Performance of Voice and Data Transmission Using the IEEE 802.11 MAC Protocol

by

Jessica M. Yeh

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author . . . .

Department of Electrical Engineering and Computer Science

May 24, 2002

Certified by . . . .

Jon Anderson

Senior Staff Systems Engineer

VI-A Company Thesis Supervisor

Certified by . . . .

Professor Vincent W.S. Chan

Director, EECS Laboratory for Information and Decision Systems

M.I.T. Thesis Supervisor

Accepted by . . . .

Arthur C. Smith

Chairman, Department Committee on Graduate Theses

Performance of Voice and Data Transmission Using the IEEE 802.11 MAC Protocol

by

Jessica M. Yeh

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2002, in Partial Fulfillment of the
Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The IEEE 802.11 wireless LAN standard attempts to provide high throughput and reliable data delivery for stations transmitting over a lossy, wireless medium. To efficiently allocate resources for bursty sources, the 802.11 Medium Access Control (MAC) sublayer uses a type of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol called the Distributed Coordination Function (DCF). The MAC protocol also includes an optional polling scheme called the Point Coordination Function (PCF) to deliver near-isochronous service to stations. This thesis analyzes the performance of these two medium access mechanisms under real-time voice and asynchronous data transmissions. Using analytical and simulative methods, the efficiency and capacity of the 802.11 protocol is determined for each type of traffic individually, as well as for a traffic mix of the two types. It is shown that the upper bound of data efficiency for DCF is 65.43% percent when transmitting maximum-sized IP packets at 11 Mbps. Furthermore, due to the difference in packet size of the two traffic types, for each additional GSM voice call (approximately 11 kbps including voice activity) to be supported using DCF, the non-real-time traffic load must decrease by approximately 250 kbps. Voice receives very little real-time Quality of Service (QoS) when using DCF to contend with constantly sending data stations. In order for 802.11 to provide real-time QoS for voice packets despite all levels of asynchronous traffic data load, the PCF mechanism can be used. By only using PCF for voice traffic, voice packets will always take priority over asynchronous data packets and receive the required real-time QoS.

VI-A Company Thesis Supervisor: Jon Anderson
Title: Senior Staff Systems Engineer

M.I.T. Thesis Supervisor: Professor Vincent W.S. Chan
Title: Director, EECS Laboratory for Information and Decision Systems

Acknowledgments

I would like to express my appreciation to my thesis advisors, Jon Anderson at Qualcomm, who supported me through all logistical obstacles, for his mentorship and motivation, and Professor Vincent W.S. Chan at M.I.T. for his continued support and advice, despite project changes.

My appreciation also goes to Rajiv Vijayan and his group at Qualcomm in San Diego for giving me the opportunity to work in such a exciting field. Special thanks to Bruce Collins for providing guidance and direction for the project.

I would also like to acknowledge Jeanne Clark at Qualcomm for all her efforts in setting up OPNET, and Mark Guzzi at Qualcomm for his helpful comments and suggestions in defining the project.

Finally, I thank my family and friends for their continued support and encouragement throughout my work on this project.

Contents

List of Figures

List of Tables

Acronyms

A definition of the listed acronym can be found on the page(s) listed in parentheses.

ACK	acknowledgment (35)
AP	access point (28)
BSS	basic service set (28)
CFP	contention-free period (33)
CP	contention period (31)
CSMA/CA	carrier sense multiple access with collision avoidance (29)
CTS	clear to send (35)
CW	contention window (32)
DCF	distributed coordination function (31)
DIFS	distributed (coordination function) interframe space (30, 31)
DS	distribution system (29)
DSSS	direct sequence spread spectrum (27)
EIFS	extended interframe space (30)
ESS	extended service set (29)
FHSS	frequency-hopping spread spectrum (27)
LAN	local area network (19)
MAC	medium access control (21)
MPDU	MAC protocol data unit (38)
NAV	network allocation vector (30)
PC	point coordinator (33)
PCF	point coordination function (33)

PHY	physical (layer) (27)
PIFS	point (coordination function) interframe space (30, 33)
RTS	request to send (35)
SIFS	short interframe space (30)
TC	traffic class (85)

Chapter 1

Overview

Network computing provide an abundance of resources to an end user on a single computer. Hardware and software applications can easily be shared among multiple personal computers. Other applications such as FTP can enable transfers of files through the network between remotely located machines. Furthermore, the rapid growth of the World Wide Web in recent years brings a wealth of information and services, all conveniently accessible through an Internet connection from a computer in the home. As we become accustomed to the benefits provided by computer networks, there is a growing desire for continuous network connection. Our busy lives demand portable devices that can keep us connected throughout the mobility of our daily lives without the hassles of cables and wires.

Wireless local area networks (LANs) provide much of the desired flexible functionality. Because they do not require an existing wired infrastructure, wireless LANs can be easily created without the need for extensive cable installation or other changes of the existing network. Furthermore, with little difficulties, they can be modified or replaced as needed, providing a convenient possibility for building simple, temporary networks. Users with portable devices may travel anywhere within the basic service area, all the while maintaining a connection to the LAN. Thus, wireless LANs can easily function as an extension of a wired LAN giving additional flexibility to the existing structure.

This thesis studies the performance of the IEEE 802.11 wireless LAN protocol.

The two medium access mechanisms of the protocol are analyzed under real-time voice and asynchronous data loading to determine the effectiveness of the protocol in offering Quality of Service for real-time traffic.

1.1 Background

1.1.1 Wireless Local Area Networks

To an end-user, wireless networks should function almost identically to wired networks. Wireless LANs must have a method of concealing the nature of the physical network and seamlessly allowing for mobility. Especially when contention for limited media resources occurs among several stations, the wireless LAN must be able to fairly and efficiently allocate these resources.

However, problems often arise with the use of a wireless environment, and these issues are resolved in the medium access protocol. For economic feasibility, wireless LAN devices cannot simultaneously listen to the medium while transmitting because they usually have only one antenna available for both sending and receiving. Thus, collision detection algorithms that continuously monitor the medium, such as those used in Ethernet, are much more difficult to implement. When switching between the circuits responsible for sending and receiving, the interface will not be able to perform either task for a certain period of time. This so-called Rx/Tx-turnaround-time places restrictions on the speed of exchange possible in the medium access protocol [10]. Furthermore, due to interference among co-located wireless LANs, the wireless channel experiences higher error rates compared to those of wired channels [10]. Thus, dropped data cannot always simply be attributed to congestion in the transmission medium.

Due to the limits in the range of signal propagation, wireless LANs encounter another issue known as the hidden-node problem. Station A may not be within receiving range of a currently sending station, B, and thus will consider the medium to be idle. Station A may, however, be within range of the receiving station C of the current transmission from B. Any attempts to initiate a new transmission from A to C may corrupt signals from both A and B. Having no means of collision detection, the current senders A and B would both continue to transmit, resulting in a lot of wasted bandwidth.

Wireless LANs have far less bandwidth available than wired LANs. Current commercial wireless LAN products support data rates up to only 11 Mbps. Furthermore, the Federal Communications Commission (FCC) allocates a relatively small amount of bandwidth for the use by wireless LANs. Because bandwidth enhancements are difficult to achieve in wireless LANs [5], this scarce resource must be used efficiently.

1.1.2 Multiaccess Schemes

When several stations share one single medium for transmission, a protocol is needed to control access to this resource. Because stations operate independently, a station will not know when another station needs to use the medium to transmit a packet. By restricting access with a specified protocol, collisions of multiple stations simultaneously attempting to transmit can be reduced. In addition, various techniques can be used to ensure the intended data is transmitted with minimal errors. For these multiaccess networks, the mechanism that governs access to the common medium resides in the Medium Access Control (MAC) Layer, the lower sublayer of the International Standards Organization (ISO) Open System Interconnection (OSI) Basic Reference Model's Data Link Control Layer (Layer 2).

MAC protocols typically can be categorized into several categories: fixed assignment, random access, and dynamic demand assignment. Fixed assignment protocols such as Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), and Code Division Multiple Access (CDMA) devote a fixed amount of resources to each user of the channel. However, these protocols often suffer from inefficient use of the resources. For a network of bursty sources, to accommodate the worst-case traffic load, much of the allocated resources would be wasted during periods of inactivity.

Random access protocols such as ALOHA, Carrier Sense Multiple Access with

Collision Detection (CSMA/CD), and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) rely on stations contending for control of the medium through stochastic means. This enables a network to fairly allocate resources as needed by each station's traffic load. These distributed medium access mechanisms require little coordination and are effective for low or medium load conditions. However, as the traffic load grows, the probability of collision during channel access contention increases, resulting in longer packet delays and throughput well less than 100% [5].

Dynamic demand assignment protocols such as Token Ring, Packet Reservation Multiple Access (PRMA), and Demand Assignment Multiple Access (DAMA) attempt to combine the deterministic behavior of fixed assignment with the flexibility of random assignment. With the expense of more coordination, better performance under higher traffic loads can be achieved.

For wireless packet data networks, fixed assignment protocols seem unsuitable because they lack the adaptability in allocating resources and allowing frequent configuration changes. Demand assignment protocols are often difficult to implement for wireless networks due to some of the requirements to accommodate mobility. For example, token-based schemes rely on knowledge of the current network topology so each station knows which stations are its current neighbors, which may be a tedious task to maintain in mobile configurations. Wireless networks need a protocol to accommodate the possibility of a constantly changing network topology. Thus, random assignment methods seem the ideal choice to allow for free movement by the mobile device. However, the tradeoff for flexibility is a non-deterministic behavior that cannot always guarantee support for a desired Quality of Service.

1.1.3 Quality of Service

Quality of Service (QoS) refers to the ability of a network to effectively provide a certain level of support for selected classes of network traffic. With QoS, the LAN features a more predictable network service by supporting dedicated bandwidth, improving loss characteristics, and setting traffic priorities across the network. In this way, QoS provides more guarantees for transmissions across the network.

Quantitatively, QoS can be described with parameters such as frame error rate, latency, jitter, and capacity. Frame error rate is the amount of frames lost or corrupted en route through the network. Latency describes the delay experienced by the traffic as it travels across the network, and jitter represents the variation of delay experienced by different frames in a stream of traffic. Capacity is the amount of useable bandwidth available for the session to transmit data. Some guarantees regarding ordered delivery of packets may also be assumed for a certain QoS.

With networks equipped to support different levels of QoS, various types of traffic can experience different forms of reliable delivery over the same network. For example, data applications and other asynchronous types of data require bandwidth for efficient transfer of large amounts of data with little packet errors, while being able to tolerate latency and jitter. On the other hand, real-time data such as voice or video need a dedicated amount of bandwidth with short latency, low jitter, little packet loss, but not necessarily completely error-free transmission. With well-designed QoS support, a network can allocate resources to perform a high-quality voice or other time-critical transmission while maintaining efficient asynchronous data traffic flow.

1.2 Project Overview

For this project, the performance of the IEEE 802.11 MAC protocol in offering QoS for various types of traffic is evaluated. Specifically, the two medium access mechanisms, the random assignment Distributed Coordination Function (DCF) and the dynamic demand assignment Point Coordination Function (PCF), are analyzed.

1.2.1 Related Studies

Previous studies have been conducted to model the performance of the IEEE 802.11 MAC protocol. Several papers investigate the performance of the Distributed Coordination Function of the protocol in an ad hoc network under asynchronous data traffic [3, 10]. These studies evaluate how certain tunable parameters of the standard such as packet size, Request To Send/Clear To Send (RTS/CTS) threshold, and fragmentation threshold affect the network throughput and delay. Through simulation, it is shown that with low channel error rates, a reasonably high channel efficiency can be achieved.

A study by Kopsel et al. [5] compares the performance of the Distributed Coordination Function with the Point Coordination Function under real-time traffic requirements. They modeled the load as a dual-source mix of voice and asynchronous data traffic and determined that DCF works well under low load conditions, but experiences throughput deterioration under high load conditions due to the increased time needed to contend for the channel. Meanwhile, the centralized-control protocol, PCF, works well under high load scenarios by optimizing channel bandwidth utilization and decreasing packet wait-time, though there is often high overhead due to unsuccessful polling attempts.

1.2.2 Project Objectives

This project investigates the performance of real-time traffic over DCF and PCF of the IEEE 802.11 MAC protocol. The primary objectives of this study include examining the throughput and capacity performance of the 802.11 MAC layer with respect to the following traffic loads: asynchronous data users, users demanding a real-time QoS, and a combination of these two user types. The ability of the 802.11 MAC protocol to deliver the QoS required of real-time traffic, as well as the degradation to asynchronous data throughput caused by supporting real-time traffic are studied.

1.3 Introduction to the Following Chapters

This chapter has introduced some of the issues of consideration in designing wireless LAN protocols. Different types of multiaccess schemes have also been described. Fixed assignment protocols guarantee a fixed amount of resources to each user of the channel, but may suffer from inefficiencies due to resources allocated to idle users. Random access protocols rely on contention for access among many users to allocate resources as needed by each station's specific traffic load, but may experience packet collisions, especially at high traffic loads. Dynamic demand assignment protocols combine the advantages of the two above protocol types, and with more coordination, may achieve better performance under high traffic loads. With these issues in mind, the different multiaccess schemes of the IEEE 802.11 MAC protocol are evaluated.

Chapter 2 gives a brief summary of the IEEE 802.11 wireless LAN standard. The basic architectural components of 802.11 are introduced, and the MAC protocol is explained. This chapter also describes the different processes by which stations may access the wireless medium and introduces the sequence of frame exchanges which may occur between stations.

Chapter 3 presents a theoretical analysis of both access mechanisms of the 802.11 MAC protocol. The overhead of the DCF mechanism is analyzed in terms of data efficiency, and an upper bound on network throughput is derived. Similar values are also calculated for the PCF mechanism, and the results are compared.

The simulation study is introduced in Chapter 4. The scope and design of the simulation is presented, and relevant assumptions are explained. The basic architecture of the network being simulated is described, as well as the simulated user types at each station. This chapter presents the different operation scenarios of the network, and describes the traffic models used. Metrics used to evaluate performance are also given.

Chapter 5 presents results of the simulation study. Data collected from the simulations is analyzed, and conclusions about the medium access protocols are drawn.

Chapter 6 presents additional mechanisms under consideration for enhancing the current standard protocol with differentiation schemes. The methods of providing service differentiation are described, and the results from simulating these mechanisms are presented.

Chapter 7 concludes with a brief summary of the results of the simulation study and suggestions for further research.

Chapter 2

IEEE 802.11 Standard

The 802.11 standard was devised under the IEEE 802 family of standards for local and metropolitan area networks, which also includes common standards such as Ethernet (802.3) and Token Ring (802.5). Similar to the other standards in the family, the 802.11 standard pertains to the Physical and Data Link layers as defined by the ISO/OSI Basic Reference Model [1].

Defined in 802.11 is the Medium Access Control (MAC) layer, MAC management and protocol services, and three physical layers (PHY) . The physical layers include an infrared baseband PHY, a frequency hopping spread spectrum (FHSS) PHY , and a direct sequence spread spectrum (DSSS) PHY . The goal of the 802.11 task group was to devise a standard to describe a wireless LAN that delivers high throughput, reliable data delivery, and continuous network connections, resembling characteristics previously only available for wired networks [6]. Currently, there have been two flavors of the standard released. 802.11a describes requirements for a high-speed physical layer in the 5 GHz band that offers transmission rates up to 54 Mbps, while 802.11b describes a high-speed physical layer in the 2.4 GHz band, offering rates up to 11 Mbps.

2.1 Architecture

The 802.11 architecture is comprised of several components: the Station (STA), the Access Point (AP), the wireless medium, the Basic Service Set (BSS), the Distribution System (DS), and the Extended Service Set (ESS).

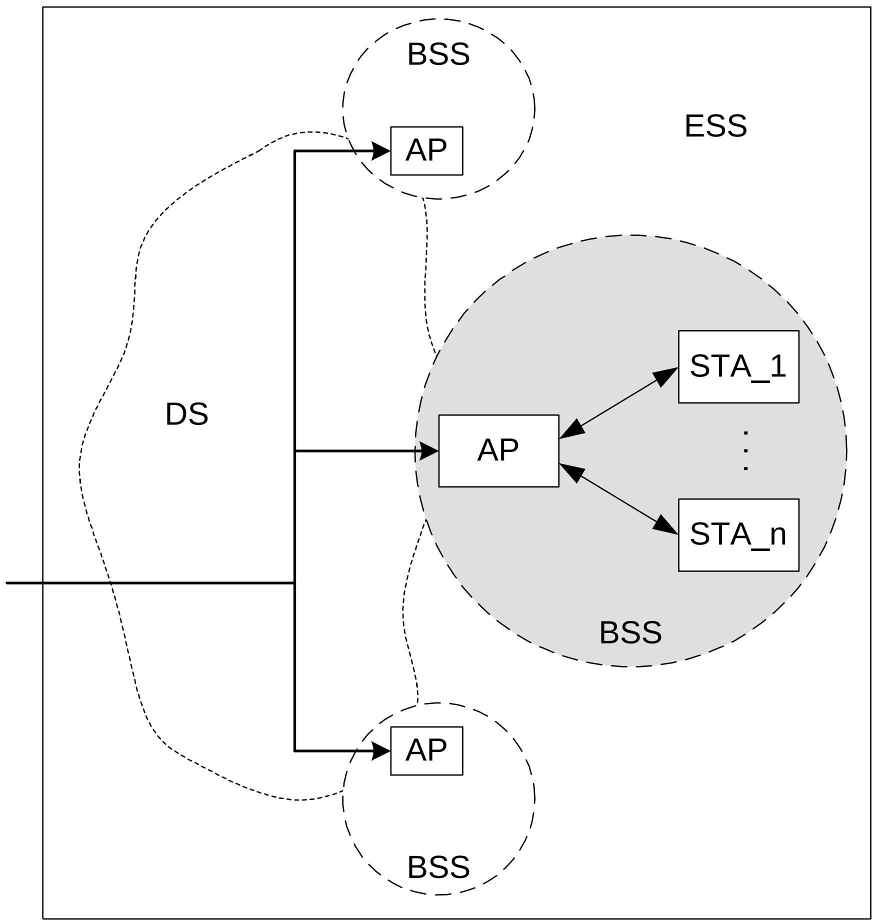

Figure 2-1: 802.11 Architecture

The Station is the component that connects to the wireless medium, typically a PC or a PCI card. The BSS is the basic network architectural component that is composed of two or more stations communicating with each other. If the stations in a BSS communicate directly with one another, they are said to be operating in ad hoc mode. When they communicate through a mediating station, they are said to be in infrastructure mode, with the mediator being known as the AP . The AP is a specialized station that can also connect a BSS to another wired or wireless network. The means by which APs communicate with each other is through an abstract medium known as the DS . This can be either a wired network such as Ethernet, or another wireless network. When several different BSSs comprise a network, they, together with the DS, form an ESS .

