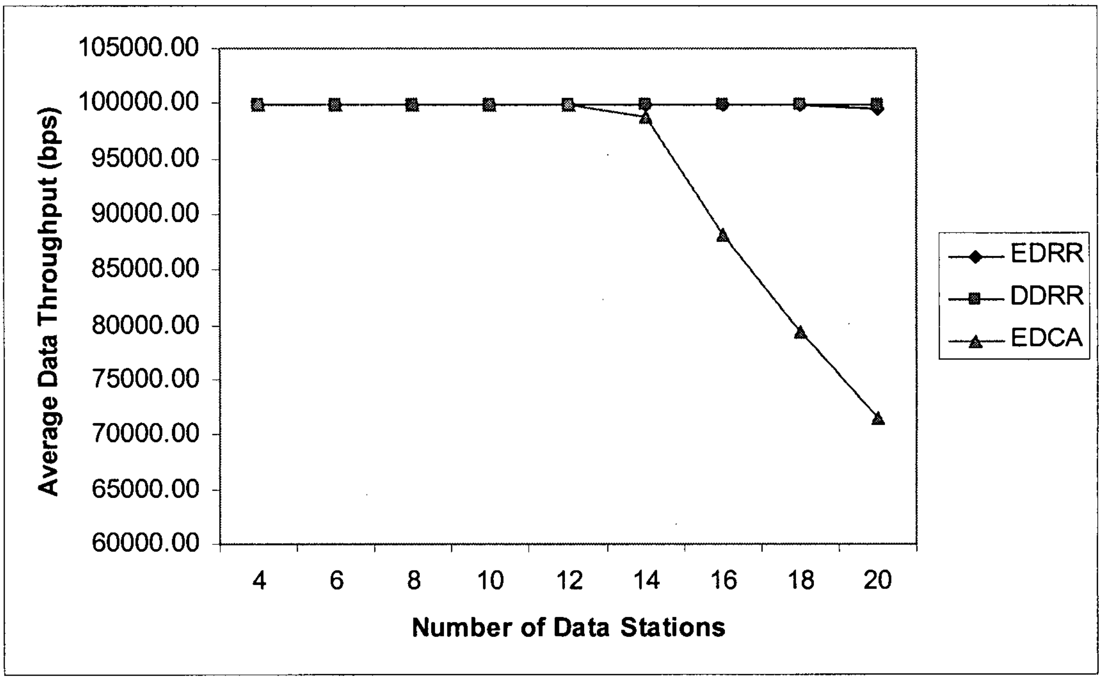

Figure 5.18. Average Throughput for Data Packets under EDRR, DDRR, and EDCA

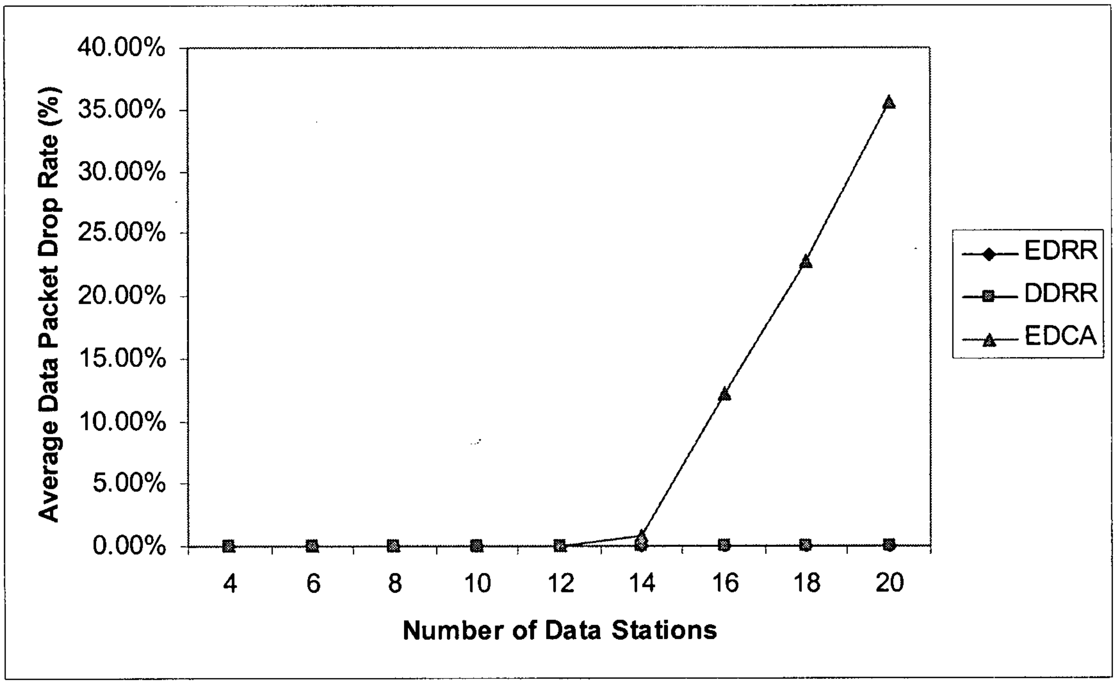

Figure 5.19. Packet Drop Rate for Data Packets under EDRR, DDRR, and EDCA

5.4.3 Discussion of Results From Scenario 2 Simulations

The simulations results from Scenario 2 once again confirm that the proposed EDRR scheme equals or outperforms DDRR and EDCA in relation to nearly all of the measured performance metrics.

Figures 5.9 and 5.10 once again show that due to lack of support for traffic flow priorities, the MAC delay and jitter values suffer for voice packets under DDRR. Due to the support for prioritized traffic in the EDRR and EDCA methods, the MAC delay and jitter values are much better than when the DDRR method is used. In fact, the EDRR and EDCA methods perform nearly the same, with the EDRR method slightly outperforming the EDCA method when the number of data stations increases above 8.

Figure 5.11 shows that the EDRR, DDRR, and EDCA all perform about the same with regards to the throughput of voice packets. Although it may appear that the throughput is lower for the EDRR method, it can be noted that this is not necessarily so because the packet drop rates for each of the methods was found to be zero for all methods. Our simulations showed that for up to 12 data stations, all three methods perform the same with respect to the packet drop rate metric. When there are more than 12 data stations, the EDCA method begins dropping voice packets due to the random nature of its retransmission mechanism. The amount of packets dropped by the EDCA was almost zero and therefore considered negligible. The EDRR and DDRR methods are able to deliver all voice packets by giving priority to retransmissions, which results from a high deficit count for retransmitted voice packets.

