3-2001

Using Direct-Sequenced Spread Spectrum in a Wired Local Area Network

Robert J. Bonner

USING DIRECT-SEQUENCED SPREAD SPECTRUM IN A WIRED LOCAL AREA NETWORK

THESIS

Robert J. Bonner, First Lieutenant, USAF

AFIT/GE/ENG/01M-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/01M-02

Using Direct Sequenced Spread Spectrum
in a Wired Local Area Network

THESIS

Robert Joseph Bonner
First Lieutenant, USAF

AFIT/GE/ENG/01M-02

AFIT/GE/ENG/01M-02

Using Direct Sequenced Spread Spectrum in a Wired Local Area Network

THESIS

Presented to the Faculty of the School of Engineering and Management
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Robert Joseph Bonner, B.S. Electrical Engineering
First Lieutenant, USAF

March 2001

AFIT/GE/ENG/01M-02

Using Direct Sequenced Spread Spectrum in a Wired Local Area Network

Robert Joseph Bonner, B.S. Electrical Engineering

First Lieutenant, USAF

Approved:

Major Rusty O. Baldwin Thesis Advisor	6 Mar 01 Date
Major Richard A. Raines Committee Member	6 Mar 01 Date
Dr. Michael A. Temple Committee Member	6 Mar 01 Date

Acknowledgements

I would like to express my sincere appreciation to my thesis advisor, Major Rusty Baldwin for his support and encouragement during the course of this thesis effort. His encouragement and keen insight helped me stay focused and get through some rather difficult hurdles to see this research to completion. I would also like to extend a special thanks to my committee members, Major Richard Raines and Dr. Michael Temple for their guidance and advice. I have relied heavily upon their support as well as the support of my fellow USAF and TUAF colleagues who assisted me during my research.

Of all those who helped me, I owe my wife the greatest debt of gratitude. She was very supportive and encouraging throughout my AFIT experience. She bore the weight of taking care of me, our pets, and our home. Due to the workload and stresses associated with this assignment, it has presented a considerable challenge to the both of us. If not for her support and understanding, I could not have completed my studies.

Robert Joseph Bonner

Table of Contents

Page

List of Figures

Figure | Page

List of Tables

List of Abbreviations

AFIT/GE/ENG/01M-02

Abstract

Computer networks dramatically impact the Air Force mission and day-to-day operations. As network speeds increase to the gigabit range and beyond, better means for arbitrating network access becomes critical for increasing performance. Conventional approaches to increase Ethernet performance include using higher bandwidth media such as fiber optic cabling. However, there is a limit to the increase of effectiveness these measures provide.

Spread spectrum multiple access techniques allow multiple users to simultaneously access the shared network resources through the use of special coding. These techniques have been principally employed in wireless networking environments to compensate for the scarce bandwidth inherent in the systems. Since cabling infrastructure upgrades may not be a viable option for increasing network performance, we apply spread spectrum techniques to a wired local area network to increase throughput and lower delay.

Using OPNET, a network simulation and design tool, to simulate a direct sequenced spread spectrum technique, demonstrated that network throughput increased linearly with the number of users while delay remained relatively constant. For a network of 15 transmitting nodes in overload (maximum bandwidth) conditions, individual station throughput increased from a 25% packet success rate for Ethernet to nearly 78% – an increase of nearly 212%. On the network level, throughput increased from 8.7 Mbps to over 28.8 Mbps – an increase of 230%. Under similar conditions, mean delay was reduced by 800% from a high of 38 msec in Ethernet to approximately 4 msec for spread spectrum. The vast performance improvement demonstrated by this research yields insight into ways to extend legacy cabling infrastructures for many years while easily accommodating newer bandwidth-intensive multimedia applications.

Using Direct Sequenced Spread Spectrum in a Wired Local Area Network

I. *Introduction*

This chapter presents an overview of this research with background information in Carrier Sense Multiple Access With Collision Detection (CSMA/CD) as well as Code Division Multiple Access (CDMA) medium access control (MAC) schemes. CSMA/CD is more popularly referred to as Ethernet. CDMA is a multiple access scheme which uses spread spectrum technology for network access. These two MAC protocols are the focus of this research.

1.1 Background

Multiple access techniques allow multiple stations to use the same media for communication. As the power of software applications and the need for enhanced multimedia functions (e.g., real-time voice and video services) continue to increase, the ability of local area networks (LANs) to meet this need is strained. As more users access network resources, throughput can diminish exponentially with the number of transmitting stations. This is in large part due to the multiple access technique employed.

For example, the IEEE 802.3 standard uses CSMA/CD for its medium access control protocol and is the predominant MAC protocol used in most LANs. In an attempt to provide fair and equitable allocation of resources, IEEE 802.3 divides the bandwidth according to time slots – similar to a time division multiplexing scheme. If a station has data to transmit, it senses the medium to detect carrier presence. If a carrier is not detected, the station will transmit and simultaneously listen to the medium. If another station has also started transmitting, a collision will be detected and both stations will stop transmitting and delay for a random amount of time before attempting retransmission. As more and more stations attempt to use the medium, the probability that two or more stations simultaneously transmit increases to the point where packet collisions significantly decrease throughput.

Code division multiple access (CDMA) is a wireless networking MAC protocol which uses spread spectrum techniques to share communication channel bandwidth through the use of orthogonal spreading codes called pseudonoise (PN) codes. This scheme allows stations to have simultaneous access to the channel without experiencing collision effects. Therefore, delays associated with random backoff times due to collisions in CSMA/CD are eliminated.

Research on more efficient broadcast channel utilization has focused primarily on existing multiple access schemes. CSMA/CD is the predominant MAC protocol in wired LANs whereas CDMA is becoming the preferred standard in wireless communications networks. However, there has not been a significant amount of research into the feasibility of using CDMA in a wired LAN.

This research proposes to use CDMA in wired LAN applications. It investigates the benefits of using wireless multiple access techniques in a wired medium and potential restrictions that result. This concept is demonstrated through simulation using the OPNET network simulation and design tool.

1.2 Research Goal

CDMA techniques have various implementations. The two most often encountered are direct sequenced and frequency hopping. The goal of this research is to model and simulate a proposed wired LAN implementation of a direct sequenced spread spectrum (DSSS) data network. Based on simulation results, a comparison is made between the performance of a CDMA network to a 10Base2 coaxial Ethernet LAN.

1.3 Organization

This thesis is divided into five chapters. The organization of the rest of this document is as follows:

- Chapter 2 presents a review of literature pertaining to Ethernet LANs and CDMA communication networks. Discussion of the underlying technology is presented.
- Chapter 3 identifies the methodology used to perform this study. Design decisions as well as identification of the factors and parameters used is research are also presented.
- Chapter 4 presents the results of the Ethernet and DSSS LAN simulations. Model verification and validation is conducted on the components used in the simulations. Analysis of the simulation results is presented through the use of statistical tests.
- Chapter 5 summarizes research conclusions and identifies areas for future research.

1.4 Conclusion

In this chapter, an introduction to the goal of increasing LAN performance through the use of CDMA techniques was presented. Additionally, the organization of this thesis was highlighted.

II. Literature Review

2.1 Introduction

This chapter describes and summarizes research on the performance of the medium access control (MAC) protocols for both carrier sense multiple access with collision detection (CSMA/CD) and direct sequence code division multiple access (DS/CDMA) communication networks. The goal of this chapter is to establish a fundamental understanding of the performance characteristics of these two MAC protocols.

2.2 Communication Networks

Communication networks for distributed computing has dramatically impacted the Air Force's mission and day-to-day operations. Networks composed of desktop computers serve not only for office automation tools such as word processors and spreadsheets, but also for communications by electronic mail (e-mail) and file transfers. These computing and communications capabilities continue to evolve. As automation tool's power and utility increase, so does the need for users to have access to them. Whether data is located on a user's desktop or on some distant server, the user has a need for ready, on-demand access to the information to perform their functions.

2.3 OSI Reference Model

Computer networks are designed using an organized series of levels or layers. This helps to reduce the complexity of designing a network. The International Standards Organization (ISO) Open Systems Interconnection (OSI) model is based on a proposal to standardize the protocols used in the various layers [Tan96]. There are seven layers in the OSI model representing the hiearchical nature of a computer network. Each is named based on the function is provides.

1. Physical Layer: Responsible for actually putting the data on the network media. It describes the physical properties of the various communications media, as well as the electrical properties and interpretation of the exchanged signals. i.e., This layer defines the length of coaxial cable and the type of connectors used to tap the channel.

2. Data Link Layer: Responsible for the physical passing of data. The Data Link Layer describes the logical organization of data bits transmitted on the network medium. i.e., This layer defines the framing, addressing and any other overhead information of Ethernet packets.

3. Network Layer: Responsible for routing data from one node to the other. The Network Layer describes how a packet will be exchanged over various links to deliver data between any two nodes in a network. i.e., This layer defines the addressing and routing structure of the network.

4. Transport Layer: Responsible for the end-to-end integrity of the data transmission. The Transport Layer describes the quality and nature of how the packet will be delivered. i.e., This layer defines if and how retransmissions will be used to ensure that the data is delivered correctly .

5. Session Layer: Responsible for establishing and maintaining the communications channels. The Session Layer describes how the exchange of data is to be conducted. i.e., This layer describes how request and reply packets are paired in a remote procedure call.

6. Presentation Layer: Responsible for managing and converting data. The Presentation Layer describes the syntax of data being transferred. i.e., This layer describes how floating point numbers can be exchanged between hosts with different math formats.

7. Application Layer: Responsible for program-to-program communication. The Application Layer describes how real work actually gets done and typically

is referenced to a program or function. i.e., This layer would implement file system operations

The data link layer contains a sublayer called the Medium Access Control (MAC) layer. The function of the MAC is vitally important to local area networks (LANs) since nearly all "LANs use a multiaccess channel as the basis of their communication" [Tan96]. The MAC sublayer determines which station is permitted to access the physical medium and when. It's function is also dominated by the type of link used. The following sections discuss the types of communication links and the MAC sublayer itself.

2.3.1 Communication Links. There are two types of communication links for computer networks: point-to-point and broadcast. A point-to-point (PPP) link consists of a single transmitter and a single receiver. However, in local area networks where many stations are connected in a geographically concentrated area, broadcast links are typically used [Tan96]. This type of link can have multiple sending and receiving nodes connected to the same broadcast medium or channel.

2.3.2 Medium Access Control. All the stations connected to a broadcast channel or the network medium are capable of transmitting data. In order to help provide a reliable data stream, data is divided into frames for transmission. These frames contain code bits for error checking and correcting along with any routing information or additional network overhead. In this way, only errored frames need to be retransmitted rather than an entire file.

Means of providing fair and equitable access to the communication channel is the task of the medium access control (MAC) protocol. Frequently, more than one station will have data to send. The MAC de-conflicts the use of the channel so stations wishing to transmit have a good probability of a successful transfer [Tan96]. This is accomplished by restricting use of the channel or medium. If two or more stations attempt to transmit at the same time, a collision will result. The time the network is occupied transferring these collided packets is wasted. Essentially, a collision 'mixes' the two packets together making both transmissions incoherent to the intended receivers. By regulating who can transmit and at what time, the MAC helps to ensure transmissions will be successful.

2.3.3 Network Performance. Network performance has grown considerably in the last twenty years largely due to increased bandwidth availability. Connection speeds of ten characters per second were considered state-of-the-art in the 1960's. In the 1970's, packet switched networks attained 64 kilobits per second (kbps) trunk speeds. The 1980's added capabilities to achieve 1.544 million bits per second or megabits per second (Mbps) which is also known as T1 speed [Kle92]. Channel speeds are now exceeding ten billion bits per second or gigabits per second (Gbps) [Zim00]. Although increasing channel bandwidth can reduce delay and increase throughput in a network, its effect is finite and may not contribute any gains at all [Kle92].

In [Kle92], Kleinrock demonstrates that the medium may be capable of transmitting gigabits of information per second, while increasing the bandwidth may not increase the speed of communication. Information travels at a speed which cannot exceed the speed of light. In fact, in coaxial cable the speed is limited to 67% of the speed of light [IEE85]. The roundtrip propagation delay time is therefore fixed and is the same for a 1 Mbps link or a 1 Gbps link for the same length of medium.

Suppose a station wishes to transmit a 1 MB file on a 1 Gbps link. Also, suppose that the distance separating the two stations is such that the propagation delay or latency is 100 ms. The effective end-to-end transfer time is

$$Transfer\ Time = Latency + \frac{1}{Bandwidth} \times Transfer\ Size \tag{2.1}$$

The effective throughput is

$$Effective\ Throughput = \frac{Transfer\ Size}{Transfer\ Time} \tag{2.2}$$

Therefore, the transmission time is $100ms + \frac{1MB}{1Gbps} = 108ms$ (note that there are 8 bits per byte) and the effective throughput for this hypothetical transmission is $\frac{1MB}{108ms} = 74.1Mbps$ – far below the capacity of a gigabit network.

Suppose the network media was upgraded to 10 Gbps. The transmission time is now $100ms + \frac{1MB}{10Gbps} = 100.8$ ms and thus the effective throughput is $\frac{1MB}{100.8ms} =$ 79.4 Mbps. Despite the order of magnitude increase in capacity, there was only a nominal increase in throughput. The effective throughput is now dominated by the latency due to propagation delay. This shows why increasing the network speed does not have an appreciable effect in increasing performance [Kle92].

Bandwidth of the transmission media is but one of the three main user selectable factors for network performance. The other two factors are network topology (e.g, Bus, Ring, Star), and the access control scheme. Because there is a large installed base of Ethernet-based LANs, and since reinstallation of the physical media to support increased bandwidth can be very costly, medium access control is a topic of prime interest in network performance studies. When analyzing general purpose communications networks and the MAC, two metrics are of prime interest: throughput and delay. These metrics are discussed in the following sections.

2.3.3.1 Packets. Computer networks communicate using packets. Packets are an organized series of bits representing the data to be transferred and a certain amount of overhead. Information on who sent the data, the size of the data, its destination, and various other identifying features form what is known as packet overhead. As a packet is transferred to one station to another and traverses the OSI layers, it constantly has overhead information added to it or stripped away.

Ethernet packets have a minimum of 27 bytes of overhead added to a piece of data whereas higher level protocols can add even more [Tan96].

This overhead reduces the effective throughput and increases the delay of a system. This is because it takes time and resources to send these extra bits of information. Furthermore, if the added overhead bits result in too large a packet, the packet will have to be broken into even smaller pieces. This process is called fragmentation [Tan96]. These fragments will then have their own set of overhead information.

2.3.3.2 Throughput. Throughput is defined as the rate (requests per unit of time) at which the requests can be serviced by the system [Jai91]. Generally, throughput is measured in bits per second (bps) and is defined with respect to user data. This is distinguished from the bits added as overhead for the particular transmission or network used. Raw capacity refers to the network speed (e.g., 10 Mbps, 100 Mbps). The maximum achieved throughput, also known as channel capacity, is

$$Capacity = \frac{frames}{second} = \frac{bits}{frame} \times \frac{frames}{second} = \frac{bits}{second}. \tag{2.3}$$

2.3.3.3 Delay. Delay can be defined in many ways. The most often used definition in performance analysis is total delay – the difference between when a packet is ready for transmission and when it is actually received [Jai91]. Total delay is defined by Jain as

$$Delay_{Total} = Delay_{Queuing} + Delay_{Transmission} + Delay_{Propagation} \tag{2.4}$$

Queueing delay is the amount of time a packet spends in a station's transmit buffer and is associated with MAC arbitration. This incorporates the time between when a packet is ready to be sent and when it is actually transmitted. Transmission delay is the time required to transmit the packet and defined in (2.1). This delay is a function of where the station is physically located on the medium. Propagation delay is the time it takes the frame to propagate across a given medium. Typically, total delay is a function of the number of stations accessing the network [Jai91].

2.3.3.4 Power Ratio. In network performance it is desirable to have high throughput and low delay. Intuitively, it seems as though increasing throughput would reduce delay. However, this is not necessarily the case. One way to increase throughput is to allow as many packets onto the network as possible. That is, if you increase the offered load to a network, the throughput will also increase. This will drive the utilization of the link to 100%. This load minimizes the possibility of an idle channel which reduces throughput. The problem with this strategy is that "increasing the number of packets in the network also increases the length of the queues" [PD96] at each station. Longer queues, in turn, mean packets are delayed longer prior to transmission resulting in longer delay.

To describe this relationship, some network designers have proposed using the ratio of throughput to delay as a metric. This ratio is sometimes referred to as the power of the network [PD96] and is

$$Power = \frac{Throughput}{Delay}. \tag{2.5}$$

The power ratio makes several assumptions which limits its usefulness. First, it assumes an M/M/1 queuing network that has infinite queues. In a real network, the network interface card (NIC) and the stations themselves have a finite amount of memory and sometimes have to drop packets. Second, power is typically defined "relative to a single connection (flow)" [PD96]. It is not clear how this extends to multiple, competing connections. Despite these limitations, however, "no alternatives have gained wide acceptance, and so power continues to be used" [PD96].

2.4 Wired and Wireless Networks

A major difference between wired and wireless networks is the transmission medium. According to Rappaport,"interference is the major limiting factor in the performance of cellular radio systems" [Rap96]. He further concludes that the main difference of wireless networks is the "extremely hostile and random nature of the radio channel." Users in wireless networks are typically mobile stations that must communicate through an unguided air interface with obstacles, limited bandwidth, and tighter power constraints. Some examples of problems with wireless communication are multipath, fading, and interference [Rap96].

Because of these limitations, wireless networks must use reduced data rates to achieve reliable communications, while continuing to accommodate an increasing number of users [Rap96]. Wired networks, on the other hand, have fixed stations connected with high quality, guided mediums that induce few errors and have a higher probability of a successful transmission [Tan96].

Due to the higher bandwidth, wired networks can afford to have relatively inefficient transfer protocols. Wired networks achieve high data rates by using more bandwidth. This is in contrast to wireless networks that do not have the high bandwidth and achieve reliable communications by using slower data rates [Tan96] [Rap96].

Packet overhead becomes an issue in both types of networks. Because the data rate is slower compared to a wired network, wireless networks cannot afford to have too much overhead added to its packets. The added bits reduce the effective throughput of the system since now resources must be used to transmit this 'extraneous' information [PD96].

2.5 Carrier Sense Multiple Access With Collision Detection

The most widely used LAN protocol is 10 and 100 Mbps Ethernet, which is based on CSMA/CD and codified in IEEE 802.3 [Chr98]. IEEE 802.3 is an international standard that specifies a 1-persistent CSMA/CD for a 10 Mbps LAN and up to 1024 stations per segment using UTP cabling and a hub in a star configuration or 30 stations per segment in a bus configuration [IEE85]. Current revisions of the standard support 100 Mbps and 10 Gbps transmit speeds. Adoption of these speeds is slow, however, and 10 Mbps is still the most widely used [Chr98] [Zim00].

2.5.1 Ethernet Operation. The Ethernet medium is time-shared and hence can be considered a form of a time-division multiple access (TDMA) system. When a station is ready to send data, it first listens to the channel to detect a carrier. If a carrier is detected, the channel is in use and the station waits until the channel is idle. When the station detects the idle channel, it transmits with probability of 1. This is referred to as 1-persistent.

There are reasons why implementations would use a probability less than one. Having $p < 1$ reduces the probability of collision when there are multiple stations waiting for a busy line to become idle. If each station transmits immediately with a probability of, "say 33%, then up to three stations can be waiting to transmit and the odds are that only one will begin transmitting when the line goes idle" [PD96]. However, under light loading levels, this can result in wasted bandwidth since there is the possibility that no station will transmit when the channel is idle. Ethernet uses 1-persistence to prevent the possibility of this wasted bandwidth.

There is an upper bound of 1500 bytes in the message of an Ethernet packet [IEE85]. Due to this limit, a station can only occupy the medium for a fixed length of time. Moreover, a station must wait at least $9.6\mu s$ before it can transmit another frame [IEE85]. This gives other stations a chance to transmit. Furthermore, it also keeps any one station from continuously transmitting and thereby preventing other stations access to the network [HM95] [RY94] [Chr96].

When two or more stations transmit at the same time, a collision is said to occur. When a collision is detected, all transmitting stations send a jam signal to alert the other stations on the network of a collision and then wait a random amount of time prior to retransmission. The immediate termination of packet transmission through collision detection saves time and bandwidth. It also minimizes delay and makes this the preferred MAC protocol in LANs versus CSMA without collision detection [Tan96].

Once a station detects that its frame is colliding with another, it transmits a 512-bit jamming sequence and then stops transmission. This 512 bit minimum jam time comes from the fact that on a maximally sized Ethernet (180 m), the delay from one end to another is 51.2μs [Tan96]. Because both stations at opposite ends of the Ethernet detect a collision has occurred, they both must transmit enough bits to fill the Ethernet pipe – 512 bits. This need to transmit 512 bits explains why every Ethernet frame must contain at least 46 bytes of data : 14 byte header + 46 byte data + 4 bytes error correction code equals 64 bytes which equals 512 bits [PD96]. Figure 2.1 illustrates Ethernet operation with a collision.

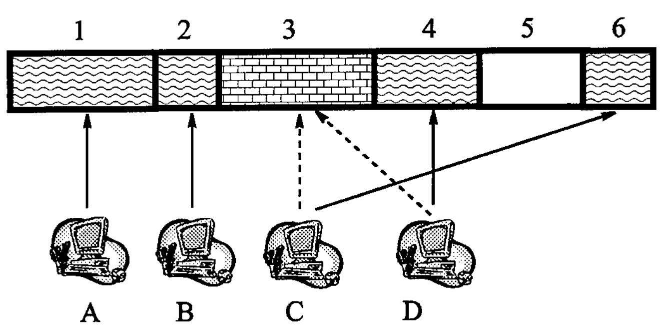

Figure 2.1. Conceptualized Ethernet Operation

Assume that the rectangular bar represents the media used for this Ethernet segment divided in time by the vertical bars. As is shown, each station, A-D, can transmit for variable lengths of time depending on how long their packets are. In this example, station A senses the channel is idle and transmits at time 1 identified by the solid arrow. Station B senses the channel is now silent since station A has finished and transmits at time 2 and station C senses the channel is idle since station B has finished and transmits at time 3. However, station D also sensed the channel was idle at time 3 and begins to transmit. The dashed arrows represent a collision. This can occur especially when the stations are separated by large distances and propagation times are significant. Since both stations C and D tried to transmit at time 3, a collision resulted rendering any information in that time slot useless. The time 3 brick pattern depicts wasted time and bandwidth since if station D did not transmit and cause this collision, station C's transmission would have been successful.

After the collision, both station C and D backoff for a random amount of time as specified by the binary exponential backoff (BEB) algorithm and attempt to transmit at a later time. The BEB in IEEE 802.3 calls for the deferring stations to wait a specified number of time slots based on the collision count. If this is the first collision for both stations C and D, then they can either defer for 0 or 1 time slots. If it is the second collision, then they can defer for 0 to 2^2 time slots.

In this example, assume station C has already had 1 collision previously and station D has none. Station C chooses to wait for 4 time periods and station D waits for 0. Since at time slot 4, no stations were transmitting, station D sensed the idle channel and began its transmission. If station C happened to have chosen 1 time slot to defer, another collision could result with other stations in Time Slot 5. Similarly, if station C chose 2 time slots, it is possible that it could have caused a collision with station A or B. Other stations continue to sense and transmit on the channel as they accumulate more data to send.

2.5.2 Ethernet Implementation Issues. Ethernets are normally implemented in a far more conservative manner than what IEEE 802.3 proposes. Most have fewer than 200 hosts while the standard allows for 1024. Similarly, Ethernets are far shorter than 1500 meters and the propagation delay is closer to $5\mu s$ rather than $51.2\mu s$ [PD96]. Generally, an Ethernet is considered heavily loaded when utilization is over 30% since most of the network's capacity is wasted on collisions. At a minimum, Ethernet frames have about 27 bytes of data overhead. In addition, for transmission purposes, Ethernet also has an interpacket gap of $9.6\mu s$. This gap equates to about 12 bytes thereby increasing the total minimum overhead to 39 bytes per packet.

2.5.3 Physical Medium. Ethernet is still the largest installed LAN topology with 10 Mbps being of the most popular [Chr98]. Despite the upgrade path to 100 Mbps Fast Ethernet and the new movement toward Gigabit Ethernet, the 10 Mbps network still compose the majority of all LANs. Even today, most LANs use coaxial cabling [Chr98]. Other mediums in current use for wired LANS are unshielded twisted pair and optical fiber.

Unshielded twisted pair (UTP), which is also found in 10 Mbps Ethernet networks, houses 4 separate cables. UTP allows 2 strands for transmission and 2 strands for receiving to provide full duplex capability in the network. UTP is implemented in such a way that each communicating station is transmitting to a central hub in a star topology and therefore has a 'virtual point-to-point (PPP)' connection.

Optical fiber poses more signaling challenges over coaxial cable or UTP and is the subject of intense research in fiber optic LANS (FO-LANs) [PSW92]. Essentially, fiber has a much larger bandwidth than other mediums and thus can accommodate a higher data rate.

In the early days of networking, 10Base5 or 'thicknet' was the predominant physical medium. That was replaced with 10Base2 or 'thinnet' which is now the predominant cable in current use with the exception of UTP [Tan96]. The cable employed in 10Base2 is RG-58. RG-58 has the specifications identified in Table 2.1 below [IEE85]

Table 2.1. RG–58 Cable Specifications

Frequency Range	0-4 GHz nominally
VSWR	1.30 max 0-4GHz
Voltage Rating	$500V_{RMS}$ max 1A DC
Contact Resistance	3 milliohms 1A DC
Impedance	50 Ohms constant for 0-4GHz
Capacitance	101 pF per meter
Attenuation	4.6dB per 100 meters @ 10MHz
Velocity Ratio	0.67

2.5.4 Past Ethernet Performance Studies. Kleinrock and Tobagi first concluded in 1975 that CSMA is "an efficient means for randomly accessing packet switched radio channels which have a small ratio (φ) of propagation delay to packet transmission time" [KT75]. CSMA efficiency can be measured in terms of φ where

$$\varphi = \frac{PropagationDelay}{PacketTransmissionTime}. \tag{2.6}$$

Since the propagation delay is

$$PropagationDelay = \frac{ChannelLength}{SpeedofLight}, \tag{2.7}$$

and the packet transmission time is

$$PacketTransmissionTime = \frac{PacketLength}{ChannelBandwidth}, \tag{2.8}$$

φ becomes

$$\varphi = \frac{ChannelLength \cdot ChannelBandwidth}{SpeedofLight \cdot PacketLength}. \tag{2.9}$$

Thus, any increase in channel length or bandwidth, or decreases in packet length will increase φ. A low φ results in reduced efficiency of the CSMA network [KT75].

Metcalfe and Boggs followed on the work of Kleinrock and Tobagi and first proposed the idea of CSMA with collision detection to increase efficiency for a prototype Ethernet system in 1976 [MB76]. This Ethernet system specified a bus topology of a 1 km cable with 100 personal workstations tapped to the bus communicating at a rate of 2.94 Mbps. The tapped bus could be any connecting medium and in current implementations is either coaxial cable (COAX), unshielded twisted pair (UTP), or fiber optic cabling [Tan96].

Further analysis of CSMA has also included performance analysis of CSMA/CD [ML83], [Tob80]. Tobagi and Hunt have shown that CSMA/CD has "improved throughput-delay characteristics over CSMA." The actual magnitude of CSMA/CD improvement over CSMA is dependent on the average retransmission delay and collision recovery time [TH80]. However, [TH80] examined a non-persistent CSMA/CD protocol versus a 1-persistent protocol with dynamic backoff.

The performance of CSMA/CD based Ethernets has been studied extensively in [SH80] and [BM88]. Shoch and Hupp measured throughput in their Ethernet implementation at 98 percent of channel capacity and demonstrated stable behavior at generated loads well over 100 percent [SH80]. This unusual behavior was due to the network traffic that was bursty in nature and composed of a combination of short packets containing computer terminal traffic and larger packets indicative of file transfers from a relatively few number of stations. Also found by Shoch and Hupp, and later refuted by Peter O'Reilly, was that "overall system performance is not significantly sensitive to the number of stations producing a specified total network load, i.e., 10 stations offering 10% of the load produces the same effect as 100 stations offering 1% of the load" [O'R83]. O'Reilly concludes that this is not necessarily true for all loading conditions and the number of stations involved in producing the offered load does in fact decrease performance [O'R83].

Ethernet was originally developed to transport non real-time traffic between users. Typical applications of this type are file transfers (FTP) and e-mail [Chr98]. Users often did not perceive the inherent latency in transferring information. However, as the needs of the network users changed, as with the increased use of real-time multimedia applications, delay becomes a large factor in the user's perception of performance and gives rise to the term "world-wide wait" [Meh96] for web-based applications.

According to Christensen, delays in an Ethernet network with a high offered load can have lengthy and highly variable packet delays. These delays can adversely affect data throughput in windowed flow controlled protocols such as TCP/IP [Chr98]. These delays are caused by a phenomenon called the 'capture effect' which has been studied extensively in [HM95] [RY94]. In a highly loaded network, one single station with packets to send can dominate or 'capture' the network such that the other stations do not have the opportunity to transmit despite the fact that they have packets to send. Christensen [Chr98] studied the effects of changing the BEB portion of Ethernet to decrease packet delay and provide better services for real-time applications.

There is no theoretical limit on maximum transfer delay. The random backoff and access algorithm in the Ethernet protocol can render the channel unstable [Pic86]. As traffic load increases and the channel becomes unstable, the number of collisions will also increase and the throughput will approach zero [Pic86]. As more stations attempt to transmit on the network, the network will become saturated and every attempt to transmit will result in a collision or backoff. Because stations cannot access the channel to transmit due to collisions or backoff, user throughput approaches zero.

The CSMA/CD protocol is very complex. Analytic models use a variety of simplifying assumptions such as "balanced-star configuration, finite populations, unimodal or constant packet lengths, small packet sizes, and no buffering to obtain tractable results" [Wan96]. Because of these assumptions, these analytic models can have very misleading results. In [TK85], the maximum achievable throughput for CSMA/CD is 60%. However, in a study by Smith and Kain [SK91] measured Ethernet performance differed significantly from predictions made by typical analytic models. In measurement of a real Ethernet, Boggs and Mogul measured a throughput of 97% for large packet sizes [BM88].

Much of the previous work in Ethernet performance was based on analytic models and simulation. However, there have been some measurement studies as well. Measurements of real Ethernet networks are needed to avoid the simplifying assumptions mentioned previously. Boggs and Mogul presented measurement data on an Ethernet in [BM88] showing the effects of packet lengths, network lengths, and numbers of hosts. It was shown that Ethernet is capable of performing "adequately" [BM88] for high-bandwidth applications when response time is not closely constrained. In [Gon87], Gonsalves measured the performance of an operational 3 and 10 Mbps bus structured Ethernet LAN. This investigation explored the effects of packet sizes and offered load on the throughput and delay metrics of the system. In general, as the packet size increased, throughput and delay also increased with respect to the offered load. The results of [Gon87] are used for the validation models used in this research.

2.5.5 Summary of Research in CSMA/CD Network Performance. The literature reviewed on CSMA/CD identified design parameters which impact throughput and delay performance. These are channel length, packet length, medium bandwidth, the number of stations connected to the medium, and the arbitration method. Changing the arbitration method from a random BEB to a more deterministic method as in [Chr98] [Chr96] served to decrease the mean and variance in packet delay in an Ethernet network. However, Ethernet can still suffer decreased throughput as offered load increases.

2.6 Code Division Multiple Access

2.6.1 Spread Spectrum. Spread spectrum (SS) communications originally began in the 1950s in development of military guidance and communication systems [Sch82]. SS is so named because the transmitted bandwidth is much wider than the minimum bandwidth required to send the information. Originally, SS was used because of its inherent noise immunity and jamming resistance. However, a relatively new multiple access technique, code division multiple access (CDMA), is one of the principle research results of SS development [Sch82]. Rather than partitioning the communications channel in time slots (TDMA) or frequency (FDMA), CDMA techniques were developed as a hybrid of the two [Skl88].

SS multiple access techniques such as CDMA allow multiple signals to occupy the same radio frequency (RF) bandwidth and be transmitted simultaneously without interfering with one another, provided orthogonal spreading sequences are used [SSM99]. There are two main types of CDMA: direct sequence (DS) and frequency hopping (FH). Direct sequence CDMA (DS/CDMA) signals are generated by adding the information bits (modulo-2) to a spreading code, or pseudonoise (PN) code. This sequence is then transmitted using traditional modulation techniques such as binary phase-shift keying (BPSK) or quadrature phase-shift keying (QPSK) [She82]. Frequency hopping CDMA (FH/CDMA) uses the PN code differently. Rather than directly spreading the transmission signal through modular addition, FH/CDMA uses the code to determine successive frequency sets, changing from one frequency to another. Thus, the signal is 'spread' or 'hopped' across many different frequencies.