2.2 Frame Format

The general MAC frame format specifies a set of fields that are present in a fixed order in all MAC frames. The general MAC frame format is shown in Figure 2-2. With the exception of the Address 4 field, all depicted fields occur in all MAC data frames. The Address 4 field is only used if the wireless network is being used to implement the DS. Other fields, such as Address 2, Address 3, Sequence Control, Address 4, and Frame Body, may be omitted in certain other frame types. (Please reference [1, 6] for definitions of each field and detailed descriptions of the formats of each individual frame type.)

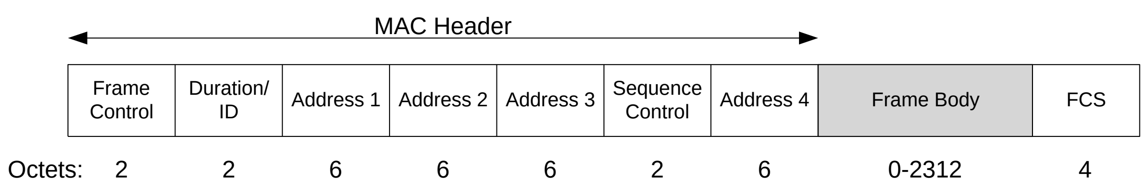

Figure 2-2: IEEE MAC Frame Format

2.3 The MAC Protocol

The IEEE 802.11 MAC layer uses a type of random assignment protocol known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with binary exponential backoff. However, it also provides an optional demand assignment scheme in order to deliver near-isochronous service to stations [6]. Both 802.11a and 802.11b use the same MAC protocol.

In CSMA, the physical layer of a station will perform carrier sensing by listening to the medium to ensure that another transmission is not already in progress before beginning its own transmission. In addition to the physical carrier sense mechanism provided by the physical layer, 802.11 also uses a virtual mechanism in an effort to avoid collisions on the wireless medium. A value in the network allocation vector (NAV) maintained by the MAC in each station indicates to the station how much longer the medium will be busy. This value is updated from duration values found in all transmitted frames. Thus, each station decodes the MAC header of every frame it hears to keep track of network activity.

2.4 Medium Access Mechanisms

The 802.11 protocol describes two medium access mechanisms: the random access Distributed Coordination Function (DCF) and the demand assignment Point Coordination Function (PCF). Five timing intervals that control access to the shared wireless medium are used to implement the two access mechanisms.

2.4.1 Timing Intervals

Figure 2-3 shows the relative lengths of the timing intervals. The shortest interval is the short interframe space (SIFS), which is the separation of frames within a transmission sequence of the frame exchange protocol. A slightly longer interval is the slot time. The PCF interframe space (PIFS) is equal to one SIFS plus one slot time, and the DCF interframe space (DIFS) is equal to one SIFS plus two slot times. The extended interframe space (EIFS) is much longer than the DIFS, and is used to allow stations to regain timing synchronization with the rest of the network when a transmission is received in error. The duration of the basic timing intervals are specified according to the particular physical layer being used.

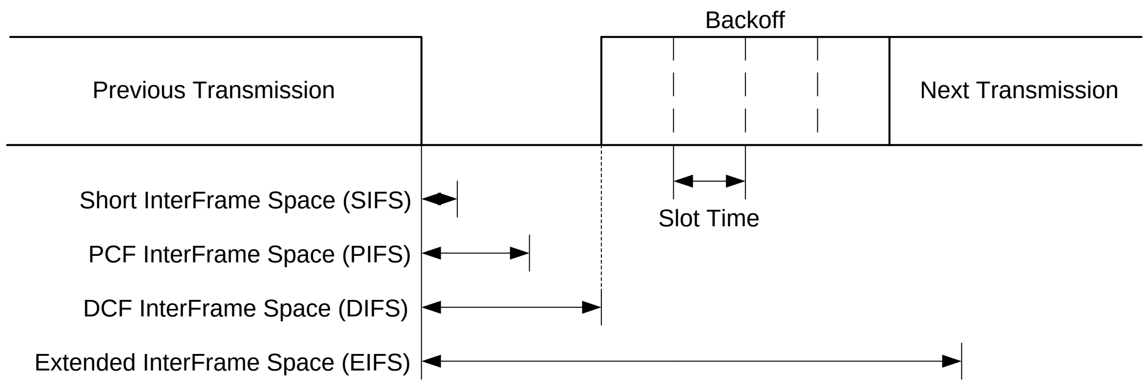

Figure 2-3: Basic Access Mechanism

2.4.2 Distributed Coordination Function (DCF)

The DCF is the basic mechanism that controls access to the wireless medium. All 802.11 stations are required to support DCF services. The period during which the DCF operates is referred to as the Contention Period (CP). After receiving a request for transmission from higher layer protocols, the MAC will check both physical and virtual carrier sense mechanisms. Once the medium is determined to be idle by both sensing mechanisms simultaneously for an interval of DIFS (or EIFS if the previous transmission contained errors), the MAC may begin transmitting the frame. If the medium is determined to be in use during the DIFS-interval, the MAC will increment the retry counter associated with that frame and defer until an idle DIFS-interval to begin backing off. Transmission of the frame can begin only when the backoff timer has expired.

2.4.2.1 Backoff

In order to prevent stations deferring to a transmission from all attempting to send data immediately following completion of the current transmission, the protocol requires stations to perform a binary exponential backoff. After sensing that the medium is idle for a DIFS-interval, a station wishing to transmit a frame will randomly select a deferral value to use as its backoff timer. The backoff timer is selected

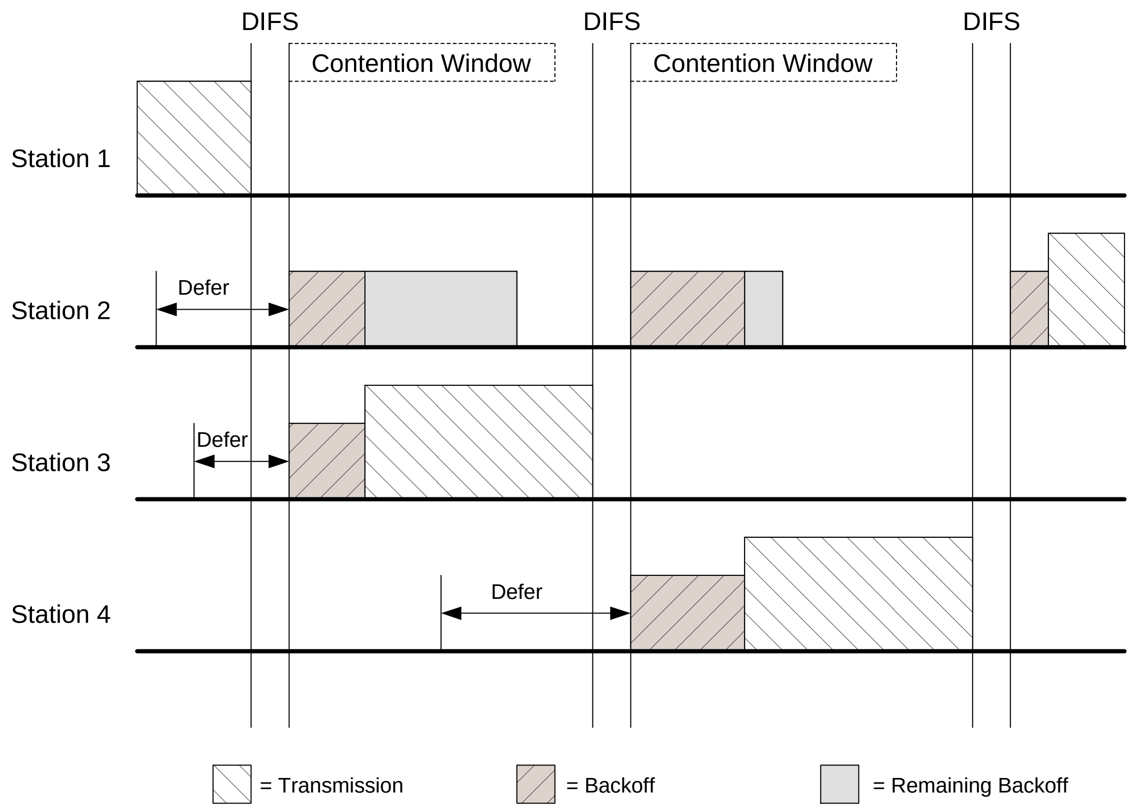

Figure 2-4: Binary Exponential Backoff

from a uniform distribution over a range known as the contention window (CW). This timer value is decremented by one for each slot time the MAC determines the medium to be idle following the idle DIFS-interval.

Should the medium become busy during backoff, the backoff timer will suspend countdown. Once the medium again becomes idle for a DIFS-interval, the station will resume counting down the backoff timer from the value when it was last suspended. The station only transmits the frame when its backoff timer expires. Figure 2-4 shows an example of how the backoff procedure works. To prevent one station with a lot of traffic from consuming all the bandwidth of the wireless medium, after a successful transmission, the station must perform backoff using a minimum-sized contention window before attempting a subsequent transmission.

If the transmission is unsuccessful (i.e. no ACK is received), a collision is assumed to have occurred (regardless of whether this actually happened). The contention window size doubles (unless it has already reached maximum size), a deferral value is selected using the new contention window, and the backoff timer begins counting down once more. The process continues until the transmission is successful, the maximum specified retry limit is reached, or the transmission is cancelled by higher layer protocols. When a successful transmission is completed, the contention window returns to its minimum size. The specific physical layer being used determines the minimum and maximum size of the contention window.

Due to the combination of contention and backoff employed by DCF, stations may experience extremely long wait-times for access to the medium. This possibly long delays as well as wide variation in delay times may be detrimental to real-time traffic. Thus, to support time-bounded traffic, the 802.11 MAC protocol also includes a centralized mode, the Point Coordination Function (PCF), that is governed by a demand assignment scheme.

2.4.3 Point Coordination Function (PCF)

The PCF is an optional mechanism that uses a poll and response method to eliminate contention for the medium. In this centrally controlled mechanism, the point coordinator (PC) located in the AP controls access to the wireless medium. The PC gains access to the medium using procedures similar to those used in DCF. However, instead of waiting for a DIFS-interval, it is only required to wait a PIFS-interval before determining the medium is idle and taking control of the medium. Once the PC has acquired the medium, it sends a beacon frame notifying stations of the beginning of the period of PCF operation known as the Contention-Free Period (CFP) . The beacon contains the maximum expected duration of the CFP, which stations use to update their NAVs.

During the CFP, the PC delivers frames to stations while also individually polling stations that have previously registered on the polling list requesting contention-free service. Each station can send one data frame for each CF-Poll received. By setting an appropriate CFP repetition interval, this mechanism can guarantee a bounded

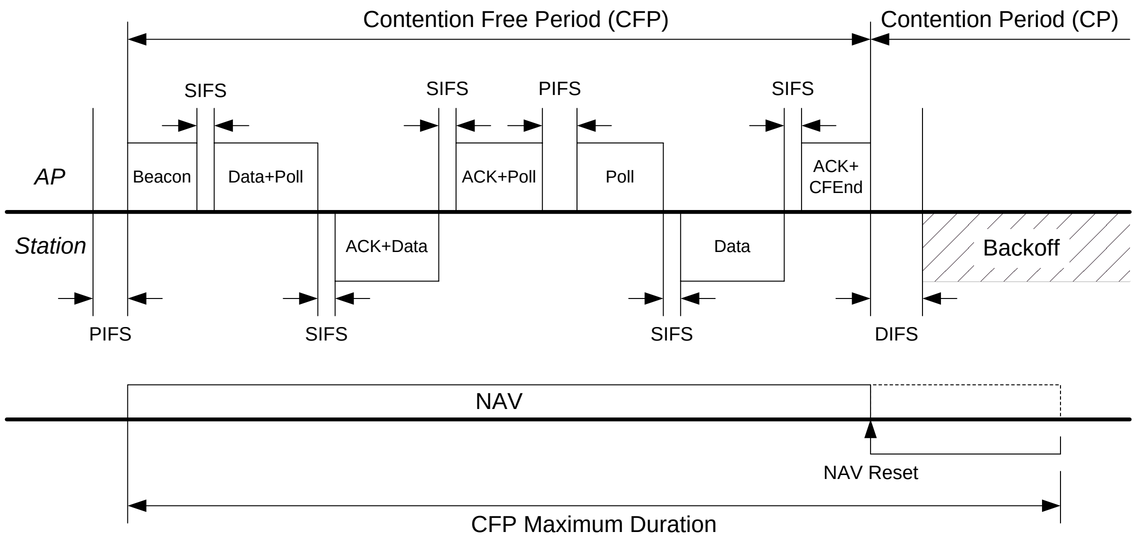

Figure 2-5: PCF Operation during the Contention Free Period (CFP)

delay for transmission of packets arriving at stations that have requested this service.

To maintain control of the medium during CFP, the PC ensures that the interval between frames is no longer than PIFS. If the PC does not receive a response to a data transmission or a CF-Poll within a period of SIFS, it will transmit its next frame before a PIFS concludes. Figure 2-5 depicts an example of possible frame transmissions during a CFP. The end of the CFP is announced when the PC sends a contention-free-end (CF-End) frame. With this frame, stations reset their NAVs and may begin competing for access to the medium under normal DCF methods.

2.4.4 Concurrent Operation of DCF and PCF

Because the PCF mechanism uses DCF methods to obtain control of the medium, it is not required that all stations support PCF services. The PC uses the PIFS interval to operate concurrently with the DCF and gain access to the medium to begin the PCF. Because the PIFS is shorter than the DIFS (used by the DCF), the PC is considered to have a higher priority to access the medium.

Parameters governing the concurrent operation of DCF and PCF, such as the CFP

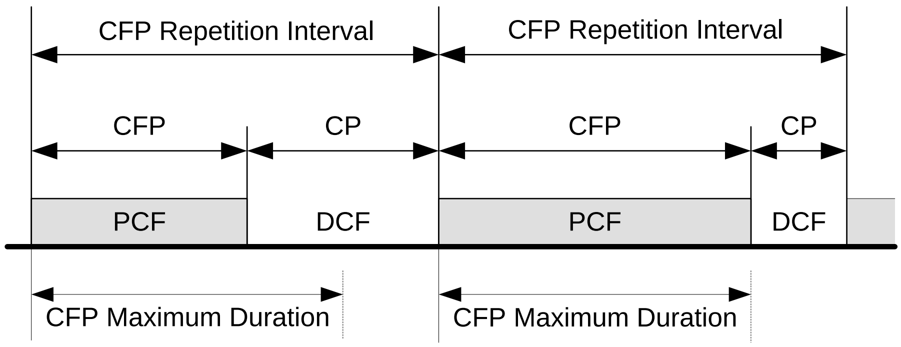

Figure 2-6: DCF and PCF Superframe Structure

Repetition Interval and the CFP Maximum Duration, can be specified to provide a certain QoS. When both DCF and PCF services are desired, durations of Contention Period and Contention-free Period alternate, as illustrated in Figure 2-6.

2.5 Basic Frame Exchange

The protocol requires that the minimal exchange between two stations consists of two frames. A data frame is sent from the source to the destination, and an acknowledgment (ACK) is returned from the destination to the source, indicating successful receipt of the data frame. Figure 2-7 illustrates a basic frame exchange. This data frame and ACK exchange is an atomic unit of exchange between two stations using the MAC protocol and cannot be interrupted by any other station.

To alleviate the problem of hidden nodes in the network, a station also has the option in the basic protocol of using two additional frames, as depicted in Figure 2-8, to notify other stations of the upcoming frame transmission so they delay their own transmissions. The source station sends a request-to-send (RTS) frame , and in response, the destination station sends a clear-to-send (CTS) frame . Upon receipt of the CTS, the source proceeds to send the data frame as above. If the destination correctly receives the frame, it sends an ACK, completing the protocol. This four-frame exchange is also an atomic unit that cannot be interrupted by any other station.