Figures 5.12 and 5.13 show that by using prioritized traffic flows, video packets are able to achieve the lowest MAC delay and jitter values under the EDRR and EDCA methods.

While all three methods achieve similar performance for MAC delay for up to 8 data stations, the EDRR and EDCA perform much better after the number of data stations increases. When there are more than 12 data stations, the EDRR method performs significantly better than the EDCA method. The spikes in the jitter values in Figure 5.13 are caused when the load on the network increases beyond the point where all packets can be delivered to their destinations. As the stations begin to drop packets, the jitter increases until the number of packets that are being dropped stabilizes. At this point, the jitter begins to drop off to a lower value. This is illustrated by noting when the packet drop rate increases beyond negligible values in Figure 5.15. It should be noted that the EDRR scheme performs the best in terms of MAC delay and jitter with respect to video packets.

The throughput for video packets is the highest when the EDRR scheme is used. The DDRR scheme has no prioritization and therefore performs the worst of the three schemes for this metric. Since the EDCA method has some level of prioritization, the video packets are able to reach their destinations more often; however, this method suffers when retransmissions become necessary. The results for the throughput of video packets are shown in Figure 5.14. As one would guess, when the throughput begins to decrease, the packet drop rate begins to increase. Figure 5.15 shows that when there are more than 12 data stations, the DDRR and EDCA schemes begin to drop packets. The EDRR method, however, is able to deliver all packets to their destinations until there are more than 16 data stations in the network.