2.6.2 DS/CDMA Transmission. The basic DS/CDMA operation is illustrated in Figure 2.2. In Figure 2.2, both a sender and receiver attempt to communicate. A binary data signal, $x(t)$, enters a non-return-to-zero (NRZ) level encoder readying the information signal to be spread. Normally, the data sequence is a binary bit stream in the form of ones and zeroes. In CSMA/CD, Manchester encoding is

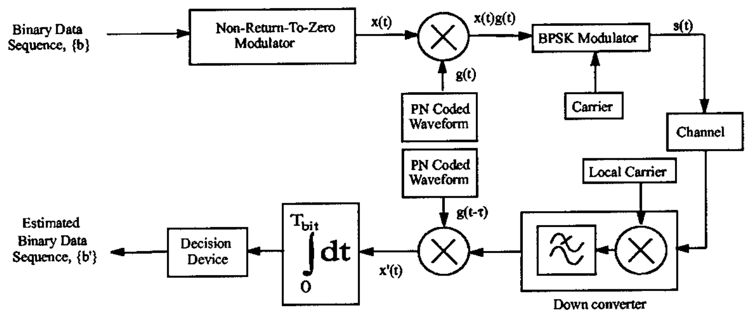

Figure 2.2. Basic DSSS System Model with BPSK Data Modulation

used to ensure that there is a signal transition for every bit of information. Further information can be found in [Tan96]. The NRZ encoder converts this stream to a series of plus and minus ones and is sometimes said to be a bipolar pulse waveform, where a binary one is a minus one and a binary zero is a plus one. The information bit stream, $x(t)$, is then multiplied with the PN code, $g(t)$. Since $g(t)$ switches at a much faster rate (i.e, has a higher frequency) than $x(t)$, the signal is said to be 'spread' since the signal is now increased or 'spread' to a higher frequency. This higher frequency is also know as the spreading or chipping rate. The signal $x(t)g(t)$ is now modulated. In this example, it is transmitted using BPSK modulation and placed on a carrier frequency. The resulting spread signal, $s(t)$, is then transmitted over a channel to a receiver, in this case free-space. While propagating through the channel, the signal can have noise added, such as background noise or interference from other transmission sources. The received signal is detected and demodulated. It is then correlated with a replica of the original spreading code $g(t)$. Because the correlation is occurring on the receiver and not on the originating station, there may be a slight synchronization problem resulting in a phase difference on the receiver. Assuming perfect synchronization, the original signal is extracted and sent to an integrator and decision device. These last two stages in Figure 2.2 serve to aid in making $x'(t)$, the received decoded information stream, estimating the original transmission. Due to noise levels and interference, it is possible for the integration and/or decision device to misinterpret the data and incorrectly data bit (a bit error).

To graphically illustrate this process, an example of a direct sequence binary phase shift keying (DS-BPSK) modulation and demodulation, as described above, is shown in Figure 2.3.

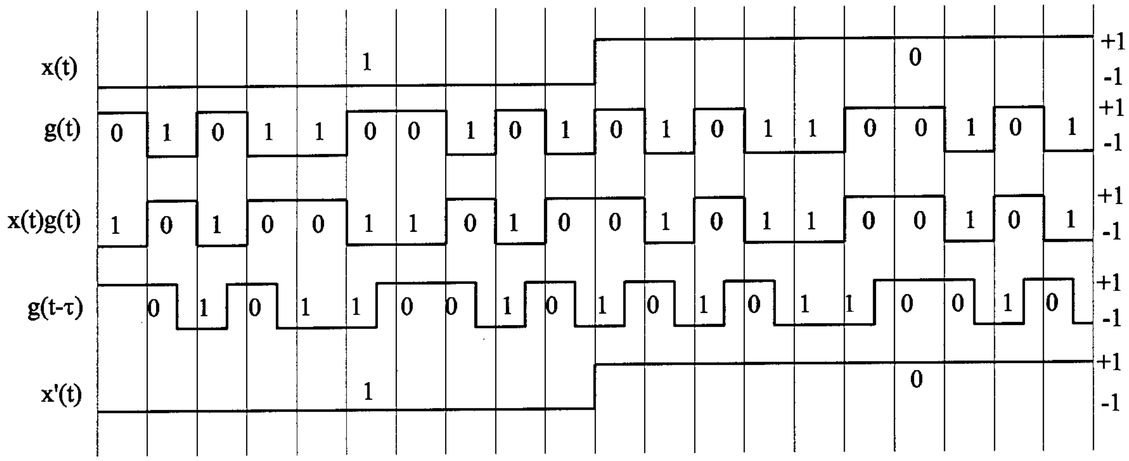

Figure 2.3. Example Modulation and Demodulation of a DSSS Signal

In this case, the PN coded waveform, $g(t)$, has 10 chips per data bit. The pulse duration of the data, x(t), is ten times longer than the duration of the spreading or chipping interval. The transmitted BPSK signal is $x(t)g(t)$. The PN coded waveform at the receiver is g(t-τ). The τ represents a time deviation since the waveform is located on a physically different station. In order to simplify calculation, it is assumed that the system employs perfect power control. In this case $\tau = 0$. Using modulo-2 addition, it can be seen that the application of the code spreads the data and the reapplication 'despreads' the signal resulting in an estimation of the original signal.

A multiuser DS/CDMA system is obtained by extending the single-user spread spectrum system through the application of different spreading codes for each user. A CDMA channel with K users sharing the same bandwidth is shown in Figure 2.4. Users $1 - K$ data signals are spread with associated spreading codes g_{1-K}. These signals are then transmitted and summarily added in the transmission channel. The channel, usually free air, also adds in a noise component as well. The signal is then

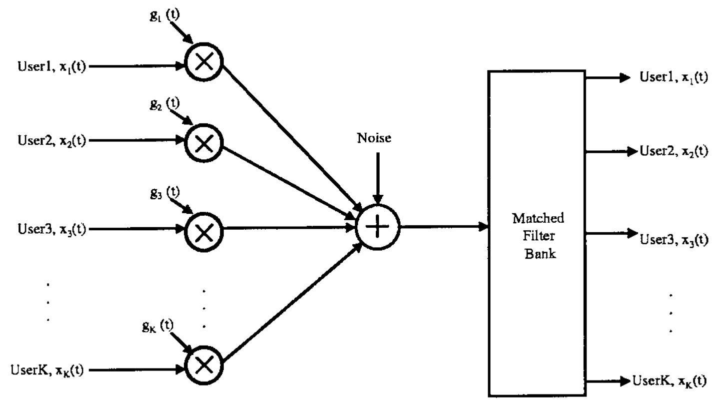

Figure 2.4. A CDMA Channel Model With K Users

received at a receiver. The received signal is simply a sum of all the spread signals and the noise. This conglomeration of signals is then passed through a matched filter bank. In this bank is the demodulation and correlation components as described in Figure 2.2. If this particular receiver is set to receive a message from User 1, then its matched filter will use User 1's spreading code to despread the signal. The despreading process will extract User 1's data stream while simultaneously reducing the interference effects of the other users since their codes do not match [Vit95]. Essentially, the despreading operation multiplies the desired signal and thereby raises it above the noise floor. At the same time, this action similarly reduces the interference and noise power components resulting in those signals being enshrouded in the noise floor. Provided that the processing gain is such that the desired signal's power is sufficiently above the noise floor, the signal will be properly received and demodulated.

2.6.3 CDMA Operation. In some CDMA systems, transmitting stations use a slotted-ALOHA based channel access mechanism to get a PN code allocation from the base arbitration station [PRAS99]. Once a code is obtained, the station transmits data on a separate transmission channel. Because transmissions occur simultaneously on the data channel (provided the noise floor is below a set threshold), "the performance of the MAC protocol should be considered for efficient data transfer" [PRAS99]. An alternative to the base arbitration station is to preassign PN codes to stations and distribute a directory of assignments among stations in the network.

2.6.3.1 Direct Sequence Versus Frequency Hopped. Direct sequence (DS) and frequency hopping (FH) are the most commonly used methods for spread spectrum communications [Vit95]. Although the basic idea is the same, these two methods have many distinctive characteristics that result in completely different operational performance.

The FH technique does not spread the signal and, as a result, there is no processing gain. Processing gain is the increase in power density obtained when a signal is despread. It improves the received signal's signal-to-noise ratio (SNR) [Vit95]. As a result, the FH technique needs more power in order to have the same SNR as a DS signal.

FH is also more difficult to synchronize since both the receiver and the transmitter must be synchronized in both time and frequency. DS, on the other hand, requires only that the timing of the chips be synchronized. The FH technique requires more time to search for the signal to lock to it. As a result, the signal latency acquisition is generally longer. DS allows a receiver to lock-in the chip sequence in just a few bits.

FH is better than the direct sequence radio when dealing with multipath, however. Since the hopper does not dwell on the same frequency, a null at one frequency in the sequence may not be a null at another frequency in the sequence. Therefore, a hopper can usually survive multipath better than a direct sequence radio.

The frequency hopper is more popular for voice than data communications. A frequency hopper can typically carry more data than a direct sequence radio since the signal is narrowband [Skl88]. A data system must have an error rate better than 10^{-4} while in general, a voice system can only survive an error rate as high as 10^{-2} [Skl88]. Voice systems can tolerate more data loss because the human brain can 'guess' between the words while a microprocessor cannot. As a result, even with implementation of error checking/correcting coding, the FH system is preferred.

Although DS typically has a lower data rate than FH, it is widely accepted to be easier to implement due to ease of synchronization. Furthermore, the use of processing gain helps to lower power consumption which is important in applications such as cellular phones. It also has better applicability to data communications due to comparably lower error rates versus FH [PZB95].

2.6.3.2 Power Control. Power control attempts to adjust signals inbound to a receiver such that they have the same power as all other signals from interfering stations. In wireless networks, power fade due to movement, distance, or location is a big problem [Skl88]. If one mobile station transmits with some power X and another with some power Y and $X \gg Y$, the intended receiver may not be able to correctly receive the communication from Y. The power in Y may be such that the multiplication by despreading will only raise the signal slightly above the large interference caused by X. And similarly, the reduction in strength of X by despreading may not be enough to detect the relatively small signal produced by Y. This is often referred to as the "near-far" problem [Sch82]. By using a power control mechanism, the near-far problem can be mitigated and provide the "optimum capacity for CDMA cellular systems" [Kas88]. Even though perfect power control is not attainable, in practice most research assumes perfect power control [Kas88] [She82].

2.6.3.3 Noise Versus Interference. Since CDMA is noise resistant, it is a good choice for wireless communications. In wireless applications, there are many sources of noise inherent in an unguided medium. Background noise, cochannel interference, jamming sources, and thermal noise all play a part in the resultant SNR. Because the PN code is noise-like, as more transmitters are added to the network, their transmissions appear to receivers as additional noise.

2.6.3.4 Pseduonoise Codes. A PN code sequence is a series of units called chips consisting of $1's$ and $0's$ (in binary form) or $-1's$ and $+1's$ (in polar form). They act as a noise-like, but deterministic, carrier used for the bandwidth spreading of the data signal [Vit95]. Spreading is achieved by combining every data symbol with a complete PN code. Since the chip rate is usually much higher than the data rate, this spreading results in the signal energy being spread across the spectrum or the data stream occupies a much larger bandwidth than it originally had. In order to be used for direct sequenced communications, PN codes must have the following properties:

- The sequences must consist of 2-leveled values, i.e., 1's and 0's or -1's and +1's depending on notation.
- The codes must produce a 1-chip wide autocorrelation peak. This facilitates code-synchronization and has properties similar to white Gaussian noise (WGN).
- The codes must have low cross-correlation values. This must be true for both full-code correlation and partial-code correlation.
- The codes should be 'balanced'. Code balance means that there can only be a difference of up to one between the number of 1's and 0's in a code. This is required to allow equal spreading of the energy over the entire frequency band.

There are two ways of separating users in CDMA: orthogonal and nonorthogonal coding. Orthogonal codes have the property that each must be synchronized with the intended receiver since each code used will only be orthogonal if they are aligned in time. Nonorthogonal codes are asynchronous. Receiving stations can synchronize with their respective transmitters by aligning on the autocorrelation peaks [VIT 95].

Hadamard-Walsh (also known as Walsh-Hadamard) (WH) codes are considered orthogonal. They have excellent cross-correlation characteristics making them extremely useful for reducing multiple access interference (MAI) from other users. Being orthogonal, there is ideally zero MAI. However, although full code cross-correlation is zero, partial code cross-correlation is not.

WH codes do not have a single auto-correlation peak. Thus, there is the possibility of multiple auto-correlation peaks making it impossible for synchronization without some external means. If perfect synchronization is not achieved, then non-zero cross-correlation peaks result attributing to a partial code synchronization. This causes unsynchronized users to interfere with each other. Other problems also arise due to the fact that WH codes do not have adequate spreading behavior. The spreading is not over the entire bandwidth. They do not spread data as well as PN sequences because the power spectral density of WH codes are concentrated in a small number of discrete frequencies [She82].

Shift register sequences are considered non-orthogonal and exhibit auto-correlation properties with relatively high cross-correlation sidelobes [She82]. However, they do have a narrow auto-correlation peak. These types of sequences are generated by using a shift register with feedback taps. By using a single shift register with specially selected feedback taps, maximum length sequences (M-sequences) can be obtained. A shift register of size n will produce a code length

$$N = 2^n - 1, \tag{2.10}$$

where N is the code length.

Gold codes are constructed from the modulo-2 addition of two M-sequences also known as a 'preferred pair'. By shifting one of the two PN sequences, a different

Gold code is produced. This property can be used to generate many codes which will permit multiple access on the channel [Skl88].

Gold codes are a popular implementation. They have low cross-correlation values and have a large family size, M, where

$$M = 2^n + 1. \tag{2.11}$$

The code length is defined by (2.10).

Gold codes have only three cross-correlation peaks as given in Table 2.2 [Hay94] where m is the shift register length and N is the period or code length. As the length

Table 2.2. Three-Level Cross-Correlation Properties of Gold Sequences

m	N	Cross-Correlation	% Occurrence
m is odd	$N = 2^m - 1$	$-\frac{1}{N}$	~ 0.50
		$\frac{-2^{\left(\frac{m+1}{2}\right)}-1)}{N}$	~ 0.25
		$\frac{2^{\left(\frac{m-1}{2}\right)}+1)}{N}$	~ 0.25
m is even and not divisible by 4	$N = 2^m - 1$	$-\frac{1}{N}$	~ 0.75
		$\frac{-2^{\left(\frac{m+1}{2}\right)}-1)}{N}$	~ 0.125
		$\frac{2^{\left(\frac{m-1}{2}\right)}+1)}{N}$	~ 0.125

of the code increases, these cross-correlation peaks become less of a problem. The most "powerful" property of Gold codes is that they have a single auto-correlation peak at zero, which makes it very effective for synchronization and detection [She82].

If a Gold code is combined with a decimated or sampled version of one of the basis M-sequences used to form it, a Kasami code is produced. Kasami codes have the same correlation properties of Gold codes. The difference is that it produces a larger set of codes. The family size of these codes is

$$M = 2^{\frac{n}{2}}\left(2^n + 1\right). \tag{2.12}$$

The larger the code family size, the more code addresses that can be created. This sets a limit on the number of users that can use the system. Also, a large family size allows selection of those codes that show desirable cross-correlation characteristics.

2.6.3.5 Synchronization. For proper operation, a DS/CDMA system requires the PN code at the receiver to be synchronized to the transmitting station. This ensures the highest autocorrelation power and thus produces the highest processing gain [Skl88]. A higher processing gain results in a higher SNR that will result in more simultaneous users.

2.6.4 CDMA Performance Studies. Traditional cellular CDMA systems are setup to have a single base station with mobile units within its transmission range. There are two frequency channels for the network; an access channel and a data channel. The access channel serves to assign a PN code to a mobile unit from the base station to either notify the unit of the PN code used for an incoming call or assign a PN code to the unit when it is ready to initiate a call. The PN code is randomly chosen from a pool of available codes. If there are no codes available, then the user may not get a dial tone or service. Once the code is received, the mobile unit can exchange information with the base station on the data channel using the assigned PN code. Communication is established from mobile unit to base station and vice versa as identified in Figure 2.5

Perez-Romero [PRAS99] found that for bursty sources such as multimedia and web service, purely random access to the access channel was efficient. However, for longer transmissions such as FTP, the randomness in attaining the PN code must be made more deterministic. Since code acquisition must be performed for each packet, once a station has successfully transmitted with a randomly attained code, it should keep that same code rather than give it up. Otherwise, collisions resulting for code allocation would increase [PRAS99].

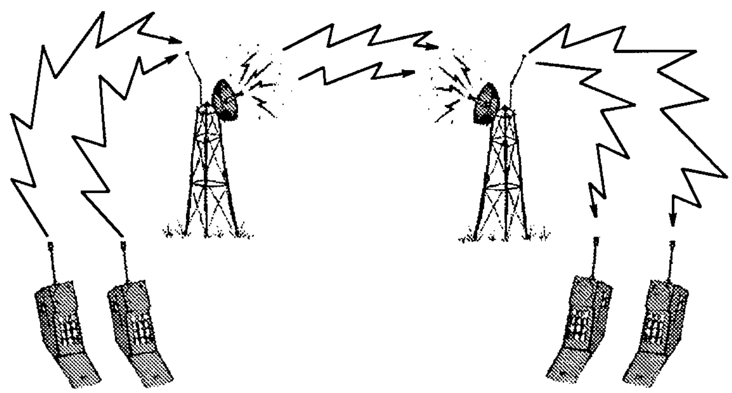

Figure 2.5. Conceptual Cell Phone Communication Link

However, dedicated code sequences assigned to separate users are inefficient in terms of channel utilization and throughput. Not every station will always have data to transmit. So if there is a user that does not have data to transmit, then the assigned code is wasted since the channel is not being utilized by that station.

Suppose there are a total of 64 codes available for a particular cell and each user has an assigned data rate of 9.6kbps. This means that there are at most 64 users who can transmit simultaneously for an overall throughput of 64 x 9.6kbps = 614.4kbps. Now suppose that there are six more stations who enter the cell and wish to transmit. Since all the codes are assigned, those six users are not able to transmit. Since the codes have been statically assigned, even when the six users currently in the cell finish their transmissions, the base station will not allocate those codes to the new users in the cell. Therefore, when the six completed users stop transmitting, the overall throughput is reduced to 58 x 9.6kbps = 556.8kbps and there is (614.4-556.8)kbps = 57.6kbps of wasted capacity.

Perez-Romereo demonstrated that if the network has few users, then the data rate for each station can be redefined to maximize the channel. Conversely, if there are a large number of users, lower data rates should be defined to maintain the same SNR level through the processing gain - assuming the bandwidth and other system parameters are held constant. Thus, if the codes are not dynamically assigned, there would be wasted utilization. This is why the assigned code allocation is inefficient [PRAS99].

DS/CDMA can have a greater capacity and support larger data rates than FDMA or TDMA [Gal94] and was chosen as the preferred radio transmission technology for the next generation wireless systems employing the IMT–2000 standard [JJ99]. However, this performance is achieved through joint-decoding at the receiver and is comparatively more complex to FDMA or TDMA.

Using the less complicated approach of linear multi–user detectors and single user decoders, Erkip and Aazhang determined that the capacity of DS/CDMA compared to TDMA and FDMA was about equal [EA98]. This study found that the probability of not being able to access the channel and the delay limited capacity was better in DS/CDMA than in the other orthogonal multiple access schemes as the offered load increased.

However, if the SNR is held constant for TDMA and compared to CDMA, Sari found that TDMA has a 10 dB advantage over CDMA and that their bit-error-rate (BER) were about equal assuming a nondispersive channel, perfect synchronization, and power control [SSM99]. Sari's results are based on an upper limit of 64 users in the cell and more ideal assumptions. Field trials conducted by Qualcomm Inc. in 1996 on an actual CDMA implementation showed 10-20 times capacity improvement over FDMA and TDMA [Ano96] and support Gallagher's claims of improved capacity with CDMA.

According to a study by Xu, "The allocation of the CDMA access channels can significantly affect the overall system performance" [Xu99]. If there are too few access channels, the system cannot be accessed even though there may be traffic channels available. Conversely, if there are too many access channels, the number of extra transmitting stations creates too much interference and wastes the RF resource. Xu found that CDMA has a 47% capacity improvement over slotted–ALOHA. He further concluded that the backoff time and number of times a station requests a channel before retransmission has a small influence on performance when the channel throughput is low or medium [Xu99].

Channel performance can be measured by channel throughput, channel traffic load, and channel capacity. The parameters which affect these performance characteristics, Xu identified, are the number of slots a stations waits after a collision is detected and how many times it probes the channel before it can retransmit [Xu99].

Sari quantifies performance for CDMA by analyzing BER with respect to the number of users. In a similar study, Zeger, identifies power and interference (both affecting BER) as principal performance parameters [ZN99]. Zeger determined as more users are added to the system, the frame error rate increases due to an increase in signal and noise power. If one holds the frame or BER constant, then the power available for other users is decreased, and thus the probability of blocking is increased since the total interference power would exceed the background noise power [ZN99] [Vit95].

In a study by Ramakrishna and Holtzman, the information rate can be increased by two methods. A single spreading code can be used. This would increase the bit rate at the expense of processing gain. Or, multiple orthogonal codes can be employed. These codes would effectively separate the single data stream into several data streams and then transmit them in parallel. This is also known as multi-code CDMA [RH98]. Either method is equally viable for increased throughput and both have the same effect on other users BER and signal-to-interference ratio (SIR). Generally, as the data rate is increased, more transmit power is needed and the interference to other users becomes larger [SK98].

The underlying factor affecting the performance of a CDMA system is the BER. The bit error rate (BER) is "one of the basic measures of performance for a CDMA system" [Let94]. Relating the signal-to-noise ratio (SNR) to the BER and applying the central limit theorem, the probability of bit error, P_e, is [Let94], [Skl88]

$$P_e = Q\left(\sqrt{2SNR}\right), \tag{2.13}$$

where Q is the complementary error function defined by as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{x^2}{2}}. \tag{2.14}$$

If the BER (or P_e) is high, this is an indication of a low SNR. This means that the signal of interest cannot be successfully extracted from the noise floor. Since the BER is fundamental in CDMA system performance, there have been numerous studies on the calculation of error probabilities for DS/CDMA systems [Let94], [LP87], [Hol92]. One of the most common approximations used for BER calculation is the Gaussian approximation

2.6.4.1 Gaussian Approximation. The predominant estimation for probability of bit error or bit error rate, P_e, is the Gaussian approximation and is [LP87]

$$P_e = Q\left(\frac{k-1}{3N} + \frac{N_0}{2E_b}\right)^{-\frac{1}{2}}, \tag{2.15}$$

where k is the number of simultaneous transmitters, N is the code length, N_0 is the noise power spectral density, and E_b is the energy per bit. However, this approximation is generally inaccurate, and particularly as N increases, the BER estimate becomes more optimistic. This yields results that give more bit errors than expected [LP87].

2.6.4.2 Improved Gaussian Approximation. An improved Gaussian approximation has been proposed [Hol92] that assumes perfect power control and random signature sequences and is

$$P_e = \frac{2}{3}Q\left[\left(\frac{k-1}{3N}+\frac{N_0}{2E_b}\right)^{-\frac{1}{2}}\right] \tag{2.16}$$

$$+\frac{1}{6}Q\left[\left(\frac{(k-1)(\frac{N}{3})+\sqrt{3}\sigma}{N^2}+\frac{N_0}{2E_b}\right)^{-\frac{1}{2}}\right]$$
$$+\frac{1}{6}Q\left[\left(\frac{(k-1)(\frac{N}{3})-\sqrt{3}\sigma}{N^2}+\frac{N_0}{2E_b}\right)^{-\frac{1}{2}}\right],$$

where

$$\sigma^2 = (k-1)\left[\frac{23N^2}{360}+\left(\frac{1}{20}+\frac{k-2}{36}\right)(N-1)\right]. \tag{2.17}$$

In [Hol92], numerical results have shown that (2.16) has accurate and consistent results compared to earlier research. However, its BER estimation underestimates BER for small values of N but is more accurrate for larger values of N.

2.6.5 Current CDMA Implementations. DS/CDMA is currently used in predominantly wireless environments. The two most prevalent of these environments are wireless personal communications systems (PCS) [RH98] and wireless Ethernet LANs as defined by IEEE 802.11 [IEE97]. Some newly emerging applications include data over cable service interface specifications (DOCSIS) [Lab99] and powerline communications networks [Str96]. Although primarily employed in the wireless arena, CDMA has shown an emergence in the wired communications realm as evidenced by the latter examples.

2.6.6 Summary of DS/CDMA Research. CDMA systems have two main channels: an access channel and a transmission channel. Many use a slotted-ALOHA MAC protocol in the access channel to gain permission from a base station to transmit on the transmission channel. Once permission to transmit has been granted, the data rate is fixed for the system. Since users share the channel simultaneously and each station's data rate is defined, there is no reduction in throughput characterized by wasted bandwidth due to collisions. However, where Ethernet has an electrical limit on the total number of stations which can access the networks at the same time, CDMA systems suffer from an abrupt drop for the entire network once a certain threshold is reached. This point of system failure can be seen when the SNR or BER is such that throughput is zero. This point is usually a design parameter of the network.

Performance studies of CDMA systems principally revolve around the number of users who can access the network simultaneously (capacity). Capacity studies are based on the power levels and their effects on the BER of transmissions. Performance is also a function of the MAC in the access channel since the MAC governs the number of users who are allowed to the system.

2.7 Summary of CSMA/CD and DS/CDMA Research

Research on both CSMA/CD and DS/CDMA shows that these technologies are the most popular in wired and wireless communication networks respectively. Neglecting the effects of increasing the bandwidth of the transmission medium, the performance drawbacks in wired Ethernet LANs stem from delay induced by increasing the number of stations and thereby increasing the offered load while decreasing data throughput. The performance of CDMA networks is a function of the number of users or the background noise in the channel and the arbitration performance of the access channel.

CSMA/CD is used primarily in wired networks and similarly, CDMA is primarily used in wireless voice and data networks. No studies were found that use CDMA in wired LANs. However, CDMA has been used in broadband cable for the transmission medium in cable television (CATV) networks and signaling in powerline networks. Little research in bridging the implementations between wired and wireless MAC protocol networks has been done.

Wireless communications is becoming a viable solution to implement computer networks [Rap96]. Optimization of channel bandwidth through innovative protocols is allowing increased data rates to rival the speeds of the early wired networks. Due to the fact that communication through the wireless channel is far more complex and has limited resources, this optimization was necessary whereas in wired networks optimization was not as large a concern. Further research is needed to bridge the gap between the wired and wireless MAC protocols to further increase wired network performance without the need to enhance or replace the already established cabling infrastructure.

2.8 OPNET

OPNET is a powerful network simulation and modeling tool [Tec97]. It uses a layered hierarchy to model the different effects a packet suffers while being transmitted over a bus, point-to-point (PPP), or radio connection. Virtually any network device, component (i.e, workstation, server, router, satellite), or communication system can be modeled to predict performance. Users may modify existing models or create whole new node or process layer models to simulate real world or prototype devices. The resulting components can then be interconnected and simulated.

2.8.1 Design Tool. Most OPNET models can be classified as systems composed of multiple subsystems that interact with each other. The subsystems' interactions rely on communication resources to exchange information. These communication resources may be required between two physically distinct entities or between logically linked entities based on their functional area. The most prevalent form of communications in OPNET models is based on messages that can carry information between subsystems called packets. Packets are data structures defined by OPNET, which are treated as "objects that can be created, modified, examined, copied, sent, received, and destroyed" [Tec97].

The basis of OPNET operation rests in C/C++ code also known as Proto-C [Tec97]. This code allows the user to precisely define how a model behaves. OPNET specific functions called Kernel procedures allow a user to manipulate packets and models to achieve the desired behavior. More specifically, operations dealing with packet transmission and reception are key to implementation of a DSSS LAN since there is a subtle difference between this type of network and a traditional radio network.

2.8.2 Transceiver Pipeline. Radio links provide a broadcast medium where each transmission can affect multiple receivers throughout the network model. The radio link in OPNET is implemented as a transceiver pipeline. This pipeline is a 14–stage process that operates on a packet as it travels from a transmitter to a receiver. The pipeline is split with 6 stages in the transmitter (Stages 0–5) and 8 stages in the receiver (Stages 6–13) as seen in Figure 2.6 [Tec97].

These stages are a series of functions designed to operate on specific packet attributes through the course of a transmission. Every type of communication link (i.e., bus, PPP, or radio) has an associated transceiver pipeline. Each pipeline differs based on the type of link in the assumptions it makes on those attributes. Since radio transmission is so complex, the radio transceiver pipeline has the most stages and does not eliminate any stages based on simplifying assumptions. This is the pipeline that is used for DSSS LAN enhancement. The simulated CSMA/CD network utilizes the simplified bus transmission pipeline.

2.8.2.1 STAGE 0 – RECEIVER GROUP. The receiver group is not actually part of the pipeline. It is executed once per transmission in order to determine eligible receivers. Even though the radio environment is broadcast in

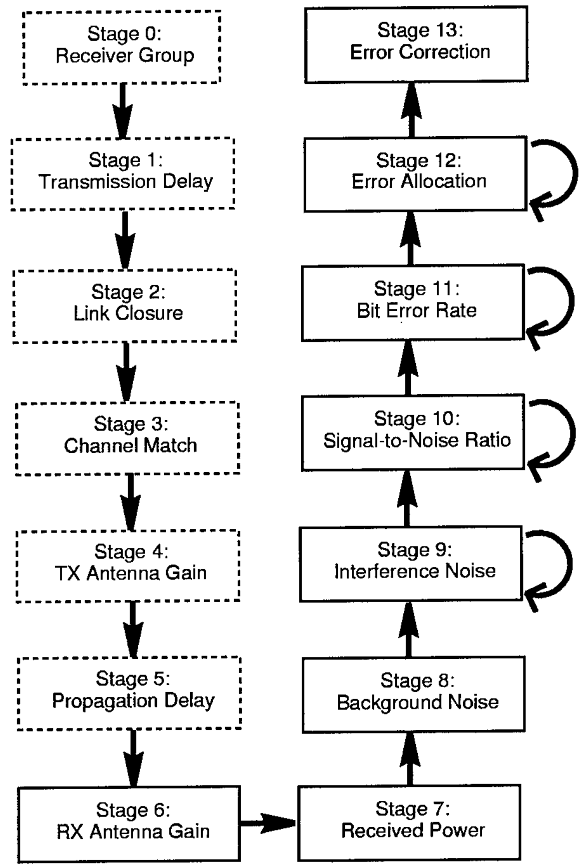

Figure 2.6. OPNET Radio Transceiver Pipeline

nature, there are reasons why certain receivers should not be considered. Some examples of this include:

- Disjoint frequency bands: If the receiver and transmitter are in two separate frequency bands, then the transmission does not affect the other as either noise or a valid signal.
- Physical separation: The receiver may be too far away from the transmitter to establish a link. This could also be due to obstacles in the environment.
- Antenna nulls: Antenna gains can be significantly reduced if directional antennas are used and not pointed to the proper location.

2.8.2.2 STAGE 1 - TRANSMISSION DELAY. The transmission delay stage is invoked for each new transmission. This calculation is shared for all resulting pipelines created between the transmitter and receiver. This stage calculates the amount of time it takes to transmit the entire packet. This result is the simulation time difference between the beginning of transmission of the first bit and the end of transmission of the last bit in the packet. The transmission delay is saved in the packet attribute *OPC_TDA_RA_TX_DELAY*.

2.8.2.3 STAGE 2 - LINK CLOSURE. The link closure stage is invoked once for each receiver in the transmitting station's receiver group. The purpose of this stage is to determine whether a particular receiver can be reached by a transmission. The ability of the transmission to reach the receiver is called closure [Tec97] where reach is defined as the point at which the transmission can be received by the intended receiver. There are several ways that a signal may not reach a receiver to include too low transmit power, obstacles in transmission path, and mismatched transmission parameters. This stage does not attempt to determine if a transmission is valid or not, but checks to see if the transmitted signal can affect a receiver channel. In effect, it applies to interference jamming as well as desired signals.

2.8.2.4 STAGE 3 - CHANNEL MATCH. The channel match stage is executed once for each receiver that satisfies the specifications in Stage 2. The purpose of this stage is to classify a transmission as either valid, noise, or ignored. Valid packets belong to a receiver channel carrying the desired signal. Noise packets may or may not be valid transmissions in the network, but are considered interference to the desired receiver since the receiver is not locked to the desired signal. Ignored packets are transmissions that do not affect the desired receiver.