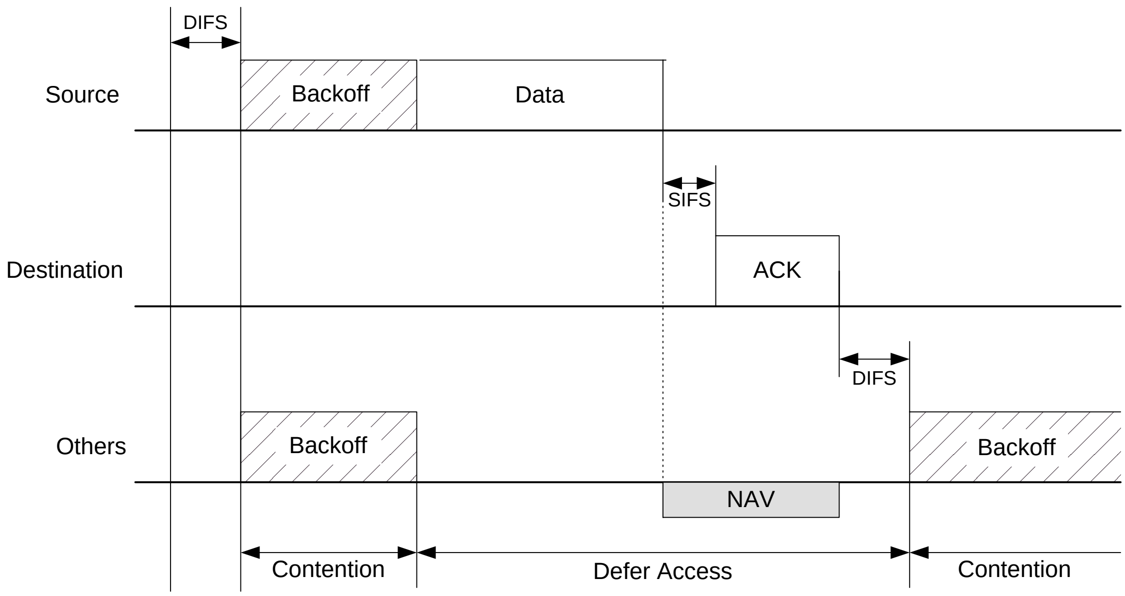

Figure 2-7: Basic Frame Exchange

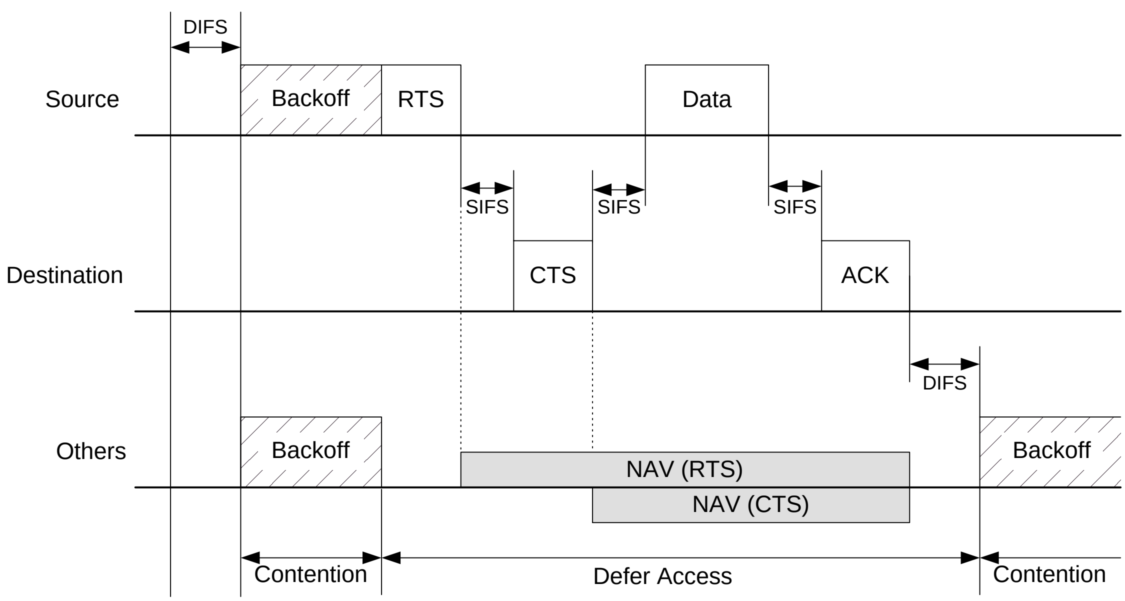

Figure 2-8: Frame Exchange Using RTS/CTS

Chapter 3

Theoretical Analysis

In this theoretical analysis of the 802.11 protocol, the efficiency of this protocol in using the wireless medium is determined. Data efficiency is defined as the percentage of total time used for successful transmission of data that the channel is occupied by the actual data. The data efficiency is analyzed for each of the two access mechanisms of 802.11. From the data efficiencies determined, an upper bound on the throughput is derived.

3.1 Analysis of the DCF

3.1.1 Data Efficiency

The data efficiency of the DCF mechanism in the 802.11 protocol can be determined by analyzing the sequence of events that occurs for a basic frame exchange over the wireless medium (illustrated in Figure 2-7). Propagation delay is assumed negligible and is ignored in this analysis. For simplicity, it is also assumed that RTS/CTS is not used. For a successful transmission, the following sequence of events occurs:

1. The medium is idle for a DIFS.

2. The sending station performs backoff.

3. The sending station transmits a packet.

4. A SIFS interval passes.

5. The receiving station transmits an ACK.

The duration of time required for the entire sequence of events can be represented by the sum of the durations of each event.

$$T_{DCF\ sequence} = T_{DIFS} + T_{backoff} + T_{packet} + T_{SIFS} + T_{ACK}$$

T_{SIFS} and T_{DIFS} are simply the inter-frame space timing interval specified by the protocol for the specific physical layer.

For simplicity, a one-stage backoff is assumed where the contention window is always at its initial, minimum size of CW_{min}. Because the backoff value is selected from a uniformly distributed interval from 0 to CW_{min}, the average selected backoff value is $\frac{CW_{min}}{2}$. The average time required for backoff can thus be calculated as

$$\begin{aligned} T_{backoff} &= average\_backoff \cdot slot\ time \\ &= \frac{CW_{min}}{2} \cdot slot\ time. \end{aligned}$$

The time required for transmission of a packet T_{packet} includes time for transmitting the actual data payload bits as well as all necessary MAC and physical layer overheads. (Please refer to the IEEE 802.11 data frame format depicted in Figure 2-2 for the fields of a MAC Protocol Data Unit (MPDU).) MAC overheads consist of the MAC header and the FCS. The entire MPDU (MAC header, payload, and FCS) is transmitted at the channel transmission rate. Physical layer overheads include the PLCP-Preamble and the PLCP-Header. Physical Layer overheads are transmitted at the basic rate [1]. The transmission time of l bits using a transmission rate of R bits per second (bps) is calculated by $\frac{l}{R}$. Thus, the time required to transmit the packet

[1] The basic rate refers to one of the rates in the BSSBasicRateSet. The BSSBasicRateSet is the set of data rates at which all stations in the BSS must be able to receive packets.

can be expressed as

$$T_{packet} = \frac{l_{MAC\ header} + l_{payload} + l_{FCS}}{R_{trans}} + \frac{l_{PLCP-Preamble} + l_{PLCP-Header}}{R_{basic}}$$

Similarly, the time required for transmission of the ACK can be calculated as the time required to send the ACK frame as well as the physical layer overheads. An ACK frame is transmitted at the basic rate [1, section 9.6] so it can be decoded by all stations. The physical layer overheads are as described for the transmission of the packet above. Thus,

$$T_{ACK} = \frac{l_{ACK}}{R_{basic}} + \frac{l_{PLCP-Preamble} + l_{PLCP-Header}}{R_{basic}}$$

To determine average data efficiency, the amount of time spent in the transmission of actual data bits T_{data} must be determined. T_{data} can be calculated by $\frac{l_{payload}}{R_{trans}}$ where $l_{payload}$ is the length of the data payload in bits, and R_{trans} is the transmission rate. Using these formulas, average data efficiency is simply

$$average\ data\ efficiency = \frac{T_{data}}{T_{DCF\ sequence}} \cdot 100\%.$$

3.1.2 Theoretical Upper Bound of Network Throughput

To calculate the theoretical upper bound of network throughput, we assume that no collisions occur and all packet transmissions are successful. The maximum throughput is achieved when, immediately upon completion of one sequence of DCF events, the following sequence begins without allowing any idle periods on the wireless medium. Furthermore, for the upper bound of network throughput, $T_{backoff} = 0$. This may occur because stations have selected a backoff value of 0 or all packets arriving from higher layers arrive at the beginning of the idle DIFS interval and thus are not required by the standard to backoff. The upper bound of network throughput can then be calculated simply as

$$\textit{upper bound (throughput)} = \textit{data efficiency}(T_{backoff} = 0) \cdot \textit{transmission rate}.$$

3.1.3 Calculated Values

Table 3.1 lists the values of protocol parameters used in the theoretical analysis. For this analysis, an 802.11b DSSS physical layer is assumed.

Parameter	**Value**
slot time	20 μsec
SIFS	10 μsec
DIFS	50 μsec
CW_{min}	31
PLCP-Preamble	144 bits
PLCP-Header	48 bits
MAC Header	24 bytes
FCS	4 bytes
ACK Frame	14 bytes

Table 3.1: Protocol Parameters for DCF using a DSSS Physical Layer

Using these values, the data efficiencies and effective throughput are calculated for different packet sizes transmitted at the various transmission rates supported by 802.11b. A basic rate of 1 Mbps is used for these calculations. Table 3.2 shows the calculated data efficiency while transmitting the maximum-sized packet (without encryption) allowed by the IEEE 802.11 protocol. Transmissions of this packet size produce the highest data efficiency of the protocol. The upper bound on data efficiency assumes a backoff of 0 slots while average data efficiency assumes a backoff of $\frac{CW_{min}}{2}$ slots.

Due to packet overheads, on average, only 94.42% of the bandwidth can be used for transmission of actual data bits. As transmission rates increase, the relative overhead from the physical layer and the ACK packet also increase because these overhead bits must still be transmitted at the slower, basic rate. This results in lower data efficiencies at higher transmission rates, with only an average of 65.40% of the bandwidth being used for payload data transmission at an 11 Mbps transmission rate.

Transmission Rate [Mbps]	1	2	5.5	11
Upper Bound on Data Efficiency [%]	95.94	93.24	84.89	74.41
Upper Bound on Data Throughput [Mbps]	0.96	1.86	4.67	8.18
Average Data Efficiency [%]	94.42	90.41	78.71	65.40
Effective Data Throughput [Mbps]	0.94	1.81	4.33	7.19

Table 3.2: Data Efficiencies and Effective Data Throughput for Sending 2304-byte Packets Using DCF

In reality, these values will actually be even lower due to other 802.11 overheads not included in this analysis such as RTS/CTS packets, Beacon packets, and other control packets such as those used for power management and BSS association, and also due to higher layer protocol overheads such as TCP and IP headers. Furthermore, failed transmissions due to channel conditions will also adversely affect throughput. Fragmentation can help to alleviate the frequent losses of packets from channel errors by minimizing the cost of each loss. With smaller packets, fewer data bits would be lost should the packet be corrupted during transmission. However, the tradeoff of using smaller packets is that the data efficiency is also decreased.

Tables 3.3 and 3.4 illustrate some of the effects of packet size on data efficiency and achievable network throughput. Table 3.3 lists the calculated data efficiency for a 1500-byte packet sent at various transmission rates. This packet size represents a maximum-sized IP datagram. Table 3.4 lists the calculated data efficiency and effective network throughput for a 32.5-byte packet. Voice calls using GSM encoding (13 kbps) with 20 msec frames have packets of this size. For both cases, the basic rate is assumed to be 1 Mbps.

Due to the high overheads required in the MAC and physical layer for each transmitted packet, transmissions of small packets can be extremely inefficient. For example, a station transmitting 32.5-byte packets over the wireless network at a rate of 11 Mbps would feel as if the network could support a transmission rate of only 290

Transmission Rate [Mbps]	1	2	5.5	11
Upper Bound on Data Efficiency [%]	93.90	89.98	78.52	65.43
Upper Bound on Data Throughput [Mbps]	0.94	1.80	4.32	7.20
Average Data Efficiency [%]	91.67	85.98	70.64	55.17
Effective Data Throughput [Mbps]	0.92	1.72	3.89	6.07

Table 3.3: Data Efficiencies and Effective Data Throughput for Sending 1500-byte Packets Using DCF

Transmission Rate [Mbps]	1	2	5.5	11
Upper Bound on Data Efficiency [%]	25.00	16.29	7.34	3.94
Upper Bound on Data Throughput [Mbps]	0.25	0.33	0.40	0.43
Average Data Efficiency [%]	19.26	11.73	4.96	2.60
Effective Data Throughput [Mbps]	0.19	0.23	0.27	0.29

Table 3.4: Data Efficiencies and Effective Data Throughput for Sending 32.5-byte Packets Using DCF

kbps. The smaller the packet, the greater the relative overhead, and thus the lower the data efficiency and data throughput.

3.1.4 DCF Contention Among Several Stations

The above analysis of average data efficiency assumes that each packet being transmitted over the wireless medium will experience an average backoff of $\frac{CW_{min}}{2}$ slot time intervals before transmission is attempted. While this is true for each station individually, due to multiplexing that occurs among several stations backing-off, the idle periods of backoff seen on the wireless medium during contention between packet transmissions is actually less than the average backoff of $\frac{CW_{min}}{2}$.

Two stations may each select a backoff value, BK_1 and BK_2, from the uniform distribution between the interval of $[0, CW_{min}]$. Suppose $BK_1 < BK_2$. Station 1's backoff timer expires first, and it transmits its packet. Station 1 then selects a new backoff value for the following packet and begins backing off once again. Meanwhile, Station 2 had also decremented its backoff counter to $(BK_2 - BK_1)$ before Station

1's transmission. Once the medium is idle again following Station 1's transmission, Station 2 continues backing off with the same backoff timer. Contention begins once again. However, this time it is between two stations where one has selected the backoff value from the uniform distribution between $[0, CW_{min}]$, and the second has effectively selected from the uniform distribution between $[0, CW_{min} - BK_1]$.

Overall, the two stations have equal access to the medium because they both contend using identical methods. Thus, for simplicity, it is assumed that the two stations alternately acquire the wireless medium. In the steady state, the duration of the contention period between packet transmission is the final continuous backoff period of a station before it acquires the medium. This final backoff period (BK) is the minimum of two random variables, one selected from the uniform interval $[0, CW_{min}]$ and the second selected from the uniform interval $[0, CW_{min} - \overline{BK}]$ where $\overline{BK}$ is the mean of the final backoff period. The average backoff period between packet transmissions on the medium $\overline{BK}$ when there are two stations contending is thus

$$\overline{BK} = \sum_{i=0}^{31-\overline{BK}} i \cdot \left[\left(\frac{1}{32}\right) \cdot \left(\frac{32 - \overline{BK} - i}{32 - \overline{BK}}\right) + \left(\frac{31 - i}{32}\right) \cdot \left(\frac{1}{32 - \overline{BK}}\right) \right]$$

where the expression between the square brackets represents $Probability\{BK = i\}$.

This expression can be extended for various numbers of stations contending for access to the wireless medium. The MATLAB code used to calculate the average final backoff period between packet transmissions for various numbers of contending stations is included in Appendix A. Figure 3-1 plots $\overline{BK}$ as it varies with the number of stations contending to access the medium. For one station, $\overline{BK} = \frac{CW_{min}}{2} = 15.5$, and it decreases as the number of stations increase.

Because the contention period between packet transmissions seen on the medium becomes shorter as the number of stations increase, a higher data throughput can be achieved. The data throughput gradually approaches the theoretical upper bound where the contention period is 0. Figure 3-2 illustrates how the maximum achievable throughput varies for stations transmitting 1500-byte packets at 11 Mbps.

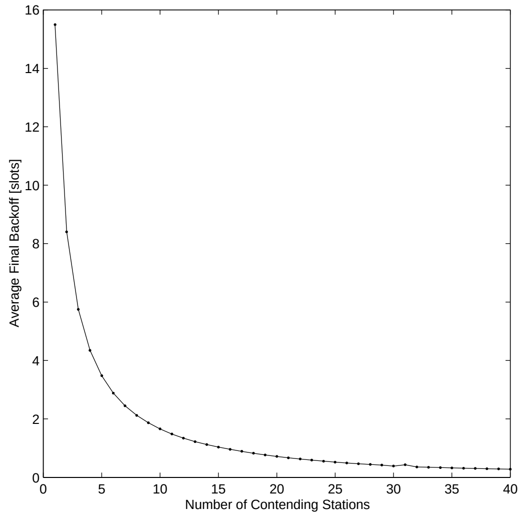

Figure 3-1: Theoretical Average Final Backoff

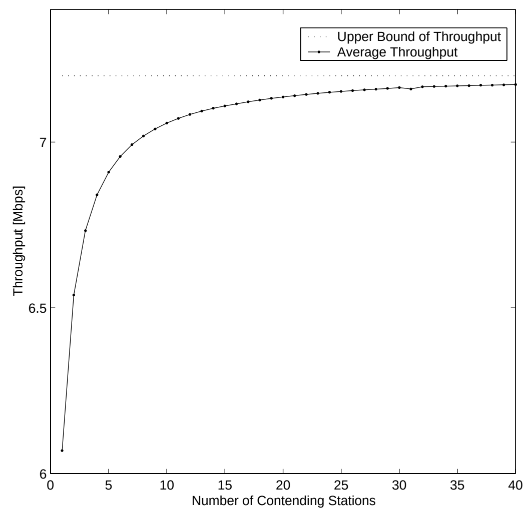

Figure 3-2: Average Throughput using Theoretical Average Final Backoff

3.2 Analysis of the PCF

3.2.1 Data Efficiency

A similar analysis of data efficiency can be performed for the PCF in the 802.11 protocol. As before, propagation delay is neglected, and RTS/CTS is not used. Initially, the scenario producing the highest data efficiency is analyzed. In this scenario, the AP always has data to send to all stations on the polling list, and all polled stations have data to send to the AP. Assuming no collisions occur, the PCF algorithm of one Contention Free Period (CFP) proceeds as follows: (refer to Figure 2-5)

1. The medium is idle for a PIFS.
2. The PC in the AP sends a Beacon frame to indicate the start of the CFP.
3. A SIFS interval passes.
4. The PC sends a Data+CF-Poll to a station on the polling list.
5. A SIFS interval passes.
6. The polled station sends a Data+CF-ACK.