While the MAC delay and jitter are not the most important metrics for data packets, it is still important to deliver data packets within a reasonable amount of time and with some regularity. Figures 5.16 and 5.17 show that the EDRR scheme achieves a nice medium between the other two methods when it comes to the MAC delay and jitter for data packets.

Due to the low priority of data packets in our network and the large amount of high priority packets, the MAC delay and jitter for data packets are very large under the EDCA. The DDRR method performs much better than this at the expense of not giving the priority to the voice and data traffic. The EDRR scheme, on the other hand, is able to provide priority to voice and video packets while still providing a small MAC delay and low jitter.

The most important metric when data is concerned is the packet drop rate. While the voice and video streams can tolerate small amounts of packet loss, data must be delivered 100% intact. All three tested methods do a good job getting data packets to their destinations when there are 12 or less data stations in the network. When there are more than 12 data stations, many data packets are dropped under the EDCA. This is due to the fact that the strict prioritization of the EDCA causes the data packets to be choked out by the large number of voice and video packets. As the load on the network increases and collisions become a problem, the data packets are dropped and never reach their destinations. Therefore, as can be seen in Figure 5.19, the packet drop rate increases sharply for data packets under the control of the EDCA once the number of data stations increases above 12. The EDRR and DDRR stations are able to achieve better performance for both throughput and data drop rates due to their use of "credits" for transmitting. As a result, the data packets are delivered to their destination with a near zero drop rate and with the maximum throughput possible for both the EDRR and DDRR schemes. Figure 5.18 illustrates this fact.

5.5 Conclusions From Simulations

As we showed through our simulations, we conclude that the proposed EDRR scheme is useful for enhancing QoS in distributed channel access WLANs. Our scheme gives an encompassing method for maintaining the desirable attributes of both the IEEE 802.11e draft standard as well as the DDRR scheme, while at the same time remaining extremely close to the standard and keeping complexity to a minimum. Through our simulations, we showed that the EDRR scheme gives significant performance gains in all of the measured metrics for a number of different traffic loads that may be applicable to real-life computing scenarios.

Multimedia network traffic demands low delay and jitter values in order to maintain an acceptable level of quality on the receiving end of the stream. Our scheme shows that small MAC delays and jitter values can be achieved for multimedia traffic under different network loads, while at the same time allowing data traffic to traverse the network error-free. The benefits of the EDRR scheme are that the service differentiation of the IEEE 802.11e draft standard is maintained, complexity is minimized, delay and jitter are enhanced for high priority traffic, and overall throughput is increased in comparison to the EDCA and DDRR methods.

Chapter 6

SUMMARY AND FUTURE WORK

With the popularity of WLANs steadily on the rise, the demand for improvements on the existing WLAN standards is more important than ever. Rising to this occasion, the IEEE 802.11 working group chartered the 802.11e task group with the responsibility of enhancing the 802.11 MAC to include bidirectional QoS to support latency-sensitive applications such as voice and video[1]. In addition to this, much research is being done in the area of improving upon the draft of the 802.11e standard.

Providing QoS in WLANs presents many challenges to be addressed by the research community. Important issues to be addressed include categorization and prioritization of packets, providing low latency and jitter for real-time packets, fair sharing of bandwidth between stations, as well as providing low drop rates for error sensitive data packets. The EDCA function that is proposed by the IEEE introduces a method for providing the prioritization of packets in WLANs. The EDCA also provides low latency and jitter for high priority traffic in certain circumstances. On the other hand, high priority packets suffer when there are too many collisions on the channel, resulting from long contention windows and backoff times (binary exponential backoff). Low priority packets suffer in a network that contains to many high priority sources, causing the packets to be choked out by the high priority packets.

In this thesis we proposed a new scheme, Enhanced EDCA Deficit Round-Robin, for improving on the areas where the EDCA has weaknesses. Using an analytical model, we validated the proposed model and its accompanying simulator. We showed through simulations that by changing the way IFS intervals and contention windows are calculated, significant improvement can be gained over the IEEE's EDCA method.

The remainder of this chapter recaps on significance of the work that we did for this thesis. Section 6.1 gives a summary of our contributions and conclusions that we are able to draw from our simulations. We conclude with a number of suggestions for future work in Section 6.2.