2.8.2.5 STAGE 4 - TRANSMITTER ANTENNA GAIN. The transmitter antenna gain stage is executed separately for each receiver except those that failed link closure in Stage 2 and channel match in Stage 3. The purpose of this stage is to compute the gain associated with the transmitter's antenna based on the direction it is pointing and its type. Antenna gain increases or reduces a transmitted signal's energy due to the physical characteristics of the antenna. Antennas that provide no gain are called isotropic since they have perfect symmetry in radiated power in all directions.

2.8.2.6 STAGE 5 - PROPAGATION DELAY. The propagation delay stage is invoked for each receiver that successfully passed the criteria for both link closure and channel match. The purpose of this stage is to calculate the amount of time required for the packet's signal to travel from the radio transmitter to the radio receiver. Generally, this result is dependent on the physical separation between the source and destination components.

2.8.2.7 STAGE 6 - RECEIVER ANTENNA GAIN. The receiver antenna gain stage is invoked for each eligible receiver. Gain is computed at the time the leading edge of the packet (i.e, the first bit) arrives at the receiver's location. It is similar to the calculations performed by Stage 4.

2.8.2.8 STAGE 7 - RECEIVED POWER. The received power stage is executed separately for each eligible receiver. The purpose of this stage is to calculate the received power of all signals arriving at the receiver. This calculation is based on the packet's transmitted power, antenna gains, separation distance, and frequency of transmission.

2.8.2.9 STAGE 8 - BACKGROUND NOISE. The purpose of the background noise stage is to characterize the effects of all noise sources except those due to concurrently arriving transmissions. Concurrently arriving transmission are accounted for in Stage 9. The typical background noise sources are thermal noise, emission from neighboring electronics, and otherwise unmodeled radio transmissions. OPNET characterizes background noise as the sum of both thermal noise and a constant ambient noise which can be considered additive white gaussian noise (AWGN). First, the thermal noise is calculated by

$$ThermalNoise = ((N_f \cdot T) + T_b)\,(W)(k), \tag{2.18}$$

where N_f is the noise figure of the receiver, T is the temperature of the receiver, T_b is the background temperature taken to be 290^o Kelvin, W is the bandwidth the transmission occupies, and k is Boltzmann's constant given as $1.379 \times 10^{-23}\frac{J}{K}$. The ambient noise is taken as a constant noise component times the bandwidth of the signal given and is

$$BackgroundNoise = NW, \tag{2.19}$$

where N is the ambient noise level, 1.0×10^{-26} Watts, and W is the bandwidth the signal occupies. The sum of (2.18) and (2.19) comprise the total background noise.

2.8.2.10 STAGES 9 – 13. Stages 9–13 were combined in this section's discussion since each stage is invoked for each collision segment. Each stage may be invoked several times depending on the number of overlapping packets colliding in the channel.

The interference noise stage (Stage 9) is invoked under two conditions: a valid packet arrives at its destination while another packet is already being received; or a valid packet is already being received when another packet whether valid or invalid arrives. Although both of these situations sound similar, the difference lies in the type of the arriving packet. If the arriving packet is valid, then collision information must be updated for both the current and the arriving packet. However, if the arriving packet is invalid (i.e., destined for different receiver or otherwise considered noise) then only collision information is updated for only the valid packet.

Essentially, if there are overlapping packets upon reception, interference noise is calculated by summing the received power of the packets that collided. This stage may be invoked several times for the same packet. That is to say that a single packet can have multiple areas of overlap with other packets at various times as seen in Figure 2.7.

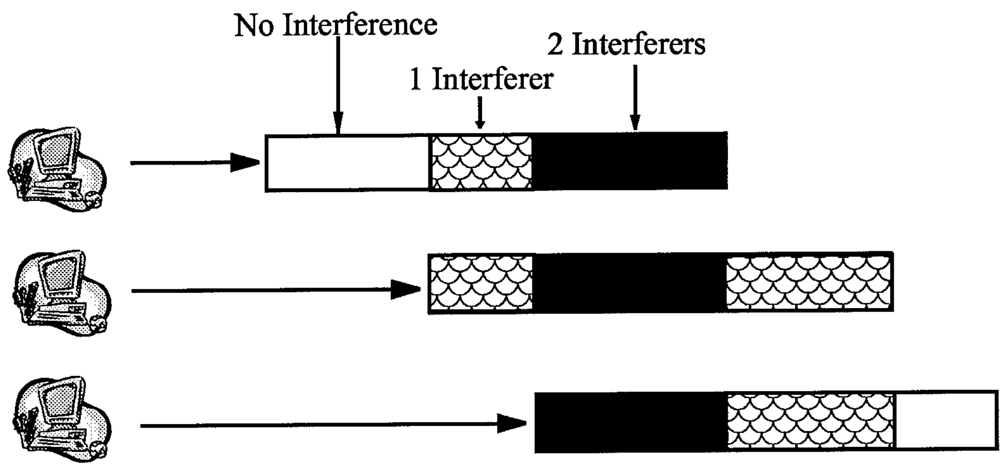

Figure 2.7. Multiple Areas of Collisions on Transmitted Packets

At each area of collision, there may be various numbers of overlap resulting in different values of interference noise power for a single packet.

The purpose of this stage is to account for the interactions between transmissions that arrive concurrently at the same receiver. The noise is accounted for by accumulating the power associated with the interfering packet in the noise value of the valid packet. Interference calculations are not needed for noise since link quality is not assessed.

The SNR stage (Stage 10) takes the interference noise values and is executed under three circumstances:

- A packet arrives at a receiver

- A packet is already being received when another packet arrives causing a collision
- A packet is already being received and another packet which is currently colliding completes reception

This stage is invoked many times depending on the number of overlapping packet transmissions. The three invocation circumstances define intervals over which the packet's average power is constant. As such, if there is interference in packet reception, the SNR must be recalculated to adjust for the increase in interference noise. The SNR value is based on earlier stages for received power, background noise, and interference noise.

The BER stage (Stage 11) is also executed under the three circumstances as defined in Stage 10. This stage may be invoked many times depending on the number of overlapping packet transmissions. The three invocation circumstances define intervals over which the packet's average power is taken to be constant. As such, if there are interference in packet reception, the BER, like the SNR, must be recalculated to adjust for the increase in interference noise. The purpose of this stage is to derive the probability of bit errors during the past interval of constant SNR. This is not the actual rate of bit errors, but the expected rate based on SNR and the type of modulation.

Following the BER stage is Stage 12, Error Allocation. The purpose of the error allocation stage is to assign the number of bits in error in a packet segment. This assignment is based on the bit error probability calculated in Stage 11 and the packet length. The higher the probability and longer the packet, the more probability of having bits in error.

Error Correction is taken care of in Stage 13. The error correction stage is invoked exactly once for each packet that is considered valid. The purpose of this stage is to determine whether or not the arriving packet will be accepted and forwarded to the respective node models. It models any error correction capability of the receiver. If the number of errors is less than or equal to a set threshold, the packet is accepted. Otherwise, it is rejected.

2.9 Summary

This chapter presented an overview of the operation of both Ethernet and CDMA communication networks. It also identified the network simulation and design tool, OPNET.

Ethernet has adapted to accommodate increasing network speeds up to 10 Gbps. However, this increase comes at the expense of replacing the installed cable base. This can result in a significant investment on the part of the network designer.

Spread spectrum communications has proven to be a viable wireless networking communication mechanism. Although primarily employed in a wireless architecture, there has been implementations in wired communications. However, these implementations do not account for the volatile nature of local area networks.

OPNET is a proven asset in network simulation and design. Its hierarchical method of model implementations makes it a powerful tool to truly emulate a full communication network. Assuming verified and validated model design, MAC level simulations of both Ethernet and the proposed DSSS LAN are possible through the use of OPNET.

III. Methodology

3.1 Background

In traditional wired LANs, the communication link between the transmitting and receiving stations is shared by allowing one station at a time access to the channel. In order to prevent simultaneous access to the medium, some form of arbitration is needed to allow each station exclusive access to the channel. This type of network access is used in the operation of Ethernet-based networks, the predominant network communication protocol used today [Chr98].

In an Ethernet network, only one station at a time is capable of transmitting data packets to another station. If the source station detects interference from a packet transmission from another station on the LAN, the source and interfering station(s) will terminate their transmissions and wait to retransmit at a later time. Therefore, information can only be sent in its entirety over the LAN if a certain amount of interference-free transmission can be sustained.

Inherent in this one-user-at-a-time protocol are several disadvantages. Even though for a small number of stations Ethernet is a suitable method of data transmission, as the numbers of stations on the LAN increases, the frequency of collisions increases as well [TH80]. This, in turn, increases the average time required for packet arrival which leads to a second disadvantage – poor performance in time-critical communication. Some forms of data communication such as email and file transfers, do not have arrival time constraints and reasonable delays can be tolerated. However, excessive delay in voice or video data renders it unusable [SH80]. One approach to solving this problem is to allow simultaneous access to the communication channel or a many-at-a-time capability. Thereby, LAN communications could become faster and more efficient [PSW92].

3.2 Problem Definition

Wired Ethernet-based LANs use carrier sense multiple access with collision detection (CSMA/CD) and binary exponential backoff (BEB) to arbitrate access to the physical medium. As more stations access the network, throughput is decreased and average delay is increased due to collisions. One obvious solution is to increase the bandwidth available in the network. This increases the number of slots available for transmission and decreases the number of collisions. Larger bandwidth results in higher data rates and therefore higher throughput. However, replacing existing cable plants with higher bandwidth media is a sizeable investment [Kle92]. Furthermore, signal propagation delay places an upper limit on network capacity in exclusive access networks whatever the bandwidth. This research explores ways to increase the performance of the wired Ethernet-based LAN without having to replace the cabling infrastructure. Rather than increasing capacity by increasing bandwidth, this research increases capacity by allowing simultaneous access to the medium through the use of spread spectrum modulation techniques.

3.3 Hypothesis

This research shows that a network using a direct sequenced code division multiple access (DS/CDMA) MAC can support more stations with a higher throughput than a comparable network using CSMA/CD. Since transmitting stations in a DS/CDMA network are separated by orthogonal spreading codes, each stations' transmission appears as noise to other stations. In this way, multiple simultaneous transmission streams can be supported. Since a successful transmission is based on the SNR of the transmission in the network, there is theoretically no limit to the number of stations that can access the network. Since there are no collisions in this network (unlike in CSMA/CD networks), as more users are added to it, there should be no degradation in user throughput.

3.4 Goals

The goal of this research is to show that the performance of a DS/CDMA multiple access protocol is significantly better than a comparable CSMA/CD network. This research also determines the practical upper limit on the number of stations that can be supported in each network.

3.5 System Definition

The systems under consideration are LANs for a small office or an academic computer lab. Both LAN configurations, Ethernet and DSSS, are modeled with a maximum of 30 network stations per network. Specifically, the performance study evaluates both configurations that contain a variable number of stations ranging from 2 to 30 stations. There are always an equal number of transmitting and receiving stations to setup virtual 'conversations' throughout the network. All stations are homogeneous. Individual capacity of the stations (i.e, processor speed, memory, etc.) were not considered as parameters in this study. The 'generic' nature of all stations lends itself to focus on the performance evaluation of the LAN configuration only. Simulation time was set at 10 minutes based on pilot studies for performance responses. No further increase in simulation time produced significant changes in the simulation results (i.e., the simulation reached steady state such that the deviation in responses as the run continued was less than 1%).

3.6 Approach

OPNET V7.0, a network simulation and design tool, is used to model and simulate both the Ethernet and DSSS LANs. First, an OPNET model of a DS/CDMA wired network was developed and tested. This model is then used to characterize the throughput, delay, and error rate of the network with respect to the offered load and number of transmitting stations. These metrics are further defined in Section 3.8.

3.7 System Boundaries

There have been many studies on the performance of Ethernet and DS/CDMA networks. However, significant use of DS/CDMA has only been in wireless or wide area networks. The operating environments are much different than that of a wired LAN.

The focus of this research is in the MAC sub-layer since this is the principal difference between CSMA/CD and DS/CDMA. The key component under study is the MAC arbitration scheme defined as either DS/CDMA or CSMA/CD. Since wireless spread spectrum networks have the ability to accommodate multiple simultaneous users in the same single transmission channel, the MAC protocol in wired networks is changed to allow this as well. Because the focus is at this particular layer, some operating assumptions have been abstracted out. The DS/CDMA network model has the following attributes:

- Each station taps a single bus channel,
- The only noise on the network is due to other transmissions,
- Perfect power control, and
- Perfect synchronization.

To baseline the performance for comparison, an IEEE 802.3 Ethernet running on 10Base2 copper wire in a bus configuration is used to characterize the CSMA/CD network. UTP and optical fiber were also considered, but are dismissed for the following reasons:

- UTP was considered since most current 10/100 Mbps Ethernet LANs use this cabling. However, it uses four separate strands of cable. If UTP were used in DS/CDMA, then there would effectively be four simultaneous channels available from the very beginning before any application of a new MAC scheme. Furthermore, UTP is employed in a star configuration creating a connection scheme which does not resemble a broadcast link. In effect, it is a point-to-point link. Since comparison of PPP to broadcast links is not under consideration, the single coaxial cable in a bus configuration more closely resembles the free space medium used in wireless applications of current DS/CDMA implementations over that of UTP. These differences complicate the analysis of the effects of the MAC analyzed in this study;

- Optical fiber was also considered, however, since the goal of this research is to implement better throughput in an existing network infrastructure, the use of fiber optic cable seems to mitigate the need for a better MAC scheme and was thus dismissed as an operating factor.

10Base2 is an RG-58 copper coaxial cable. Signals propagating within it can have the same characteristics as those signals traveling in an antenna for wireless transmissions [Lab99]. Since it has the closest resemblance to free space versus UTP or fiber and due to its large installed base, it is used as the model for the physical medium. A 10 Mbps, 10Base2 coaxial LAN is chosen as the baseline architecture. It can then be easily ported to accommodate DS/CDMA wireless–to–wired implementation.

3.8 System Services and Performance Metrics

Both the CSMA/CD and DS/CDMA protocols provide access to the broadcast medium in the computer network. The stations connected to this medium access the bus to transmit variable amounts of data. The system service is simply to provide an efficient communication link for each individual station's transmission. Once a transmission has occurred there are three possible outcomes:

1. The bits arrive correctly,
2. The bits arrive with errors,
3. The bits do not arrive at all.

Since the networks are wired-based, their bit error rates are close to zero as opposed to a wireless implementations where BER defines system performance [Rap96] [PD96]. The service of bits arriving with errors is thus not a service which needs to be measured. If there are errors, it is due to collisions in the network and will be attributed to outcome three: bits do not arrive at all.

For each transmission, the data transfer rate, or user throughput and end-to-end delay of the DS/CDMA and CSMA/CD networks are collected. This is related to the load of the system defined as the number of stations transmitting in the network. This leads to the following performance metrics:

- Throughput (bps) directly measures the rate at which bits arrive correctly and corresponds to outcome one. Indirectly, with offered load, it also measures the service outcome number three of bits not arriving at all. In the CSMA/CD, the aggregate network throughput is constrained by the capacity of the network media. However, in the DS/CDMA network, the aggregate network throughput is limited by the sum total of the capacities of each individual station since all stations can transmit concurrently at the same time,
- Mean Delay (msec) is the time from when a packet is placed in the transmitter's transmission queue to when the final bit is received at the receiver, and
- Power Ratio which relates throughput and delay and measures outcomes one and three and indirectly relates to outcome two. As throughput increases, the power ratio also increase. As delay increases, power decreases. Even though throughput may increase at the expense of delay, the ratio of the two shows a performance decrease.

The throughput and mean delay are used to compare performance between both the CSMA/CD and DS/CDMA LAN implementations. They are also used to compute the power ratio of the network to give another venue to characterize the performance.

Other metrics recorded are queue size, queuing delay, and average packet size. Queueing delay is a component of the mean end-to-end (ETE) delay metric and influences outcome three. If packets cannot queue, then they are dropped which reduces throughput. Packet sizes will aid in determining the overall effect of the queueing delay, ETE delay, and throughput.

3.9 Parameters

3.9.1 System. The system parameters affecting the performance of the network and packet transmission delay are:

- Speed of the network: 10 Mbps [IEE85]
- Bandwidth of the physical medium: 400 MHz [Lab99] [Tan96]
- Length of physical medium: 180 m [IEE85]
- Type of MAC: CSMA/CD or DS/CDMA
- SNR
- Eb/No
- Number of stations connected to the network
- Transmission Buffer size: 8 KB [Kil98]

The parameters that are common to both Ethernet and DSSS LANs are:

- All stations are homogeneous
- Packets arrive at each station according to a Poisson distribution.
- Packets are serviced on a first-come-first-served (FCFS) basis

Parameters that are specific to Ethernet:

- IEEE 802.3 Standard
- 10Base2 bus configuration

- Maximum channel bandwidth is 10 Mbps
- Binary Exponential Backoff (BEB) used for retransmission delay interval when a collision occurs

Parameters specific to DSSS:

- Direct sequenced spread spectrum
- No coding or error correction
- Binary phase shift keying (BPSK) modulation
- Processing gain is defined as channel bandwidth divided by data rate

$$ProcessingGain = GP = \frac{Bandwidth}{DataRate}. \tag{3.1}$$

- Sources of interference are
  - Background thermal noise
  - Transmitting sources not in same code family
  - Multiple access interference due to transmission from other stations in the same code family
- Code family employed are Gold codes
  - Length, N=513
  - 513 preferred m-sequences accommodates more than the 2–30 accessing stations
  - Maximum cross correlation bounded at a nominal value of about 0.0176 based on the expected value of the equations defined in Table 2.2. This parameter is explained further below.

3.9.1.1 Code Selection. Gold codes are modeled in the DS/CDMA simulations. Since the operation of CDMA is not actually under review, the choice of a spreading code is only made to help define what chipping rate is used and to quantify the cross-correllation effect of the codes on multiple access interference (MAI). For simulation purposes, the actual spreading code is irrelevant since it's only impact for OPNET modeling is how it affects the noise resulting from cross-correlation MAI. Each code family (i.e., WH, Gold, or Kasami) has it's own unique characteristics and it is these families which govern CDMA performance in a simulation study. Furthermore, since Gold codes are so well behaved with its 3-valued correlations, the expected value is used based on Table 2.2. Since OPNET uses the final sum total of the multiple access interference, this assumes the cross-correlation values occurs according to their percentages.

Because the choice of Gold codes, synchronization is possible through the code itself. However, since this research is concerned with the MAC sublayer, synchronization is not an issue and therefore we assume stations to be perfectly synchronized.

3.9.2 Workload. The workload is defined in the OPNET simulation by specifying the distribution of the size of and interarrival times of the packets. The workload is a function of the number of users and the amount of data that must be transmitted which are factors of this study. It is assumed that all stations have an identical offered load to the system and that increasing the number of stations increases the offered load to the network for the same packet interarrival time. This is done to facilitate the ideal separation of the factor effects in the analysis. The workload parameters affecting performance are:

- Packet interarrival times which are exponentially distributed, and
- Packet size which is geometrically distributed with mean of 791 bytes as defined by the IEEE 802.3 [IEE85].

3.10 Factors

The factors chosen for this study are the following:

- Type of MAC scheme used: DS/CDMA and CSMA/CD,
- Number of stations transmitting on the network: The upper limit on Ethernet for a 10Base2 Network is 30 nodes per segment [IEE85]. Five different values will be used for the number of stations, n=2,4,8,16,30
- Workload: the offered load to the network will be 25%, 50%, 75%, 100%, 200%, and 400% of the network capacity. Each station on the network will contribute an equal share of the offered load.

3.11 Exponential Distribution for Workload

Since network traffic is a factor in this research, specific values for the mean have been chosen to represent a range of light to heavy loading levels. The value of the mean is set to provide the network load of 25%, 50%, 75%, 100%, 200%, and 400% channel bandwidth utilization. The latter three levels represent an overload condition.

These loading levels are dependent on the number of transmitting stations in the network. For an overall network utilization of 25% for a 10 Mbps Ethernet, stations on the network must generate an offered load to total 2.5 Mbps or 0.25 times 10 Mbps. For example, if there are 10 stations on the network, then each station must offer 0.25 Mbps of load to the network or $\frac{2.5 \text{ Mbps}}{10 \text{ stations}}$. Using the minimum and maximum values for the packet size defined in [IEE85], the average packet length is about 791 bytes. Since there are 8 bits per byte, it is a simple matter to convert the number of bits of offered load to the number of packets of offered load. The offered load per station is

$$OfferedLoad\left(\frac{packets}{sec \cdot station}\right) = \frac{\left(\frac{\%Utilization}{100}\right)\left(\frac{NetworkCapacity}{\#Stations}\right)}{(PacketLengthinBytes)\left(\frac{8bits}{Bytes}\right)}. \quad (3.2)$$

The interarrival time is simply the inverse of (3.2) and gives seconds per packet for each station. The values for the interarrival times for the various configurations of the simulation trials are given in Table 3.1.

Table 3.1. Packet Interarrival Times (msec)

	Number of Users				
Offered Load	2	4	8	16	30
25%	2.53	5.06	10.12	20.25	37.97
50%	1.27	2.53	5.06	10.12	18.98
75%	0.84	1.69	3.37	6.75	12.66
100%	0.63	1.27	2.53	5.06	9.49
200%	0.32	0.63	1.27	2.53	4.75
400%	0.16	0.32	0.63	1.27	2.37

The DSSS network does not require dividing the the total load amongst its users since there is ideally no conflict in resource allocation. It is possible that every station in the DSSS network each could transmit to the maximum capacity of the network. Even so, in order to maintain consistency between the two different networks, every station will have the interarrival times outlined in Table 3.1 regardless of MAC scheme used.

The data rate is a user defined factor in the DS/CDMA network. Depending on the amount of processing gain required to maintain a SNR capable of sustaining a BER of 1×10^{-4}, the data rate can be set to achieve the required processing gain. Since the Ethernet station could ideally handle a data rate of 10 Mbps, the DSSS stations were designed to also accommodate a 10 Mbps data rate although the DSSS could be configured for a higher data rate at the expense of the processing gain.

3.12 Evaluation Technique

This research will use simulation and analysis to compare CSMA/CD to DS/CDMA. The simulation is conducted using OPNET MODELER [Tec97], a network simulation and design tool. Models of both Ethernet and DSSS networks for the 2–30 connected stations are created using the graphical user interface in OPNET MODELER and simulated on Ultra10 SparcStations running SunOS Release 5.7 operating system.

3.13 Models Used

This research used the OPNET provided CSMA/CD models to build a 10Base2 bus-type Ethernet network. These components include OPNET's *ethcoax_station_adv* models for the stations, *eth_coax_adv* for the channel, and *eth_tap_adv* for the bus tap.

The DS/CDMA models were created using OPNET's simple radio transmitter and receiver models. These are built using three components: a source, a queue, a transmitter/receiver, and an antenna. Most studies and models for DS/CDMA networks use a wireless channel which assumes transmission in a lossy, unguided medium. This research first developed the DS/CDMA network as a wireless network following the specifications of generic direct-sequenced spread spectrum communications. This model was then validated for proper operation by comparison to analytic models. The free-space model for the transmission channel in the wireless implementation was then modified to have characteristics of the RG–58 (10Base2) copper cable. This, then, changed the wireless DS/CDMA implementation and turned it into a wired LAN for comparison to the Ethernet LAN. The rest of this chapter explains how the DSSS models were developed as well as detail the analysis process which will be covered in-depth in Chapter 4.

3.14 OPNET Implementation of DSSS

There are many characteristics that separate DSSS from normal radio transmission. This includes differences in the amount of required transmitter power, signal bandwidth, interference, and noise reduction. Channel attributes normally defined for an unguided medium such as the air interface for radio transmissions must be changed to emulate a guided medium such as coaxial cable. In order to accommodate these changes, several transceiver pipeline stage modifications are required. Specific code changes are listed in Appendix D. The addition of certain model attributes is needed to successfully simulate a DSSS network. These attributes were added to both the DSSS transmitter and receiver and are identified in Table 3.2 and further explained in the following subsections.

Table 3.2. Extended Receiver Attributes

Attribute	**Variable Type**	**Default Value**
Code_Family	Integer	0(Gold Code)
Cross_Correlation	Double	1
Spreading_Gain	Double	1

The *Code_Family* attribute was added to provide a flag for OPNET to identify that a DSSS transmission is taking place. If this attribute is not set, then OPNET assumes that the transmission is not a spread signal and follows the default radio calculations in the pipeline. The variable type is an *integer* to easily differentiate the type of code family employed. There are several types of spreading codes used in DSSS communications – each having its own unique characteristics. The default value is defined as 0 representing a Gold code family. Other values which can be assigned different integer values are WH codes, Kasami codes, and any number of others. However, for this research, only the attributes for Gold codes have been defined.

The code family chosen for spreading impacts the value of the *Cross_Correlation* attribute. The cross correlation identifies how much of a de-spread signal's power will pass on to the receiver as noise. Various code families have different values of cross correlation. Gold codes have very 'well behaved' tri-level cross correlation values and is easily inputed into the *Cross_Correlation* attribute. The variable type is a *double* since cross-correlation is essentially a percentage of passed interference power.

The *Spreading_Gain* attribute is also a *double* value. This attribute performs an adjustment to the overall processing gain of the system based on the type of interference noise encountered in a transmission. When the de-spreading operation is applied to the signal in the presence of narrowband noise, there is a different processing gain associated with it versus a wideband or pulsed interference signal. The interfering signal does not necessarily get reduced by the full processing gain of the system as do regular MAI signals [PZB95]. The *Spreading_Gain* attribute compensates for this by scaling the overall processing gain of the system. One of the operating assumptions is that the only source of interference is due to MAI which is roughly equivalent to wideband noise. Since this type of noise does not scale as do narrowband or pulsed noise, it's value is 1 which means that the amount of processing gain into the system with this type of interference is the same as the processing gain out of the system. In examining the other types of interference, the *Spreading_Gain* attribute can have a range from 0 to 1 where 1 is no reduction in gain and 0 means that the noise is resistant to this spreading technique.

3.14.1 Stage 2 – Link Closure. It is assumed for simulation purposes that link closure is established. All stations in the network are capable of sending and receiving information to any other station. As such, the *closure-all* pipeline stage is employed. Hence, there are no calculations performed for this stage other than assigning a true value to the link closure attribute.

3.14.2 Stage 3 – Channel Match. The channel match stage ensures that the transmitter and receiver have compatible characteristics for communication. Where the traditional radio characteristics leave off, spread spectrum requires a few more properties that must 'match'. Spread spectrum's most notable difference between traditional radio is the use of a 'spreading code'. These codes possess unique properties depending on the family of code membership. Auto and cross-correlation values exist between any two codes. This auto and cross-correlation is essentially the degree to which the codes are similar. The actual cross-correlation value is based on spreading code length and other properties [Skl88]. A family of codes possesses the unique qualities of minimal cross correlation values between codes in the family and reduced correlation between codes of a different family.

OPNET already has an attribute in the radio link for a spreading code. This attribute is used as a way to identify an individual station – a virtual address. Assigning a value to this attribute does not make the transmission spread spectrum communication. In fact, it does not affect the operation at all [Tec97]. Since this research is not actually modulating data streams or spreading transmissions, the need for an actual spreading code is a moot point. However, from a simulation standpoint, it can be used to designate a particular code from a predefined family for the network for use by a certain station for multiple access considerations. Therefore, in order to implement a DSSS LAN, a code family must be defined to differentiate the spreading code attribute as a member of a spread spectrum network versus a radio network.

The attribute *Code_Family* is added to facilitate this function. This attribute must match on both the transmitter and receiver end. If they match, then the receiver channel will be tagged as a valid transmission and be processed by subsequent stages. However, if they don't match, then it is assumed that the packet is noise and be processed in the Interference Noise stage.

3.14.3 Stages 4 & 6– Transmitter and Receiver Antenna Gain. This stage is modified to always assign isotropic antenna transmission characteristics to the receiver channel. This facilitates the ultimate wired implementation.

3.14.4 Stage 7 – Received Power. One of the operating assumptions of the DSSS LAN is perfect power control. In order to facilitate this characteristic, the received power calculation is manipulated such that the user defined transmitter power is the same as the resulting received power at the receiver. The relationship between frequency, distance and power is therefore eliminated.

3.14.5 Stage 9 – Multiple Access Interference Noise. This stage was renamed to better explain its function. Assuming that the only sources of noise are thermal and background noise, then stations produce interference noise. This stage was changed to accumulate noise power according to the type of interference. All stations in the network can be considered wideband jamming sources if their signals are tagged as interference since transmissions occupy the full bandwidth of the medium.

To simplify calculations of the SNR and more closely model the wired medium of RG–58 cabling, it is assumed that the only noise source is the other transmitting nodes. Thus, the noise floor will be composed of other transmissions simultaneously accessing the network. If there are no transmissions, the cable will be silent (no signals). This is not an unreasonable assumption since the stations on a wired media experience near zero noise. Current OPNET models do not account for noise in the Ethernet-Bus state processes or the bus transmission pipeline [Tec97]. Noise, in the Ethernet bus, will result in a collision on the network. This is also the operating assumption for the DS/CDMA network when there are multiple simultaneous transmissions, rather than a collision occurring, the noise level is increased.

In order to understand how the interference noise attribute is accumulated, consider the diagram in Figure 3.1. If the receiver is looking for $s(t)$ and there are no other transmissions occurring, the signal-to-noise (SNR) for the signal of interest before transmission is

$$SNR = \frac{S}{N}, \tag{3.3}$$

where N is the average noise power in the system and S is the signal power of the signal of interest. For a DSSS binary phase shift keyed (BPSK) modulation in a additive white gaussian noise channel, the SNR at the input to the demodulator is

$$SNR_{input} = \frac{S'}{N'} = GP(SNR) = GP\frac{S}{N}, \tag{3.4}$$

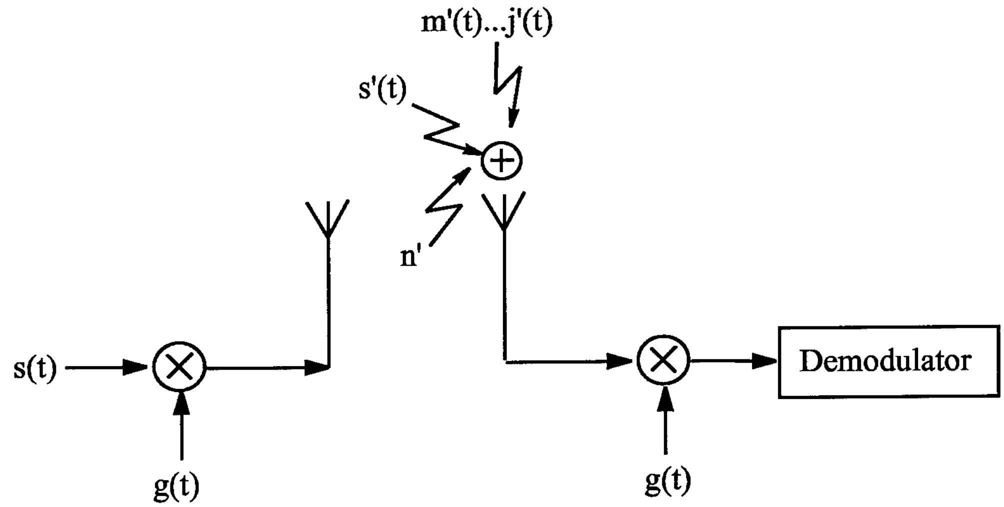

Figure 3.1. Simplified Diagram of Multiple Access Interference

where GP is the processing gain of the system and is defined in (3.1) and the primes indicate the received power of the respective signals (either signal of interest, S, or noise, N).

Consider a signal, $m(t)$, that is in the same code family as the intended signal, $s(t)$. Since $m(t)$ is in the same family, there will be some cross-correlation in the de-spreading process and it can be considered as a multiple access interferer (MAI). This cross-correlation coefficient serves to reduce the processing gain by some amount dictated by the coefficient's value. Following the same reasoning for as (3.4), the signal-to-multiple-access-interferer can be described by

$$SNR_M = \frac{S'}{M'} = (\alpha)(GP)\frac{S}{M}, \tag{3.5}$$

where α is the cross-correlation coefficient between the codes used to spread and despread respectively and M is the received power of an MAI station.

Now consider another signal, $j(t)$, arrives at the receiver. This time, it is some arbitrary signal that may or may not be spread but is not in the same code family as $s(t)$. This signal can be considered a generic jamming source. There are no cross-correlation values to deal with in this case. However, depending on the type of jamming (i.e., wideband, narrowband, pulsed), the processing gain associated with this signal is different. Following (3.4) and (3.5), a signal-to-jamming ratio can be defined as

$$SNR_J = \frac{S'}{J'} = (GP_j)\frac{S}{J}. \tag{3.6}$$

The signal of interest plus MAI stations and jamming sources are potential transmissions that could be encountered in a DSSS network. To relate all three signals together, consider the signal–to–interference ratio (SIR) at the input to the demodulator. The numerator is the power of the signal of interest, S. The denominator is the sum of the noise signal powers, M, J, and noise N

$$SIR = \frac{S'}{M' + J' + N'}. \tag{3.7}$$

Simplifying (3.7) using (3.4), (3.5), and (3.6) yields

$$SIR = \frac{(GP)S}{N + \left(\frac{GP}{GP_j}\right) J + (\alpha(GP))M}, \tag{3.8}$$

Generalizing (3.8) to account for multiple jammers and MAIs yields

$$SIR = \frac{(GP)S_i}{N + \sum_{p=1}^{r} \left(\frac{GP}{GP_j}\right) J_p + \sum_{q=1}^{t} (\alpha(GP))M_q}, \tag{3.9}$$

where k is the total number of transmitting stations, S_i is the power of the signal of interest, r is the number of jammers, and t is the number of MAIs.