7. A SIFS interval passes.

8. The PC sends a CF-End+ACK frame to indicate the end of the CFP.

Events 4 through 7 are repeated for each station polled during a CFP, as permitted by CFPMaxDuration, the parameter specifying the maximum duration of the CFP. As for the DCF analysis above, the duration of time required for the entire sequence of events can be represented by the sum of the durations of each event.

$$\begin{aligned} T_{PCF\ sequence} &= T_{PIFS} + T_{Beacon} + T_{SIFS} \\ &\quad + N \cdot (T_{Data+CF-Poll} + T_{SIFS} + T_{Data+CF-ACK} + T_{SIFS}) \\ &\quad + T_{CF-End+ACK} \end{aligned}$$

where N is the number of stations polled during a CFP.

T_{PIFS} and T_{SIFS} are simply the PIFS and SIFS interval specified by the standard according to the particular physical layer being used. T_{Beacon} is 96 bits (64-bit time-stamp, 16-bit beacon interval, and 16-bit capability information) transmitted at the basic rate plus MAC header (24 bytes) and FCS (4 bytes) overheads, and $T_{CF-End+ACK}$ is 20 bytes transmitted at the basic rate. $T_{Data+CF-Poll}$ and $T_{Data+CF-ACK}$ are both $l_{MAC\ header} + l_{payload} + l_{FCS}$ bits transmitted at the chosen transmission rate. All transmitted packets also incur the physical layer overhead of the PLCP-Preamble and PLCP-Header being transmitted at the basic rate. From these values, the data efficiency can be derived.

$$data\ efficiency = \frac{N(2T_{data})}{T_{PCF\ sequence}} \cdot 100\%$$

3.2.2 Calculated Values

In addition to the values listed in Table 3.1, Table 3.5 lists the protocol parameters for PCF. Table 3.6 shows the calculated data efficiencies for a polling list consisting of one station sending maximum 2304-byte packets at various transmission rates, and Table 3.7 shows corresponding values for when the station sends 32.5-byte packets.

Parameter	**Value**
PIFS	30 μsec
Beacon	40 bytes
Data+CF-ACK, Data+CF-Poll, Data+CF-ACK+CF-Poll	28 bytes + payload
Null Frame	29 bytes
CF-ACK, CF-Poll, CF-ACK+CF-Poll	29 bytes
CF-End, CF-End+ACK	20 bytes

Table 3.5: Protocol Parameters for PCF using a DSSS Physical Layer

Transmission Rate [Mbps]	1	2	5.5	11
Data Efficiency [%]	95.45	92.33	82.83	71.30
Effective Data Throughput [Mbps]	0.95	1.85	4.56	7.84

Table 3.6: Data Efficiencies and Effective Data Throughput for PCF: 2304-byte Packets, 1 poll per CFP

Transmission Rate [Mbps]	1	2	5.5	11
Data Efficiency [%]	22.85	14.51	6.37	3.39
Effective Data Throughput [Mbps]	0.23	0.29	0.35	0.37

Table 3.7: Data Efficiencies and Effective Data Throughput for PCF: 32.5-byte Packets, 1 poll per CFP

The effective overhead of PCF can be further reduced by increasing the number of stations on the polling list. With more stations sending data, the overhead of the frames indicating the start and end of the CFP is not as significant. Figures 3-3 and 3-4 illustrate the effect of varying the number of station polled each CFP on the data efficiency for PCF. In both cases, stations transmit packets at 11 Mbps, and the basic rate is 1 Mbps.

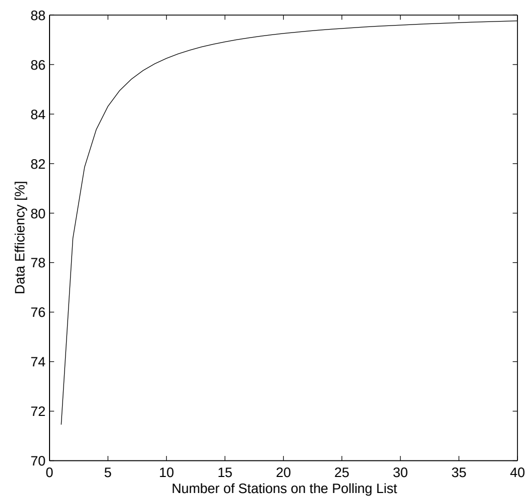

Figure 3-3: Data Efficiency of PCF When Transmitting 2304-byte Packets

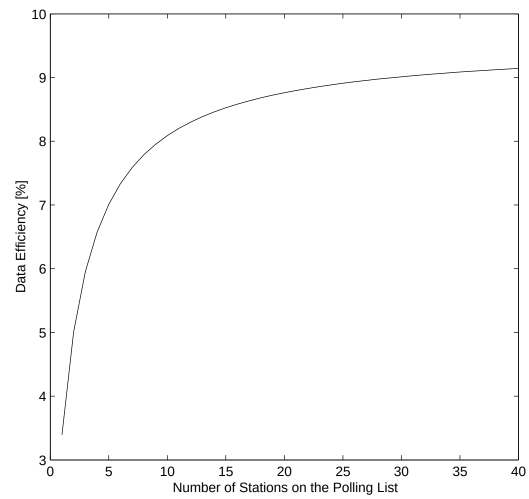

Figure 3-4: Data Efficiency of PCF When Transmitting 32.5-byte Packets

With 40 stations being polled in each CFP, stations transmitting 2304-byte packets can achieve a data efficiency of 87.76%, producing an effective transmission rate of 9.65 Mbps. If the stations send 32.5-byte packets, an efficiency of only 9.19% can be achieved to produce an effective rate of 1.01 Mbps. Thus, if more stations are polled in each CFP, the effective overhead of each packet is reduced. However, the number of stations polled in each CFP is limited by the maximum CFP duration parameter set by the network administrator. The more stations on the polling list, the longer the period of wait between polls experienced by each station (assuming a fair polling scheme). This may be detrimental to stations sending real-time traffic that require dedicated bandwidth.

3.2.3 Effects of Traffic Activity

For random access protocols such as DCF, the differing levels of traffic resulting from voice activity is easily accommodated. With little coordination, resource allocations adapt as loads on each station change. Because control of the medium is acquired through stochastic means, greater portions of the network's resources are easily allocated to stations experiencing higher traffic loads.

On the other hand, dynamic demand assignment protocols such as PCF require more coordination to achieve the flexibility to adapt under changing traffic load conditions. For PCF, this overhead appears in the form of polls to stations that do not have traffic to send. These stations must respond to the poll by transmitting a Null frame to decline the poll, further contributing to protocol overhead. These overhead frames occupy bandwidth that could otherwise be used for transmission of data bits.

The data efficiency of PCF for a traffic stream with periods of alternating activity and inactivity is calculated. For this analysis, a simple two-state Markov model is used. The source generator in the station is assumed to be in the ON-state with a probability $P(On)$ and in the OFF-state with a probability $P(Off)$. Stations in which the source is in the ON-state will have a packet awaiting transmission in their send buffer, while stations in the OFF-state will respond to CF-Polls with a Null frame. For each station polled by the PC, the AP will have a corresponding traffic source modeled by the two-state Markov chain. Thus, when the PC polls a station, there is a $P(On)$ chance that it will also have a data packet to send with the poll to the station.

The data efficiency is calculated in a similar manner as before, factoring in traffic activity:

$$data\ efficiency = \frac{P(On) \cdot N(2T_{data})}{T'_{PCF\ sequence}} \cdot 100\%$$

where

$$
\begin{aligned}
T'_{PCF\ sequence} &= T_{PIFS} + T_{Beacon} + T_{SIFS} \\
&\quad + N \cdot (E\{T_{Poll}\} + T_{SIFS} + E\{T_{Response}\} + T_{SIFS}) \\
&\quad + T_{CF-End+ACK}.
\end{aligned}
$$

For the above equation, $E\{T_{Poll}\}$ is the expected length of the frame sent from the AP in polling the station, and $E\{T_{Response}\}$ is the expected length of the frame sent from the station to the AP in response to a CF-Poll. Because the length of a CF-Poll frame is the same as that of a CF-Poll+ACK, and the length of a Data+CF-Poll frame is the same as that of a Data+CF-ACK+CF-Poll frame (refer to Table 3.5), the value of $E\{T_{Poll}\}$ can be derived ignoring whether or not the AP is piggy-backing an ACK onto the poll. Thus, the value depends only on whether the AP has any data to send to the station being polled.

$$E\{T_{Poll}\} = P(On) \cdot T_{Data+CF-Poll} + P(Off) \cdot T_{CF-Poll}$$

Similarly, because the length of a Null frame and the length of a CF-ACK frame are identical, and the length of a Data frame and that of a Data+CF-ACK frame are also the same, the value of $E\{T_{Response}\}$ is derived ignoring whether or not the station is acknowledging a received data frame from the AP. The value depends only on whether the station has any packets to send to the AP.

$$E\{T_{Response}\} = P(On) \cdot T_{Data} + P(Off) \cdot T_{Null}$$

Using the above equations, the data efficiency of PCF for varying traffic activity is calculated when $P(On) = 42.55\%$ and $P(Off) = 1 - P(On) = 57.45\%$. [2] Table 3.8 lists the calculated values at various transmission rates for one station polled each CFP where stations transmit 32.5-byte packets. Compared to the values listed in

[2] These values correspond to a voice model by Paul T. Brady [2] cited in [8, pg. 493].

Table 3.7, these values are much lower, illustrating the excess overhead caused by exchanges between the AP and polled station of Null packets that do not contain data.

Transmission Rate [Mbps]	1	2	5.5	11
Data Efficiency [%]	11.14	6.72	2.81	1.47
Effective Data Throughput [Mbps]	0.11	0.13	0.15	0.16

Table 3.8: Data Efficiencies and Effective Data Throughput for PCF: 32.5-byte Packets with Varying Traffic Activity, 1 poll per CFP

3.3 Comparison of DCF and PCF

Comparing the values in Tables 3.6 and 3.7 with those of Tables 3.2 and 3.4, using PCF achieves higher data efficiency on average than using DCF when polled stations always have packets to send, regardless of packet size. This can be attributed to the higher overheads needed using DCF such as performing backoff before transmission. Furthermore, PCF enables the uplink and downlink to use piggy-backed frames. These frames (such as an ACK+Data or a CF-Poll+Data) are the same size as a regular Data frame transmitted during DCF, and allow the overheads of polls and ACKs to be effectively eliminated. By allowing the transmission of these types of frames, the PCF reduces the effective overhead of each data frame transmitted during the CFP.

When the effects of traffic activity are factored in, the advantages of using PCF are not as great. Comparing the values in Table 3.8 to those listed in Table 3.4 where DCF is used to transmit 32.5-byte packets, using PCF actually produces a lower data efficiency when only one station is polled during a CFP. This is due to the high overheads required of coordinating the polling mechanism.

Figure 3-5 shows how the data efficiencies of DCF and PCF with traffic activity vary for different numbers of stations. Stations are assumed to be transmitting 32.5-byte packets. The overhead of beacons have been removed from the analysis for PCF

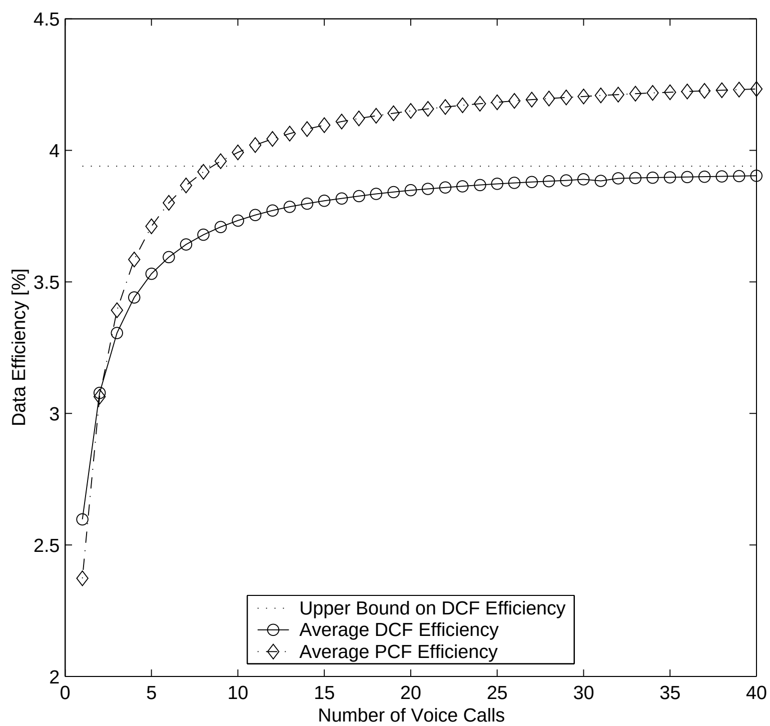

Figure 3-5: Data Efficiencies of DCF and PCF, Varying the Number of Stations

for a more accurate comparison with the DCF analysis.

During DCF, the number of contending stations determines the average length of the backoff contention period between successive packet transmissions. Similarly, using PCF, the more stations on the polling list, the less the relative overhead of starting and ending the CFP. As seen in Figure 3-5, for low traffic levels, the random assignment DCF mechanism performs better. It is only when three or more stations are polled during a CFP that using the PCF produces a higher average data efficiency than using DCF. When there are nine or more stations polled during a CFP, the data efficiency using PCF is higher than even the upper bound on data efficiency for DCF. In general, it is at higher traffic levels when the reduction of acknowledgment overheads outweigh the overheads of coordinating the polling mechanism that PCF performs more efficiently.

Chapter 4

Simulation Setup

The simulation is constructed using the OPNET Modeler network simulating tool. A single infrastructure basic service set (BSS) is modeled under five operational scenarios. (Please refer to the shaded area in Figure 2-1.) Under the first scenario, stations simulating asynchronous data users [1] contend for the medium using only the distributed coordination function (DCF). This establishes a baseline for the performance of the random access mechanism of the 802.11 protocol in a network loaded entirely with asynchronous data traffic. The ability of the DCF to support real-time traffic is evaluated in the second scenario. Stations transmit only time-critical traffic while operating under DCF. To represent real-time traffic, the most basic form of time-critical traffic, packetized encoded voice, is used. The third scenario evaluates how well the DCF access mechanism can support a mix of traffic demanding real-time Quality of Service (QoS) and asynchronous data traffic. The tradeoff of bandwidth between asynchronous traffic and real-time traffic is established. The fourth scenario evaluates the performance of the PCF access mechanism in offering real-time QoS to time-critical traffic. In this scenario, the network, operating under both PCF and DCF, is loaded with only voice traffic. The fifth scenario evaluates the ability of 802.11 to offer distinct QoS to two different types of users. Under this scenario, the BSS operates under both PCF and DCF, with asynchronous data and time-critical

[1]Asynchronous data users will refer to traditional network users generating TCP/IP data traffic that is tolerant of delay.

transmissions. In addition to evaluating how well 802.11 supports real-time traffic, this scenario determines the effects of real-time transmissions on asynchronous traffic.

4.1 Assumptions

The simulations focus only on the performance of the MAC protocol. Idealized traffic generators model packets arriving from higher network layers. Detailed characteristics of the physical layer also are not simulated. Thus, these simulations simply model performance of the medium access control layer, and may not reflect performance of an actual 802.11 network with higher layer protocols such as TCP/IP or UDP.

To reduce complexity in the simulation models, several assumptions are made:

- All stations are stationary and do not move in, out, or within the BSS.
- The BSS is considered an isolated network. There is no interference caused by neighboring BSSs (e.g. reusing the same DSSS spreading sequence)
- The "hidden terminal" problem is not addressed. All stations in the BSS are assumed to be within range of all other stations within the BSS.
- The RTS/CTS frame exchange enhancement was designed to address the hidden node problem. Because the simulated BSS does not contain hidden nodes, RTS/CTS frames are not transmitted. The RTS/CTS parameter is turned off for all frames, simulating the dot11RTSThreshold attribute with the default value of 2347 bytes.
- Evaluation of protocol performance is assumed to occur after association services have been performed. Thus, the BSS is assumed to have already been functioning for a period of time so traffic patterns represent steady state.
- No stations are in power save mode. Thus, stations are available to receive packets at all times, and the AP need not buffer packets for "sleeping stations.

- In an infrastructure BSS, all mobile stations communicate with the AP [6]. Thus, all traffic is sent first to the AP, and the AP forwards the packets to their appropriate destinations. No packets are sent directly from station to station in this infrastructure BSS.

- The simulation assumes an 802.11b Direct Sequence Spread Spectrum (DSSS) physical layer at the bottom of the protocol stack. The MAC access protocol uses parameters specified for this physical layer.

- In accordance with the assumed physical layer, the basic rate set of the BSS includes 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. Stations in the simulated BSS send data traffic over the channel at 11 Mbps for the duration of the simulation. Control traffic such as Beacons and ACKs are transmitted at the basic rate of 1 Mbps. Furthermore, all packets transmitted over the channel will experience an additional delay representing the DSSS PLCP-Preamble and PLCP-Header being transmitted at the basic rate of 1 Mbps [6].

- No multicast or broadcast data frames are sent. Only directed packets with one specific destination are transmitted.

- For simplicity, each station has an infinite FIFO transmit buffer. No packet drops are due to buffer overflow.

- The simulations assume an errorless channel. All transmitted packets experience a BER = 0 during transmission unless a collision of more than one station sending at the same time has occurred.

- When packets are being transmitted over a lossy channel, longer packets may be fragmented into smaller packets, which have a higher probability of errorless transmission. Because the simulation assumes an errorless channel, the benefits of using fragmentation cannot be evaluated. Thus, fragmentation is turned off for all packets, simulating the dot11FragmentationThreshold attribute being set to the default value of 2346 bytes.