6.1 Summary

Addressing the issues that will enable the acceleration of multimedia into the realm of WLANs is one of extreme importance. In order to do this, there is much research to be done to improve on some of the issues that arise from using a wireless channel. The primary objective of this thesis is to address some of these issues through an investigation into medium access schemes for supporting QoS. Through our research, we propose a new scheme for improving on a number of performance metrics that are important for providing end-to-end QoS in 802.11 wireless networks.

By implementing a distributed version of the DRR scheduling method, the EDRR scheme is able to provide QoS for high priority traffic in a distributed manner without compromising the performance of low priority traffic. Performance requirements such as MAC delay and jitter, which are important to time-sensitive applications including VoIP and streaming video, are considered and addressed without compromising the requirements of lower priority data traffic. Our scheme is able to achieve these results through the use of transmission "credits" which ensure the timely delay of high priority traffic through the use service quanta. Another attribute that results of the service quanta is that lower priority will still receive its "fair share" of the channel bandwidth that is proportional to its quantum of service.

Through the use of OPNET's Modeler application, we were able to test a few different methods of wireless channel access methods against each other to see how they stacked up against one another. We performed simulations of two different ad-hoc networks under a number of different traffic loads to see how they performed for a defined set of performance metrics. Our simulations revealed that the EDRR method achieves superior performance for both high priority and low priority traffic with regards to all measured performance metrics.

The information that we have gathered through our research leads us to the conclusion that the EDRR scheme is a viable method for integrating QoS into IEEE 802.11 WLANs. The "fair" distribution of bandwidth, the closeness to the IEEE standard, and the performance improvements over other schemes provide strong evidence that the EDRR scheme is a practical and feasible alternative for improving QoS in 802.11 wireless networks.

6.2 Future Work

Admission control is an important aspect when it comes to providing QoS in wireless networks. Providing some sort of admission control would offer a nice platform for choosing the service quanta that the EDRR scheme relies on for maintaining the "fair" distribution of bandwidth. It would also ensure that the network does not get overloaded and end up failing completely.

Our research focused purely on ad-hoc WLANs. A natural extension of our work would be to simulate EDRR in infrastructure wireless LANs. The EDRR scheme could be used in place of the EDCA scheme for contention-based access within an access point, as well as stations that are communicating through an AP. Investigation into the way that the AP would gain access to the channel in order to maintain the "fairness" would also be of interest.

The retransmission technique that our method uses is the binary exponential backoff algorithm that is used by the 802.11 standard. There are proposals for more efficient retransmission mechanisms in CSMA/CA networks. Studying the way these other mechanisms work in conjunction with EDRR may also be attractive.

EDRR uses scaling factors in order to calculate the IFS and CW parameters. In order to improve performance, these values could be chosen adaptively. There is a trade-off between throughput and the "fairness" of the scheduling algorithm for EDRR. Choosing the scaling factors adaptively will allow the IFS and CW values to be increased during times of high channel contention and decreased during times of low channel activity. This will result in a "happy medium" between fairness and throughput for all possible channel conditions.

Appendix A

OVERVIEW OF OPNET

All simulations for this project were run in Optimized Network Engineering Tools (OPNET) Modeler 10.0. OPNET is the industry's leading environment for network modeling and simulation, allowing you to design and study communication networks, devices, protocols, and applications [37]. OPNET is an event-driven simulation system, which means that time progresses as scheduled events occur. The basic OPNET program includes models for most common network protocols including ATM, IP, Ethernet and IEEE 802.11. OPNET allows users to modify the built-in protocols in order to try new implementations, hopefully improving on the standard model. New models can also be created by building user defined project models, node models and process models. These components make up the hierarchical structure of OPNET.

Models in OPNET are organized in a three level hierarchical fashion. Each level of the hierarchy describes different aspects of the model that is being simulated. The highest level is the Project object, which is composed of Node and Link objects. A node object is made from a Process object. Figure A-1 shows the hierarchical structure of OPNET models.