A *Spreading_Gain* attribute is added to the receivers' list of attributes to accommodate the varying types of interference and is defined as

$$SpreadingGain = \frac{GP}{GP_j}, \tag{3.10}$$

which is simply a ratio of processing gains where GP is the standard processing gain defined by (3.1) and GP_j is the processing gain experienced by a jammer source. Wideband interference results in a spreading gain approximately equal to one [PZB95].

This pipeline stage is responsible for calculating the denominator of (3.9). Arriving packets are checked to determine if the packets overlap in time or have collided. If they do and they are members of the matched receiver's channel, then they are considered noise. Based on the flags set in the channel match stage, the noise is applied as either MAI noise or jammer noise. MAI noise is enhanced by the system's GP but also decreased by the cross-correlation coefficient. Jammer noise is decreased by the newly-defined spreading gain ratio. This accumulated sum is the resulting *accum-noise* attribute of the packet.

3.14.6 Stage 10 – Signal to Noise Ratio. OPNET's default radio transceiver pipeline stores the SNR before applying GP. When plotting the SNR statistic, this value would result in less than expected numbers for the DSSS network. Therefore, a modification was made to this stage to add in the processing gain.

3.15 Experimental Design

A full factorial experimental design with replications is chosen for the experimental design. Factors and their associated levels are identified in Table 3.3.

Table 3.3. Factors and Levels

Factors	MAC	Number of Stations	Network Offered Load
Levels	DS/CDMA CSMA/CD	2 4 8 16 30	25% 50% 75% 100% 200% 400%

There are a total of three factors, each of which has two, five, and six respective levels. In order to characterize the effects of these factors, a minimum of $2 \times 5 \times 6 = 60$ experiments are needed for every combination of these factors at the prescribed levels. In order to isolate experimental error, three replications have been chosen for a total of $60 \times 3 = 180$ experiments.

3.15.1 Regression Model. The model for this design consists of three replications of each of the 60 experiments corresponding to the a=5 levels (2, 4, 8, 16, 30) of the *# Stations* factor, b=6 levels (25%, 50%, 75%, 100%, 200%, 400%) of the *Network Offered Load* factor, and c=2 levels (DS/CDMA,CSMA/CD) of the MAC protocol used. There are a total of seven. effects. This includes;

- Three main effects:
  1. MAC Scheme
  2. # Stations
  3. Network Offered Load
- Three Two-Factor Interactive Effects:
  1. MAC Scheme and # Stations
  2. MAC Scheme and Network Offered Load
  3. # Stations and Network Offered Load
- One Three-Factor Interactive effect of all three factors

To analyze the throughput and delay of this research, the model is

$$y_{abcr} = \mu + \alpha_a + \beta_b + \xi_c + \gamma_{ab} + \gamma ac + \gamma_{bc} + \gamma_{abc} + e_{abcr}, \tag{3.11}$$

where

- y_{abcr} = the response of the r^{th} replication of the experiment with the three factors at the a, b, and c levels respectively
- μ = mean response,
- α_a = effect of # Stations at level a,
- β_b = effect of Offered Load at level b,
- ξ_c = effect of MAC scheme at level c,
- γ = interactive effects of factors at given levels,
- e_{abcr} = error of the response of the r^th replication at the factors a, b, and c levels.

The effects of the factors in this research are analyzed using an ANalysis Of VAriance (ANOVA) table. This provides for the calculation of effects of not only the factors, but the effects due to the interaction of these factors as well as characterizing experimental error. Further, the percentage of variation explained by the # Stations, Offered Load, MAC scheme, their interactions, and experimental error are identified.

3.15.2 Verification and Validation of Models. Much research has been conducted in Ethernet LANs. The study by Gonsalves [Gon87] was used to validate the *ethcoax_station_adv* model as well as the underlying network media. These models are provided by OPNET and have already undergone extensive testing as a sound simulation model of a coaxial-based 10 Mbps Ethernet station. Comparison of the results obtained from verification and validation tests to Gonsalves's study is needed to ensure proper setup of the Ethernet networks used in this research. The results of this analysis is in Chapter IV.

Due to the complexity of DS/CDMA wireless channels, most previous research uses statistical approximations rather than exact analysis to model these networks. In order to verify that the OPNET models created for this research behave as they should, model verification is accomplished by comparison to these approximations.

For a CDMA system, the BER defines the performance of the system. Approximations of the BER are typically used in verification and/or validation of DS/CDMA research.

The BER is related to the SNR as seen in (2.13) This is also the relationship that OPNET uses to do a table lookup for the BER value in Stage 11 of its transceiver pipeline [Tec97].

The improved Gaussian approximation along with the standard Gaussian approximation are both used to verify and validate that the OPNET DS/CDMA models created for this research is operating correctly. This was done by comparing the results of the analytic models to that of the models in this research and statistically determine if they are different. If they are not statistically different, then the DS/CDMA models can be assumed to operate validly based on the assumptions of this research.

3.16 Summary

This chapter outlined the methodology used to analyze the data resulting from this research. The experimental design was presented along with definition of this research's goals, boundaries, factors, and input parameters.

OPNET's radio transceiver pipeline is modified to perform spread spectrum communication functionality. The Ethernet models are taken from OPNET's default library and compared for proper performance.

A multilinear regression is constructed to characterize the throughput and delay responses of the systems under study. The comparison of both systems is facilitated through the use of an ANOVA table.

IV. Results and Analysis

This chapter presents the results obtained from this research. Verification and validation of the DS/CDMA and CSMA/CD LAN models were conducted and summarized. Results from experimental runs are also presented. An analysis of variation was conducted to quantify the effects of the *MAC scheme*, *# of Stations*, *Network Offered Load*, and the interactions between these factors to the throughput and delay metrics.

4.1 Verification and Validation

The simulation of both the DS/CDMA and CSMA/CD models must be designed such that the results are representative of the real systems they model. This is accomplished using a two-stage approach. The first stage is verification, the second is validation.

Since there are a number of assumptions made to implement the respective LANs, the first step is to determine if the model behaves correctly. This is referred to as 'verification' which ensures that "the model does what it is intended to do" [Jai91]. There are three tests which can be performed to aid in verification [Jai91].

1. Consistency: When offered similar workloads, the results should be similar as well.
2. Seed Independence: Changing the random seed between replications of a particular configuration should produce similar results. This ensures that the random-number generation does not effect the final result.
3. Degeneracy: The model should work for extreme cases and produce results. This aids in the debugging process to ensure proper model behavior.

The second step is called 'validation'. Validation is concerned with the "representativeness of the assumptions" [Jai91]. In other words, validation ensures that the model produces results consistent with those observed in a real system. There are two comparisons which aid in the validation process [Jai91].

1. Real-System Measurements: Comparison to real systems is "the most reliable and preferred way to validate" [Jai91]. However, this may not be possible for all situations.
2. Theoretical Results: The system which is being modeled may be analytically modeled using simplifying assumptions. The results obtained from an analytical model may be compared to validate the simulation model, but there is a danger that the analytic model is not valid. If so, the simulation model will not be valid either.

It is important to note that although a model may be verified and validated, it is very difficult to have a 'fully validated' model. The validation tests are only valid for the configurations in those tests. Since the verification and validation runs cover the extreme cases this research will encounter, the simulation models are assumed to operate as a real implementation of these systems under the same assumptions.

4.1.1 CSMA/CD. CSMA/CD verification tests were conducted using OPNET's *ethcoax_station_adv* models for the stations, *eth_coax_adv* for the channel, and *eth_tap_adv* for the bus tap. These models are used to build and simulate a 10Base2 coaxial Ethernet LAN. Verification is accomplished in four stages:

1. Network containing two stations transmitting at sub-capacity.
2. Network containing two stations transmitting over capacity.
3. Network containing four stations transmitting at sub-capacity.
4. Network containing four stations transmitting over capacity.

The first and second stages verify that a single communication link is established. Since the medium is a coaxial wire, the error rate can be considered zero.

Therefore, the number of sent packets should equal the number of received packets provided the arrival rate of the packets do not saturate the network. If the network has more packets arriving than can be transmitted, as is the case in Stage 2, then there will be packet drops at the transmitting station. Since the transmitters have a finite memory size (8 KB) [Kil98] and the network is saturated, the packets will have to enqueue in the transmitter's buffer. Once the buffer fills, the packet must be discarded resulting in a buffer overflow. However, since the buffer is continually full, the transmitter will always have a packet to send and thus there will be nonstop transmission. The throughput will be equal to the capacity of the channel which is 10 Mbps. Furthermore, since packets are queuing in Stage 2, the end-to-end delay will also increase in this configuration.

The third and fourth stages verify that two communication links are established. However, since there are two simultaneous transmissions occurring at the same time, there is an increased probability for collisions to result. Since bandwidth is wasted whenever there is a collision, the throughput will be decreased compared to Stages 1 and 2. In the overloaded case for Stage 4, packets will be lost not only due to collisions but also due to transmitter buffer overflows. Delay will also increase since packets are queueing.

Stages 1 and 2 are setup according to Figure 4.1. Stages 3 and 4 are setup according to Figure 4.2. Stations designated with a 'TX' are setup to only transmit whereas 'RX' designate stations configured to only receive. The transmitters are assigned interarrival times to generate loads below the capacity of the network. For verification purposes, three trials were conducted with interarrival times constant and set at 2.5 ms for Stage 1 and 5 ms for Stage 3 . This generates an average total offered load of about 2.5 Mbps which is below the 10 Mbps capacity of the network. The packets are constant sized at 791 bytes. For the overload cases in Stages 2 and 4, the interarrival times are constant and set at 0.3 ms and 0.6 ms respectively to

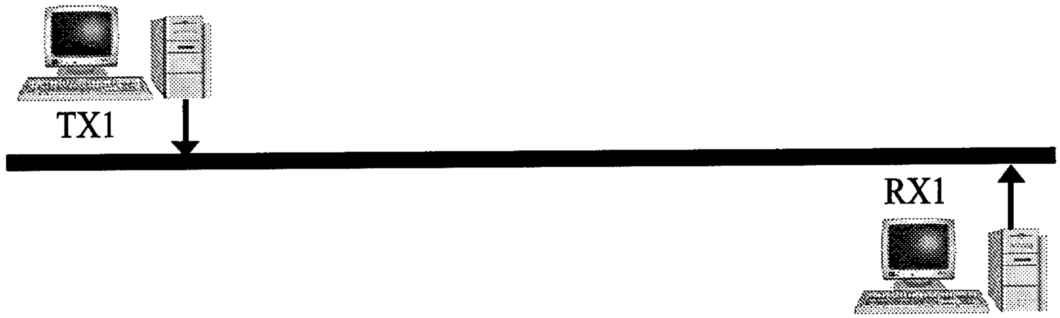

Figure 4.1. Ethernet Station Verification Setup: 2 Stations

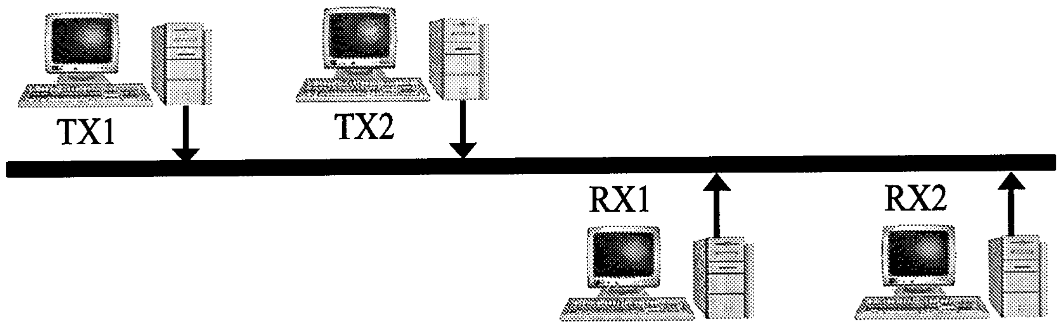

Figure 4.2. Ethernet Station Verification Setup: 4 Stations

generate a total offered load of 20 Mbps – twice the capacity of the network. The configurations for the verification runs are summarized in Table 4.1

Verification of the CSMA/CD simulation model is accomplished using the consistency, seed independence, and degeneracy tests. Consistency is established for the workload in that one station offering the network a total of 2.5 Mbps has approximately the same result as two stations offering a total of 2.5 Mbps even though each station is only offering $\frac{2.5Mbps}{2} = 1.25$ Mbps each to the network. Changing the random seed between the 3 replications produced similar results. Overloading the network shows that for the extreme loading case, the model still produces consistent results. Extreme underloading was not considered since the 2.5 Mbps offered load is an underloading case.

It is clear that the CSMA/CD models operate according to the behavior of Ethernet operation. Since there was a finite simulation time (10 minutes), the de-

Table 4.1. Ethernet Verification Configuration Parameters

Parameter	**Value**
Packet Size	Constant 791 Bytes
Interarrival Rate	
Stage 1	2.5 msec
Stage 2	5 msec
Stage 3	0.3 msec
Stage 4	0.6 msec
Buffer Size	Infinite
Simulation Time	5 minutes

lay measurement had a maximum value. However, since there was also an infinite buffer size, the delay values would have continued to increase without bound as the simulation is increased. Collisions were indirectly measured based on the reduction in the throughput measurements. Since the Ethernet channel has a near zero BER, losses are attributed solely to collisions. Since there was a loss in throughput, there must be an increase in the number of collisions. All values recorded were averaged over the 10 minute simulation time for these verification trials. Based on the results in Table 4.2, therefore, the model is verified.

Validation of the CSMA/CD network is accomplished by comparing the results to data obtained from a real system. In a study by Gonsalves [Gon87], the throughput and delay metrics were measured in a 10 Mbps Ethernet LAN configured using values listed in Table 4.3.

This research duplicated this configuration and recorded throughput and delay. The data was then compared using paired observations where the result of each configuration in the simulation was paired with the data point corresponding to the similar configuration in Gonsalves' study. The performance differences in these paired observations produces a new set of data which a confidence interval for the mean can be calculated. This analysis is presented in Appendix B. The results obtained from validation simulation runs are seen in Figure 4.3.

Table 4.2. Ethernet Verification Results

Metric	Replication 1	Replication 2	Replication 3
Throughput			
Stage 1	2.47 Mbps	2.49 Mbps	2.48 Mbps
Stage 2	2.48 Mbps	2.49 Mbps	2.47 Mbps
Stage 3	9.50 Mbps	9.50 Mbps	9.50 Mbps
Stage 4	9.26 Mbps	9.25 Mbps	9.24 Mbps
Delay			
Stage 1	0.77 msec	0.77 msec	0.77 msec
Stage 2	0.78 msec	0.78 msec	0.77 msec
Stage 3	6.54 msec	6.54 msec	6.54 msec
Stage 4	11.18 msec	11.19 msec	11.11 msec
Queue Size			
Stage 1	0.05 packets	0.05 packets	0.05 packets
Stage 2	0.01 packets	0.01 packets	0.01 packets
Stage 3	9.81 packets	9.80 packets	9.81 packets
Stage 4	7.87 packets	7.62 packets	7.73 packets

Table 4.3. Gonsalves Ethernet Configuration

Parameter	Value
Bandwidth	10 Mbps
Packet Sizes	Constant 512 and 1500 Bytes
Normalized Offered Load	10-1000%
Number of Stations	30-38
Buffer Size for Each Station	1 packet

As can be seen, the simulated results tracked the measured responses in Gonsalves' study fairly closely. A confidence interval was constructed on the mean difference between these two systems. Since the confidence interval included zero, the systems are not statistically different. Therefore, with 95% confidence, the simulation behaves the same as a real 10 Mbps Ethernet system.

4.1.2 DS/CDMA. The DS/CDMA models are adapted from the simple radio transmitter and receiver models in OPNET. These are built using three components: a source, a queue, a transmitter/receiver, and an antenna. These stations are built as shown in Figures 4.4 and 4.5.

CSMA/CD Validation Plots

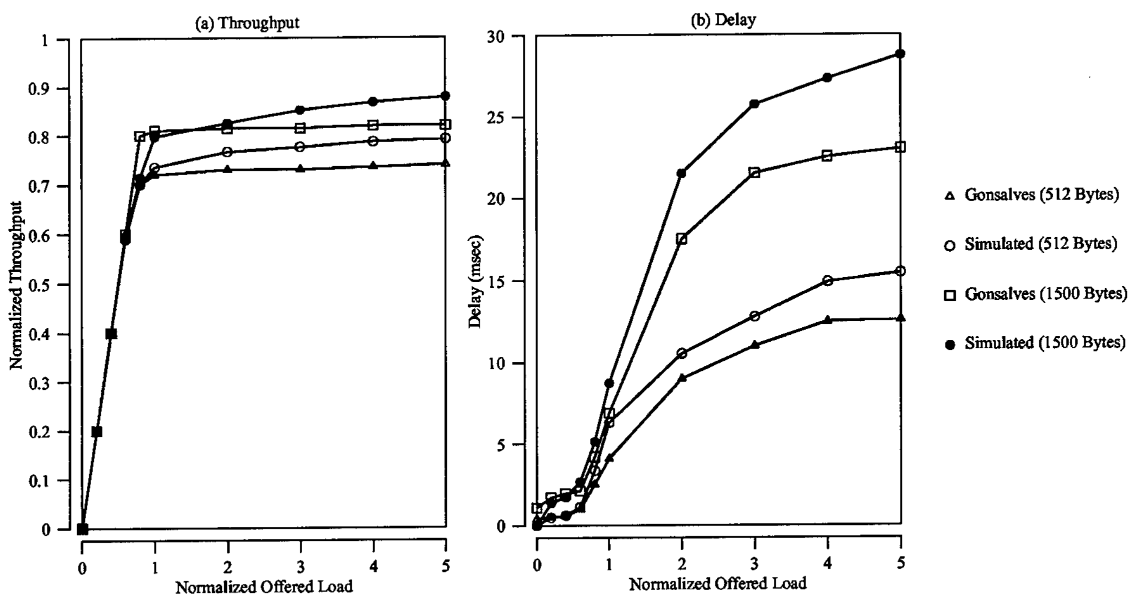

Figure 4.3. Ethernet Validation Responses

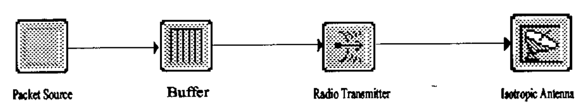

Figure 4.4. Node Model of DSSS Transmitter

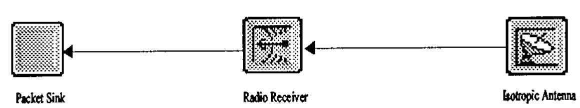

Figure 4.5. Node Model of DSSS Receiver

The difference between these models and the default radio models lies in the implementations present in the radio transceiver pipeline stages. The modified code is listed in Appendix D. In order to implement a finite buffer representing memory limitations in the stations, a simple first-in-first-out (FIFO) queue is added between the source and the transmitter in the DSSS TX model.

The verification of the DS/CDMA models was not as straightforward as the CSMA/CD models. This is principally due to the fact that a wired LAN implementation of DS/CDMA was not available for comparison. Additionally, the upper layers of the OSI model created additional levels of complexity. Since this research is at the MAC and subsequent layers, the higher layers were abstracted out. Error checking and correction coding was also not considered in this research. If the BER rose above 1×10^{-4}, the packet was discarded.

To begin the verification and validation process, the DS/CDMA system was first designed to operate in a wireless environment. Since the literature did not reveal any studies using spread spectrum communication in a wired environment, validation tests would not be based on analytic or real system data. However, there have been numerous studies using spread spectrum in the wireless environment. Therefore, wireless DS/CDMA models were constructed to facilitate tests based on analytic models for the bit error rate and the wireless radio transmission models. If the unguided free space characteristics of the wireless model were made to emulate the guided medium characteristics of a coaxial cable, then the model is assumed to be valid for this configuration.

It was assumed that connections have already been established between stations. Therefore, time for initial acquisition of the data signal is omitted from the ETE delay calculation. A 1×10^{-4} BER threshold was used. Therefore, if the BER rose above this threshold, then the SNR level is assumed too low and the packet is lost. Throughput drops to zero at this point. Perfect power control is also assumed to mitigate the 'near-far' effects of transmissions in the network.

The principal use of spread spectrum communications, with respect to this research, is to provide a multiple access capability using an existing communication channel. A station is not only able to send packets to a receiver when it is the only transmitter, but if there are multiple transmitters, all the 'conversations' can be successfully received at the same time. In effect, collisions are removed from the network when a DS/CDMA MAC scheme is present. Therefore, it is expected that the throughput of one connection (one transmitter and one receiver) will not decrease in the presence of another transmitter. This is in contrast to the CSMA/CD MAC scheme where if there was a collision, the bandwidth is wasted and therefore, the throughput would decrease.

Another aspect of the DS/CDMA system is the relative immunity to noise. Noise can come from a variety of sources, but it is assumed in this research that noise is only due to the other transmitting stations in the network. To simulate this effect, a single transmitter and a single receiver are setup as in Figure 4.6 using OPNET's default radio models.

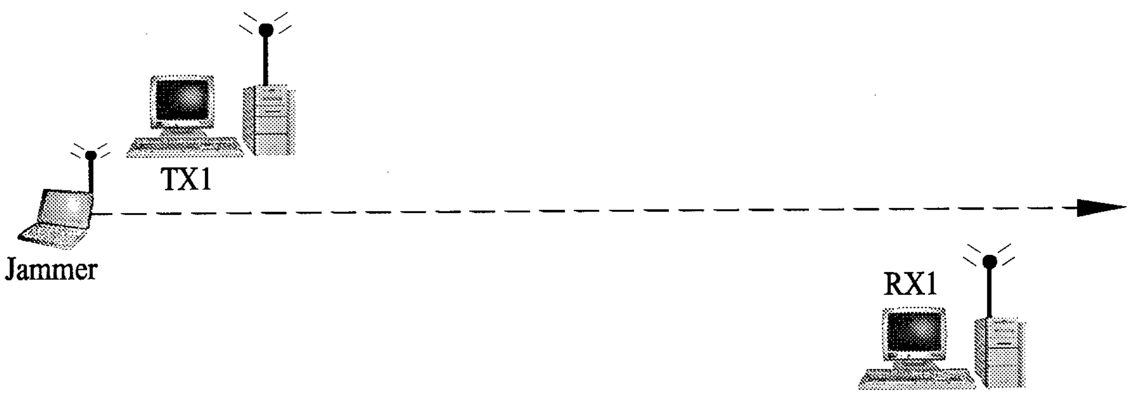

Figure 4.6. DSSS Jamming Verification

The jammer source is broadcasting white noise in the full bandwidth of the network. The characteristics of this setup are listed in Table 4.4.

Table 4.4. DSSS Verification Configuration

Bandwidth	100 kHz
Packet Sizes	Constant 1024 bits
Packet Interarrival Time	1 packet per second
Transmit Power	1 Watt
Jammer Power	1 Watt
Data Rate	1024 bps
Buffer Size	Infinite

As the jammer comes closer to the receiver, the noise power received becomes greater than the received signal power, thus reducing the SNR. This reduction results in an increase in BER. This increase translates to reduced throughput since the signal can no longer be effectively received at the receiver. Since the BER threshold is set to zero, all the bits in a packet must be received correctly in its entirety in order to be counted. Therefore, when the BER is above zero, the throughput is zero. This can be seen in Figure 4.7.

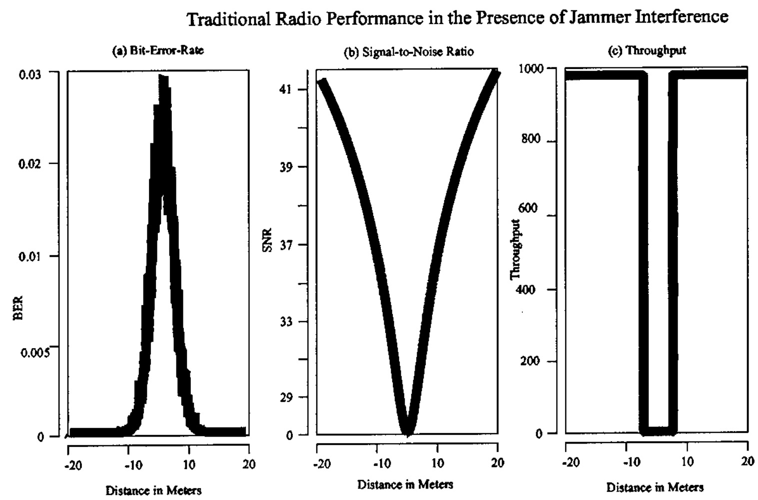

Figure 4.7. Traditional Radio BER, SNR, and Throughput Performance in the Presence of Jammer Interference

If, however, the DS/CDMA model is used in this same configuration, the effects of a jammer are mitigated by the processing gain of the system. Throughput should not suffer since there is an increase in the effective SNR level at the receiver. This translates into an insignificant effect to the BER of the communication link and thus, throughput remains constant. This can be seen in Figure 4.8.

Rerunning these two configurations for an offered load greater than the capacity of the stations should show an increase in ETE delay. This is due to the queueing of packets at the transmitter's buffer. The throughputs will be constant with a value equal to the capacity of the station since the stations always have a packet to transmit at any given time. This was verified with consistent throughput responses of about 10 Mbps and linearly increasing ETE delay measurements.

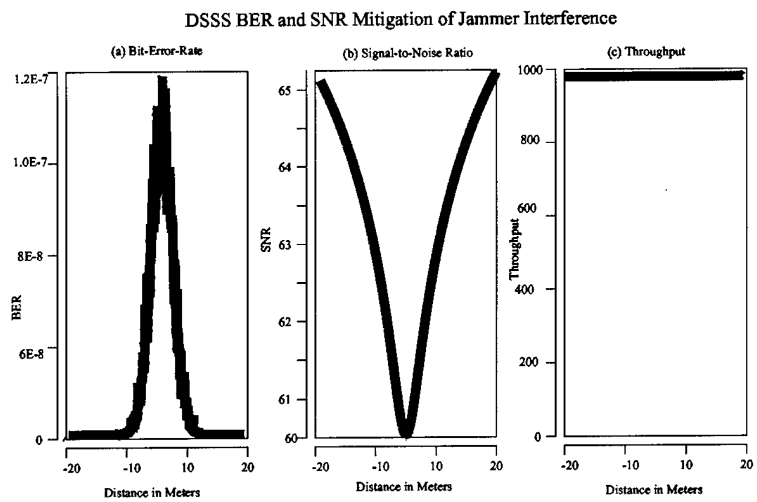

Figure 4.8. DSSS BER and SNR Mitigation of Jammer Interference

Note that the capacity for the DS/CDMA system is defined with respect to the station rather than the network medium. This is due to the difference in the MAC schemes. CSMA/CD is a TDMA-type system where each station must take turns to access the medium. However, the DS/CDMA stations can transmit at whatever data rate is defined for the system regardless of the number of stations. Also, it can transmit whenever it chooses versus having to wait for a time slot. When the offered load is greater than the defined data rate of the station, packets will queue. This is similar to CSMA/CD where the offered load is greater than the capacity of the entire network and queueing results.

The next scenario in this verification process involves multiple simultaneous transmitters. When two or more stations broadcast at the same time in the network, whether in a wireless or wired environment, packets overlap in the medium and creates collisions. In the case of a radio, if the transmissions are in disjoint frequency bands then the power of the signals of interest do not affect one another and thus, multiple conversations can be supported. This is the same situation as being able to tune to various radio stations on a home stereo. However, if there is a single channel or frequency band, then the power of the signals will 'mix' and the information will be unintelligible to the receivers.

This can be divided into two situations for the DS/CDMA system. The first situation is when the multiple transmissions are within the same code family as the signal of interest. When the signals are in the same code family, there is 'bleed-over' of power from the interfering station into the noise summation of the receiver. This is representative of the cross-correlation value introduced in Section 2.6.3.4 and implemented in (3.5). The power of the MAI will reduce the effective processing gain present in the receiver. The second situation involves communications when the transmissions are in different code families. In this case, the power of the interferer is considered wideband noise and reduced by the processing gain without the effect of cross-correlation.

Consider the configuration in Figure 4.9 using the default non DS/CDMA radio models. The simultaneous transmissions will interfere with each other similar to the

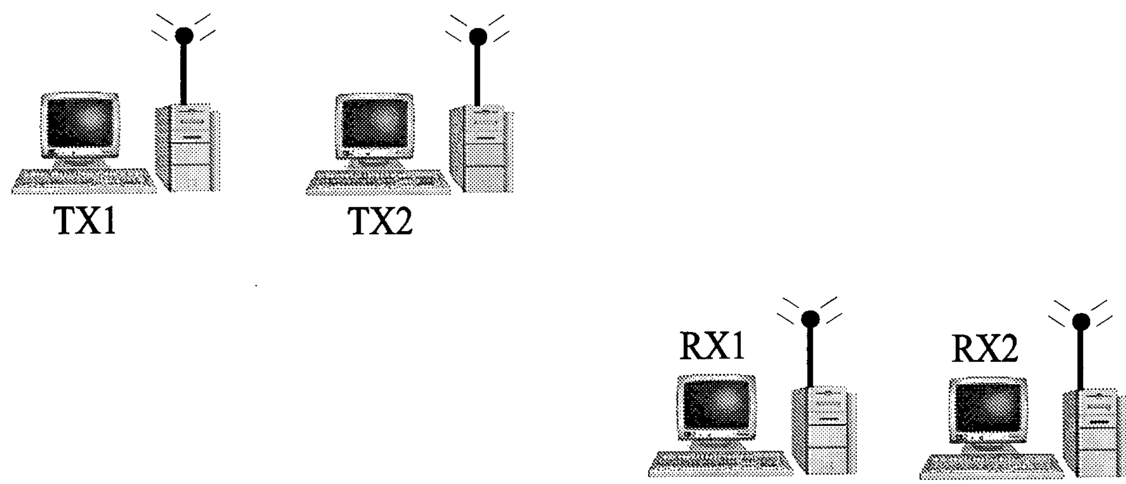

Figure 4.9. Two Simultaneous Wireless Radio Communication Links

jammer in Figure 4.6. However, the stations are stationary, so TX1's transmission will interfere with TX2's transmission throughout the entire simulation run and throughput will be zero. The results of these runs are summarized in Table 4.5.

Comparing the SNR in Figures 4.7 and 4.8 and Table 4.5, note how the SNR in DS/CDMA is higher than the traditional radio. This is a result of the processing gain introduced by spreading the signal. The SNR of DS/CDMA when the interfering

Table 4.5. MAI Verification Responses

Metric	Value
Throughput	1,024 bps
Delay	1 second
SNR	65.5 dB
BER	0.0

signal is in the same code family is lower than the SNR of DS/CDMA when the interfering signal is not in the family like the jammer. The SNR due to MAI is approximately 65 dB neglecting distance between interferes. The SNR of a station not in the same code family characterized by the wideband jammer is approximately 70 dB, again neglecting distance between receiver and interferer. The difference of 5 dB is due to the cross-correlation involved with the MAI signal will increase the effective noise passed to the demodulator in the receiver. This reduces the SNR.

The ETE delay will be some constant value representing the propagation and transmission components of ETE delay. Queuing delay will not be significant since the loading is low compared to the capacity of the station. The multiple simultaneous transmission configurations are also rerun to demonstrate increased ETE delay due to increased offered load. Queueing delay is introduced in this configuration resulting in increased ETE delay. ETE was measured to be a constant one second. This is attributed to the low 1 pps data rate of the transmissions. At the higher loading levels, the delay was on a continuous increase thus verifying the model assumptions.

Since the DS/CDMA models are able to mitigate the noise effects of both additive white noise (from the jammer configurations) and multiple access interference (the multiple simultaneous configurations), the model behavior is verified. Furthermore, similar to the experimental runs conducted in the verification of the CSMA/CD, consistency, seed independence, and degeneracy were also tested and verified.

Validation consisted of comparing the results to an analytic model of BER performance for a DS/CDMA system using the Gaussian and Improved Gaussian approximations. The parameters for these validation runs are listed in Table 4.6.