- When a station receives a packet, it will not send a corresponding acknowledgment (ACK) if the packet has been corrupted due to the occurrence of a collision. A frame is considered to be corrupt if it contains one or more bit errors. If the sending station does not receive an uncorrupted ACK, it retransmits the same data according to the backoff procedure specified in the standard. The maximum allowable number of retransmissions is Short_Retry_Limit (because RTS/CTS is not being used). If the station cannot successfully transmit the packet in the maximum allowable number of retransmissions, the packet is dropped, and no further retransmissions are attempted.

4.2 Default Simulation Parameters

Table 4.1 lists the parameters used in the simulations.

Description	Value
Transmit Buffer Size	infinite
Fragmentation Threshold	2346 bytes
RTS Threshold	2347 bytes
Short Retry Limit	7
Physical Layer	DSSS
PLCP Preamble Length	144 bits
PLCP Header Length	48 bits
Slot Time	20 μsec
SIFS Time	10 μsec
DIFS Time	50 μsec
PIFS Time	30 μsec
CFP Max Duration	18 msec
CFP Repetition Interval	20 msec

Table 4.1: Simulated Protocol Parameters

4.3 Radio Channel Model

The radio channel will be modeled with the default wireless LAN physical layer model available in the OPNET wireless LAN models.

Stations will be able to receive traffic over the simulated wireless channel only from other stations in the same BSS. All stations within the BSS will be able to receive transmissions from all other stations in the same BSS. No stations will be "hidden". Traffic will experience a nonzero transmission delay determined by the length of the packet being sent and the data rate being used for transmission. A nonzero propagation delay calculated from the distance between the sending transmitter and receiver is also imposed. However, BSSs are assumed to be no larger than 100 feet in diameter. Thus, stations will need to transmit only up to this distance, and these delays will be negligible. Regardless of the SNR ratio obtained by the OPNET simulator, the BER of the packet will be zero, modeling an errorless channel. This will remove effects of noise in the channel on the evaluation of the performance of the protocol.

4.4 Asynchronous Data Traffic Model

For simplicity and to minimize simulation time, a bursty source will produce the traffic load to simulate asynchronous data users. Two types of users will be modeled: users producing traffic as a random process and users who continuously have traffic to send. The first type of user will help evaluate how the network handles various traffic loads with a fixed number of users, while the second type of user will demonstrate how many users can effectively saturate the network.

4.4.1 Data Packet Size

For both types of asynchronous data users, packet payloads will be a constant 1500 bytes to simulate the maximum size of an IP datagram [7]. This packet size is chosen to produce the worst-case bandwidth-usage by data packets. Thus, these large packets will simulate the worst-case interference by non-real-time packets to real-time traffic.

4.4.2 Data Packet Destination

According to the 802.11 standard, when a station transmits one packet in an infrastructure BSS, the AP is responsible for forwarding the packet to the appropriate station if the station resides in the same BSS, or forwarding the packet through the distribution system (DS) to the remote AP servicing the BSS of the destination station [1]. All asynchronous data traffic is assumed to have destinations outside the BSS of the sending station. Because the DS is not modeled in the simulations, all data traffic has the AP as its destination.

4.4.3 Data Performance Metrics

Asynchronous data traffic is usually tolerant of delay, but requires bandwidth for efficient transfer of large amounts of data with minimal errors. Thus, average packet delay and data throughput are used as metrics to evaluate the performance of the 802.11 protocol to support data traffic. Packet delay is one-way delay measured as the period of time from packet creation at the source to packet receipt at the destination station. Throughput is measured in bits per second as the rate of data bits correctly received in errorless frames at the destination.

4.5 Asynchronous Data User Types

4.5.1 Poisson Arrival User

A two-state Markov chain with on-state and off-state will model the data user. The duration of On- and Off- periods will be exponentially distributed with means of 100 seconds and 1 second respectively. Packet interarrival times in the On-state will also be exponentially distributed. The mean of this distribution will be scaled to produce the various desired levels of traffic load for simulation. No packets will be generated during the Off-periods. Each simulated user begins producing traffic a time interval after the start of the simulation selected from an exponential distribution with a mean of 1 second.

4.5.2 Constantly Sending User

The second type of user being modeled will be one that continuously has a packet in its sending buffer. Thus, it will constantly be trying to access the medium to send this packet. This models a user using a protocol such as FTP to transmit an extremely large file. A similar two-state Markov chain will model the constantly sending user. However, the model will always be in the On- state, as the duration of the Off-period will be a constant 0 seconds. Packet interarrival times in the On-state will be a constant 2 msec so stations never have an empty send-buffer. Each simulated constantly sending user begins generating traffic an exponentially distributed random period with a mean of 2 msec after the start of the simulation.

4.6 Real-time Traffic Model

The simplest form of real-time traffic, packetized voice calls, will be used to represent the time-critical traffic in these simulations. A voice station generates a "call" that begins a random time after the start of the simulation, and continues throughout the duration of the simulation. Each voice call consists of two traffic streams, a stream of voice traffic from the source station to the destination station, and a statistically identical stream from the destination station to the source station. The streams' characteristics are as described below, and each operates independently of the stream in the opposite direction.

4.6.1 Voice Packet Stream Characteristics

A voice traffic user is modeled with a simple on-off speech process [3, 9] to simulate voice activity. A two-state Markov chain describes the speech model, with a talk state in which the source produces packets every 20 msec, and a silent state in which no packets are generated. The amount of time spent in each state is exponentially distributed, with the mean duration of the talk-spurt period being 1 second and the mean duration of the silence period being 1.35 seconds. Two different vocoders are simulated, providing low and high coding rates. GSM vocoding produces 32.5-bytes packets every 20 msec for a rate of 13 kbps, while G.711, a vocoder commonly used for land-based phone systems, produces 160-byte packets every 20 msec for a rate of 64 kbps.

4.6.2 Voice Packet Destination

Because the stations of a BSS are assumed to be at most 100 feet apart, the need for a voice connection over this short distance is assumed to be unnecessary, or highly unlikely. Thus, all voice traffic in the simulation originating in the BSS will be assumed to have destinations outside the BSS, and will be transmitted to the AP for forwarding into the DS. For the simulations, all stations simulating real-time users will send packets destined for the AP. In return, the AP will send voice packets back to each station performing a call, simulating packets from stations in neighboring BSSs that have been routed through the DS to the current BSS.

4.6.3 Voice Performance Metrics

Real-time traffic needs a dedicated amount of bandwidth with short latency, low jitter, and little packet loss. Thus, the QoS metrics used to evaluate the real-time performance of the protocol are one-way packet delay and jitter of the voice traffic. Delay is measured as the time from which the packet is created at the source above the MAC sublayer to receipt at the destination terminal. The ITU G.114 specification recommends a maximum one-way delay no longer than 25 msec, and no more than 150 msec if echo cancellers are used, for excellent quality voice (quoted in [8]). Jitter is the variation of the delay experienced by successive packets in the same packet stream. The standard deviation of the delay experienced by voice packets is used to measure jitter. Voice QoS requires that delay variations remain less than 100 msec, any more of which cannot be effectively compensated for by jitter buffers.

4.7 BSS Operation Scenarios

4.7.1 Asynchronous Data Transmission Using DCF

The primary goal of the simulations under this scenario is to evaluate the performance of the contention access mechanism of the protocol known as the distributed coordination function (DCF). These simulations analyze the effectiveness of the protocol in resolving contention of several stations accessing the shared wireless medium. Furthermore, the overhead due to control frames (Beacons, ACKs) used in the protocol is examined.

4.7.1.1 Constant Number of Poisson Arrival Users

In this simulation, the data throughput achievable under various levels of network traffic load is determined for a fixed number of contending stations. The BSS consists of an AP and 20 fixed stations operated by Poisson arrival asynchronous data users. (See Section 4.5.1 above.) For simplicity, all stations (except the AP) will generate identically distributed traffic loads but will operate independently. The mean of the packet arrival distribution at each station is scaled to produce an aggregate traffic load ranging from 1 Mbps to 10 Mbps in increments of 1 Mbps. All stations transmit over the channel at a rate of 11 Mbps.

4.7.1.2 Varying Number of Constantly Sending Users

In this simulation, the effects on the maximum achievable throughput of a varying number of contending stations is simulated. This simulation also demonstrates how the bandwidth is divided among users when all users are trying to obtain as much of the bandwidth as possible. Each of the users in this simulated BSS will be of the constantly sending user type, described in Section 4.5.2 above. The simulation is run, varying the number of sending users from 1 to 20.

4.7.2 Real-time Transmission Using DCF

The goal of the simulations for this scenario is to evaluate the performance of the random access mechanism to offer QoS for real-time traffic. The model represents a varying number of stations in the BSS transmitting real-time voice packets. As described above, all packets will be sent to the AP, and the AP will also send packets to each active station simulating traffic routed from neighboring BSSs. Stations operate independently and are homogeneous in traffic generation. All stations in a simulation run use the same vocoder coding scheme, and this scheme is unchanged for the duration of the simulation.

4.7.3 Supporting Two Types of Traffic Using Only DCF

This scenario is designed to evaluate what type of QoS real-time traffic, in the presence of asynchronous data traffic, experiences on an 802.11 network operating under DCF. The throughput cost to asynchronous traffic for supporting a real-time voice stream is also investigated. Voice traffic is deemed broken when greater than 1% of the real-time packets experience delay longer than 25 msec.

4.7.3.1 Voice Traffic Contending with Poisson Arrival Users

These simulations consist of 20 stations, all generating traffic and using DCF to access the medium. Each station generates only one type of traffic: Poisson arrival asynchronous data or real-time voice. The performance of the network is evaluated for 1, 5, and 9 voice stations. Thus, in each of those cases, there are 19, 15, and 11 stations transmitting asynchronous data respectively. The voice traffic generated by real-time stations is that described in Section 4.6. The simulations are run with both GSM and G.711 encoding, with all voice stations using the same encoding in any given simulation run. The asynchronous data traffic is that described in Section 4.5.1 with constant 1500 byte packets with exponentially distributed interarrival times. All asynchronous data user stations generate identically distributed traffic loads with the aggregate network data load ranging from 1 Mbps to 10 Mbps in increments of 1

Mbps.

4.7.3.2 Voice Traffic Contending with Constantly Sending Users

This scenario consists of a fixed number of voice stations and a varying number of data stations, all using DCF. The scenario is run with 1, 5, and 9 voice stations, and in each case, each voice stations represents a voice "call" consisting of traffic streams transmitting to and from the AP. The number of contending data stations in the BSS simulating users of the type described in Section 4.5.2 is increased until the voice streams are broken.

4.7.4 Real-time Traffic Using Mostly PCF

For this scenario, the ability of the PCF access mechanism to deliver real-time QoS is studied. The bandwidth consumed by the overhead for the polling mechanism is examined. The performance of the PCF mechanism will be evaluated for a variable number of transmitting stations.

A fixed number of transmitting voice stations make up the BSS in this scenario. However, for each simulation run, only a subset of those stations will have traffic to send, and only those stations will be included on the polling list. The protocol parameters are chosen such that the majority of the bandwidth is reserved for the contention-free period while the length of the contention period is minimal. The standard requires that the contention period be long enough to allow transmission of one maximum-sized packet [1].

4.7.4.1 PCF Polling List

The Point Coordinator (PC) in the AP maintains a list of stations requiring real-time QoS services. The simulation assumes that stations requiring this service have already been included in the polling list, and details of registering to be added to the list are not included in the simulation. Real-time stations are assumed to be making one continuous voice-call that lasts the entire duration of the simulation. The number of stations generating real-time traffic varies for each simulation run, and only those real-time stations are included in the polling list. For simplicity, stations are arranged on the polling list in order of increasing address, with the lowest-addressed station at the top of the list.

4.7.4.2 PCF Polling

Details about the polling scheme are considered beyond the scope of the standard [1]. However, the polling scheme used in the simulations is described as below.

At the designated beginning of each Contention-Free Period (CFP), the AP takes control of the medium using standard DCF methods. The stations on the polling list are polled in order, beginning with the first station on the list, regardless of where the PC left off at the end of the previous CFP. For each CF-Poll received, a voice station may send one frame in response. Each station on the polling list is polled at most once during each CFP. If there is time remaining in the CFP after the PC has finished polling each station on the list, the PC relinquishes control of the medium to normal DCF contention. If the CFP has exceeded the specified dot11CFPMaxDuration attribute, the PC ends the CFP immediately without any further polls. In the next CFP, the PC begins polling anew from the top of the list.

4.7.4.3 PCF Protocol Parameters

To accommodate the QoS requirements of real-time voice, the CFP-Repetition interval is set at 20 msec. Voice packets in the On-state are generated at the rate of one every 20 msec, so this would prevent any unnecessary buffering at the sending stations. The dot11CFPMaxDuration attribute is set at 18 msec to satisfy the requirement that the contention period be long enough for transmission of one maximum-length MPDU. The maximum packet payload size specified by the standard is 2304 bytes if encryption is not used.

4.7.5 Supporting Two QoS Using DCF and PCF

The goal of the simulations for this scenario is to evaluate the performance of the 802.11 MAC sublayer under two distinct QoS requirements. The concurrent operation of the PCF and DCF is studied. The simulation evaluates the ability of the protocol to deliver the necessary QoS requirements to the real-time voice packets without adversely affecting asynchronous data throughput.

For simplicity, each station in the BSS transmits either voice packets or asynchronous data packets, but not both. Stations operate under both medium access mechanisms, but stations simulating data users only transmit using DCF while stations making voice calls transmit using only PCF. Thus, only voice stations are included on the polling list. During the CFP, only time-critical voice data is transmitted, while only asynchronous data is sent during the CP.

4.7.5.1 Voice Using PCF with Poisson Arrival Data Users Using DCF

The model consists of 20 fixed stations. This scenario is run with 1, 5, and 9 stations making voice calls using GSM encoding. The asynchronous data users are of the type described in Section 4.5.1.

4.7.5.2 Voice Using PCF with Constantly Sending Data Users Using DCF

This model consists of a fixed number (1, 5, and 9) of voice stations while the number of data stations is varied. The simulations are run assuming there are a maximum of 20 stations in the BSS. Voice stations are as described in Section 4.6, while data stations are of the type described in Section 4.5.2 with a continuously non-empty send buffer.

Chapter 5

Simulation Results and Analysis

This chapter presents and analyzes the results of the simulation study described in Chapter 4.

5.1 Asynchronous Data Transmission Using DCF

5.1.1 Varying Data Load with a Constant Number of Users

This simulation demonstrates the network throughput achievable under a varying network load as described in Section 4.7.1.1. Figure 5-1 shows the simulated average network throughput achieved for 20 stations transmitting 1500-byte packets as the mean interarrival time of packets arriving at each station decreases, and Figure 5-2 shows the average one-way packet delay for the same scenario.

For small load levels, the throughput increases linearly with the traffic load level. For loads up to 6 Mbps, packets experience an average delay no more than 12 msec long. All packets transmitted successfully, as no packets are dropped due to the station exceeding the packet retry limit. From the shape of the graph, the network becomes saturated for traffic loads exceeding 6 Mbps. The maximum network throughput is only 6.6 Mbps. At these high loads, the network is operating under unstable conditions as the arrival rate of packets exceeds the departure rate. Sending buffers fill at the stations, causing packet delay times to also increase.

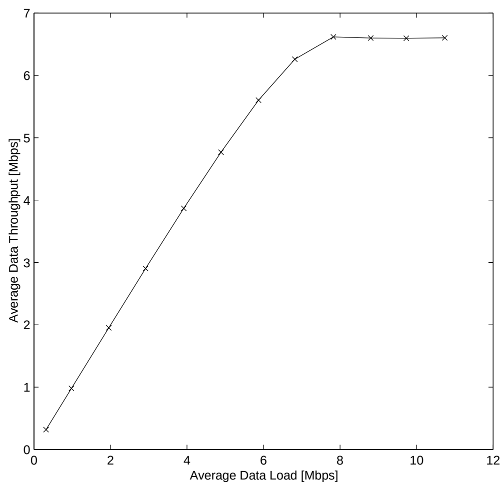

Figure 5-1: Average Data Throughput for Poisson Arrival Stations Using DCF

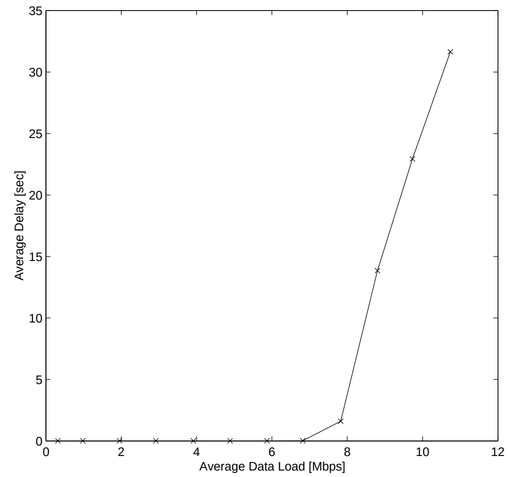

Figure 5-2: Average Data Packet Delay for Poisson Arrival Stations Using DCF

However, the protocol performs well in resolving contention among stations, even at these high traffic levels. No packets are dropped due to excessive retransmissions from collisions on the wireless medium. Packets simply spend more time in the sending queue before eventually being transmitted successfully. Furthermore, each station acquires an equal share of the usable bandwidth.

5.1.2 Constant Data Load with a Varying Number of Users

This simulation demonstrates how the protocol performs when different numbers of users each try to acquire as much of the bandwidth as possible, as described in Section 4.7.1.2. In order to simulate all stations continuously having packets in their transmit-buffer to send, the network operates under unstable conditions in which the arrival rate of packets exceeds the network departure rate.