A project is a graphical representation of a communications network topology. A network topology consists of node objects and link objects. OPNET's project editor provides a geographical context with physical characteristics reflected appropriately in simulation of both wired and mobile/wireless networks.

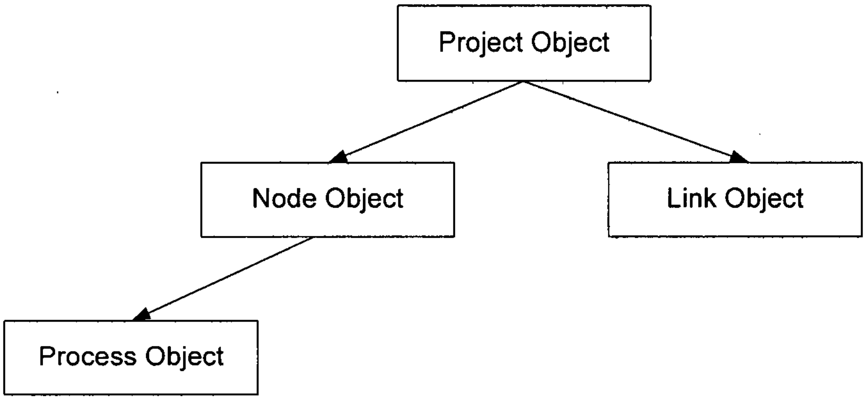

Figure A-1. Hierarchical structure of OPNET models

A node in OPNET allows you to capture the architecture of a network device or system by depicting the flow of data between functional elements, called "modules" [37]. Each module can generate, send, or receive packets to other modules in the node in order to perform its function. Typical modules include applications, protocol stacks, algorithms, and physical resources. Each module is comprised of a process object which defines its function.

Process objects are finite state machines (FSMs) that are used to support the specification of protocols, resources, applications, algorithms, and queuing policies. The process editor provides a graphical representation of the FSM. The states and transitions of the FSM are programmed using standard C/C++ code. OPNET also has an extensive library of functions designed for protocol programming, known as Kernel Procedures.

The standard models that are provided with OPNET make a good starting point for doing research into new protocols where the original model is modified in order to implement a new scheme. One such model that is provided with OPNET is the IEEE 802.11 model, which we used as a starting point for our research.

REFERENCES

[1] P. Roshan, J. Leary, "802.11 Wireless LAN Fundamentals – A practical guide to funderstanding, designing, and operating 802.11 WLANs", Cisco Press, pp. xvii, 2004.

[2] IEEE Std. 802.11a-1999, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 1: High-speed Physical Layer in 5 GHz Band, 1999.

[3] IEEE Std. 802.11b-1999, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 2: High-speed Physical Layer (PHY) extension in the 2.4 GHz band – Corrigendum 1, 1999.

[4] C. Heegard, et.al, "High-Performance Wireless Ethernet", IEEE Communications Magazine, pp. 64 – 73, November, 2001.

[5] IEEE Std. 802.11i-2004, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 6: Medium Access Control (MAC) Security Enhancements.

[6] IEEE Std. 802.11g-2003, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical

Layer (PHY) specifications – Amendment 4: Further High-speed Physical Layer (PHY) extension in the 2.4 GHz band.

[7] IEEE Std. 802.11e-2004, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Draft Amendment D7.0: Medium Access Control (MAC) Quality of Service (QoS) enhancements, January 2004.

[8] "IEEE 802 General Information Website", http://www.ieee.org/portal/index.jsp?pageID=corp_level1&path=about/802std&file=index.xml&xsl=generic.xsl

[9] B.P. Crow, I. Widjala, J.G. Kim, P.T. Sakai, "IEEE 802.11 Wireless Local Area Networks", IEEE Communications Magazine, pp. 116-126, September 1997.

[10] J. L. Sobrinho, A. S. Krishnakumar, "Distributed multiple access procedures to provide voice communications over IEEE 802.11 wireless networks," Bell Laboratories – Lucent Technologies, May 1996.