Table 4.6. DS/CDMA Validation Configuration

N: Code Length	31
E_b/N_o	12 dB
k: Number of Users	2-20

The Gaussian approximation generally underestimates the BER for a small number of users resulting in a higher analytic value than can be expected from a real system. However, for larger numbers of users, the approximation tends to overestimate the BER. The improved Gaussian approximation has an almost opposite result. It overestimates the BER for small numbers of users and underestimates the BER for larger numbers of users resulting in lower analytic value than can be expected from a real system with a large number of users. Both of these analytic models have been used in previous studies for model validation and will also be used for validation for this research.

Using a paired-sample comparison analysis, the differences between the analytic models and the responses of this research were investigated. Using a zero-mean confidence interval test, the three traces in Figure 4.10 were found not to be different at a 95% confidence level.

A detailed presentation of this validation is in Appendix B. The conclusion is that the wireless DS/CDMA behaves as it should and produces results consistent with a spread spectrum model.

Definitive validation is not possible for the wired DS/CDMA implementation due to insufficient research in this area. However, coaxial cable as employed by cable television operators has properties similar to radio antennas. Cable television is used to transport television signals originally designed for broadcast communication.

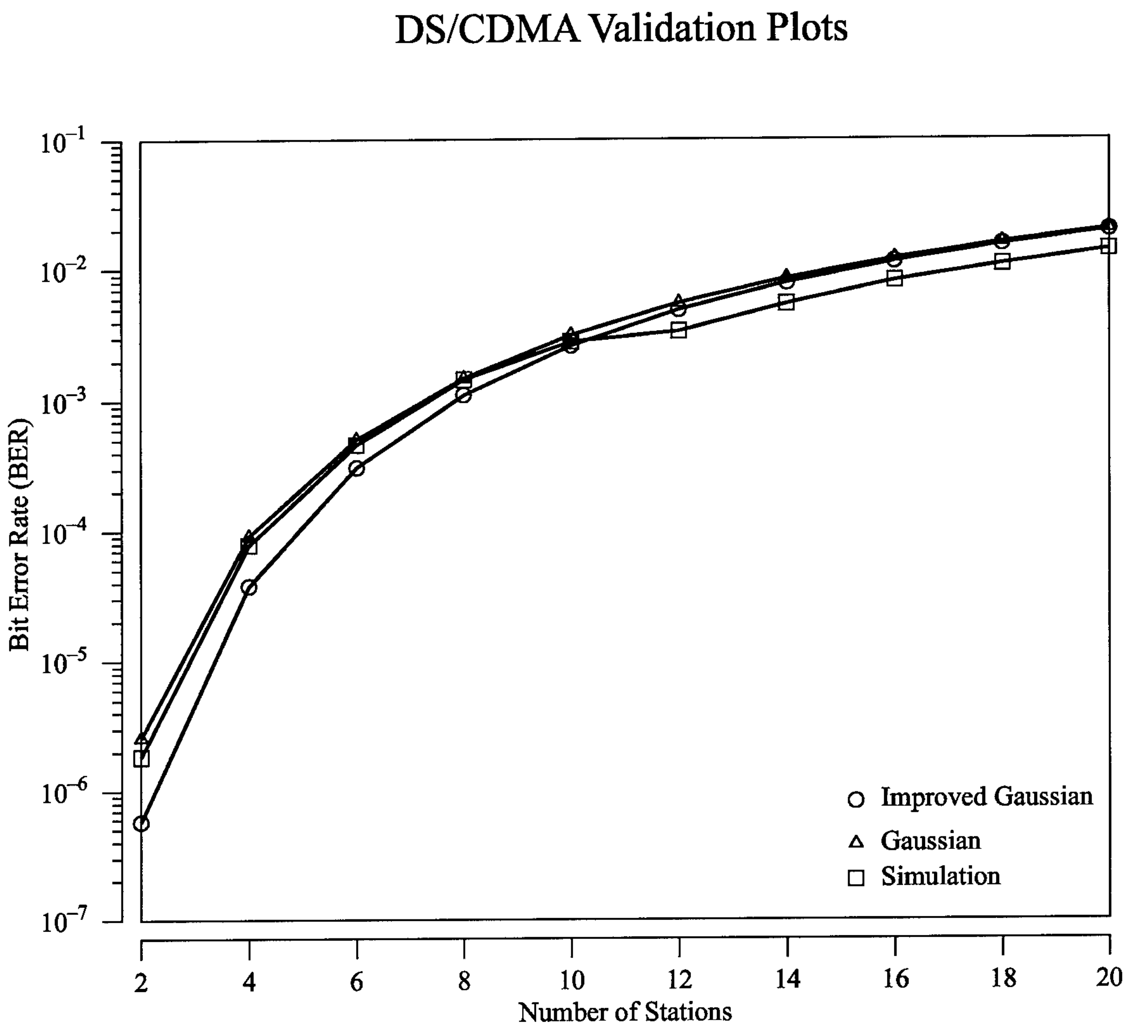

Figure 4.10. DSSS Validation Comparison to Analytic Models

Because there is virtually no loss in signal strength and quality, cable television has been the preferred means of receiving television programming [Lab99].

Radio communication requires the use of an antenna to transmit the signal to a receiver also equipped with an antenna. By extending this communication link, it is possible to accomplish the same conveyance using one long antenna with the transmitting and receiving stations on either end. This long antenna can now be replaced with a coaxial cable. Additionally, since the frequency spectrum is not regulated in a wire as opposed to the frequency management for transmission in free space, more bandwidth is available in a coaxial line. This results in more channels available for cable television at a higher fidelity than the antiquated 'antenna rabbit ears'.

Using this reasoning, the differences between a radio transmission and a wired transmission are not fundamental. The principal difference is that radio does not have physical links whereas wired does. However, the communication is virtually the same. Therefore, since the wireless DS/CDMA system is verified and validated, substituting characteristics of a coaxial cable for the free space characteristics will result in a system that behaves accordingly. Thus, it is assumed, that the wired DS/CDMA system is a valid simulation model.

4.2 CSMA/CD Analysis

4.2.1 Throughput. The throughput for the Ethernet simulations followed the expected trends. There were slight deviations from published results when higher loading levels were introduced. In general, throughput decreases as more stations access the network. This is due to the increased number of collisions as more stations access the channel. But, for two or more transmitting stations, this did not hold. In fact, 15 transmitting stations at a loading of 200% (20 Mbps) had a higher throughput than four or eight transmitting stations at similar offered loads as seen in Figure 4.11. However, with 95% confidence, these values were not statistically significant.

This result differs from theoretical results from analytic models [TK85] [TH80]. In these studies, stations had infinite buffers. As such, at offered load exceeding network capacity, the end-to-end delay will increase without bounds and never reach a steady state. As packets arrive to the transmitting station, they queue and wait for transmission. Since the service rate of the stations cannot exceed the speed of the network and the offered load exceeds this capacity, more packets will continue to enqueue. This results in an infinite delay time for this network.

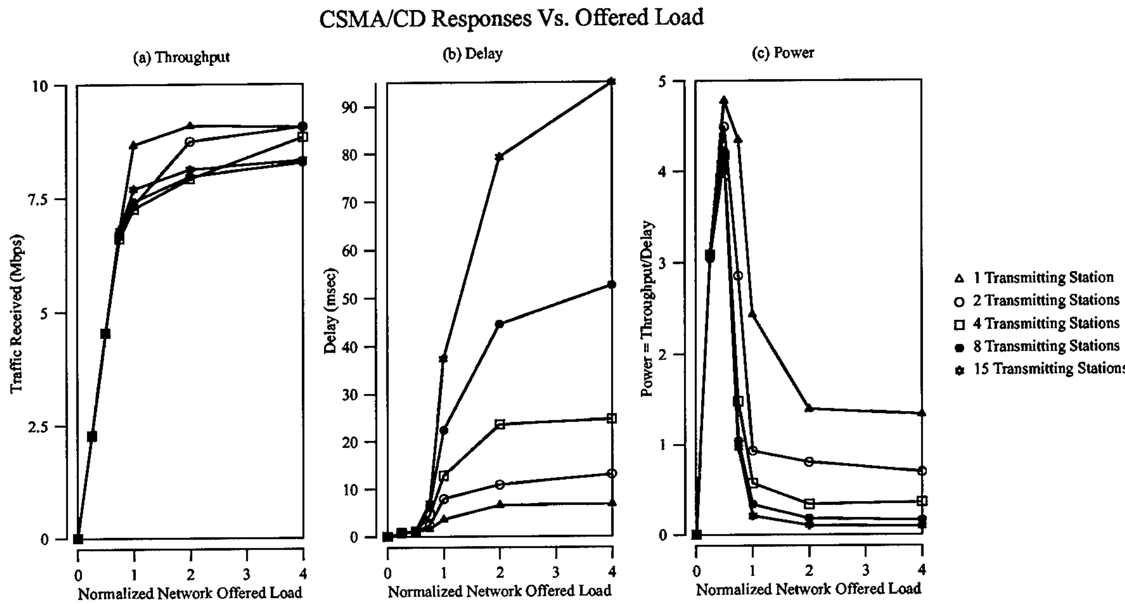

Figure 4.11. CSMA/CD Responses With Respect to Load

Limiting the buffer size is one way to bound delay. In measured studies of Ethernet performance by Boggs et al. and Gonsalves et al. [BM88] [Gon87], the network interface cards on the stations have some defined amount of memory (although these values were not specified in Boggs' study). This amount of memory limits the buffer size and therefore limits the number of packets that may wait for transmission. Therefore, regardless of offered load, there is a finite value for the maximum end-to-end delay.

The results of this research had an 8 KB limit for the buffer size. Therefore, when the buffer was full, arriving packets were dropped. This decreased the overall throughput and followed the predictions of the empirical models. However, when two or more simultaneous stations transmitted packets, it is seen in Fig. 4.11 that the throughput is actually more for 15 stations than for four or eight transmitting stations despite an increase in number of collisions. This was found to be statistically insignificant with a confidence level of 95%.

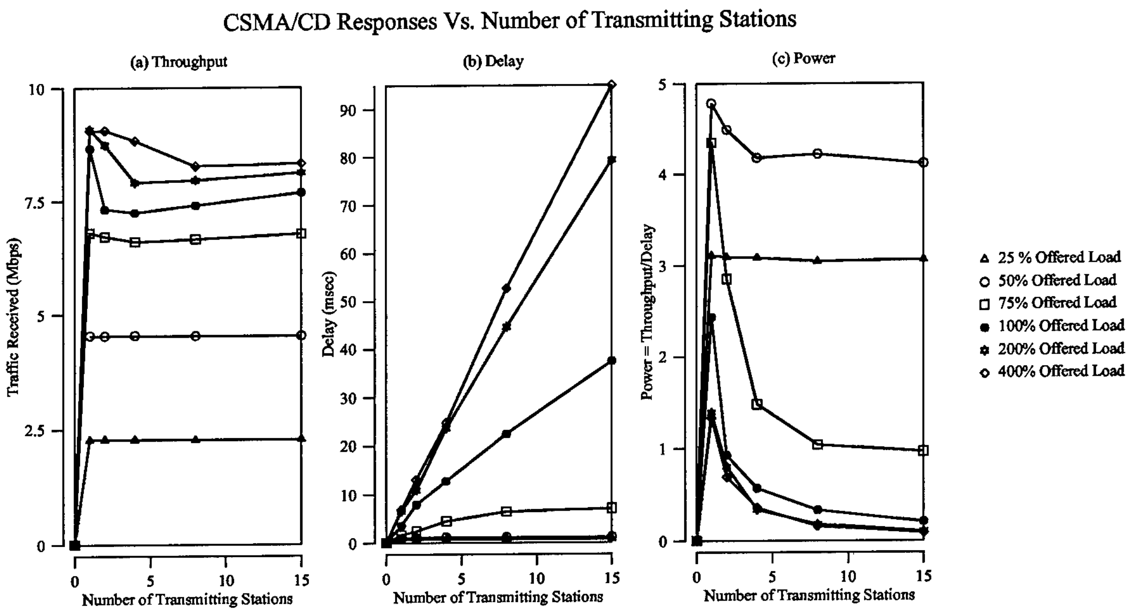

Figure 4.12. CSMA/CD Responses With Respect to Number of Stations

However, this could possibly be a result of the random distribution of packet sizes and the arrival of those packets. As more stations are added to the network and the offered load for the system is held constant, then each individual station's packet interarrival time is scaled back such that the average number of packets transmitted is reduced by a corresponding ratio to the number of added stations. For instance, if the network offered load is to be about 20 packets per second (pps), then if there is only one transmitting station, it will transmit 20 pps. If there are two transmitting stations, then each will transmit 10 pps. Similarly, if there are four transmitting stations, then each will transmit 5 pps. Reducing the number of packets that need to be transmitted per second per user increases the interarrival time of the packet arrivals.

This reduced number of packets per second each station transmits is then combined with the packet size. Two transmitting stations yields an average throughput of about 7 Mbps for a 100% offered load to the network. This is contrasted with 15 transmitting stations at a similar load attaining a throughput of about 9.4 Mbps. Generally, as more users are added to the system, the throughput decreases.

The explanation may lie in the 8 KB transmission buffer. Since packets are arriving at a lower rate per station, the transmission buffers do not fill as fast compared to network configurations with fewer stations for the same load. Since the buffers have more room to hold packets, not as many packets will be discarded. As the loading level is increased, the buffers fill faster and similar performance should be noticed between these configurations. However, if the packet size is taken into account, the drop in throughput for fewer stations can be seen. Smaller packets will fill the buffers whereas larger packets are discarded. For networks with more stations, packet interarrival times are larger and have less impact.

At loads above the capacity of the network, throughput characteristics begin to change. The throughput increases when the loading increases above the stated capacity of the network. Initially, the network is able to accommodate the traffic being submitted to it. However, once the buffer fills on each NIC, the station is forced to discard packets. This is evident since there is no queue space for the packet to wait for transmission. However, the buffer is not necessarily filled to capacity; i.e., since the packet sizes are variable, they will not evenly divide the entire buffer memory space. Therefore, smaller sized packets may be accepted into the 'left-over' memory space and queued for transmission while larger sized packets are immediately discarded. This situation effectively increases the utilization of the NIC's buffer space resulting in an increase in throughput in spite of the increased loading level. As the loading level is increased even further, this situation is more pronounced since the arrival of packets to the station is so fast that the buffer is filled practically the entire simulation.

4.2.2 Delay. The delay response of the CSMA/CD networks produced consistent results as predicted. Generally, as the loading increased, the mean ETE delay also began to increase. This trend in evident in Figures 4.11b and 4.12b.

Since there was a finite transmit buffer, the ETE delays stabilized at the points in the above mentioned plots. However at high loading levels, this was an unstable region and the delay metrics are volatile.

4.2.3 Power. Another view of the performance of the system is its Power curve. Power was defined previously in Section 2.3.3.4 as the ratio of throughput to delay. When examining power in Figures 4.11c and 4.12c with respect to the offered load, the maximum point on the curve is the optimum operating point in the network for the given configuration.

In this case, it is evident that the optimum loading level is about 70-80%. In looking at power with respect to number of stations, the fewer number of stations results in the most power. The network has its maximum power when the number of stations is limited to about two to four transmitting stations.

Thus, although throughput may increase as the load increases, the power ratio will show the detrimental effect of increased delay at these higher loading levels.

4.3 DS/CDMA Analysis

The throughput and delay times were not what was expected. Throughput was generally higher than what the network should have been capable of passing and similarly, delay was longer due to inferences made by a wireless transmission mechanism. However, upon further investigation, these effects were characteristic of proper operation. These results are discussed in the following sections.

4.3.1 Throughput. The throughput measurements of the DS/CDMA network was higher than expected. Defining the station's maximum throughput at 10 Mbps and specifying the interarrival rate to produce packets at or above a 10 Mbps rate, the throughput was about 10 Mbps. When examining actual data throughput or what some call "goodput" [PD96], this was not expected. However, when the data must be encapsulated in frame formats, such as Ethernet, there is a certain amount of overhead which should be associated with it. In the case of the CSMA/CD experiments, this is taken care of by the Ethernet packet format. At a minimum, Ethernet adds about 39 bytes of overhead to each packet (27 bytes of header and trailer plus 12 bytes of interpacket gap). This equates to about $\frac{39}{791} = 5\%$ overhead to a sample of packets with an average size of 791 bytes. Thus, Ethernet cannot achieve a full 10 Mbps throughput since 5% of the bandwidth must be used to transmit this overhead information. The link throughput (what the network medium actually transfers) may reach a full 10 Mbps, however, overall station throughput will be about 5% less giving a theoretical maximum reported network throughput of 10 Mbps-(0.05*10 Mbps) = 9.5 Mbps based on an average 791 byte packet size.

Since the DS/CDMA network was designed to closely match the CSMA/CD Ethernet LAN, the measured network throughput should have been scaled back by 5%. Otherwise, performance results identifying variations due to the factors in this research would be aliased with the omission of overhead resulting in degradation in user throughput. This scaling factor is an estimate in the amount of overhead required in the system. Rather than simply subtracting out 5% from the capacity of each network run resulting in $10Mbps-(0.05*19Mbps) = 9.5Mbps$ as the data rate of each individual station, 5% of the resulting value was subtracted out to characterize the losses that would have been noticed had overhead bits been implemented. So, if a station is offering 2.5 Mbps total throughput, then the resulting data throughput would be at most $2.5Mbps - (0.05 * 2.5Mbps) = 2.375Mbps$ which is consistent with the results of the Ethernet.

Upon further investigation, this throughput scaling was dismissed. This research assumes that the communication links have already been established. In the DS/CDMA case, each station already has knowledge of which station it wishes to transmit to. Since each transmitter/receiver pair does not have to deal with MAI due to the ideal separation in spread spectrum communications, there is little, if any, control information in the packet. As such, there is almost no packet overhead.

Source and destination addresses information is not needed since transmission negotiation would have occurred before the broadcast commenced. Additionally, only the intended receiver will be able to decode the transmission since it is encoded using the receiver's PN code sequence. Thus, the data indirectly has a destination address built into the packet through the use of the spreading code. All the control information which Ethernet must embed in the overhead portion of its packets is taken into account in the station-to-station negotiation which is assumed to occur prior to the start of the transmission links investigated in this research. There could be error detection and correction coding in the packet overhead, but this is beyond the scope of this research. Thus, there is no overhead needed in the DS/CDMA network and is not scaled out for Ethernet comparison. The throughput responses of the DS/CDMA network are in Figures 4.13a and 4.14a

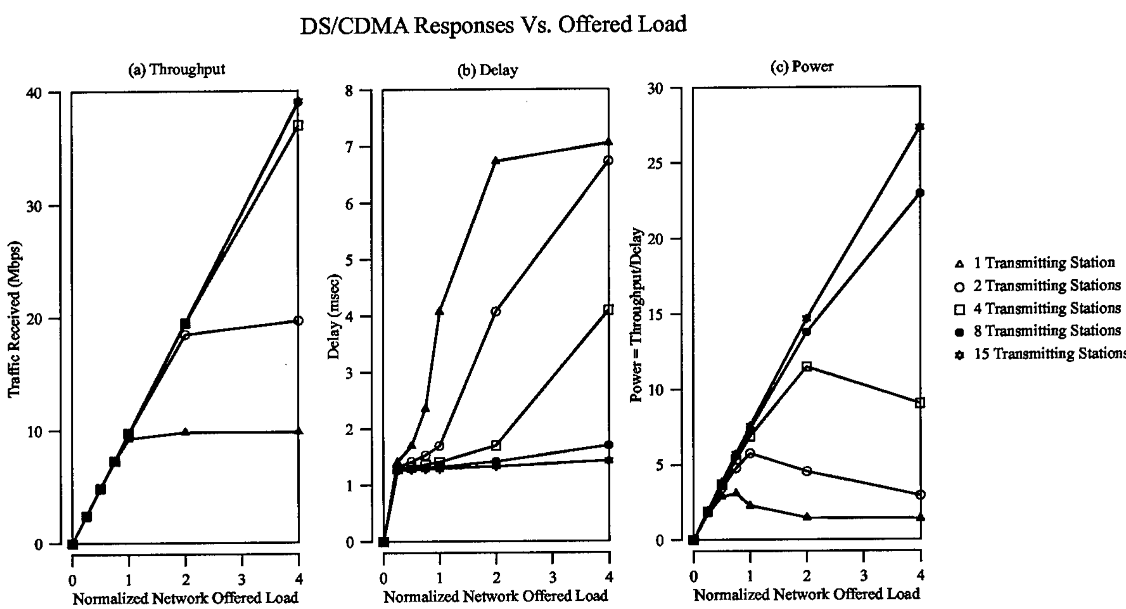

Figure 4.13. DS/CDMA Responses With Respect to Load

As can be seen, the throughput curves are not bounded. In fact, the capacity of the network is indeed defined by the sum of the capacity of each station in the network. So one station in the network results in a capacity of 10 Mbps. Two transmitting stations increase the network capacity to 20 Mbps and so on. In the

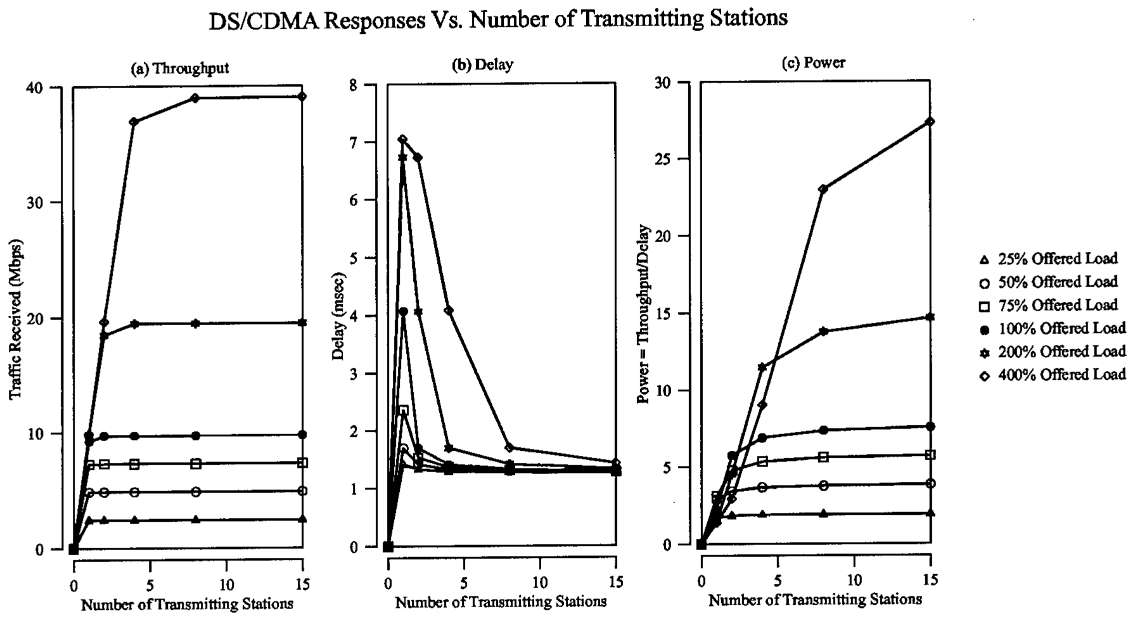

Figure 4.14. DS/CDMA Responses With Respect to Number of Stations

overload conditions, packets may be thrown out at the transmitter queue, however, the throughput will be at its maximum value.

4.3.2 Delay. Delay measurements in DS/CDMA was generally higher than expected but followed the expected trends similar to the CSMA/CD network. This is due to the transmission and propagation delay calculations in OPNET. The propagation delay calculations in the bus pipeline uses a *unit_delay* attribute that represents a normalized delay increment which equates to the speed of light calculations in the radio model. Since the DS/CDMA wired implementation is an adaptation of the wireless, model, the radio pipeline stages are used versus the bus stages used in the CSMA/CD implementations. Although the calculations are nearly identical, the radio calculations take into account a form of processing delay in the transmitter.

In the CSMA/CD models, the bus transmitter has a 'virtual queue' in which it stores packets it receives from the source to transmit. If the transmitter is currently busy sending a packet, the newly arrived packet waits before it is sent. OPNET has a defined time associated with this wait hidden from the user. In the bus models, this delay is about 0.1ms. This value is taken from the queueing delay measured in the MAC of the *ether_coax_adv* model. This value is not constant and its variability is a function of the size of the packet that is to be transmitted.

In the DS/CDMA models, this delay is higher. Analysis shows that there is anywhere from 0.5 to 0.7 ms of delay associated with the transmitter's queueing delay statistic as well as the queue component used to simulate the station's finite memory buffer. These delays in addition to the propagation delay are the ETE delay statistic in the absence of contention (the two station network configuration). The difference of 0.4-0.6 ms between the CSMA/CD and DS/CDMA models is attributed to the different transmitters used.

The transmitter component in the bus model assumes that the packet it receives for transmission is ready to be placed on the physical line [Tec97]. There is a certain amount of delay associated with this. In Ethernet, Manchester encoding is used. This basically takes the raw bits in the packet and converts them to a series of binary transitions to be placed on the cable. However, in the radio transmitter, the assumption is a little different. The data it receives from the source is assumed to be the raw bits for transmission like in the bus transmitter, but rather than doing a simple encoding scheme, the data is also modulated onto a carrier frequency [Tec97]. In this case, BPSK modulation is used. This information is then sent to the antenna component which places the data onto the medium. In the wireless case, the medium is air. In the DS/CDMA system in this research, the air has attributes consistent with a a coaxial cable. Thus, the differences in these queueing delays results in the CSMA/CD system having a slightly better delay response than the comparably configured DS/CDMA system. However, this difference is finite and overshadowed by the large queueing delay imposed at higher loading levels. The delay responses are plotted in Figure 4.13b and 4.14b

4.3.3 Power. When examining power in Figures 4.13c and 4.14c with respect to the offered load, the maximum point on the curve is the optimum operating point in the network for the given configuration. In this case, the optimum loading level is about 70% for a single transmitter. But, as more users are added to the system, the 'knee' in the power curves is shifted to the right resulting in higher power levels than CSMA/CD. In fact, there does not seem to be a maximum value for power for eight and fifteen transmitting stations for these loading conditions and configurations. A failure point stress test is explained in the following section to determine when the system performance would degrade.

4.3.4 Failure Point. The DS/CDMA system holds much promise in expanding the capabilities of the installed base of legacy LAN implementations. The simulation analysis conducted for the comparison portion of this research placed an upper limit on the total number of users which can simultaneously communicate on a maximally sized coaxial Ethernet segment. This is due to electrical concerns of the IEEE working group which drafted the IEEE 802.3 standard. However, neglecting this limit and assuming that an infinite number of users may access the medium, investigation as to the maximum number of users that could communicate was conducted. The theoretical limit has a ceiling of 513 users since this is the total number of spreading codes available for the Gold code family chosen for this research. Any additional users would violate the assumption of the value used for the cross-correlation coefficient. However, using a BER of 1×10^{-4} as the minimum acceptable error rate, it is shown in Appendix C that the actual limit is 98 concurrent communication links. This limit could be increased by choosing a different code family, however, the cross-correlation values must be recalculated as well to meet the requirements of the simulation model.

The loading level is set for each station to continuously transmit constant sized (791 bytes) packets at a rate of 10 Mbps. Additional stations are added incrementally until a failure point is reached. Failure is defined as a loss of throughput for any particular station. This point is reached when the SNR is so low at the receiver that it cannot discern the signal data from the noise. BER will no longer equal zero as well. It was found that the addition of the 97^{th} user caused the entire network throughput to drop to zero with a threshold set to 1×10^{-4} . Since perfect power control for the network has been assumed, loss of throughput for one link will also indicate a loss of throughput for any other link. This is due to the assumption that all the stations are identical and perfect power control is established. Power received at one station is the same as power received at another station. So, when one station has a low SNR, they all have a low SNR because every station's transmission interferes with every other station's transmissions.

4.4 System Comparison

An ANOVA analysis was conducted to determine the percent variation due to the factor effects. This analysis is detailed in Appendix A. Furthermore, zero mean confidence interval tests were conducted to determine if there were statistically significant differences in the responses of the networks. Figures 4.15 and 4.16 present the data collected in simulation trials.

4.4.1 Throughput. The throughput response had a wide range in variability. The responses ranged from 2.27 Mbps to about 38.55 Mbps. A square root transform was initially applied to minimize the variation in the data. The principle source of variation is due to the *Network Offered Load* which accounted for about 63% of the variation. The *MAC Scheme* only accounted for about 8% and the *Number of Stations* accounted for less than 2%.

It is widely known and accepted that throughput is directly related to the offered load of the system [PD96]. Increasing the offered load increases the throughput. Therefore, *Network Offered Load* was removed from the analysis and the ANOVA was recalculated to determine the true effects of the *MAC Scheme* and *Number of*

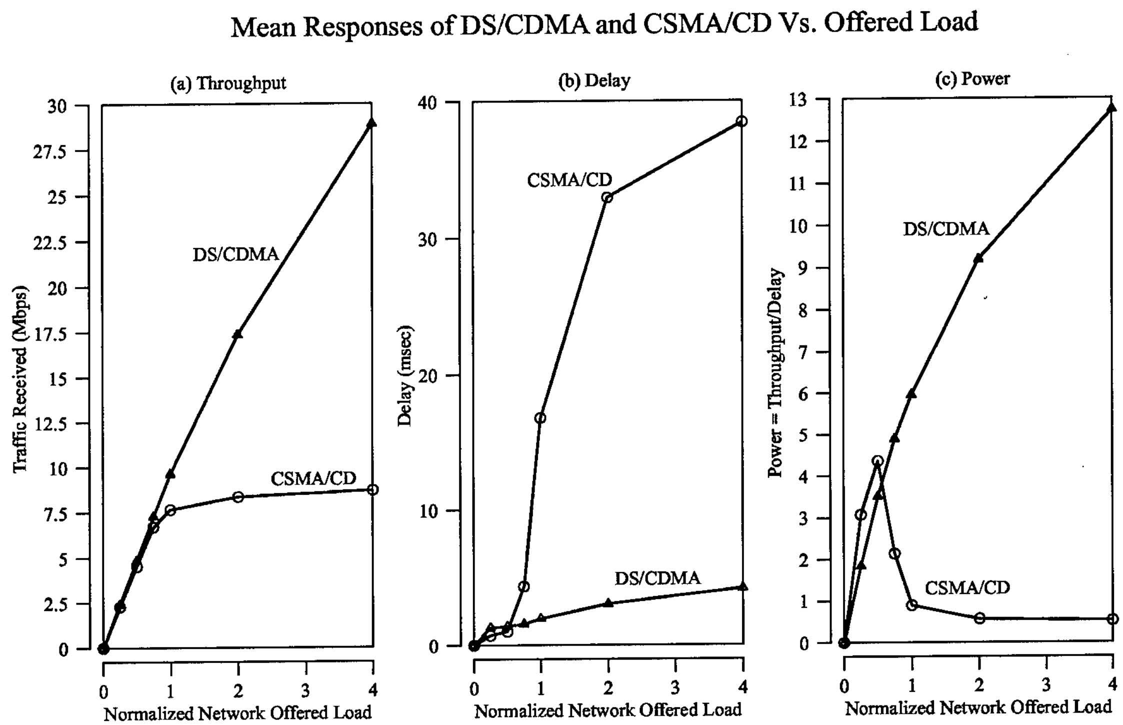

Figure 4.15. Comparison of DS/CDMA and CSMA/CD Responses Vs. Network Offered Load

Stations. At a low 25% loading level, the *MAC Scheme* constituted nearly 99% of the variation whereas the *Number of Stations* accounted for nearly less than 0.2% and their interactive affects amounted to about 0.4%. The latter effects were not statistically significant at a 95% confidence level. This clearly shows that the *MAC Scheme* is a predominant factor in throughput responses for the systems at low load.