Figure 5-3 illustrates the simulated average throughput achieved for different numbers of stations in the BSS using DCF to constantly contend for access to the wireless medium. The curve from the data efficiency theoretical model depicted in Figure 3-2 is also shown, scaled to account for the simulated beacon transmissions. The scaling

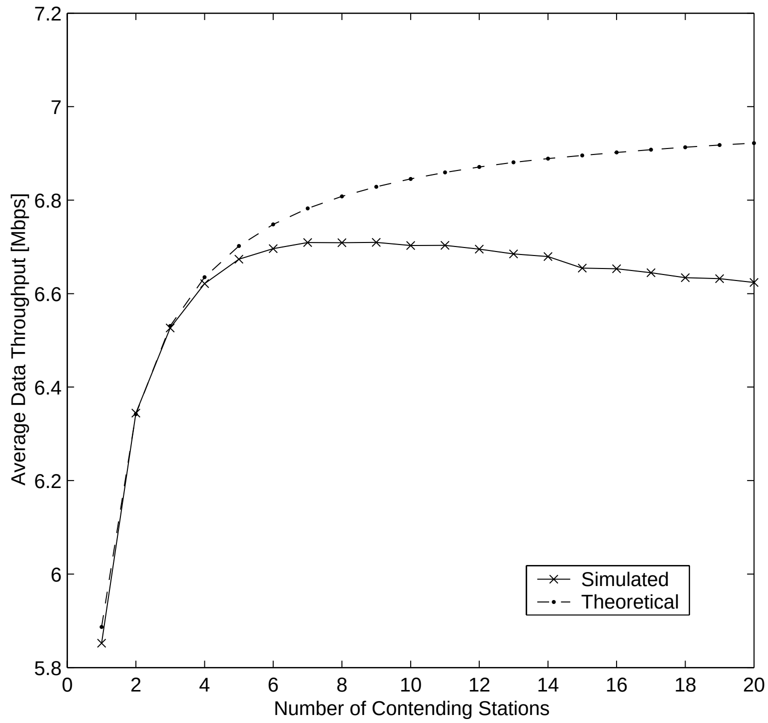

Figure 5-3: Average Throughput for Constantly Sending Stations Using DCF

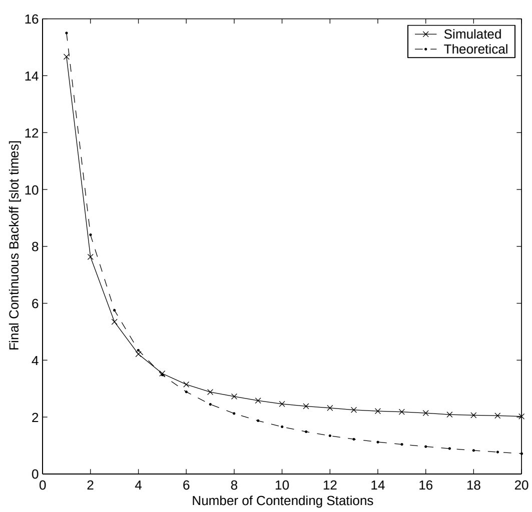

Figure 5-4: Average Final Continuous Backoff for Constantly Sending Stations Using DCF

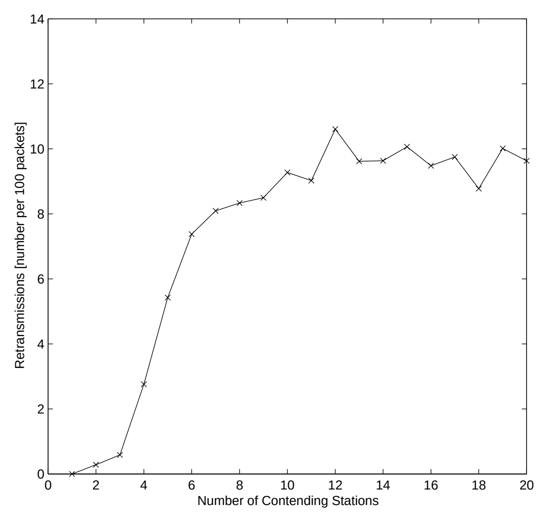

Figure 5-5: Average Number of Retransmissions at One Station for Constantly Sending Stations Using DCF

factor is calculated using

$$\text{Scaling Factor} = 1 - \frac{T_{beacon}}{T_{beacon\ interval}}$$

where T_{beacon} is that calculated in Section 3.2.2 and $T_{beacon\ interval} = 20$ msec.

For a small number of contending stations, the achievable throughput increases as the number of stations increases due to the multiplexing of the backoff times that occurs. The simulated decrease in the final continuous backoff period is illustrated in Figure 5-4. The theoretical backoff period derived in Section 3.1.4 is also shown. In simulation, the length of the final continuous backoff period before a station acquires the medium and transmits a packet decreases from 7.6 slots for two contending stations to 2.9 slots when there are seven contending stations. This shorter backoff period between packet transmissions accounts for the increasing throughput levels seen from one to nine contending stations.

However, at higher contention levels, the curve of simulated results begins to deviate from the theoretical curve. Data throughput starts to decrease as the number of stations increase. Because theoretical models only approximate reality, in certain situations, some approximations no longer hold and the model begins to fail. This difference between theory and reality is seen in Figure 5-3. In this situation, the assumption used in the theoretical model that all transmissions are successful no longer holds. As the number of contending stations increase, collisions caused by the backoff timers of several stations all expiring in the same slot increase. As a result, stations must retransmit packets that have collided, resulting in a lower data throughput on the network. The increase in retransmissions experienced by the continuously sending data stations in simulation is shown in Figure 5-5. In simulation, with four contending stations, one station averages 2.76 retransmissions per 100 packets, while with fifteen contending stations, the average is 10.06 retransmissions per 100 packets.

For constantly sending stations, the highest data throughput is achieved when nine stations contend for access to the medium. However, this maximum value does not differ by much from the next highest throughput levels. The throughput of six to twelve stations differ by less than 1%, indicating that the backoff scheme is relatively successful in adjusting to alleviate collisions for different numbers of contending stations.

5.2 Real-time Voice Transmission Using DCF

This simulation reveals how an 802.11 BSS performs when transmitting only real-time voice traffic using DCF. As described in Section 4.7.2, the number of simulated stations making voice calls in the BSS is increased from two to forty by increments of two, and for each scenario, traffic is transmitted for a simulated five minutes, with each station producing approximately 6000 data samples. Figure 5-6 shows the Cumulative Distribution Function (CDF) of voice packet delay when using GSM encoding, and Figure 5-7 shows the similar data when using G.711 encoding. Figure 5-8 compares the two vocoding schemes and illustrates the probability of voice packet delay being less than 25 msec for eighteen to thirty stations.

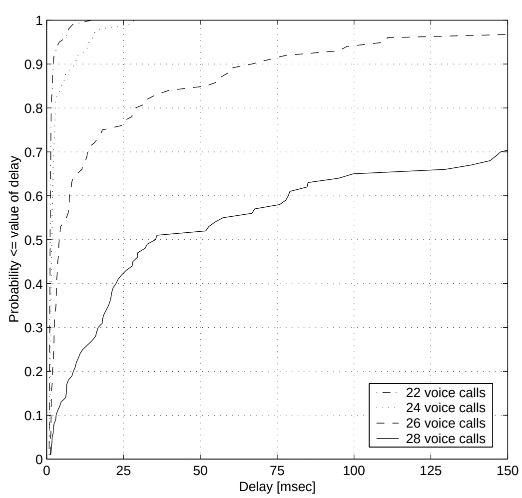

Figure 5-6: CDF of Voice Using DCF, GSM encoding

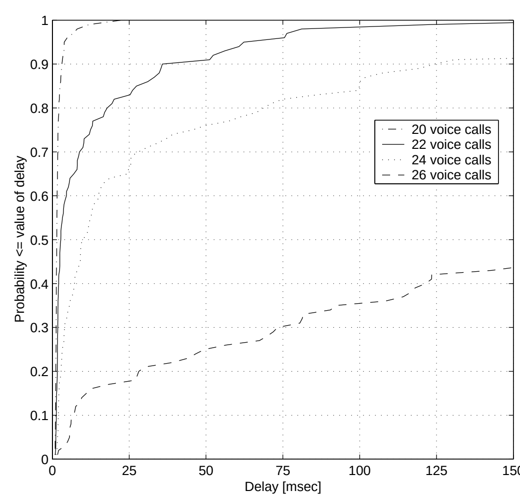

Figure 5-7: CDF of Voice Using DCF, G.711 encoding

Assuming jitter buffers are not used, DCF can support up to 24 GSM voice calls before more than 1% of voice packets experience delays greater than 25 msec. Even when using higher rate vocoders such as G.711, DCF is still able to support up to

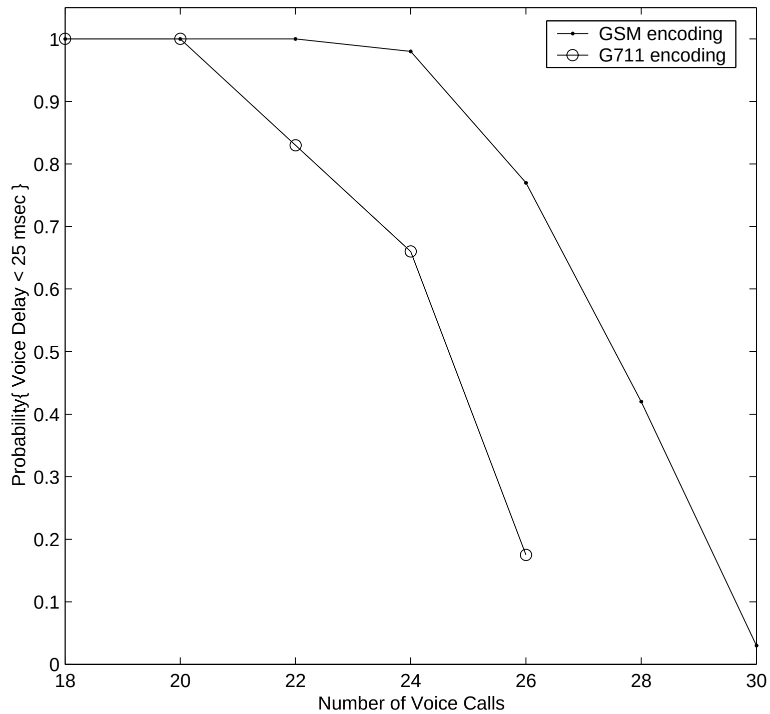

Figure 5-8: Probability of Voice Packets Meeting Delay Threshold of 25 msec

20 voice calls. As the number of voice calls increases, the percentage of packets meeting the delay threshold of 25 msec gradually decreases. Because of the high packet overheads required for the 802.11 protocol and the high transmission rates used, the differences between transmitting 32.5-byte GSM packets and 160-byte G.711 packets are minimal. GSM performs slightly better and is able to support more voice calls due to its smaller packet sizes.

In both cases, the excessive delay that causes the voice calls to fail is due to the funneling of several voice calls into the BSS through only the AP. Because the AP is simply another station, each time the AP acquires the medium, the DCF protocol allows it to transmit only one packet. When there are N stations in the BSS performing voice calls, the AP has N-times the traffic load as each of the other real-time stations for which it must contend for access to the medium. Thus, incoming packets to the BSS must wait in the sending queue of the AP for their turn to be transmitted. In the simulations, only incoming packets to the BSS experience delays greater 25 msec. The delays of packets being transmitted to the AP for forwarding outside the BSS average only approximately 3 msec for up to 40 voice calls.

5.3 Supporting Two Types of Traffic With DCF

5.3.1 Voice Traffic Contending With Poisson Arrival Users

This simulation demonstrates how many real-time voice calls can be supported in the presence of various asynchronous data traffic loads when both user types are using only DCF. The BSS operation scenario is described in Section 4.7.3.1. Figures 5-9, 5-11, and 5-13 show the CDF of voice packet delay for BSSs with 1, 5, and 9 GSM real-time stations respectively contending with Poisson arrival asynchronous data users. Figures 5-10, 5-12, and 5-14 show similar data for when G.711 vocoding is used. A comparison of the percent of voice packets experiencing delay less than 25 msec for both encoding types is shown in Figure 5-15, and a summary of the maximum asynchronous data traffic levels possible on the network without breaking the voice calls can be found in Table 5.1. For this case, it is assumed that jitter buffers are not used so a voice call is considered broken when greater than 1% of voice packets experience delay longer than 25 msec.

	Vocoder	
	GSM	**G.711**
Mean Data Load with 1 Voice Call	5 Mbps	5 Mbps
Mean Data Load with 5 Voice Calls	4 Mbps	4 Mbps
Mean Data Load with 9 Voice Calls	3 Mbps	2 Mbps

Table 5.1: Poisson Arrival Data Stations and Voice Stations Both Using DCF

Due to the large packet overheads required by the 802.11 protocol, as well as the small size of the real-time packets, the low-rate vocoding scheme GSM and higher-rate vocoding scheme G.711 perform very similarly. In general, data packets are much larger than voice packets, and the difference in size between voice packets of different encoding schemes is very small in comparison. GSM and G.711 differ only in the

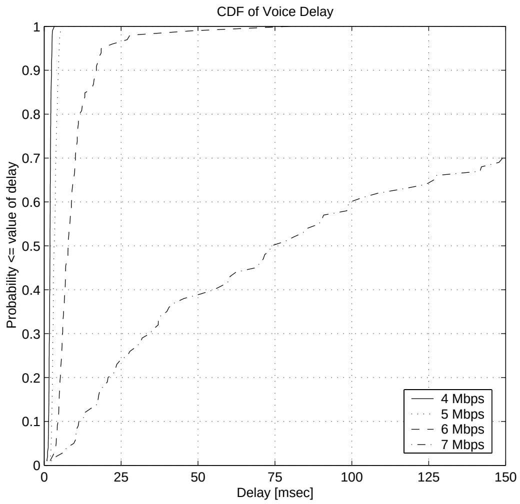

Figure 5-9: CDF of 1 Voice Call Contending With Poisson Arrival Data Stations Varying the Mean Data Load, GSM encoding

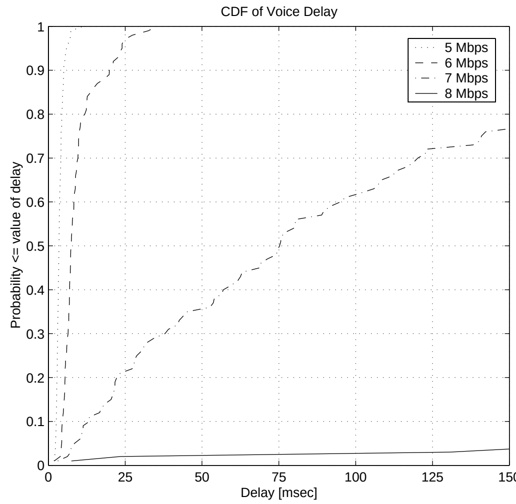

Figure 5-10: CDF of 1 Voice Call Contending With Poisson Arrival Data Stations Varying the Mean Data Load, G.711 encoding

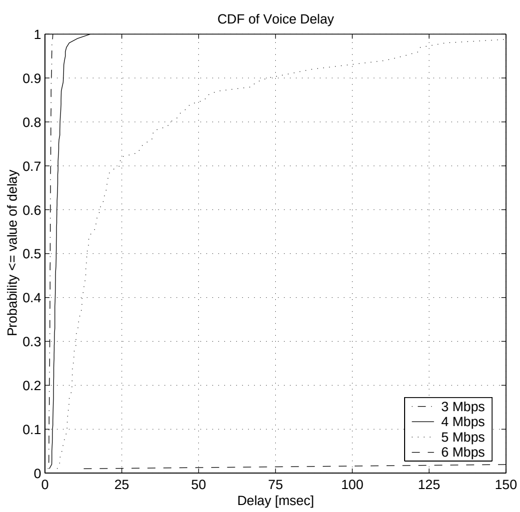

Figure 5-11: CDF of 5 Voice Calls Contending With Poisson Arrival Data Stations Varying the Mean Data Load, GSM encoding

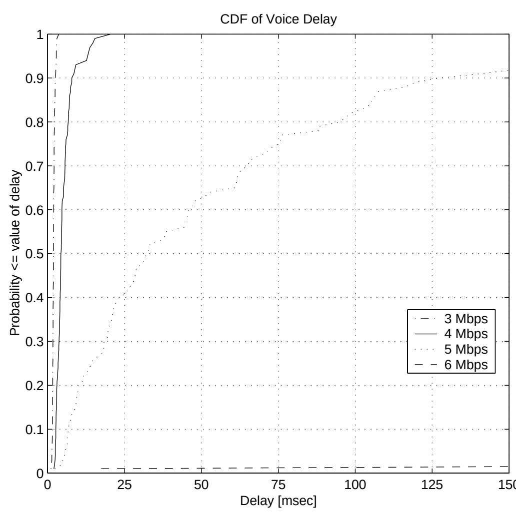

Figure 5-12: CDF of 5 Voice Calls Contending With Poisson Arrival Data Stations Varying the Mean Data Load, G.711 encoding

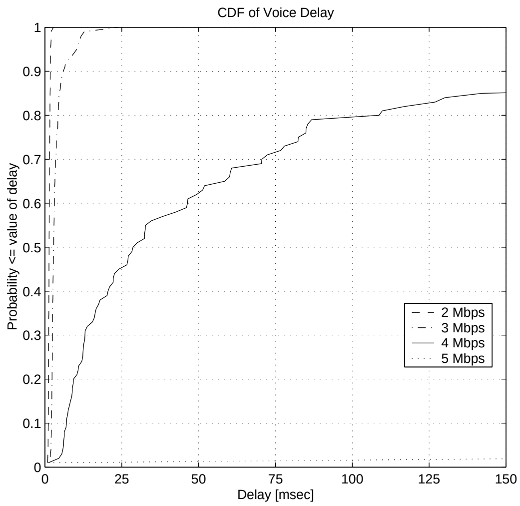

Figure 5-13: CDF of 9 Voice Calls Contending With Poisson Arrival Data Stations Varying the Mean Data Load, GSM encoding