[11] S. Chung, K. Piechota, "Understanding the MAC Impact of 802.11e: Part 2", Silicon and Software Systems, Commsdesign.com, October 2003.

[12] W. Pattara-Atikom, P. Krishnamurthy, S. Banerjee, "Starvation Prevention and Quality of Service in Wireless LANs," *The 5th International Symposium on Wireless Personal Multimedia Communications*, vol. 3, pp. 1078 – 1082, October 2002.

[13] I. Aad, C. Castelluccia, "Differentiation mechanisms for IEEE 802.11", in *Proceedings of IEEE INFOCOM 2001,* Anchorage, AK, April 2001.

[14] A. Veres, A. Campbell, M. Barry, L. Sun, "Supporting service differentiation in wireless packet networks using distributed control", IEEE *Journal on Selected Areas in Communications,* vol. 19, no. 10, pp. 2081 – 2093, 2001.

[15] A. Banchs, X. Perez, "Distributed weighted fair queuing in 802.11 wireless LAN", in *ICC,* 2002, vol. 5, pp. 3121 – 3127.

[16] N. H. Vaidya, P. Bahl, S. Gupta, "Distributed fair scheduling in a wireless LAN", in *Mobile Computing and Networking Conference,* 2000, pp. 167 – 178.

[17] Xiao, Y. "Enhanced DCF of IEEE 802.11e to Support QoS," In *Proc. IEEE Wireless Communications and Networking Conference* (WCNC), vol. 2, pp. 1291-1296, March 2003.

[18] Zhu, H. and Chlamtac, I. "An Analytical Model for IEEE 802.11e EDCF Differential Services," In *Computer Communications and Networks* (ICCN), pp. 163-168, October 2003.

[19] Robinson, J. and Randhawa, S. "Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 917-928.

[20] IEEE Std. 802.11-1997, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.

[21] J. Zyren, A. Petrick, "IEEE 802.11 Tutorial", YDI Wireless, May 13, 2004.

[22] M. Ergen, "IEEE 802.11 Tutorial", University of California, Berkeley, June 2002.

[23] IEEE Std. 802.11-1999, Information technology – Telecommunications and information exchange between systems- Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.

[24] Bing, Benny, *Wireless Local Area Networks*, Wiley-Interscience, Inc, 2002.

[25] M. G. Arranz, R. Aguero, L. Munoz, P. Mahonen, "Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks", In 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 72-77, 2001.

[26] M. Veeraraghavan, N. Cocker, T. Moors, "Support of voice services in IEEE 802.11 wireless LAN", IEEE Infocom 2001.

[27] A. Grilo, M. Nunes, "Performance Evaluation of IEEE 802.11e", Proceedings of the IEEE PIMRC 2002.

[28] M. Shreedhar, G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin", *IEEE/ACM Transactions on Networking*, vol 4, no. 3, pp. 375 – 385, June 1996.

[29] Kanhere, S. S. and Sethu, H. "Fair, Efficient and Low-Latency Packet Scheduling Using Nested Deficit Round Robin," Proceedings of IEEE Workshop on High Performance Switching and Routing, Dallas TX, 2001, pp. 6-10.

[30] C. Semeria, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines", Jupiter Networks Inc, December 2001.

[31] OPNET Wireless LAN Model User Guide, OPNET Technologies, Inc.

[32] P. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversation", *Bell Syst. Tech Journal,* vol 48, no. 7, pp. 2245 – 2272, Sept. 1969.

[33] G. Bianchi, "Performance analysis of the IEEE 802.11 DCF," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535-547, March 2000.

[34] Mathematica 5.0, Wolfram Research Inc, © 2003.

[35] A. Helal, B. Haskell, J. Carter, R. Brice, D. Woelk, M. Rusinkiewicz, "Any Time Anywhere Computing: Mobile Computing Concepts and Technology", Kluwer Academic Publishers, October 1999.

[36] N. Desai, V. Verma, S. Helal, "Infrastructure for Peer-to-Peer Applications in Ad-Hoc Networks", Computer and Information Science and Engineering Department, University of Florida, 2003.

[37] http://www.opnet.com/products/modeler/home.html