In an overload condition (Network Offered Load = 400%), the number of stations had more of an impact with approximately 15% of the variation. The *MAC Scheme* comprised about 66% and their interactive effects was nearly 19%. This shows that at high loading levels, the impact of the *Number of Stations* becomes more significant while the *MAC Scheme* continues to be a dominant factor in the responses. Interestingly, the interaction between these two factors increased as well. This is attributed to the nature of the MAC protocol being a multiple access scheme which must accommodate as many users as possible. At low load, there is not as

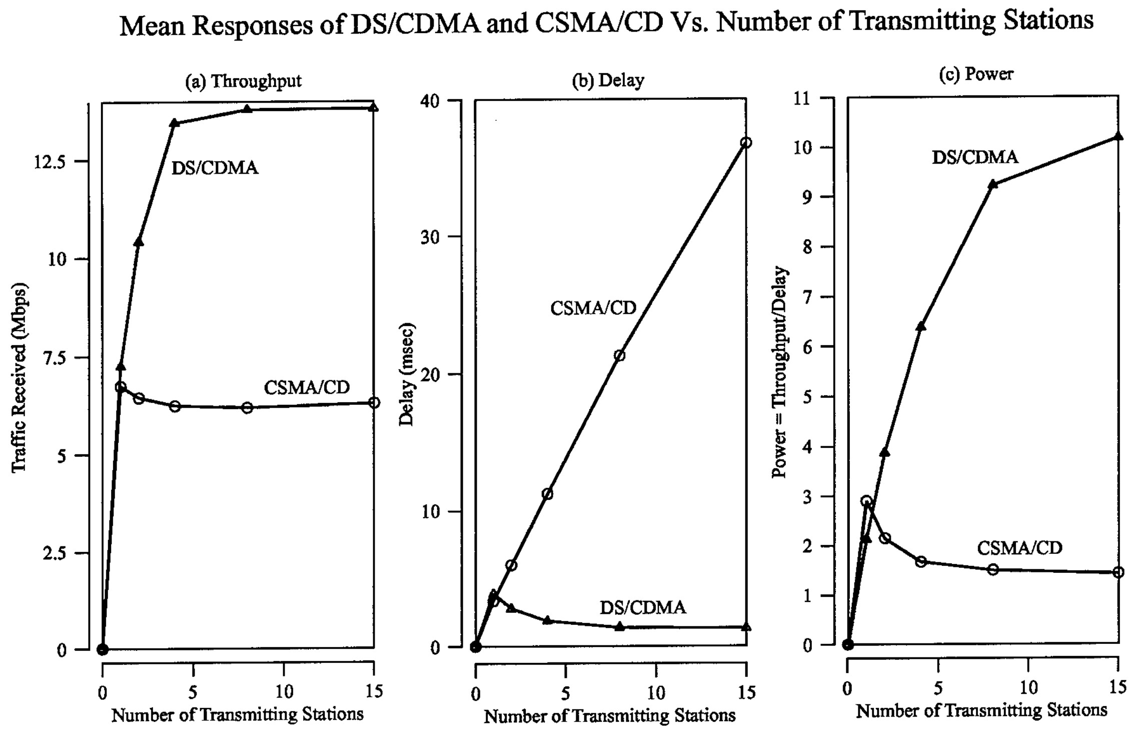

Figure 4.16. Comparison of DS/CDMA and CSMA/CD Responses Vs. Number of Stations

much contention for the medium. However, at higher loads, each station always has data to transmit, so the arbitration of the medium becomes more important. A summary of the percent variation due to the factors is given in Table 4.7.

Table 4.7. Percent Variation of Factor Effects on Throughput

Percentage of Variation Due to Factor Effects						
Factor	**25% Load**	**50% Load**	**75% Load**	**100% Load**	**200% Load**	**400% Load**
# Stations	0.16	0.23	1.31	2.43	10.49	15.02
MAC	98.57	99.39	96.71	86.21	73.59	66.16
# Stations and MAC	0.35	0.13	1.88	11.34	15.91	18.81
Error	0.91	0.24	0.10	0.02	0.00	0.00

Confidence levels were constructed on the mean difference between the DS/CDMA and CSMA/CD networks throughput responses. With 95% confidence, the DS/CDMA network had a better mean throughput response.

The general trend in Figure 4.17 shows that the throughput in the DS/CDMA network does not degrade as quickly as the CSMA/CD network. Although there is a clear distinction in throughput performance on the aggregate network performance, the effects on a per station basis is more pronounced. Comparing station-to-station throughput, the throughput was increased by nearly 250% at high loads. The CSMA/CD network was only capable of successfully transmitting approximately 25% of its data whereas the DS/CDMA network was able to sustain a rate of about 78% Mbps at a similar load.

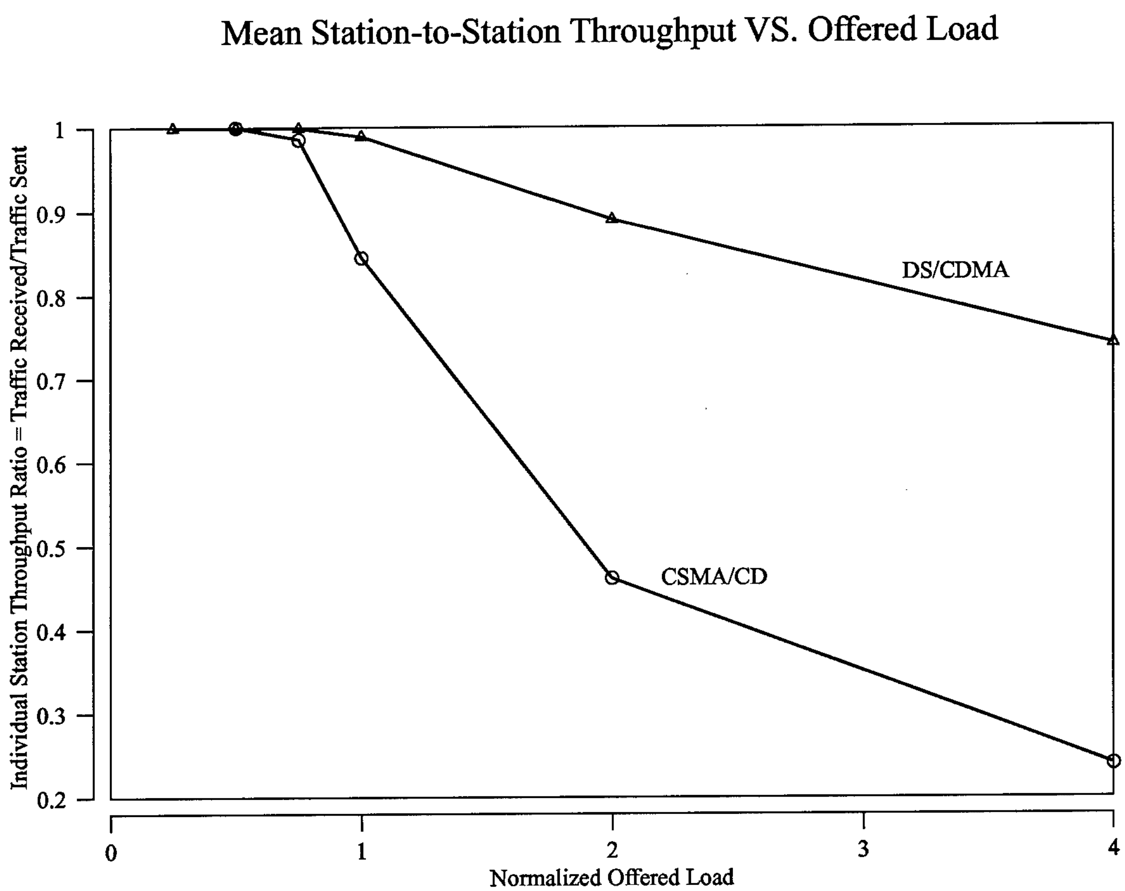

Figure 4.17. Comparison of DS/CDMA and CSMA/CD Mean Station Percent Throughput Vs. Load

4.4.2 Delay. The delay had a similarly large range in responses. The responses ranged from 0.73 msec to about 94.99 msec. The principle source of variation was again attributed to the *Network Offered Load.* This was not verified due to the fact that the data did not allow the use of an ANOVA analysis on the complete data set. Multiple transforms were applied to the data, however the assumption of errors being independent and normally distributed was violated. However, separating out the load effects and transforming the resulting data allowed for an ANOVA analysis to quantify the effects of the *MAC Scheme* and *Number of Stations.*

It is widely known and accepted that delay is also directly related to the offered load of the system [PD96]. Increasing the offered load increases the amount of delay in the network due to the queueing of packets at the transmitting stations. Therefore, *Network Offered Load* was removed as a factor and the ANOVA analysis was then possible on the resulting data set. This allowed for the calculation to determine the true effects of the *MAC Scheme* and *Number of Stations.* At a low 25% loading level, the *MAC Scheme* constituted nearly 98% of the variation. The effects due to the *Number of Stations* and their interactive effects comprised less than 2% of the total variation. This clearly shows that the *MAC Scheme* is the dominant factor in delay responses for the systems at low load.

In an overload condition (Network Offered Load = 400%), the number of stations had more of an impact with approximately 9% of the variation. The *MAC Scheme* comprised about 53% and their interactive effects was nearly 38%. This shows that at high loading levels, the impact of the *Number of Stations* becomes more significant while the *MAC Scheme* continues to be a dominant factor in the responses. Interestingly, the interaction between these two factors increased as well. This is attributed to the nature of the MAC protocol being a multiple access scheme which must accommodate as many users as possible. At low load, there is not as much contention for the medium. However, at higher loads, each station always has data to transmit with even more data arriving. Queueing delays are introduced since not every station can transmit whenever it needs to. It must wait its turn to gain access to the channel. Thus, the arbitration of the medium becomes more important. A listing of the percent variation due to the factors is given in Table 4.8

Table 4.8. Percent Variation of Factor Effects on Delay

Percentage of Variation Due to Factor Effects						
Factor	**25% Load**	**50% Load**	**75% Load**	**100% Load**	**200% Load**	**400% Load**
# Stations	0.79	6.12	0.90	0.77	0.60	9.32
MAC	98.15	69.63	54.58	72.69	66.36	53.29
# Stations and MAC	1.06	24.11	44.51	26.54	33.04	37.39
Error	0.00	0.14	0.01	0.00	0.00	0.00

Confidence levels were constructed on the mean difference between the DS/CDMA and CSMA/CD networks throughput responses. With 95% confidence, the DS/CDMA network had a lower mean delay response. Delay was reduced from an average of 38 ms to about 4 ms resulting in an improvement of nearly 850%.

An ANOVA analysis was not performed on the power metric since both of its components have already been analyzed in the previous sections. However, a zero mean difference confidence interval was calculated to determine which network had a better power response. With 95% confidence, the DS/CDMA had a better mean power performance than a comparably configured CSMA/CD system.

4.5 Summary

This chapter presented the results of the research. Verification and validation of the simulation models was first conducted. Simulation analysis of Ethernet and DSSS LANs were then discussed as well as a comparison as to which had a better delay-throughput characteristic. It was found that the DS/CDMA implementation was significantly different than a CSMA/CD implementation with higher throughput and lower delay at a 95% confidence level. Similarly, power was also higher at the same confidence level in a DS/CDMA network. Analysis of this conclusion was also presented.

V. Conclusions and Recommendations

This chapter summarizes this research's efforts. A restatement of the goal and summary of the results are presented. Recommendations for future work completes this chapter.

5.1 Goal

The goal of this research was to model and simulate a proposed wired LAN implementation of a direct sequenced spread spectrum data network. Based on simulation results, a comparison was made between the performance of a CDMA network to a 10Base2 coaxial Ethernet LAN. This research also determined the practical upper limit on the number of stations that can be supported in each network.

5.2 Results

The DS/CDMA wired network demonstrated better throughput and delay response than a comparably configured CSMA/CD network. The maximum number of users that can be accommodated in the DS/CDMA system is approximately 98 simultaneous communication links which equates to 196 total stations. The IEEE 802.3 places a limit of 30 total stations per Ethernet segment. Thus, the DS/CDMA system can accommodate nearly 4 times the number of users.

5.3 Conclusions

This research proposed a wired implementation of a wireless medium access control protocol to increase LAN performance. The OPNET simulation results show that DS/CDMA provides higher throughput and lower delay than a comparably configured CSMA/CD LAN. The main factor affecting these results is the amount of network offered load which constituted approximately 60% of the variation in overall throughput responses. With constant loading, the MAC scheme constituted the overwhelming majority of the response variation for both throughput and delay. This substantiates the claim that the MAC protocol is responsible for differences in throughput and delay.

Throughput increased by nearly 250% at higher loading levels from nearly 8.7 Mbps to well over 28 Mbps over standard CSMA/CD. Delay was reduced from an average of 38 ms to about 4 ms, an improvement of nearly 850%. These improvements were only limited based on the research bounds. These reported improvements could certainly be increased if the per station data rates and PN code sequences were chosen differently.

5.4 Implications and Impact

The implications involved in using DS/CDMA in a wired LAN are tremendous. Not only is it possible to sustain more stations in a given network segment, but this performance increase is available without a costly cabling infrastructure upgrade. No longer must users endure lengthy file transfers or suffer reduced productivity based on excessive network delay.

Current coaxial-based Ethernets can be revitalized to accommodate new multimedia functionality where it wasn't practical before. Delay limited applications, e.g., voice over IP (VoIP) or internet telephony, may be possible due to the reduction of mean end-to-end delay. Large file transfers such as downloading entire theater-quality movie files may not be so prohibitive due to the increase in mean throughput. The video rental industry could loan videos via a simple Internet download. Secure videoconferencing is also possible due to the inherent security offered by spread spectrum techniques.

Military application potential is also impacted. The U.S. Air Force, and the Department of Defense as a whole, has a need to maintain a high quality, highly reliable internetwork. Costly upgrades impair this level of readiness largely due to the inability to upgrade the cabling infrastructure. Replacing a network interface card in a personal computer is far easier and could potentially be done at a much lower expense. Increased multimedia capabilities, as described previously, can also be implemented. These added functions will enhance productivity and readiness.

Spread spectrum communication mechanisms have the potential to enhance virtually any type of currently fielded time-division-based data networks. The costs associated with this type of upgrade is nominal when considering the enormous throughput and delay improvement to be gained with this medium access control protocol.

5.5 Recommendation for Future Work

The DS/CDMA model of this research used design parameters of a baseband 10Base2 RG-58 coaxial cable. This cable is capable of gigabit per second rates [Tan96] using digital signaling. Synchronization time was neglected but could impact the end-to-end delay measurements reducing the average difference between the CSMA/CD and DS/CDMA models. Synchronization may be divided into coarse and fine, coarse is used for initial signal acquisition and fine is used to continuously track the signal. Losses can be incurred if synchronization is not maintained. Power control was also assumed for this research which would require some sort of overhead and was not addressed.

Many networks use UTP and broadband cabling. UTP has four strands of copper wire each of which could be designated a specific channel on its own. Control and synchronization information could be used on separate lines to allow a dedicated communication line as simulated for this research. Broadband cable has the capability of having multiple frequency bands while still maintaining a large bandwidth of about 300-450MHz [Tan96]. Rather than using UTP for the extra channels, the bandwidth of a broadband cable could be divided into smaller frequency bands (FDMA) for the extra control information needed for true implementation of a DS/CDMA LAN.

This research was a proof of concept for using spread spectrum communication techniques in a wired LAN environment. Future research in this area could involve adapting the models of this study and implementing synchronization timing issues as well as other conversation setup functions. The 'near-far' effect could also be incorporated into the existing models and the lack of perfect power control evaluated in the absence of a control channel. The channel medium could also be investigated for adaptation onto a UTP or broadband cabling and address both of these issues. Once overhead and link setup are properly addressed, another area of study is the implementation of a DS/CDMA testbed to experiment with actual communication station hardware.

Characteristics for Gold codes was employed in this research using an expected value for the cross-correlation coefficient defined in Table 2.2. Another venue for future research would be to implement the models created for this research for DSSS and dynamically calculate the cross-correlation between codes. This would entail actual assignment of PN codes to the different stations rather than simply identifying the station by some arbitrary reference address. Performance of a dynamic calculation as to the cross-correlation between a specific pair of codes for each occurrence of a 'collision' in the network would then be implemented for evaluation.

5.6 Summary

The research goals have been met. Spread spectrum techniques hold promise for upgrading legacy coaxial cable local area networks. It is possible for users traversing the internet to not suffer from the tedium of lengthy downloads. Spread spectrum internet applications may one day provide an end to the "world-wide wait".

Appendix A. Model Analysis

This section elaborates on the analysis conducted in Chapter 4. An ANOVA table analysis was conducted. In using the ANOVA, a number of assumptions were made which required testing. If the tests do not hold, then the conclusions drawn from the ANOVA would be compromised. These assumptions are

- There is a linear relationship between the factors and the response.
- The errors are statistically independent
- The errors are normally distributed with zero mean and constant standard deviation.

The first response analyzed is the throughput metric. The measurements obtained from the simulations are listed in Table A.1

Table A.1. Throughput Measurements in Mbps

		Number of Transmitting Stations (B)									
		2		4		8		16		30	
		MAC(C)		MAC(C)		MAC(C)		MAC(C)		MAC(C)	
		DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD
Network Offered Load (A)	25	2.44	2.27	2.43	2.28	2.42	2.28	2.42	2.27	2.44	2.28
		2.45	2.30	2.43	2.27	2.44	2.29	2.43	2.29	2.45	2.27
		2.44	2.28	2.43	2.28	2.43	2.28	2.43	2.29	2.44	2.25
	50	4.87	4.54	4.86	4.53	4.85	4.54	4.85	4.54	4.88	4.54
		4.89	4.58	4.86	4.55	4.86	4.54	4.86	4.53	4.89	4.54
		4.87	4.56	4.88	4.53	4.87	4.53	4.87	4.55	4.89	4.53
	75	7.26	6.81	7.30	6.72	7.29	6.61	7.30	6.66	7.31	6.78
		7.27	6.84	7.30	6.74	7.30	6.62	7.30	6.68	7.31	6.77
		7.27	6.82	7.31	6.73	7.30	6.61	7.31	6.68	7.32	6.73
	100	9.24	8.66	9.73	7.33	9.73	7.25	9.73	7.41	9.75	7.68
		9.23	8.67	9.74	7.27	9.75	7.24	9.74	7.40	9.73	7.67
		9.24	8.66	9.76	7.34	9.76	7.27	9.76	7.40	9.76	7.70
	200	9.83	9.09	18.47	8.74	19.50	7.91	19.49	7.96	19.51	8.12
		9.83	9.09	18.46	8.73	19.49	7.87	19.48	7.94	19.49	8.13
		9.83	9.09	18.46	8.74	19.50	7.88	19.49	7.94	19.50	8.12
	400	9.83	9.05	19.67	9.06	36.93	8.83	38.98	8.27	39.05	8.33
		9.83	9.05	19.67	9.06	36.94	8.84	38.98	8.27	38.95	8.32
		9.83	9.05	19.67	9.06	36.91	8.85	38.99	8.29	39.01	8.32

The normal Q-Q plot revealed that the linearity assumption between the factors and the throughput response was violated. In analyzing the throughput responses in Figure 4.15 and 4.16 it is evident that there is not a linear relationship for the mean throughput performance.

This resulted in a regression model with an R^2 value of about 90%. This in and of itself was not necessarily a bad model. At lower loading levels, the relationship was very linear. However, at higher loads, the relationship looks more logarithmic. The ratio of $\frac{y_(max)}{y_(min)}$ was also very large. To reduce the range of the throughput response, a square root transform was applied to the data.

This transform resulted in a regression model having an R^2 value of approximately 98% thus making the relationship of the responses to the predictor variable more linear. An ANOVA table was constructed on this transformed data with results identified in Table A.2.

Table A.2. ANOVA Table of Throughput

Source of Variation	Sum of Squares	Percentage Variation	Degrees of Freedom	Mean Square	F- Computed	F- Statistic	Significant
Load	125.27	62.58	5	25.05	4.78E+06	2.45	Yes
Stations	3.07	1.54	4	0.77	1.47E+05	2.61	Yes
MAC	15.51	7.75	1	15.51	2.96E+06	4.08	Yes
Load X Stations	9.79	4.89	20	0.49	9.35E+04	1.84	Yes
Load X MAC	30.76	15.37	5	6.15	1.17E+06	2.45	Yes
Stations X MAC	4.94	2.47	4	1.23	2.36E+05	2.61	Yes
Load X Stations X MAC	10.84	5.42	20	0.54	1.03E+05	1.84	Yes
Error	0.00	0.00	120	0.00			
Total	200.19	100.00	179				

Since the *Network Offered Load* factor constituted about 62% of the variability, separate ANOVA analysis was conducted at each of the loading levels (25% through 400% respectively). The ANOVA tables for these loading levels are presented in Table A.3.

The delay measurements also had a significant range in responses from about 0.7 msec to above 94 msec of mean ETE delay. This is shown in Table A.4.

Table A.3. ANOVA Tables of Throughput at the 5 Loading Levels

ANOVA for Throughput @ 25% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.00	0.18	4.00	0.04	9.81E-01	2.87	No
MAC	0.18	98.57	1.00	98.57	2.17E+03	4.35	Yes
Stations X MAC	0.00	0.35	4.00	0.09	1.91E+00	2.87	No
Error	0.00	0.91	20.00	0.05			
Total	0.19	100.00	29.00				

ANOVA for Throughput @ 50% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.00	0.23	4.00	0.06	1.27E+00	2.87	No
MAC	0.80	99.39	1.00	99.39	2.19E+03	4.35	Yes
Stations X MAC	0.00	0.13	4.00	0.03	7.35E-01	2.87	No
Error	0.00	0.24	20.00	0.01			
Total	0.81	100.00	29.00				

ANOVA for Throughput @ 75% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.03	1.31	4.00	0.33	7.20E+00	2.87	Yes
MAC	2.50	96.71	1.00	96.71	2.13E+03	4.35	Yes
Stations X MAC	0.05	1.88	4.00	0.47	1.04E+01	2.87	Yes
Error	0.00	0.10	20.00	0.00			
Total	2.59	100.00	29.00				

ANOVA for Throughput @ 100% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.83	2.43	4.00	0.61	1.34E+01	2.87	Yes
MAC	29.39	86.21	1.00	86.21	1.90E+03	4.35	Yes
Stations X MAC	3.86	11.34	4.00	2.83	6.25E+01	2.87	Yes
Error	0.01	0.02	20.00	0.00			
Total	34.09	100.00	29.00				

ANOVA for Throughput @ 200% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	86.63	10.49	4.00	2.62	5.78E+01	2.87	Yes
MAC	607.57	73.59	1.00	73.59	1.62E+03	4.35	Yes
Stations X MAC	131.38	15.91	4.00	3.98	8.77E+01	2.87	Yes
Error	0.00	0.00	20.00	0.00			
Total	825.58	100.00	29.00				

ANOVA for Throughput @ 400% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.37	15.02	4.00	3.76	8.28E+01	2.87	Yes
MAC	1.64	66.16	1.00	66.16	1.46E+03	4.35	Yes
Stations X MAC	0.47	18.81	4.00	4.70	1.04E+02	2.87	Yes
Error	0.00	0.00	20.00	0.00			
Total	2.48	100.00	29.00				

It was also found that the normality assumption was also violated for these response as well. Again, the relationship varied depending on the offered load to the network. Numerous transforms were applied to the data, but none resulted in a model with an R^2 greater than about 50%. Removing the *Network Offered Load* factor from the analysis, an ANOVA analysis was possible at the low to high loading levels.

Table A.4. Delay Measurements in msec

		Number of Transmitting Stations (B)									
		1		2		4		8		15	
		MAC Scheme(C)		MAC Scheme(C)		MAC Scheme(C)		MAC Scheme(C)		MAC Scheme(C)	
		DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD	DS/CDMA	CSMA/CD
Normalized Network Offered Load (A)	25%	1.41	0.73	1.33	0.74	1.29	0.74	1.27	0.74	1.26	0.74
		1.41	0.73	1.32	0.74	1.29	0.74	1.27	0.74	1.26	0.75
		1.41	0.73	1.33	0.74	1.29	0.74	1.27	0.74	1.26	0.74
	50%	1.70	0.95	1.42	1.01	1.33	1.09	1.29	1.07	1.27	1.10
		1.70	0.96	1.41	1.01	1.33	1.05	1.29	1.07	1.27	1.09
		1.70	0.95	1.42	1.00	1.33	1.06	1.29	1.10	1.27	1.12
	75%	2.36	1.57	1.53	2.36	1.37	4.47	1.31	6.41	1.28	7.02
		2.36	1.58	1.53	2.39	1.37	4.45	1.31	6.11	1.28	7.34
		2.37	1.56	1.53	2.43	1.37	4.36	1.31	6.28	1.28	6.80
	100%	4.07	3.56	1.70	7.90	1.41	12.75	1.33	22.39	1.29	37.34
		4.07	3.63	1.70	8.06	1.42	12.57	1.33	22.22	1.29	37.17
		4.07	3.60	1.70	7.96	1.41	12.60	1.32	22.64	1.29	36.85
	200%	6.73	6.53	4.07	10.86	1.70	23.56	1.41	44.52	1.33	79.37
		6.73	6.53	4.08	10.87	1.70	23.67	1.41	44.44	1.33	79.75
		6.73	6.54	4.07	10.88	1.70	23.64	1.41	44.42	1.33	79.47
	400%	7.05	6.80	6.73	13.04	4.09	24.65	1.70	52.59	1.43	94.99
		7.05	6.80	6.73	13.05	4.09	24.76	1.70	52.57	1.43	94.96
		7.04	6.80	6.73	12.99	4.09	24.58	1.70	52.53	1.43	94.94

The ANOVA table using the raw and transformed delay measurements are identified in Table A.5.

Although the raw responses did not follow a linear relationship to the factors, transformation of the data helped to satisfy the ANOVA assumptions and allowed analysis using this approach. Curvilinear regression was a possibility. However, since only the variation due to the factors was needed and prediction of future responses were not, an ANOVA analysis using R^2 values from 85% and higher were assumed adequate.

Table A.5. ANOVA Tables of Delay at the 5 Loading Levels

ANOVA for Delay @ 25% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.02	0.79	4.00	0.20	1.15E+03	2.87	Yes
MAC	2.47	98.15	1.00	98.15	5.73E+05	4.35	Yes
Stations X MAC	0.03	1.06	4.00	0.26	1.55E+03	2.87	Yes
Error	0.00	0.00	20.00	0.00			
Total	2.51	100.00	29.00				

ANOVA for Delay @ 50% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.08	6.12	4.00	1.53	8.94E+03	2.87	Yes
MAC	0.96	69.63	1.00	69.63	4.07E+05	4.35	Yes
Stations X MAC	0.33	24.11	4.00	6.03	3.52E+04	2.87	Yes
Error	0.00	0.14	20.00	0.01			
Total	1.37	100.00	29.00				

ANOVA for Delay @ 75% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.02	0.90	4.00	0.22	1.31E+03	2.87	Yes
MAC	0.98	54.58	1.00	54.58	3.19E+05	4.35	Yes
Stations X MAC	0.80	44.51	4.00	11.13	6.50E+04	2.87	Yes
Error	0.00	0.01	20.00	0.00			
Total	1.79	100.00	29.00				

ANOVA for Delay @ 100% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.13	0.77	4.00	0.19	1.12E+03	2.87	Yes
MAC	11.86	72.69	1.00	72.69	4.24E+05	4.35	Yes
Stations X MAC	4.33	26.54	4.00	6.64	3.88E+04	2.87	Yes
Error	0.00	0.00	20.00	0.00			
Total	16.32	100.00	29.00				

ANOVA for Delay @ 200% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.11	0.60	4.00	0.15	8.77E+02	2.87	Yes
MAC	12.28	66.36	1.00	66.36	3.88E+05	4.35	Yes
Stations X MAC	6.11	33.04	4.00	8.26	4.82E+04	2.87	Yes
Error	0.00	0.00	20.00	0.00			
Total	18.51	100.00	29.00				

ANOVA for Delay @ 400% Load							
Source of Variation	**Sum of Squares**	**Percentage Variation**	**Degrees of Freedom**	**Mean Square**	**F- Computed**	**F- Statistic**	**Significant**
Stations	0.16	9.32	4.00	2.33	1.36E+04	2.87	Yes
MAC	0.91	53.29	1.00	53.29	3.11E+05	4.35	Yes
Stations X MAC	0.64	37.39	4.00	9.35	5.46E+04	2.87	Yes
Error	0.00	0.00	20.00	0.00			
Total	1.72	100.00	29.00				

Appendix B. Model Validation

This section elaborates on the model validation conducted in Chapter 4. There are three types of tests conducted to ensure the model assumptions were not violated. These tests include:

- Visual Residual Plots
- R^2 Test for Linearity
- Levene's Test for Homogeneity of Variances

B.1 Normal Quantile-Quantile Plot

The F-test used in the ANOVA analysis assumes a linear relationship between the responses and the factors and that the errors are normally distributed. To test normality, a normal quantile-quantile (Q-Q) plot is constructed. The x-axis is the quantilies of the normal distribution with respect to the number of observations in the analysis. The y-axis is the rank-ordered responses. If the responses do not follow a linear relationship with respect to the normal quantiles, then the assumption is violated and the conclusions drawn from the ANOVA are compromised [Jai91].

B.2 R^2 Linearity Test

The normal Q-Q plot is a visual test for normality. However, a visual determination of a linear relationship is very subjective. The coefficient of determination statistic, more commonly known as the R-squared value, can be interpreted as the proportion of the variation in the dependent variable that is statistically explained by the associated independent variable from the regression model [Jai91]. It is a more analytic method of determining linearity and is defined as

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} \tag{B.1}$$

An R^2 value of 1 indicates that 100% of the variation of the dependent variable may be explained by the linear relationship with the independent variable. An R^2 of 0% indicates that there is no relationship between the dependent and independent variables. Another way of interpreting the R^2 value is that the closer it is to 100%, the closer the relationship between the response and the factors is linear. So, R^2 is a measure of linearity.

B.3 Residual Scatter Plots

Another F-test assumption for ANOVA is that the errors, or residuals, are independent and have a constant standard deviation. A scatter plot of the residuals versus the predicted values can be used as a visual test. If there are no trends in the points, the assumption of independence is upheld. Furthermore, if the residual deviation from zero is relatively constant, then the assumption of constant standard deviation is upheld. Instead of the predicted values, the scatter plot can also be used by plotting the residuals versus the experiment number. This also will produce trends, if applicable, that will test for assumption violation [Jai91].

B.4 Normality Test Results

Visual residual plots present the residuals (errors) of the data and plot them with respect to quantiles of a specified distribution, fitted (average) responses, or experiment number. The purpose of these plots is to notice a trend in the residuals which will either support of contradict the assumptions of the ANOVA analysis. The first of these tests is the assumption of normally distributed errors. This can be tested using a normal quantile-quantile plot of the residuals versus quantiles of the the normal distribution. These plots are identified in Figures B.1 and B.2.

Initially, the R^2 value for the delay response was 0.57 which is not very close to 1. A value of 1 indicates near perfect linearity. Many transforms were applied, but none resulted in R^2 much better than the untransformed results. As identified

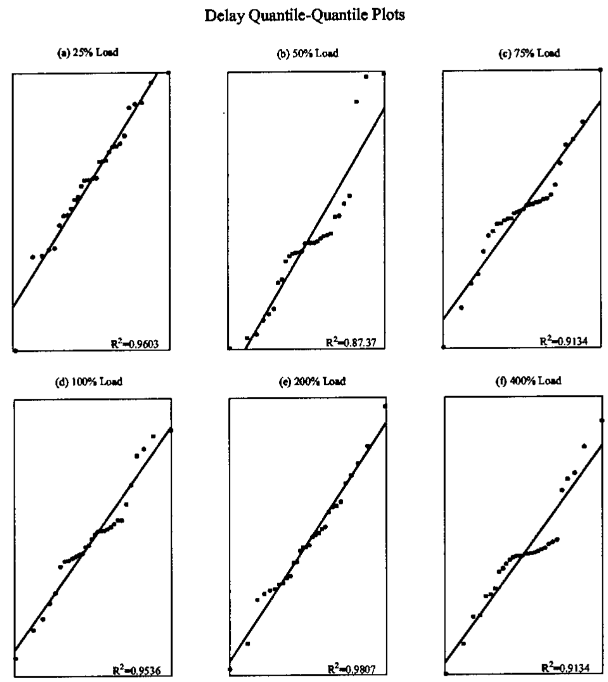

Figure B.1. Normal Q-Q Plot of the Delay Residuals

in Appendix A, the *Network Offered Load* factor was removed from consideration and ANOVA was conducted on the 25% through 400% offered load levels. The 25% offered load resulted in an R^2 value of 96%, thus no transform was needed. Without transforming the other levels, most had R^2 values far below 90%. Due to the high variability at the higher loading levels, a few transforms were needed. These are identified in Table B.1.

Table B.1. Data Transforms

Loading Level	**Transform**
25%	None
50%	None
75%	$y = x^{\frac{-5}{4}}$
100%	$y = LOG(x-1)$
200%	$y = LOG(x-1)$
400%	$y = x^{\frac{-11}{17}}$

These transforms result in the normal Q-Q plots in Figure B.1. These were the best fitted transforms that could be found. Since prediction of responses based on the regression equation is not needed, an R^2 value of 85% and above was considered acceptable to satisfy the ANOVA assumptions.