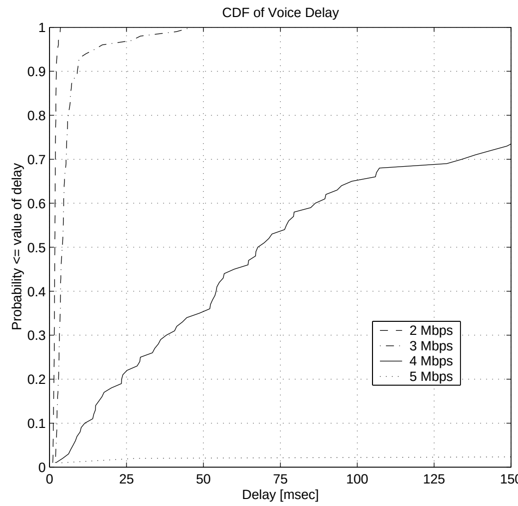

Figure 5-14: CDF of 9 Voice Calls Contending With Poisson Arrival Data Stations Varying the Mean Data Load, G.711 encoding

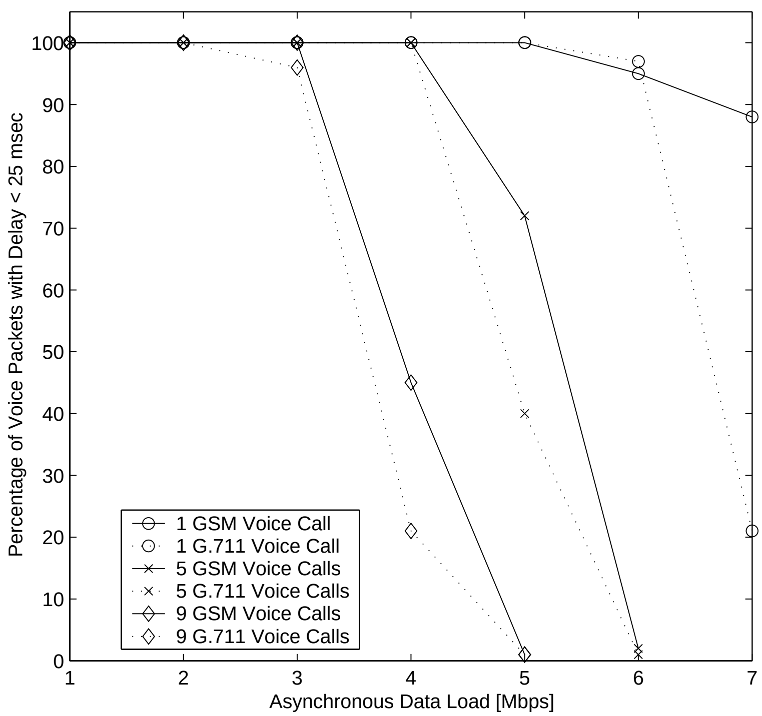

Figure 5-15: Probability of Voice Packets Meeting Delay Requirement of 25 msec when Contending with Poisson Arrival Users

scenario involving nine voice calls. With so many voice calls occurring, the larger packet sizes of G.711 occupy more bandwidth. Thus, nine GSM calls can contend with 3 Mbps asynchronous data load while nine G.711 calls can only contend with a 2 Mbps load before the calls fail. Packetized voice streams using different different encoding schemes can be expected to perform very similarly when there are few voice streams. It is only when there are a significant number of voice streams that the difference in vocoding schemes is large enough to be evident.

From the values summarized in Table 5.1, the tradeoff between real-time GSM voice and asynchronous data traffic is approximately

$$\text{11 kbps voice } \approx \text{250 kbps data} .$$

For each additional 11 kbps voice call to be supported by DCF, the simulations indicate that the contending asynchronous data load must be decreased by 250 kbps. The large tradeoff in supported load is mainly due to the difference in packet sizes of the two types of traffic. For each acquisition of the medium, an asynchronous data station transmits 1500 bytes of actual data, while a real-time station transmits only 32.5 bytes of actual voice data. This difference in payload sizes of the packets accounts for the real-time/asynchronous data tradeoff. Comparing the data efficiencies calculated in Chapter 3 results in a ratio very close to this tradeoff.

$$\begin{aligned} Tradeoff: \quad & \tfrac{11kbps}{250kbps} && = 4.4\% \\ Data\ Efficiency: \quad & \tfrac{2.60\%}{55.17\%} && = 4.7\% \end{aligned}$$

5.3.2 Voice Contending With Constantly Sending Users

This simulation demonstrates how voice transmissions perform on an 802.11 network when contending with continuously sending data stations. Please refer to Section 4.7.3.2 for a description of this scenario. Each constantly sending data station produces 6 Mbps of data. Thus, one station by itself can completely saturate the network. Figures 5-16, 5-18, and 5-20 show the CDF of voice packet delay for 1, 5, and

9 GSM voice stations respectively contending with varying numbers of continuously sending data stations. Figures 5-17, 5-19, and 5-21 show similar data for G.711 voice stations. Note that Figures 5-18, 5-19, 5-20, and 5-21 are plotted on a logarithmic x-axis in order to accommodate the wide range of delay values experienced by the voice packets. A comparison of the two encoding types can be found in Figure 5-22 which shows the percentage of voice packets experiencing delay less than 25 msec.

As before, the two encoding schemes perform very similarly for lower traffic loads. However, with greater numbers of voice calls and more contending data stations, G.711 voice calls experience a lower percentage of packets meeting the required delay than that experienced by GSM calls due to the smaller packet sizes of GSM. In general, as the number of contending data stations increases, the percentage of packets meeting the delay requirement decreases. This can be attributed to the relationship that the greater the number of stations attempting to acquire the medium, the higher the probability that voice stations wait longer before being able to transmit voice packets. The simulated data show two deviations from this trend in 5 and 9 GSM voice calls seen in Figure 5-22. These simulation runs exhibit a sort of plateau occurring as the number of contending data stations increase. At this time, this trend is attributed to artifacts of the simulation, and more investigation is required before it can be explained.

A summary of the number of data stations possible without breaking the voice streams in each scenario is shown in Table 5.2. As before, for this case, a voice call is considered broken when greater than 1% of voice packets experience delay longer than 25 msec. From the results shown in the table, very few voice calls can be supported when the voice traffic must contend with constantly sending data stations. Only two such data stations can be present in the BSS if even just one voice call is to be supported. To support five voice calls, only one constantly sending station can be permitted.

The scenarios simulating constantly sending users represent a pessimistic bound on the contention of asynchronous data users with voice calls. In reality, this type of data traffic pattern is probably only exhibited by stations performing a FTP of a large file, and due to transmission windows of higher layer protocols, this traffic pattern would not be maintained for a very long duration of time. However, it is possible for the AP to exhibit this type of traffic pattern if it is servicing the transmission of large quantities of data from outside the BSS to stations within this BSS. Under this possible scenario, very few voice calls would be given adequate QoS.

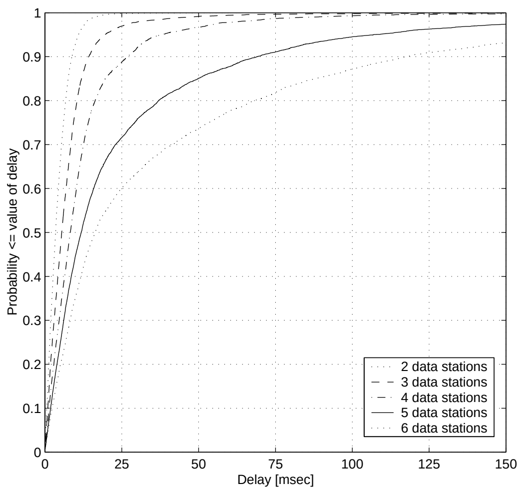

Figure 5-16: CDF of 1 Voice Call Contending With Constantly Sending Data Stations, GSM encoding

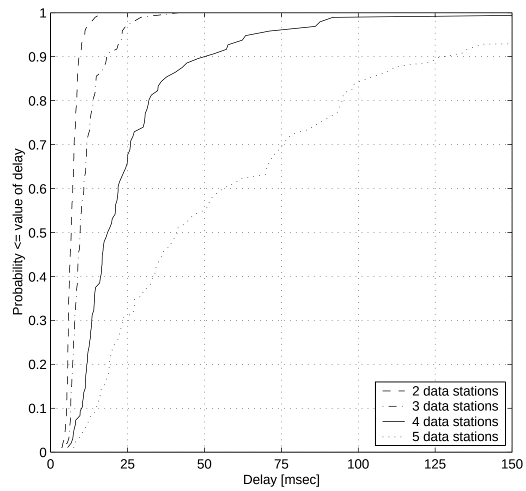

Figure 5-17: CDF of 1 Voice Call Contending With Constantly Sending Data Stations, G.711 encoding

	Vocoder	
	GSM	**G.711**
Data Stations Contending with 1 Voice Call	2	2
Data Stations Contending with 5 Voice Calls	1	1
Data Stations Contending with 9 Voice Calls	0	0

Table 5.2: Number of Constantly Sending Data Stations Contending With Voice Stations Without Breaking the Voice Call, All Using DCF

5.4 Real-time Voice Traffic Using Mostly PCF

This simulation demonstrates the ability of the PCF access mechanism to deliver real-time QoS to voice calls. The BSS operation scenario is described in Section 4.7.4.

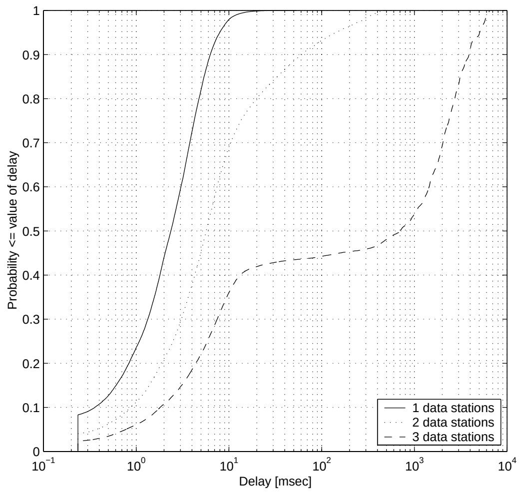

Figure 5-18: CDF of 5 Voice Calls Contending With Constantly Sending Data Stations, GSM encoding

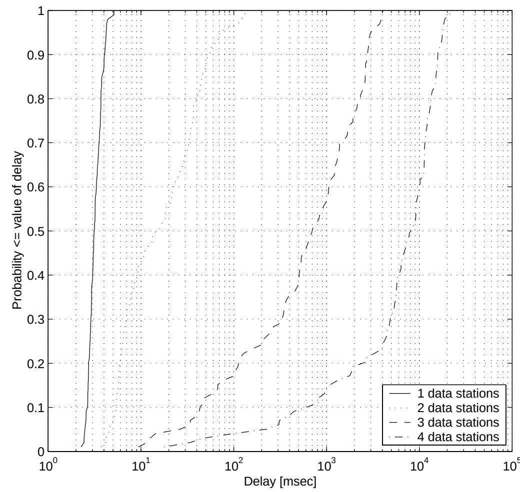

Figure 5-19: CDF of 5 Voice Calls Contending With Constantly Sending Data Stations, G.711 encoding

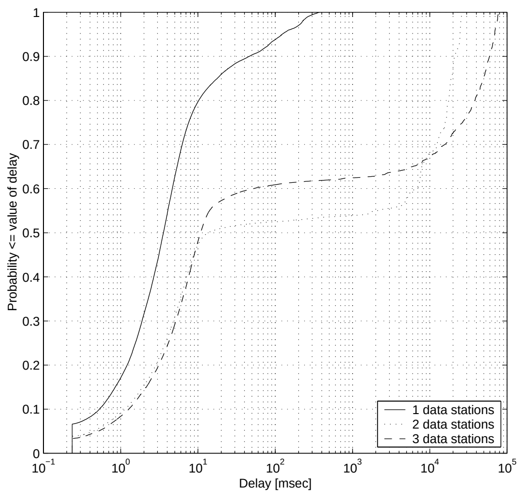

Figure 5-20: CDF of 9 Voice Calls Contending With Constantly Sending Data Stations, GSM encoding

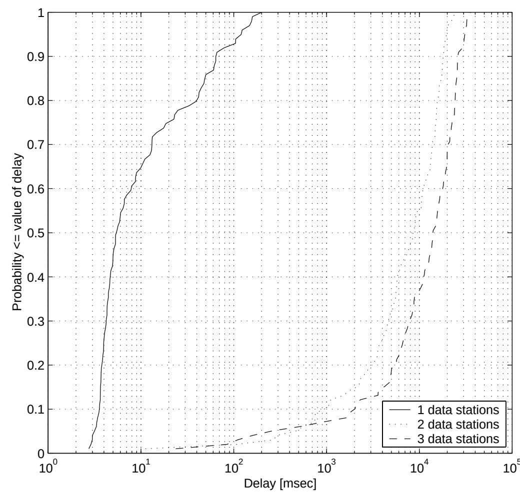

Figure 5-21: CDF of 9 Voice Calls Contending With Constantly Sending Data Stations, G.711 encoding

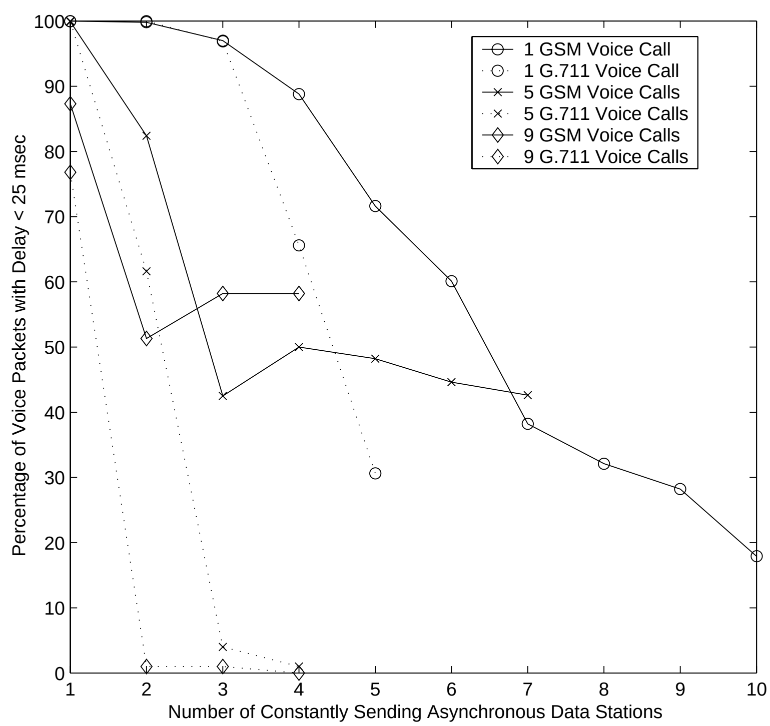

Figure 5-22: Probability of Voice Packets Meeting Delay Requirement of 25 msec when Contending with Constantly Sending Users

Stations producing packetized voice traffic transmit only during the CFP using the PCF. No packets are transmitted during the CP, although the duration of the CP is long enough to permit transmission of one maximum-sized MPDU, as required by the protocol.

Figure 5-23 shows the delay characteristics of GSM voice packets transmitted in the BSS as the number of voice calls occurring in the BSS increases, and Figure 5-24 shows the CDF of voice packet delay. Voice packets using G.711 encoding experienced similar delay characteristics.

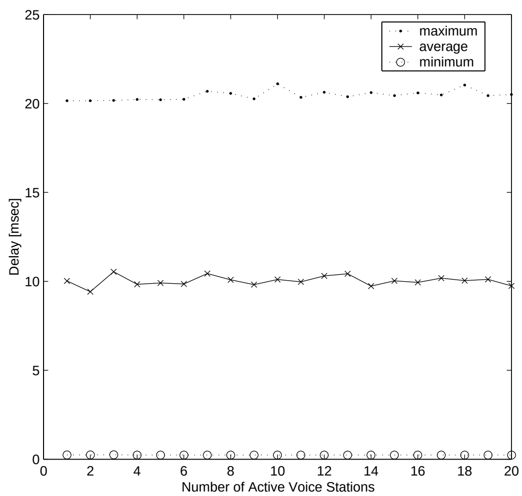

Figure 5-23: Delay Characteristics of Voice Calls Using PCF, GSM encoding

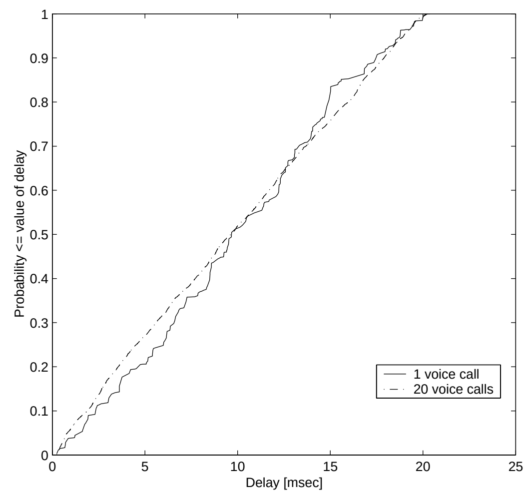

Figure 5-24: CDF of Voice Calls Using PCF, GSM encoding

All voice packets experience delay less than 25 msec. As long as the CFP is long enough for the PC to poll all stations on the polling list, voice traffic from polled stations experience the same delay, regardless of how many other stations are also making voice calls. Furthermore, because the PC in the AP coordinates the CFP, it has higher priority access to the medium. It can thus more easily transmit its higher traffic load of incoming packets to the BSS, enabling incoming and outgoing packets to experience the same delay characteristics.

Using the polling procedure described in Section 4.7.4.2, the PCF can support up to 36 stations making GSM real-time voice calls, and 28 stations making G.711 voice calls. In both cases, scenarios with more than this maximum number failed because the CFPMaxDuration parameter did not allow enough time for the PC to poll all stations on the polling list. Consequently, stations with high numbered addresses did not receive enough CF-Polls to transmit their voice packets within a 25 msec delay. Nevertheless, by using PCF, more voice calls in the BSS can be supported than when only DCF is used.

5.5 Supporting Two QoS Using DCF and PCF

These simulations study the concurrent operation of the DCF and PCF in offering distinct QoS to two classes of traffic. The BSS operates as described in the scenarios in Sections 4.7.5.1 and 4.7.5.2. Voice traffic is given priority access to the medium by using the PCF mechanism while asynchronous data traffic only uses the DCF access mechanism. The results of these simulations show that because the PCF has priority over DCF, a portion of the bandwidth is devoted especially to voice. Neither type of asynchronous data user hinders the transmission of voice traffic.

Number of Voice Calls	1	5	9
Expected CFP Duration	1.370 msec	3.232 msec	5.095 msec
Theoretical Percentage Reduction	93.15%	83.84%	74.52%
Simulated Percentage Reduction	89.3 %	79.4 %	68.8 %
Simulated Maximum Data Throughput	5.92 Mbps	5.28 Mbps	4.61 Mbps
Data Throughput (voice using DCF)	5 Mbps	4 Mbps	3 Mbps

Table 5.3: Asynchronous Data Throughput Using DCF with Voice Using PCF

Voice traffic experiences delay characteristics similar to those shown in Figures 5-23 and 5-24 for PCF transmitting only voice where all voice packets meet the real-time voice delay threshold of 25 msec. However, this real-time QoS support comes at the expense of asynchronous data throughput. Table 5.3 list the maximum data throughput simulated for both types of asynchronous data user, and the percentage of throughput reduction due to the CFP. The theoretical percentage of throughput reduction is also included in Table 5.3 for comparison. [1] Nevertheless, the resource allocation of PCF allows higher asynchronous data throughput without breaking the voice calls than if DCF is used for both voice and asynchronous data packets. (Values from Table 5.1 of voice and data stations both using only DCF are repeated here for comparison.) In conclusion, if real-time QoS is desired in face of heavy asynchronous data traffic, PCF should be used to transmit the real-time packets to ensure that they receive the necessary QoS.

[1] Theoretical values are calculated by

$$Theoretical\ Percentage\ Reduction = \frac{Expected\ CFP\ Duration}{CFP\ Repetition\ Interval}$$

where $Expected\ CFP\ Duration$ is calculated according to the methods described in Section 3.2 and $CFP\ Repetition\ Interval = 20msec.$

Chapter 6

Enhancements for Service Differentiation

This chapter investigates other methods of priority access to the wireless medium that have been proposed for 802.11. The IEEE Task Group E is currently working on a draft standard to add proposed enhancements for QoS into 802.11 [4]. These enhancements involve tuning certain parameters of the current protocol to support up to eight different traffic classes (TC). Packets of class $TC = 7$ have the highest priority, while packets of class $TC = 0$ have the lowest priority. Using simulations, the separate effectiveness of two of the proposed enhancements is determined.

6.1 Interframe Space

One proposed enhancement involves configuring the duration of the interframe space that the medium must be idle before the station can begin backing off or transmitting a packet. In the standard 802.11 protocol, this required interframe space is the DIFS for the DCF, and the PIFS for the PCF. In the draft standard 802.11e, this period, referred to as the Arbitration InterFrame Space (AIFS), can be set individually for each traffic class, with the shortest possible AIFS being equal in duration to the DIFS. Traffic classes with the highest priority will have the shortest AIFS interval. For the simulations, the eight AIFS durations are set according to the equation

$$AIFS[TC] = DIFS + (7 - TC) * slot\ time \text{ where } TC = 0, \ldots, 7.$$

6.2 Minimum Contention Window Size

The minimum contention window size CW_{min} is another parameter considered to differentiate access between different traffic classes. The contention window specifies the upper bound of the uniformly distributed interval which a station uses to select a backoff value. CW_{min} specifies the initial size of the contention window when a station begins to attempt transmission, and also the size to which the contention window returns after a successful transmission. In the standard, for the 802.11b DSSS physical layer, CW_{min} has a value of 31. For the simulations, the minimum contention window sizes are set such that

$$CW_{min}[TC] = 31 - 2 \cdot TC \text{ where } TC = 0, \ldots, 7.$$

6.3 Simulation Setup

The simulations investigating these protocol enhancements evaluate the level of priority access that occurs due to the enhancements. The simulations consist of two stations with identical traffic generation patterns sending traffic to the AP. Both stations simulate the Constantly Sending Asynchronous Data User described in Section 4.5.2 where each always has a 1500-byte packet in its buffer to send so is continuously trying to access the medium. One station always produces traffic of the highest traffic class (TC = 7) while the traffic class of packets sent by the other station is varied from (TC = 7) to (TC = 0). The percentage of total throughput consisting of high-priority traffic is used to quantify the effectiveness of the differentiation mechanism.

6.4 Results

Figure 6-1 shows the percentage of total throughput that is high-priority traffic when one station produces high-priority traffic and one station produces low-priority traffic. Figure 6-2 shows the results for a similar scenario in which one high-priority station contends with two low-priority stations.

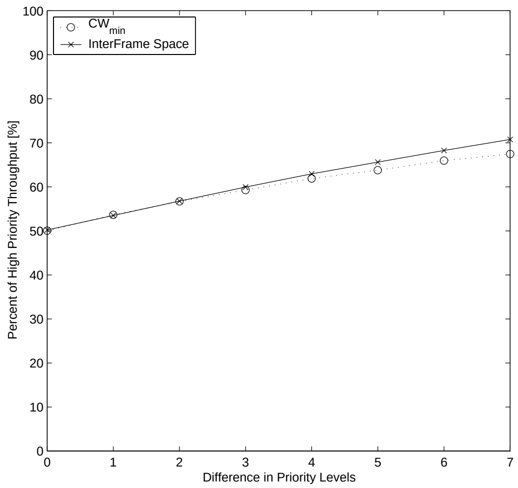

Figure 6-1: Percent of High Priority Throughput, 1 high priority station and 1 low priority station

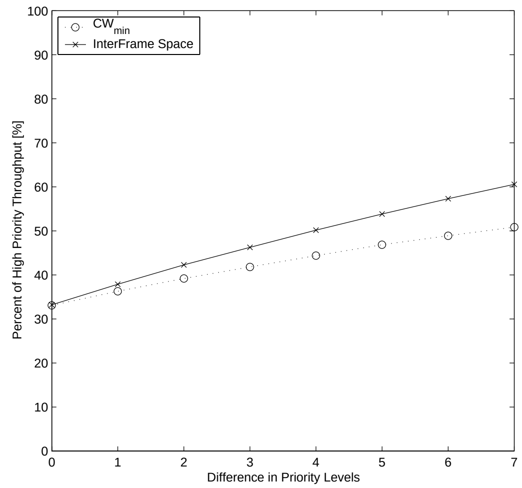

Figure 6-2: Percent of High Priority Throughput, 1 high priority station and 2 low priority stations

When the stations are both sending data with TC = 7, each station uses equal portions of the bandwidth. This result is not surprising, considering both stations use the same protocol of accessing the medium. As the difference in traffic classes of the high and low priority stations increase, the higher priority station takes a larger portion of the bandwidth. With the parameters configured as described above, varying the required interframe space is the more effective method of priority differentiation.

Chapter 7

Summary and Conclusion

With the recent emergence of wireless LAN technology, there is a growing need for a standard to ensure compatibility between products of competing vendors. The 802.11 protocol is the standard for wireless LANs adopted by IEEE. With more and more media forms using digital communications, wireless networks must be able to support various types of traffic. This thesis evaluates the performance of two traffic types, packetized voice and asynchronous data, using the 802.11 MAC protocol.

To support different types of traffic, the 802.11 MAC protocol has two medium access mechanisms: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The DCF is a random multiaccess scheme in which most decision-making is performed by the stations. Stations contend for access to the medium using stochastic means. Thus, they may wait for long periods of time before gaining access to the wireless medium, and even after transmitting, their packets may experience collisions requiring further retransmissions. Stations desiring near-isochronous service may use the optional PCF mechanism. PCF is a centrally controlled mechanism in which stations register to be included in a polling list. During contention-free periods, the mediator of the PCF polls members of the polling list, permitting them to transmit packets without contention.

Due to packet overheads, the highest data efficiency that can be achieved when using DCF to transmit maximum-sized packets permitted by the protocol is 95.94%. This efficiency decreases as transmission rates increase because certain overhead bits must still be transmitted at the slower, basic rate. An 11 Mbps transmission rate of maximum-sized 2304-byte packets produces an upper bound on data efficiency of 74.41%, and an average data efficiency of only 65.40%, corresponding to an effective throughput of only 7.19 Mbps. Transmitting smaller-sized packets such as 1500-byte maximum-sized IP datagrams further reduces the average data efficiency to 55.17% to produce an effective throughput of only 6.07 Mbps.

The number of stations contending for access to the medium also affects the achievable throughput. Due to multiplexing of the backoff periods, throughput increases as the number of stations increase. This increase in throughput is confirmed in simulation, though eventually, the effects of collisions become significant and throughput again decreases.

For few stations on the polling list, PCF has worse data efficiency than DCF. However, if there are three or more stations on the polling list, the benefits of using PCF outweigh the polling overheads,and the average efficiency of PCF becomes higher than that of DCF.

DCF performs rather poorly in providing real-time Quality of Service (QoS) for voice calls. If stations transmitting packetized voice traffic use only DCF, the entire Basic Service Set (BSS) can only have an aggregate load of 5 Mbps of asynchronous data traffic if one voice call is to be supported. For each additional voice call that is desired, the asynchronous traffic throughput level must be reduced by approximately 250 kbps.

Furthermore, voice performs very badly when contending with constantly sending asynchronous data users (such as a user FTPing a large file or the Access Point (AP) funneling a large amount of data traffic into the BSS). There can be at most four of this type of user if one voice call is to be supported, assuming jitter buffers are used, and only two of this type of user if jitter buffers are not used. Thus, PCF must be used if real-time voice is to be transmitted in the presence of an appreciable level of asynchronous data traffic using the 802.11 MAC protocol.

Service differentiation is possible using the DCF by configuring protocol parameters such as the InterFrame Space and the minimum contention window size (CW_{min})

differently for separate classes of traffic. However, additional research is needed to determine how these differentiated access mechanisms perform in comparison to PCF in efficiently giving real-time QoS support to voice traffic.

From the results presented in this thesis, a large part of the shortcomings of 802.11 in supporting real-time traffic stems from the high overheads of the protocol. Future research may focus on other methods of reducing per packet overheads in order to better support real-time traffic such as sending small packets together as one large packet. Furthermore, future work can also investigate whether restrictions (such as limitations on packet size) need to be placed on the use of the PCF mechanism so that stations not requiring strict dedicated bandwidth cannot register for PCF if they do not require those services.

Appendix A

Calculating Backoff

This is the MATLAB script used for calculating the final backoff period, taking into account backoff multiplexing. A DSSS physical layer is assumed so the CW_{min} parameter is assumed to be 31.

```
function [answer] = new_bkoff_mult(num_stations)
 % This function takes one argument specifying the number of contending
 % stations and returns the length of the final continuous backoff period
 % in units of slot-time.
 % Assumptions: CW_min = 31
 answer = 0;
 num = [0];
 den = 1;
 for index = 0:num_stations-1
   num_multiple = [1];
   for boundary = 0:num_stations-1
     if (index == boundary)
       continue;
     elseif (boundary < index)
       num_multiple = doMultiply([-boundary 32; 0 -1], num_multiple);
```

```
      else
        num_multiple = doMultiply([-boundary 31; 0 -1], num_multiple);
      end
    end
    num = addMatrices(num, num_multiple);
    den = conv([-index 32], den);
  end
  old_answer = 0;
  for z = 0:10
    sum_num = 0;
    [rows cols] = size(num);
    for i=1:(31-(num_stations-1)*z)
      for k=1:rows
        item = 0;
        for j=1:cols
          item = item+num(k,j)*z^(cols-j);
        end
        sum_num = sum_num + item * i^k;
      end
    end
    sum_den = 0;
    limit = length(den);
    for k=1:limit
      sum_den = sum_den + den(k)*z^(limit-k);
    end
    final_answer = sum_num / sum_den;
    if (final_answer < z)
      answer = findIntersect([z-1 old_answer], [z final_answer]);
%       disp(strcat('Found Answer at z = ', num2str(z)));
      break
```

```
    else
      old_answer = final_answer;
    end
  end

function [sum] = doMultiply(arg1, arg2)
  % This function performs multiplication of polynomials of two variables
  % Usage:   largest power of z in low indexed columns
  %          smallest power of i in low indexed rows
  % Example: arg1 = [0 5 -2; 0 3 -1; 2 1 3]
  %               = (2z^2 + z + 3) i^2 + (3z - 1) i + (5z -2)
  %          arg2 = [0 3; 1 0]
  %               = (z) i + 3
  %          sum  = [0 0 15 -6; 0 5 7 -3; 0 9 2 9; 2 1 3 0]
  %               = (2z^3 + z^2 + 3z) i^3 + (9z^2 + 2z + 9) i^2 +
  %                   (5z^2 + 7z - 3) i + (15z - 6)
  [row1 col1] = size(arg1);
  [row2 col2] = size(arg2);
  sum = [0];
  for i=1:row1
    for j=1:row2
      term = conv(arg1(i,:), arg2(j,:));
      sum = addToSum(sum, term, (i+j-1));
    end
  end

function [new_sum] = addToSum(old_sum, new_term, index)
  % This function adds the elements of a row vector to the specified
  % row in a matrix.  The return value is the resulting matrix,
  % possibly larger in dimension.
```

```
% Arguments: old_sum - matrix in which vector is to be added
%             new_term - row vector to be added
%             index - row of matrix in which vector is to be added
% Example:  addToSum([1 2; 3 4], [1 1 1], 2) = [0 1 2; 1 4 5]
[row col] = size(old_sum);
new_row = max(row, index);
new_col = max(col, length(new_term));
% make sure column dimensions match
if (length(new_term) > col)
  add_col = length(new_term) - col;
  old_sum = [zeros(row, add_col) old_sum];
elseif (col > length(new_term))
  add_col = col - length(new_term);
  new_term = [zeros(1, add_col) new_term];
end
% make sure row dimensions match
if (index>row)
  diff = index - row;
  old_sum = [old_sum; zeros(diff, new_col)];
end
new_sum = old_sum;
new_sum(index,:) = old_sum(index,:) + new_term;

function [sum] = addMatrices(matrix1, matrix2)
  % This function adds two matrices.  If the matrices are not the same
  % size, they are added with the element in the upper right corner matching.
  % Example: addMatrices([1 2; 3 4], [1 1 1; 1 1 1; 1 1 1])
  %              = [1 2 3; 1 4 5; 1 1 1]
  sum = 0;
  [row1 col1] = size(matrix1);
```

```
  [row2 col2] = size(matrix2);
  new_row = max(row1, row2);
  new_col = max(col1, col2);
  if (row1 > row2)
    diff = row1 - row2;
    matrix2 = [matrix2; zeros(diff, col2)];
  elseif (row2 > row1)
    diff = row2 - row1;
    matrix1 = [matrix1; zeros(diff, col1)];
  end
  if (col1 > col2)
    diff = col1 - col2;
    matrix2 = [zeros(new_row, diff) matrix2];
  elseif (col2 > col1)
    diff = col2 - col1;
    matrix1 = [zeros(new_row, diff) matrix1];
  end
  sum = matrix1 + matrix2;

function [pt] = findIntersect(pt1, pt2)
  % Finds the intersection of the line determined by the two
  % argument points with the line y = x.
  x1 = pt1(1, 1);
  y1 = pt1(1, 2);
  x2 = pt2(1, 1);
  y2 = pt2(1, 2);
  pt = (x2 * y1 - x1 * y2) / (x2 - x1 - y2 + y1);
```

Bibliography

[1] ANSI/IEEE. Std 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. 1999.

[2] Paul T. Brady. A model for generating on-off speech patterns in two-way conversation. *Bell Systems Technical Journal*, 48(7), 1969.

[3] B. P. Crow, I. Widjaja, J. G. Kim, and P. Sakai. Investigation of the IEEE 802.11 medium access control (MAC) sublayer functions. In *Proceedings of INFOCOM 97*, 1997.

[4] IEEE Task Group E. IEEE draft std 802.11e/d1: Medium access control (MAC) enhancements for quality of service (QoS). 2001.

[5] Andreas Kőpsel, Jean-Pierre Ebert, and Adam Wolisz. A performance comparison of point and distributed coordination function of an IEEE 802.11 WLAN in the presence of real-time requirements. In *Proceedings of the 7th Intl. Workshop on Mobile Multimedia Communications (MoMuC2000)*, 2000.

[6] Bob O'Hara and Al Petrick. *The IEEE 802.11 Handbook: A Designer's Companion*. IEEE Press/Standards Information Network, New York, NY, 1999.

[7] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[8] Malathi Veeraraghavan, Nabeel Cocker, and Tim Moors. Support of voice services in IEEE 802.11 wireless LANs. In *Proceedings of IEEE INFOCOM 2001*, 2001.

[9] Matthijs A. Visser and Magda El Zarki. Voice and data transmission over an 802.11 wireless network. In *Proceedings of IEEE Personal, Indoor, and Mobile Radio Conf. (PIMRC) 1995*, 1995.

[10] Jost Weinmiller, Morten Schläger, Andreas Festag, and Adam Wolisz. Performance study of access control in wireless LANs - IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN. *Mobile Networks and Applications*, 1997.