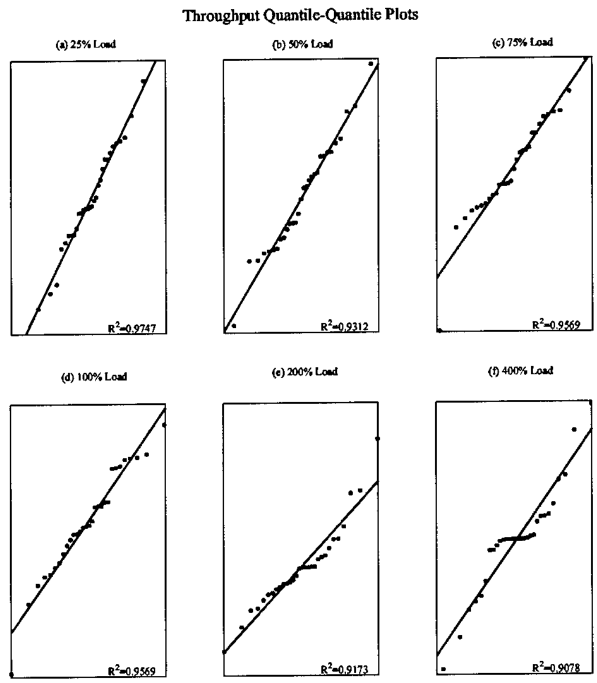

Figure B.2. Normal Q-Q Plot of the Throughput Residuals

Figure B.2 shows the normal Q-Q plot of the raw and transformed throughput data. The use of a transform was only needed at the 400% loading level. Initially, the raw data resulted in an R^2 value of less than 56%. Applying a $y = LOG(x)$ resulted in an R^2 value of 91%. Using the same reasoning as above, the normal Q-Q plots all pass the test. Thus, the use of the visual plots and the R^2 test verified that the assumption of normally distributed error for the respective ANOVA analyses.

B.5 Scatter Plot Test Results

The next assumption deals with the independently distributed errors with constant standard deviation. This can be verified with a scatter plot of the residuals with respect to the fitted values or the experiment number. For a numerical equivalent, the Levene test for homogeneity may also be employed. The plots for this verification is in Figures B.3, B.4, B.5, and B.6.

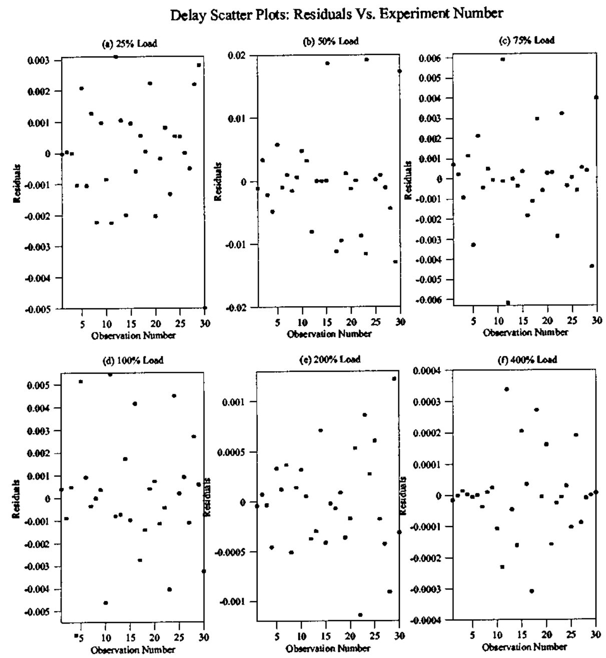

Figure B.3. Scatter Plot of Residuals Vs. Experiment Number of Delay Responses

As can be seen, most of the scatter plot of the residuals of both the delay and throughput in Figures B.5 and B.6 do not show any noticeable trend. Some look as though there may be some clustering when plotting with respect the fitted values. In these circumstances, Levene's test is needed to quantify the trend if any. However, when plotting with respect to the observation number, there are no noticeable trends.

B.5.0.1 Levene Test for Homogeneity of Variances. An analytic test to determine the constant standard deviation of errors is to test for homogeneity of the variances. Homogeneity of variances is roughly equivalent to equal standard variations across samples. Since the F-test in ANOVA is particularly sensitive to the assumption of constant or equal standard deviations, Levene's test can be used to analytically determine the conclusion which could be drawn from a scatter plot [Mil86].

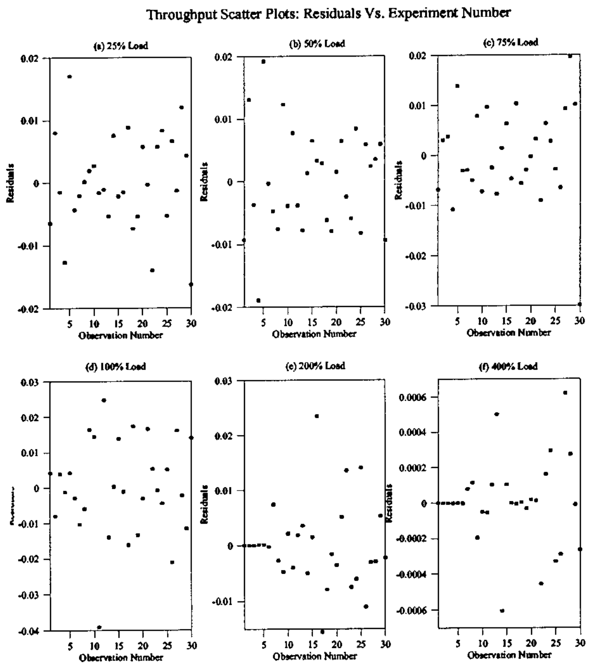

Figure B.4. Scatter Plot of Residuals Vs. Experiment Number of Throughput Responses

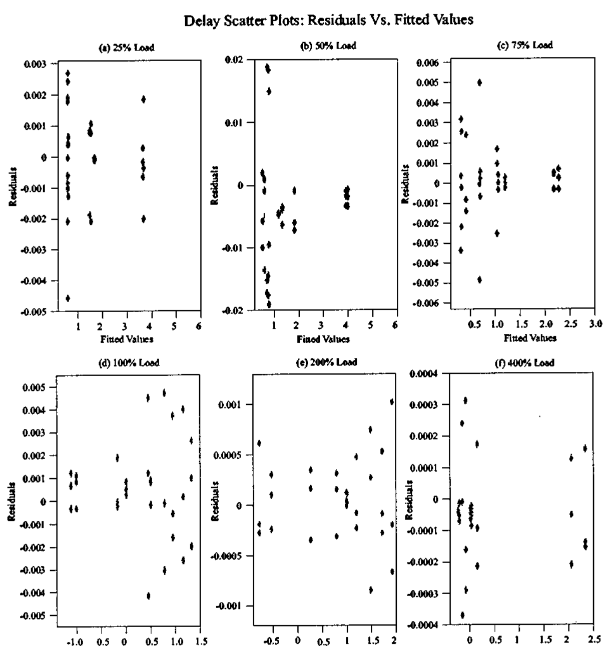

Figure B.5. Scatter Plot of Residuals Vs. Fitted Values of Delay Responses

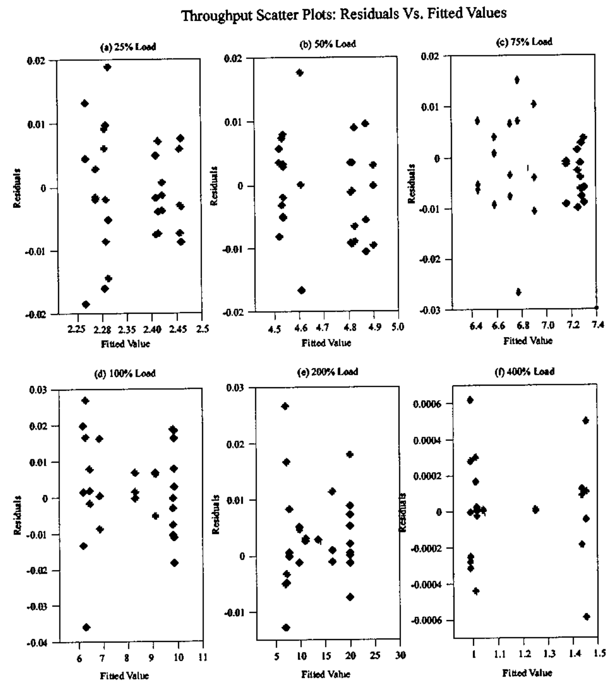

Figure B.6. Scatter Plot of Residuals Vs. Fitted Values of Throughput Responses

Levene's test is an alternative to Bartlett's test which is most commonly used. Levene's test is preferred since it is not as sensitive to the assumption of normality as is Bartlett's test. Levene's test uses the variable W defined as

$$W = \frac{(N-k)\sum_{i=1}^{k} N_i \left(\bar{Z}_{i.} - \bar{Z}_{..}\right)^2}{(k-1)\sum_{i=1}^{k}\sum_{j=1}^{N_i}\left(\bar{Z}_{ij} - \bar{Z}_{i.}\right)^2} \tag{B.2}$$

where N is the sample size, k is the number of subgroups, N_i is the sample size of the i^{th} subgroup and Z is defined as

$$\bar{Z}_{ij} = |Y_{ij} - \bar{Y}_{i.}| \tag{B.3}$$

where Y is the given variable and $\bar{Y}_{i.}$ can be either the mean or the median value. If the variances are not homogenous or the standard deviations are not equal, then

$W > F_{(1-\alpha),(k-1),(N-k)}$ where F is the F-distribution with $k-1$ and $N-1$ degrees of freedom at a significance level of α.

Visually, it seems that these responses have no real trend. But this is a subjective conclusion. Using the Levene Test for Homogeneity of Variances, it can be seen in Table B.2 that all the responses have a constant standard deviation.

Table B.2. Levene Test Results

Test Plot	**W-Calculated**	**F-Statistic**	**Constant st. dev.?**
Throughput 25% Load	0.01	2.18	Yes
Throughput 50% Load	0.01	2.18	Yes
Throughput 75% Load	0.08	2.18	Yes
Throughput 100% Load	0.16	2.18	Yes
Throughput 200% Load	0.73	2.18	Yes
Throughput 400% Load	1.10	2.18	Yes
Delay 25% Load	0.05	2.18	Yes
Delay 50% Load	0.41	2.18	Yes
Delay 75% Load	0.00	2.18	Yes
Delay 100% Load	0.05	2.18	Yes
Delay 200% Load	0.04	2.18	Yes
Delay 400% Load	0.64	2.18	Yes

The ANOVA assumptions hold and the confidence in the results is substantiated.

Appendix C. DS/CDMA Failure Point Prediction

This section details the analysis used to calculate the maximum number of simultaneous communication links that the DS/CDMA system employed in this research could maintain. It was found that 98 is the theoretical maximum limit this system may hold. This is far less than the 513 available spreading codes in the Gold code family chosen. Increases in the number of users is possible by redefining the parameters such as the code family, the available bandwidth, and the data rate. Special calculation decisions were made due to how OPNET uses the radio transceiver pipeline to perform the SNR and BER calculations.

The DS/CDMA system should fail when the bit error rate rises above 1×10^{-4}. This is the upper limit when the number of bit errors exceeds the maximum that a data network can tolerate [Skl88]. Since there was no special coding involved in the transmission, BPSK is used since it is the optimum uncoded modulation scheme [Skl88]. The system is operating at baseband which limits the effective bandwidth and processing gain of the signal. Consider the following frequency response of a baseband signal. Since it is baseband, the signal is centered about 0 Hz in a $sinc^2$ curve as depicted in Figure C.1

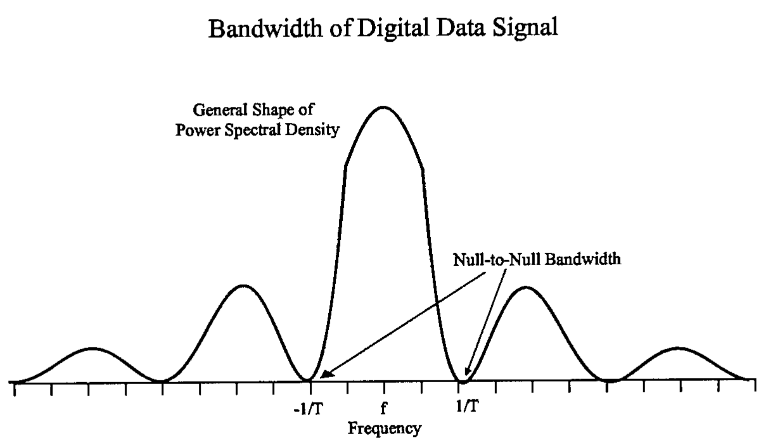

Figure C.1. Baseband BPSK Signal Spectrum

The signal actually occupies both positive and negative frequencies. However, only positive frequencies are of interest. This effectively eliminates half the power of the signal. Frequently, the ratio $\frac{E_b}{N_o}$ is used to calculate a probability of error. For BPSK modulation, probability of error, P_e, or BER is [Skl88]

$$P_e = Q\left(\sqrt{2\left(\frac{E_b}{N_o}\right)}\right) \tag{C.1}$$

where Q is the complementary error function already presented in Chapter 3. This calculation for BER assumes additive white gaussian noise, matched filter detection, and a constant noise power spectral density of $\frac{N_o}{2}$.

Since OPNET assumes a baseband signal and calculates a signal-to-noise ratio, conversion of $\frac{E_b}{N_o}$ to SNR is a simple algebraic manipulation. E_b is the energy per bit and can be calculated by dividing the baseband signal power by the data rate or

$$E_b = \frac{S_{BB}}{R} \tag{C.2}$$

where S is the power of the signal of interest and R is the data rate of the signal and the subscript BB denotes a baseband signal. The inverse of R yields the symbol time, T_s which is the duration, in seconds, of the data bit. Since the data rate is fixed at 10 Mbps, and assuming the modulation yields $1\frac{bit}{Hz}$ results in a $R = 10$ MHz. This yields a T_s of 100 ns. Average baseband noise is calculated as

$$N_{BB} = \frac{N_o W_{BB}}{2} \tag{C.3}$$

where N_o is the noise power spectral density and W_{BB} is the bandwidth the baseband signal occupies. From Figure C.1, the baseband null-to-null bandwidth of the data signal is simply

$$W_{BB} = \frac{1}{T_s} = R_s \tag{C.4}$$

Without the inclusion of the processing gain (this will be included later in the calculations), the $\frac{E_b}{N_o}$ is

$$\frac{E_b}{N_o} = \frac{\frac{S_{BB}}{R}}{\frac{N_{BB \times 2}}{W_{BB}}} = \frac{S_{BB}}{2 \times N_{BB}} \times \frac{W_{BB}}{R_s} = \frac{SNR_{BP}}{2} \tag{C.5}$$

where the subscript BP represents a bandpass signal. Thus, the BER is

$$(P_e)_{BB} = Q\left(\sqrt{2\frac{E_b}{N_o}}\right) = \left(Q\sqrt{SNR_{BP}}\right) \tag{C.6}$$

The ultimate effect of spreading the signal across the full capacity of the channel is a reduction in the noise power by the processing gain of the system [Skl88]. In effect, the effective SNR is

$$SNR_{Effective} = \frac{S}{\frac{N}{GP}} = \frac{S}{N} \times GP = \frac{S}{N}_{dB} + GP_{dB} \tag{C.7}$$

The noise in the system is only due to the transmission of the other users in the network. Hence, every transmission other than the signal of interest is considered a noise source. Since each user is also in the same gold code family, there will be some extra power allowed to pass on to the demodulator with its effect quantified by the cross-correlation coefficient, $\alpha = 0.0176$. The total amount of power received is the portion, α, of the sum of the powers of all other transmissions reduced by processing gain of the system. Recall that the processing gain of the system given in EQ 3.1 is the ratio of the bandwidth divided by the data rate where the bandwidth is the full capacity of the RG-58 (400 MHz) and the data rate is 10 Mbps. This processing gain is effectively halved due to the use of a baseband signal. Therefore, a factor of two must be removed in the OPNET calculations to compensate for this type of signalling. Thus, the total calculated noise system power from OPNET due to the multiple access interferes is

$$M_{BB} = \frac{\frac{Power}{User} \times \text{Number of Interfers} \times \alpha}{\frac{GP}{2}} \tag{C.8}$$

Since the upper limit on the BER is 1×10^{-4}, it can be found that the argument of the Q function must be no less than 3.5. Thus

$$\sqrt{SNR_{BP}} \geq 3.5 \rightarrow SNR \geq 12.25$$

The power received is a constant 1 Watt due to the assumption of perfect power control. Thus, the total number of users that may be permitted in the system without increasing the BER above 1×10^{-4} is

$$N \leq 1W/12.25 = 81.63 \text{ mW}$$

$$\text{Number of users} \geq \frac{N \times \left(\frac{GP}{2}\right)}{\frac{Power}{User} \times \alpha}$$

$$= \frac{81.63mW \times \left(\frac{40}{2}\right)}{1W \times 0.0176} \cong 98 \text{ maximum users}$$

Thus, the total number of users this system can support is 98 concurrent communication links within its own LAN. This maximum number could be increased redefining the Gold code used. This will change the total number of available codes and the associated cross-correlation coefficient. Also, the data rate could be decreased which increases the processing gain of the system. This value was verified in the simulation.

Appendix D. Pipeline Stage Code

The following pages list the code used in the DSSS pipeline stages. All 14 stages of the pipeline are listed.

```
/************************************************/
/* STAGE 0                                      */
/* Filename: dsss_rxgroup.ps.c                  */
/* Description:                                 */
/*   Receiver group model for wireless          */
/*   spread spectrum link Transceiver Pipeline  */
/* Author: Robert J. Bonner                     */
/************************************************/

#include "opnet.h"

#if defined (__cplusplus)
extern "C"
#endif
int
dsss_rxgroup (Objid tx_obid, Objid rx_obid)
	{
	/** Determine the potential for communication between   **/
	/** given transmitter and receiver channel objects.     **/
	FIN (dsss_rxgroup (tx_obid, rx_obid));

	/* By default, all receivers are considered         */
	/* as potential destinations.                        */
	FRET (OPC_TRUE);
	}
```

```
/**********************************************/
/* STAGE 1                                    */
/* Filename: dsss_txdel.ps.c                  */
/* Description:                               */
/*   Transmission delay model for wireless    */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                   */
/* History:                                   */
/*   - Original Opnet code: "dra_txdel.ps.c"  */
/*   - Modified for Spread Spectrum: 2Oct00   */
/**********************************************/

#include "opnet.h"

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_txdel (Packet * pkptr)
	{
	int		pklen;
	double		tx_drate, tx_delay;

	/** Compute the transmission delay associated with the transmission of a
	    packet over a radio link. **/

	FIN (dsss_txdel (pkptr));

	/* Obtain the transmission rate of that channel.	*/
	tx_drate = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_DRATE);

	/* Obtain length of packet.	*/
	pklen = op_pk_total_size_get (pkptr);

	/* Compute time required to complete transmission of packet.	*/
	tx_delay = pklen / tx_drate;

	/* Place transmission delay result in packet's reserved transmission data
	   attribute.	*/
	op_td_set_dbl (pkptr, OPC_TDA_RA_TX_DELAY, tx_delay);

	FOUT;
	}
```

```
/***********************************************/
/* STAGE 2                                     */
/* Filename: dsss_closure_all.ps.c             */
/* Description:                                */
/*   Closure model for wireless spread spectrum */
/*   link Transceiver Pipeline. This model assumes */
/*   that all receivers are capable of receiving */
/*   a transmission from any transmitter       */
/* Author: Robert J. Bonner                    */
/* History:                                    */
/*   - Original Opnet code: "dra_closure_all.ps.c" */
/*   - Modified for Spread Spectrum: 2Oct00    */
/***********************************************/

#include "opnet.h"

/***** pipeline procedure *****/

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_closure_all (Packet * pkptr)
	{
	/** Guarantee closure between transmitter and receiver. **/

	FIN (dsss_closure_all (pkptr));

	/* Place closure status in packet transmission data block. */
	op_td_set_int (pkptr, OPC_TDA_RA_CLOSURE, OPC_TRUE);

	FOUT;
	}
```

```
/**********************************************/
/* STAGE 3                                          */
/* Filename: dsss_chanmatch.ps.c                    */
/* Description:                                     */
/*   Channel match model for wireless               */
/*   spread spectrum link Transceiver Pipeline      */
/* Author: Robert J. Bonner                         */
/* History:                                         */
/*   - Original Opnet code: "dra_chanmatch.ps.c"    */
/*   - Modified for Spread Spectrum: 2Oct00         */
/*   - Updated: 3Oct00                              */
/*   - - Added in "code_family" check for match     */
/*   - Updated: 4Oct00                              */
/*       - - Revised to support if "code_family"    */
/*               attribute doesn't exist.           */
/**********************************************/
#include "opnet.h"
#if defined (__cplusplus)
extern "C"
#endif
void
dsss_chanmatch (Packet * pkptr)
	{
	int		tx_code_family, tx_id, my_id, rx_code_family, rx_id;
	double		tx_freq, tx_bw, tx_drate, tx_code;
	double		rx_freq, rx_bw, rx_drate, rx_code;
	Vartype		tx_mod, rx_mod;
	/*Determine the compatibility between transmitter and receiver channels.
	FIN (dsss_chanmatch (pkptr));
	/* Obtain transmitting channel attributes. */
	tx_freq		= op_td_get_dbl (pkptr, OPC_TDA_RA_TX_FREQ);
	tx_bw		= op_td_get_dbl (pkptr, OPC_TDA_RA_TX_BW);
	tx_drate	= op_td_get_dbl (pkptr, OPC_TDA_RA_TX_DRATE);
	tx_code		= op_td_get_dbl (pkptr, OPC_TDA_RA_TX_CODE);
	tx_mod		= op_td_get_ptr (pkptr, OPC_TDA_RA_TX_MOD);
	my_id		= op_td_get_int (pkptr, OPC_TDA_RA_TX_OBJID);
	tx_id		= op_topo_parent (my_id);
	/* Obtain receiving channel attributes. */
	rx_freq		= op_td_get_dbl (pkptr, OPC_TDA_RA_RX_FREQ);
	rx_bw		= op_td_get_dbl (pkptr, OPC_TDA_RA_RX_BW);
	rx_drate	= op_td_get_dbl (pkptr, OPC_TDA_RA_RX_DRATE);
	rx_code		= op_td_get_dbl (pkptr, OPC_TDA_RA_RX_CODE);
	rx_mod		= op_td_get_ptr (pkptr, OPC_TDA_RA_RX_MOD);
	my_id		= op_td_get_int (pkptr, OPC_TDA_RA_RX_OBJID);
	rx_id		= op_topo_parent (my_id);
	/* If code_family is specified, then get value, otherwise transmission is
```

```
   not spread.*/
if (op_ima_obj_attr_exists(tx_id, "code_family"))
        {op_ima_obj_attr_get (tx_id, "code_family", &tx_code_family);}
if (op_ima_obj_attr_exists(rx_id, "code_family"))
        {op_ima_obj_attr_get (rx_id, "code_family", &rx_code_family);}
/* For non-overlapping bands, the packet has no       */
/* effect; such packets are ignored entirely.         */
if ((tx_freq > rx_freq + rx_bw) || (tx_freq + tx_bw < rx_freq))
        {op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS,
                        OPC_TDA_RA_MATCH_IGNORE);
         FOUT;}
/* Otherwise check for channel attribute mismatches which would  */
/* cause the in-band packet to be considered as noise.           */
/* If the code_family attribute is specified, then the signal is spread
/* and check to see if transmitting in the same code family */
if ((op_ima_obj_attr_exists(rx_id, "code_family")) &&
   (op_ima_obj_attr_exists(tx_id, "code_family")))
        {if ((tx_freq != rx_freq) || (tx_bw != rx_bw) ||
                (tx_drate != rx_drate) || (tx_code != rx_code) ||
                (tx_code_family != rx_code_family) || (tx_mod != rx_mod))
                {op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS,
                                OPC_TDA_RA_MATCH_NOISE);
        FOUT; };
        }
/* If the code_family is not specified, then don't perform the code_family
   attribute check */
if ((tx_freq != rx_freq) || (tx_bw != rx_bw) ||
   (tx_drate != rx_drate) || (tx_code != rx_code) || (tx_mod != rx_mod))
        {
        op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS,
                        OPC_TDA_RA_MATCH_NOISE);
        FOUT;
        }
/* Otherwise the packet is considered a valid transmission which    */
/* could eventually be accepted at the error correction stage.      */
op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS,
               OPC_TDA_RA_MATCH_VALID);
FOUT;
}
```

```
/**********************************************/
/* STAGE 4                                     */
/* Filename: dsss_tagain.ps.c                  */
/* Description:                                */
/*   Transmitter antenna gain model for wireless */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                    */
/* History:                                    */
/*   - Original Opnet code: "dra_tagain.ps.c"  */
/*   - Modified for Spread Spectrum: 2Oct00    */
/**********************************************/

#include "opnet.h"
#include <math.h>

/***** constants *****/

#define RAD_TO_DEG      (180.0 / 3.1415927)
#define DEG_TO_RAD      (1.0 / RAD_TO_DEG)

/***** pipeline procedure *****/

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_tagain (Packet * pkptr)
	{
	double		tx_x, tx_y, tx_z;
	double		rx_x, rx_y, rx_z;
	double		dif_x, dif_y, dif_z, dist_xy;
	double		rot1_x, rot1_y, rot1_z;
	double		rot2_x, rot2_y, rot2_z;
	double		rot3_x, rot3_y, rot3_z;
	double		cos_pt_th, sin_pt_th;
	double		cos_sw_th, sin_sw_th, cos_sw_ph, sin_sw_ph;
	double		rx_phi, rx_theta, point_phi, point_theta;
	double		bore_phi, bore_theta, lookup_phi, lookup_theta, gain;
	Vartype		pattern_table;
	double		sweep_phi, sweep_theta;

	/** Compute the gain associated with the transmitter's antenna. **/

	FIN (dsss_tagain (pkptr));

	/* Obtain handle on receiving antenna's gain. */
```

```
pattern_table = op_td_get_ptr (pkptr, OPC_TDA_RA_TX_PATTERN);

/* Special case: by convention a nil table address indicates an isotropic
/* antenna pattern. Thus no calculations are necessary.
/* For a wired spread spectrum implementation in Opnet, be sure to set the
/* antenna pattern attribute to "isotropic" to nullify the calculations
   for receiver  antenna gain.                                      */
if (pattern_table == OPC_NIL)
        {
        /* Assign zero dB gain regardless of transmission direction. */
        op_td_set_dbl (pkptr, OPC_TDA_RA_TX_GAIN, 0.0);
        FOUT;
        }

/* Obtain the geocentric coordinates of the transmitter. */
tx_x = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GEO_X);
tx_y = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GEO_Y);
tx_z = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GEO_Z);

/* Obtain the geocentric coordinates of the receiver. */
rx_x = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_X);
rx_y = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_Y);
rx_z = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_Z);

/* Compute the vector from the transmitter to the receiver. */
dif_x = rx_x - tx_x;
dif_y = rx_y - tx_y;
dif_z = rx_z - tx_z;

/* Determine phi, theta pointing directions for antenna.            */
/* These are computed based on the target point of the antenna      */
/* module and the position of the transmitter.                          */
point_phi = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_PHI_POINT);
point_theta = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_THETA_POINT);

/* Determine antenna pointing reference direction                   */
/* (usually boresight cell of pattern).                             */
/* Note that the difference in selected coordinate systems          */
/* between the antenna definiton and the geocentric axes,           */
/* is accomodated for here by modifying the given phi value.        */
bore_phi = 90.0 - op_td_get_dbl (pkptr, OPC_TDA_RA_TX_BORESIGHT_PHI);
bore_theta = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_BORESIGHT_THETA);

/* Setup a new coord. system where x axis is in same theta plane    */
/* as pointing direction. This allows simple computation of effect*/
/* of phi rotation on the transmission vector.                      */
```

```
cos_pt_th = cos (DEG_TO_RAD * point_theta);
sin_pt_th = sin (DEG_TO_RAD * point_theta);
rot1_x = dif_x * cos_pt_th - dif_y * sin_pt_th;
rot1_y = dif_x * sin_pt_th + dif_y * cos_pt_th;
rot1_z = dif_z;

/* Rotate the boresight direction into the pointing direction  */
/* and compute the effect of this on the transmission vector.*/
sweep_phi = bore_phi - point_phi;
sweep_theta = bore_theta - point_theta;
cos_sw_th = cos (DEG_TO_RAD * sweep_theta);
cos_sw_ph = cos (DEG_TO_RAD * sweep_phi);
sin_sw_th = sin (DEG_TO_RAD * sweep_theta);
sin_sw_ph = sin (DEG_TO_RAD * sweep_phi);
rot2_x = (rot1_x * cos_sw_ph - rot1_z * sin_sw_ph) * cos_sw_th + rot1_y *
            sin_sw_th;
rot2_y = rot1_y * cos_sw_th - (rot1_x * cos_sw_ph - rot1_z * sin_sw_ph) *
            sin_sw_th;
rot2_z = rot1_x * sin_sw_ph + rot1_z * cos_sw_ph;

/* Reverse the initial coordinate system transform */
/* which was done to permit proper phi rotation. */
rot3_x = rot2_x * cos_pt_th + rot2_y * sin_pt_th;
rot3_y = rot2_y * cos_pt_th - rot2_x * sin_pt_th;
rot3_z = rot2_z;

/* Determine x-y projected distance. */
dist_xy = sqrt (rot3_x * rot3_x + rot3_y * rot3_y);

/* For the vector to the receiver, determine phi-deflection from        */
/* the x-y plane (in degress) and determine theta deflection from      */
/* the positive x axis.
        */
if (dist_xy == 0.0)
        {
        if (rot3_z < 0.0)
                rx_phi = -90.0;
        else
                rx_phi = 90.0;
        rx_theta = 0.0;
        }
else
        {
        rx_phi = RAD_TO_DEG * atan (rot3_z / dist_xy);

        if (rot3_y > 0.0)
```

```
		rx_theta = -RAD_TO_DEG * acos (rot3_x / dist_xy);
	else
		rx_theta = RAD_TO_DEG * acos (rot3_x / dist_xy);
	}


/* Setup the angles at which to lookup gain.                    */
/* In the rotated coordinate system, these are really           */
/* just the angles of the transmission vector. However,         */
/* note that here again the difference in the coordinate        */
/* systems of the antenna and the geocentric axes is            */
/* accomodated for by modiftying the phi angle.                 */
lookup_phi = 90.0 - rx_phi;
lookup_theta = rx_theta;

/* Obtain gain of antenna pattern at given angles. */
gain = op_tbl_pat_gain (pattern_table, lookup_phi, lookup_theta);
/* Set gain=0 due to wired DSSS implementation */
gain = 0.0;
/* Set the tx antenna gain in the packet's transmission data attribute. */
op_td_set_dbl (pkptr, OPC_TDA_RA_TX_GAIN, gain);

FOUT;
}
```

```
/**********************************************/
/* STAGE 5                                     */
/* Filename: dsss_propdel.ps.c                 */
/* Description:                                */
/*   Propagation delay model for wireless      */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                    */
/* History:                                    */
/*   - Original Opnet code: "dra_propdel.ps.c" */
/*   - Modified for Spread Spectrum: 2Oct00    */
/**********************************************/
#include "opnet.h"
/***** constants *****/
/* propagation velocity of signal in wire is 2/3 speed of light(m/s) */
#define     PROP_VELOCITY   2.0E+08
/***** pipeline procedure *****/
#if defined (__cplusplus)
extern "C"
#endif
void
dsss_propdel (Packet * pkptr)
	{
	double		start_prop_delay, end_prop_delay;
	double		start_prop_distance, end_prop_distance;

	/** Compute the propagation delay separating the   **/
	/** radio transmitter from the radio receiver.      **/
	FIN (dsss_propdel (pkptr));

	/** If the transmitter is mobile, then there will be a start distance and
	    an end distance. If the transmitter is not moving then start and end
	    distance will be the same. Get the start distance between transmitter
	    and receiver. **/

	start_prop_distance = op_td_get_dbl (pkptr, OPC_TDA_RA_START_DIST);
	/* Get the end distance between transmitter and receiver. */
	end_prop_distance = op_td_get_dbl (pkptr, OPC_TDA_RA_END_DIST);
	/* Compute propagation delay to start of reception. */
	start_prop_delay = start_prop_distance / PROP_VELOCITY;
	/* Compute propagation delay to end of reception. */
	end_prop_delay = end_prop_distance / PROP_VELOCITY;
	/* Place both propagation delays in packet transmission data attributes.
	op_td_set_dbl (pkptr, OPC_TDA_RA_START_PROPDEL, start_prop_delay);
	op_td_set_dbl (pkptr, OPC_TDA_RA_END_PROPDEL, end_prop_delay);
	FOUT;
	}
```

```
/***********************************************/
/* STAGE 6                                      */
/* Filename: dsss_ragain.ps.c                   */
/* Description:                                 */
/*   Receiver gain model for wireless           */
/*   spread spectrum link Transceiver Pipeline  */
/* Author: Robert J. Bonner                     */
/* History:                                     */
/*   - Original Opnet code: "dra_ragain.ps.c"   */
/*   - Modified for Spread Spectrum: 2Oct00     */
/***********************************************/

#include "opnet.h"
#include <math.h>

/***** constants *****/

#define RAD_TO_DEG      57.29578
#define DEG_TO_RAD      (1.0 / 57.29578)

/***** pipeline procedure *****/

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_ragain (Packet * pkptr)
        {
        double          tx_x, tx_y, tx_z;
        double          rx_x, rx_y, rx_z;
        double          dif_x, dif_y, dif_z, dist_xy;
        double          rot1_x, rot1_y, rot1_z;
        double          rot2_x, rot2_y, rot2_z;
        double          rot3_x, rot3_y, rot3_z;
        double          cos_pt_th, sin_pt_th;
        double          cos_sw_th, sin_sw_th, cos_sw_ph, sin_sw_ph;
        double          tx_phi, tx_theta, point_phi, point_theta;
        double          bore_phi, bore_theta, lookup_phi, lookup_theta, gain;
        Vartype         pattern_table;
        double          sweep_phi, sweep_theta;

        /** Compute the gain associated with the receiver's antenna. **/

        FIN (dsss_ragain (pkptr));

        /* Obtain handle on receiving antenna's gain. */
```

```
pattern_table = op_td_get_ptr (pkptr, OPC_TDA_RA_RX_PATTERN);

/* Special case: by convention a nil table address indicates an isotropic
/* antenna pattern. Thus no calculations are necessary.
/* For a wired spread spectrum implementation in Opnet, be sure to set the
/* antenna pattern attribute to "isotropic" to nullify the calculations
   for receiver antenna gain.                                              */
if (pattern_table == OPC_NIL)
        {
        /* Assign zero dB gain regardless of transmission direction. */
        op_td_set_dbl (pkptr, OPC_TDA_RA_RX_GAIN, 0.0);
        FOUT;
        }

/* Obtain the geocentric coordinates of the transmitter. */
tx_x = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GEO_X);
tx_y = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GEO_Y);
tx_z = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GEO_Z);

/* Obtain the geocentric coordinates of the receiver. */
rx_x = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_X);
rx_y = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_Y);
rx_z = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_Z);

/* Compute the vector from the receiver to the transmitter. */
dif_x = tx_x - rx_x;
dif_y = tx_y - rx_y;
dif_z = tx_z - rx_z;

/* Determine phi, theta pointing directions for antenna.          */
/* These are computed based on the target point of the antenna    */
/* module and the position of the receiver.                       */
point_phi = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_PHI_POINT);
point_theta = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_THETA_POINT);

/* Determine antenna pointing reference direction                 */
/* (usually boresight cell of pattern).                           */
/* Note that the difference in selected coordinate systems        */
/* between the antenna definiton and the geocentric axes          */
/* is accomodated for here by modifying the given phi value.      */
bore_phi = 90.0 - op_td_get_dbl (pkptr, OPC_TDA_RA_RX_BORESIGHT_PHI);
bore_theta = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_BORESIGHT_THETA);

/* Setup a new coord system where x axis is in same theta plane   */
/* as pointing direction. This allows simple computation of       */
/* effect of phi rotation on the transmission vector.             */
```

```
cos_pt_th = cos (DEG_TO_RAD * point_theta);
sin_pt_th = sin (DEG_TO_RAD * point_theta);
rot1_x = dif_x * cos_pt_th - dif_y * sin_pt_th;
rot1_y = dif_x * sin_pt_th + dif_y * cos_pt_th;
rot1_z = dif_z;

/* Rotate the boresight direction into the pointing direction  */
/* and compute the effect of this on the transmission vector.*/
sweep_phi = bore_phi - point_phi;
sweep_theta = bore_theta - point_theta;
cos_sw_th = cos (DEG_TO_RAD * sweep_theta);
cos_sw_ph = cos (DEG_TO_RAD * sweep_phi);
sin_sw_th = sin (DEG_TO_RAD * sweep_theta);
sin_sw_ph = sin (DEG_TO_RAD * sweep_phi);
rot2_x = (rot1_x * cos_sw_ph - rot1_z * sin_sw_ph) * cos_sw_th + rot1_y *
          sin_sw_th;
rot2_y = rot1_y * cos_sw_th - (rot1_x * cos_sw_ph - rot1_z * sin_sw_ph) *
          sin_sw_th;
rot2_z = rot1_x * sin_sw_ph + rot1_z * cos_sw_ph;

/* Reverse the initial coordinate system transform    */
/* which was done to permit proper phi rotation.      */
rot3_x = rot2_x * cos_pt_th + rot2_y * sin_pt_th;
rot3_y = rot2_y * cos_pt_th - rot2_x * sin_pt_th;
rot3_z = rot2_z;

/* Determine x-y projected distance. */
dist_xy = sqrt (rot3_x * rot3_x + rot3_y * rot3_y);

/* For the vector to the transmitter, determine phi-deflection        */
/* from the x-y plane (in degress) and determine theta-               */
/* deflection from the positive x axis.                               */
if (dist_xy == 0.0)
        {
        if (rot3_z < 0.0)
                tx_phi = -90.0;
        else
                tx_phi = 90.0;
        tx_theta = 0.0;
        }
else
        {
        tx_phi = RAD_TO_DEG * atan (rot3_z / dist_xy);

        if (rot3_y > 0.0)
                tx_theta = -RAD_TO_DEG * acos (rot3_x / dist_xy);
```

```
        else
                tx_theta = RAD_TO_DEG * acos (rot3_x / dist_xy);
        }

/* Setup the angles at which to lookup gain.                    */
/* In the rotated coordinate system, these are really           */
/* just the angles of the transmission vector. However,         */
/* note that here again the difference in the coordinate        */
/* systems of the antenna and the geocentric axes is            */
/* accomodated for by modifying the phi angle.                          */
lookup_phi = 90.0 - tx_phi;
lookup_theta = tx_theta;

/* Obtain gain of antenna pattern at given angles. */
gain = op_tbl_pat_gain (pattern_table, lookup_phi, lookup_theta);
/* Set Gain=0 for wired DSSS implementation */

gain = 0.0;
/* Set the rx antenna gain in the packet's transmission data attribute. */
op_td_set_dbl (pkptr, OPC_TDA_RA_RX_GAIN, gain);

FOUT;
}
```

```
/**********************************************/
/* STAGE 7                                     */
/* Filename: dsss_power.ps.c                   */
/* Description:                                */
/*   Received Power model for wireless spread  */
/*   spectrum link Transceiver Pipeline.       */
/* Author: Robert J. Bonner                    */
/* History:                                    */
/*   - Original Opnet code: "dra_power.ps.c"   */
/*   - Modified for Spread Spectrum: 2Oct00    */
/**********************************************/

#include "opnet.h"
#include <math.h>

/***** constants *****/

#define C                   3.0E+08       /* speed of light (m/s) */
#define SIXTEEN_PI_SQ       157.91367     /* 16 times pi-squared */

static const char*    PowI_Err_Hdr = "Error in radio power computation pipeline stage
(dsss_power):";

/***** pipeline procedure *****/

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_power (Packet * pkptr)
	{
	double       prop_distance, rcvd_power, path_loss;
	double       tx_power, tx_base_freq, tx_bandwidth, tx_center_freq;
	double       lambda, rx_ant_gain, tx_ant_gain;
	Boolean      sig_lock;
	Objid        rx_ch_obid;
	double       in_band_tx_power, band_max, band_min;
	double       rx_base_freq, rx_bandwidth;

	/** Compute the average power in Watts of the     **/
	/** signal associated with a transmitted packet.  **/

	FIN (dsss_power (pkptr));

	/* If the incoming packet is 'valid', it may cause the receiver to       */
	/* lock onto it. However, if the receiving node is disabled, then        */
```

```
 /* the channel match should be set to noise.                    */
 if (op_td_get_int (pkptr, OPC_TDA_RA_MATCH_STATUS) ==
OPC_TDA_RA_MATCH_VALID)
        {
        if (op_td_is_set (pkptr, OPC_TDA_RA_ND_FAIL))
                {
                /* The receiving node is disabled.  Change   */
                /* the channel match status to noise.        */
                op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS,
                                OPC_TDA_RA_MATCH_NOISE);
                }
        else
                {
                /* The receiving node is enabled.  Get       */
                /* the address of the receiver channel.      */
                rx_ch_obid = op_td_get_int (pkptr, OPC_TDA_RA_RX_CH_OBJID);

                /* If the receiver channel is already locked,           */
                /* the packet will now be considered to be noise.       */
                /* This prevents simultaneous reception of multiple */
                /* valid packets on any given radio channel.            */
                if (op_ima_obj_attr_get (rx_ch_obid, "signal lock", &sig_lock)
                        == OPC_COMPCODE_FAILURE)
                        {
                        op_sim_end (PowI_Err_Hdr,
                                "Unable to get signal lock flag from channel
                                 attribute.",
                                OPC_NIL, OPC_NIL);
                        }
                if (sig_lock)
                        op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS,
                  OPC_TDA_RA_MATCH_NOISE);
                else
                        {
                        /* Otherwise, the receiver channel will become         */
                        /* locked until the packet reception ends.             */
                        sig_lock = OPC_BOOLINT_ENABLED;
                        if (op_ima_obj_attr_set (rx_ch_obid, "signal lock",
                            sig_lock) == OPC_COMPCODE_FAILURE)
                                {
                                op_sim_end (PowI_Err_Hdr,
                                        "Unable to set receiver channel attribute
                                         (signal lock).",
                                        OPC_NIL, OPC_NIL);
                                }
                        }
```

```
            }
        }

    /* Get power allotted to transmitter channel. */
    tx_power = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_POWER);

    /* Get transmission frequency in Hz. */
    tx_base_freq = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_FREQ);
    tx_bandwidth = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_BW);
    tx_center_freq = tx_base_freq + (tx_bandwidth / 2.0);

    /* Calculate wavelength (in meters). */
    lambda = C / tx_center_freq;

    /* Get distance between transmitter and receiver (in meters). */
    prop_distance = op_td_get_dbl (pkptr, OPC_TDA_RA_START_DIST);

    /* When using TMM, the TDA OPC_TDA_RA_RCVD_POWER will already      */
    /* have a raw value for the path loss. */
    if (op_td_is_set (pkptr, OPC_TDA_RA_RCVD_POWER))
        {
        path_loss = op_td_get_dbl (pkptr, OPC_TDA_RA_RCVD_POWER);
        }
    else
        {
        /* Compute the path loss for this distance and wavelength. */
        if (prop_distance > 0.0)
            {
            path_loss = (lambda * lambda) /
                (SIXTEEN_PI_SQ * prop_distance * prop_distance);
            }
        else
            path_loss = 1.0;
        }

    /* Determine the receiver bandwidth and base frequency. */
    rx_base_freq = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_FREQ);
    rx_bandwidth = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_BW);

    /* Use these values to determine the band overlap with the transmitter.    */
    /* Note that if there were no overlap at all, the packet would already     */
    /* have been filtered by the channel match stage.                               */

    /* The base of the overlap band is the highest base frequency. */
    if (rx_base_freq > tx_base_freq)
        band_min = rx_base_freq;
```

```
else
        band_min = tx_base_freq;

/* The top of the overlap band is the lowest end frequency. */
if (rx_base_freq + rx_bandwidth > tx_base_freq + tx_bandwidth)
        band_max = tx_base_freq + tx_bandwidth;
else
        band_max = rx_base_freq + rx_bandwidth;

/* Compute the amount of in-band transmitter power. */
in_band_tx_power = tx_power * (band_max - band_min) / tx_bandwidth;

/* Get antenna gains (raw form, not in dB). */
tx_ant_gain = pow (10.0, op_td_get_dbl (pkptr, OPC_TDA_RA_TX_GAIN) /
         10.0);
rx_ant_gain = pow (10.0, op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GAIN) /
         10.0);

/* Calculate received power level. In order to simulate perfect power
   control,let the received power equal the transmitted power
   where transmitted power is defined to be the effective received power
   to the input to the receiver. This also neglects power loss due to
   attenuation in the coaxial cable */
rcvd_power = tx_power;

/* Assign the received power level (in Watts) */
/* to the packet transmission data attribute. */

op_td_set_dbl (pkptr, OPC_TDA_RA_RCVD_POWER, rcvd_power);
FOUT;
}
```

```
/**********************************************/
/* STAGE 8                                    */
/* Filename: dsss_bkgnoise.ps.c               */
/* Description:                               */
/*   Background noise model for wireless      */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                   */
/* History:                                   */
/*   - Original Opnet code: "dra_bkgnoise.ps.c" */
/*   - Modified for Spread Spectrum: 2Oct00   */
/**********************************************/
#include "opnet.h"

/***** constants *****/
#define BOLTZMANN       1.379E-23
#define BKG_TEMP        290.0
// Background noise is near zero for a wire, so ambient noise level  must be VERY small to
  facilitate OPNET's calculations.
#define AMB_NOISE_LEVEL     1.0E-26

/** Procedure **/
#if defined (__cplusplus)
extern "C"
#endif
void
dsss_bkgnoise (Packet * pkptr)
        {
        double      rx_noisefig, rx_temp, rx_bw;
        double      bkg_temp, bkg_noise, amb_noise;

        /*Compute noise sources other than transmission interference.*/
        FIN (dsss_bkgnoise (pkptr));

        /* Get receiver noise figure. */
        rx_noisefig = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_NOISEFIG);

        /* Calculate effective receiver temperature. */
        rx_temp = (rx_noisefig - 1.0) * 290.0;

        /* Set the effective background temperature. */
        bkg_temp = BKG_TEMP;

        /* Get receiver channel bandwidth (in Hz). */
        rx_bw = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_BW);

        /* Calculate in-band noise from both background and thermal
```

```
   sources. */
bkg_noise = (rx_temp + bkg_temp) * rx_bw * BOLTZMANN;

/* Calculate in-band ambient noise. */
amb_noise = rx_bw * AMB_NOISE_LEVEL;

/* Put the sum of both noise sources in the packet transmission
   data attribute. */
op_td_set_dbl (pkptr, OPC_TDA_RA_BKGNOISE, (amb_noise +
      bkg_noise));
FOUT; }
```

```
/****************************************************/
/* STAGE 9                                          */
/* Filename: dsss_mai_noise.ps.c                    */
/* Description:                                     */
/*   Multiple access interference noise model       */
/*   for wireless spread spectrum link              */
/*   Transceiver Pipeline.                          */
/* Author: Robert J. Bonner                         */
/* History:                                         */
/*      - Original Opnet code: "dra_inoise.ps.c"    */
/*      - Modified for Spread Spectrum: 2Oct00      */
/*      - Updated: 9Oct00                           */
/*      - - Modified to accommodate family code     */
/*      - Updated: 12Oct00                          */
/*      - - Revised to make seperate calculations for */
/*      - - MAI and regular non-spread interferers. */
/****************************************************/

#include "opnet.h"
#include "math.h"

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_mai_noise (Packet * pkptr_prev, Packet * pkptr_arriv)
	{
	int		arriv_match, prev_match, chip;
	int		my_id, tx_id_a, rx_id_a;
	int		tx_id_p, rx_id_p;
	int		tx_code_family, rx_code_family;
	double		prev_rcvd_power, arriv_rcvd_power, correlation, noise;
	double		arriv_noise_accum, prev_noise_accum, proc_gain,
			spread_gain, gain;

	/** Evaluate a collision due to arrival of 'pkptr_arriv'	**/
	/** where 'pkptr_prev' is the packet that is currently	**/
	/** being received.					**/

	FIN (dsss_mai_noise (pkptr_prev, pkptr_arriv));

	/* If the previous packet ends just as the new one begins, this is not	*/
	/* a collision (just a near miss, or perhaps back-to-back packets).	*/

	if (op_td_get_dbl (pkptr_prev, OPC_TDA_RA_END_RX) != op_sim_time ())
		{
```

```
/* Increment the number of collisions in previous packet. */
op_td_set_int (pkptr_prev, OPC_TDA_RA_NUM_COLLS, op_td_get_int
               (pkptr_prev, OPC_TDA_RA_NUM_COLLS) + 1);

/* Increment number of collisions in arriving packet. */
op_td_set_int (pkptr_arriv, OPC_TDA_RA_NUM_COLLS, op_td_get_int
               (pkptr_arriv, OPC_TDA_RA_NUM_COLLS) + 1);

/* Determine if previous packet is valid or noise. */
prev_match = op_td_get_int (pkptr_prev, OPC_TDA_RA_MATCH_STATUS);

/* Determine if arriving packet is valid or noise. */
arriv_match = op_td_get_int (pkptr_arriv, OPC_TDA_RA_MATCH_STATUS);

/* Get received power levels for both packets. */
prev_rcvd_power = op_td_get_dbl (pkptr_prev, OPC_TDA_RA_RCVD_POWER);
arriv_rcvd_power = op_td_get_dbl (pkptr_arriv,
                                  OPC_TDA_RA_RCVD_POWER);

/* Get Object IDs of TX/RX pairs for previous and arriving packet.
my_id      = op_td_get_int (pkptr_prev, OPC_TDA_RA_TX_OBJID);
tx_id_p    = op_topo_parent (my_id);
my_id      = op_td_get_int (pkptr_prev, OPC_TDA_RA_RX_OBJID);
rx_id_p    = op_topo_parent (my_id);

my_id      = op_td_get_int (pkptr_arriv, OPC_TDA_RA_TX_OBJID);
tx_id_a    = op_topo_parent (my_id);
my_id      = op_td_get_int (pkptr_arriv, OPC_TDA_RA_RX_OBJID);
rx_id_a    = op_topo_parent (my_id);

/* If the arriving packet is valid, then calculate interference of
   previous packet on arriving one.    */
if (arriv_match == OPC_TDA_RA_MATCH_VALID)
    {
    /* If the spread spectrum atrtributes exists, then calculate
       spread noise effects. */
     if ((op_ima_obj_attr_exists(rx_id_a, "code_family")) &&
        (op_ima_obj_attr_exists(tx_id_p, "code_family")) &&
        (op_ima_obj_attr_exists(rx_id_a, "correlation_coeff")))
           {
           /* Get Spread Spectrum Attributes. */
           op_ima_obj_attr_get (tx_id_p, "code_family",
                                  &tx_code_family);
           op_ima_obj_attr_get (rx_id_a, "code_family",
                                  &rx_code_family);
           op_ima_obj_attr_get (rx_id_a, "correlation_coeff",
```

```
                                &correlation);
            op_ima_obj_attr_get (rx_id_p, "spreading_gain",
                                &spread_gain);
            proc_gain = op_td_get_dbl (pkptr_arriv,
                                     OPC_TDA_RA_PROC_GAIN);
            /* If the code families match, then interference is due
               to multiple access. */
            if (tx_code_family == rx_code_family)
                {
                arriv_noise_accum = op_td_get_dbl (pkptr_arriv,
                                     OPC_TDA_RA_NOISE_ACCUM);
                op_td_set_dbl (pkptr_arriv,
                               OPC_TDA_RA_NOISE_ACCUM,
                               arriv_noise_accum +
                               (prev_rcvd_power * correlation *
                               proc_gain));
                                                    }
            /* Else, the interference is considered a result of
               jamming (Narrowband or Wideband). */
            else
                {
                arriv_noise_accum = op_td_get_dbl (pkptr_arriv,
                                     OPC_TDA_RA_NOISE_ACCUM);
                op_td_set_dbl (pkptr_arriv,
                               OPC_TDA_RA_NOISE_ACCUM,
                               arriv_noise_accum +
                               (prev_rcvd_power * spread_gain));
                }
            }
        /* Else, the transmission is occurring between non-spread
           stations, so use default radio calculations. */
        else
            {
            arriv_noise_accum = op_td_get_dbl (pkptr_arriv,
                                 OPC_TDA_RA_NOISE_ACCUM);
            op_td_set_dbl (pkptr_arriv, OPC_TDA_RA_NOISE_ACCUM,
                           arriv_noise_accum + prev_rcvd_power);
            }
        }

/* If the previous packet is valid, then calculate the interference
   of arriving packet on previous one. */
if (prev_match == OPC_TDA_RA_MATCH_VALID)
        {
        /* If the spread spectrum atrtributes exists, then calculate
           spread noise effects. */
```

```
if ((op_ima_obj_attr_exists(rx_id_p, "code_family")) &&
    (op_ima_obj_attr_exists(tx_id_a, "code_family")) &&
   (op_ima_obj_attr_exists(rx_id_p, "correlation_coeff")) &&
   (op_ima_obj_attr_exists(tx_id_p, "chip_rate")))
        {
        /* Get Spread Spectrum attributes. */
        op_ima_obj_attr_get (tx_id_a, "code_family",
                                    &tx_code_family);
        op_ima_obj_attr_get (rx_id_p, "code_family",
                                    &rx_code_family);
        op_ima_obj_attr_get (rx_id_p, "correlation_coeff",
                                    &correlation);
        op_ima_obj_attr_get (rx_id_p, "spreading_gain",
                                    &spread_gain);
        proc_gain = pow (10.0, op_td_get_dbl (pkptr_prev,
                                OPC_TDA_RA_PROC_GAIN) / 10.0);
        /* If the code families match, then interference is due
           to multiple access. */
        if (tx_code_family == rx_code_family)
                {
                prev_noise_accum = op_td_get_dbl (pkptr_prev,
                                        OPC_TDA_RA_NOISE_ACCUM);
                op_td_set_dbl (pkptr_prev,
                                    OPC_TDA_RA_NOISE_ACCUM,
                                    prev_noise_accum +
                                    (arriv_rcvd_power * correlation *
                                    proc_gain ));
                        }
        /* Else, the interference is considered a result of
          jamming (Narrowband or Wideband). */
        else
                {
                prev_noise_accum = op_td_get_dbl (pkptr_prev,
                                        OPC_TDA_RA_NOISE_ACCUM);
                op_td_set_dbl (pkptr_prev,
                                    OPC_TDA_RA_NOISE_ACCUM,
                                     prev_noise_accum +
                                    (arriv_rcvd_power * spread_gain));
                }
        }
/* Else, the transmission is occurring between non-spread
   stations, so use default radio calculations. */
 else
        {
        prev_noise_accum = op_td_get_dbl (pkptr_prev,
                                OPC_TDA_RA_NOISE_ACCUM);
```

```
                op_td_set_dbl (pkptr_prev, OPC_TDA_RA_NOISE_ACCUM,
                        prev_noise_accum + arriv_rcvd_power);

                }
            }
        }

    FOUT;
    }
```

```
/*****************************************************/
/* STAGE 10                                          */
/* Filename: dsss_sir.ps.c                           */
/* Description:                                      */
/*   Signal-to-Interference Ratio (SIR) model for wireless spread  */
/*   spectrum link  Transceiver Pipeline. SIR is the effective SNR  */
/*   at IF filter output. Probability of error is based on this SIR  */
/*   rather than SNR.                                */
/* Author: Robert J. Bonner                          */
/* History:                                          */
/*       - Original Opnet code: "dra_snr.ps.c"       */
/*       - Modified for Spread Spectrum: 2Oct00      */
/*              - Modified: 12Oct00                  */
/*              - - Incorporated processing gain noise  */
/*              - - reduction calculation            */
/*****************************************************/
#include "opnet.h"
#include <math.h>
#if defined (__cplusplus)
extern "C"
#endif
void
dsss_sir (Packet * pkptr)
        {
        double bkg_noise, accum_noise, rcvd_power, proc_gain, eff_snr, snr, noise;
        /** Compute the signal-to-noise ratio for the given packet. **/
        FIN (dsss_sir (pkptr));
        /* Get the packet's received power level. */
        rcvd_power = op_td_get_dbl (pkptr, OPC_TDA_RA_RCVD_POWER);
        /* Get the packet's accumulated noise levels calculated by the
           interference and background noise stages. */
        accum_noise = op_td_get_dbl (pkptr, OPC_TDA_RA_NOISE_ACCUM);
        bkg_noise = op_td_get_dbl (pkptr, OPC_TDA_RA_BKGNOISE);
        noise = accum_noise+bkg_noise;
        /* Get the processing gain associated with the packet. Assigned as dB        */
        proc_gain = op_td_get_dbl (pkptr, OPC_TDA_RA_PROC_GAIN);
        /* Calculate SNR and convert to dB  */
        snr = 10.0 * log10 (rcvd_power / noise);
        /*      Calculate effective SNR incorporating processing gain. */
        eff_snr = snr + proc_gain;
        /* Assign the effective SNR in dB. */
        op_td_set_dbl (pkptr, OPC_TDA_RA_SNR, eff_snr);
        /* Set field indicating the time at which SNR was calculated. */
        op_td_set_dbl (pkptr, OPC_TDA_RA_SNR_CALC_TIME, op_sim_time ());
        FOUT;
        }
```

```
/*********************************************/
/* STAGE 11                                  */
/* Filename: dsss_ber.ps.c                   */
/* Description:                              */
/*   Bit-Error-Rate (BER) model for wireless */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                  */
/* History:                                  */
/*   - Original Opnet code: "dra_ber.ps.c"   */
/*   - Modified for Spread Spectrum: 2Oct00  */
/*        - - Revised Comments               */
/*        - Updated 21Oct00                  */
/*        - - Removed effective SNR calculation */
/*        - - and used it in Stage 10 (SNR)  */
/*********************************************/

#include "opnet.h"

#if defined (__cplusplus)
extern "C"
#endif

void
dsss_ber (Packet * pkptr)
	{
	double		ber, snr, proc_gain, eff_snr,test;
	Vartype			modulation_table;

	/** Calculate the average bit error rate affecting given packet. **/
	FIN (dsss_ber (pkptr));
	/* Determine current value of Signal-to-Noise-Ratio (SNR). */
	snr = op_td_get_dbl (pkptr, OPC_TDA_RA_SNR);
	/* Determine address of modulation table. */
	modulation_table = op_td_get_ptr (pkptr, OPC_TDA_RA_RX_MOD);
	/* Derive expected BER from effective SNR. */
	ber = op_tbl_mod_ber (modulation_table, snr);
	/* Place the BER in the packet's transmission data. */
	op_td_set_dbl (pkptr, OPC_TDA_RA_BER, ber);
	FOUT;
	}
```

```
/***********************************************/
/* STAGE 12                                    */
/* Filename: dsss_error.ps.c                   */
/* Description:                                */
/*   Error Allocation model for wireless       */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                    */
/* History:                                    */
/*   - Original Opnet code: "dra_error.ps.c"   */
/*   - Modified for Spread Spectrum: 2Oct00    */
/***********************************************/

#include "opnet.h"
#include <math.h>

/* Define a convenient macro for computing     */
/* factorials using the gamma function.        */
#define log_factorial(n)      lgamma ((double) n + 1.0)
extern double lgamma (double);

#if defined (__cplusplus)
extern "C"
#endif
void
dsss_error (Packet * pkptr)
	{
	double		pe, r, p_accum, p_exact;
	double		data_rate, elap_time;
	double		log_p1, log_p2, log_arrange;
	int		seg_size, num_errs, prev_num_errs;
	int		invert_errors = OPC_FALSE;

	/** Compute the number of errors assigned to a segment of bits within
	/** a packet based on its length and the bit error probability.       **/

	FIN (dsss_error (pkptr));

	/* Obtain the expected Bit-Error-Rate 'pe' */
	pe = op_td_get_dbl (pkptr, OPC_TDA_RA_BER);

	/* Calculate time elapsed since last BER change */
	elap_time = op_sim_time () - op_td_get_dbl (pkptr,
			OPC_TDA_RA_SNR_CALC_TIME);

	/* Use datarate to determine how many bits in the segment. */
	data_rate = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_DRATE);
```

```
seg_size = elap_time * data_rate;

/* Case 1: if the bit error rate is zero, so is the number of errors.      */
if (pe == 0.0 || seg_size == 0)
        num_errs = 0;

/* Case 2: if the bit error rate is 1.0, then all the bits are in error.
/* (note however, that bit error rates should not normally exceed 0.5).
else if (pe >= 1.0)
        num_errs = seg_size;

/* Case 3: The bit error rate is not zero or one. */
else
        {
        /* If the bit error rate is greater than 0.5 and less than 1.0,
           invert the problem to find instead the number of bits that are not in
           error in order to accelerate the performance of the algorithm.  Set a
           flag to indicate that the result will then have to be inverted. */
        if (pe > 0.5)
                {
                pe = 1.0 - pe;
                invert_errors = OPC_TRUE;
                }

        /* The error count can be obtained by mapping a uniform random
            number  in [0, 1[ via the inverse of the cumulative mass function (CMF)
            for the bit error count distribution.  Obtain a uniform random number in [0, 1[ to
            represent the value of the CDF at the outcome that will be produced.     */
        r = op_dist_uniform (1.0);

        /* Integrate probability mass over possible outcomes until r is
           exceeded.  The loop iteratively corresponds to "inverting" the CMF since it
           finds the bit error count at which the CMF first meets or exceeds the
           value r.      */
        for (p_accum = 0.0, num_errs = 0; num_errs <= seg_size; num_errs++)
                {
                /* Compute the probability of exactly 'num_errs' bit errors
                    occurring. The probability that the first 'num_errs' bits will be in
                    error is given by pow (pe, num_errs). Here it is obtained in
                    logarithmic form to avoid underflow for small 'pe' or large 'num_errs'. */
                log_p1 = (double) num_errs * log (pe);

                /* Similarly, obtain the probability that the remaining bits
                   will not be in error. The combination of these two events
                   represents one possible configuration of bits yielding a
                   total of 'num_errs' errors.*/
```

```
            log_p2 = (double) (seg_size - num_errs) * log (1.0 - pe);

            /* Compute the number of arrangements that are possible with
               the same number of bits in error as the particular case
               above. Again obtain this number in logarithmic form (to
               avoid overflow in this case). This result is expressed as
               the logarithmic form of the formula for the number N of
               combinations of k items from n:  N = n!/(n-k)!k!          */
            log_arrange =  log_factorial (seg_size) -
                                        log_factorial (num_errs) -
                                        log_factorial (seg_size - num_errs);

            /* Compure the probability that exactly 'num_errs' are present
               in the segment of bits, in any arrangement.                    */
            p_exact = exp (log_arrange + log_p1 + log_p2);

            /* Add this to the probability mass accumulated so far for
               previously tested outcomes to obtain the value of the CMF
               at outcome = num_errs.*/
            p_accum += p_exact;

            /*'num_errs' is the outcome for this trial if the CMF meets or
               exceeds the uniform random value selected earlier. */
            if (p_accum >= r)
                  {  break; }
            }

        /* If the bit error rate was inverted to compute correct bits
           instead, then Reinvert the result to obtain the number of bits in
           error.                          */
        if (invert_errors == OPC_TRUE)
                num_errs = seg_size - num_errs;
        }

    /* Increase number of bit errors in packet transmission data attribute. */
    prev_num_errs = op_td_get_int (pkptr, OPC_TDA_RA_NUM_ERRORS);
    op_td_set_int (pkptr, OPC_TDA_RA_NUM_ERRORS, num_errs + prev_num_errs);

    /* Assign actual (allocated) bit-error rate over tested segment. */
    if (seg_size != 0)
            op_td_set_dbl (pkptr, OPC_TDA_RA_ACTUAL_BER, (double) num_errs /
              seg_size);
    else op_td_set_dbl (pkptr, OPC_TDA_RA_ACTUAL_BER, pe);

    FOUT;
}
```

```
/**********************************************/
/* STAGE 13                                   */
/* Filename: dsss_ecc.ps.c                    */
/* Description:                               */
/*   Error Correction model for wireless      */
/*   spread spectrum link Transceiver Pipeline */
/* Author: Robert J. Bonner                   */
/* History:                                   */
/*   - Original Opnet code: "dra_ecc.ps.c"    */
/*   - Modified for Spread Spectrum: 2Oct00   */
/**********************************************/
#include "opnet.h"
#if defined (__cplusplus)
extern "C"
#endif
void
dsss_ecc (Packet * pkptr)
  {
	int		pklen, num_errs, accept;
	Objid		rx_ch_obid;
	double		ecc_thresh;

	/** Determine acceptability of given packet at receiver. **/
	FIN (dsss_ecc (pkptr));
	/* Do not accept packets that were received  */
	/* when the node was disabled.                       */
	if (op_td_is_set (pkptr, OPC_TDA_RA_ND_FAIL))
		accept = OPC_FALSE;
	else
		{
		/* Obtain the error correction threshold of the receiver. */
		ecc_thresh = op_td_get_dbl (pkptr, OPC_TDA_RA_ECC_THRESH);

		/* Obtain length of packet. */
		pklen = op_pk_total_size_get (pkptr);

		/* Obtain number of errors in packet. */
		num_errs = op_td_get_int (pkptr, OPC_TDA_RA_NUM_ERRORS);

		/* Test if bit errors exceed threshold. */
		if (pklen == 0)
			accept = OPC_TRUE;
		else
			accept = ((((double) num_errs) / pklen) <= ecc_thresh) ?
		   OPC_TRUE : OPC_FALSE;
		}
```

```
/* Place flag indicating accept/reject in transmission data block. */
op_td_set_int (pkptr, OPC_TDA_RA_PK_ACCEPT, accept);

/* In either case the receiver channel is no longer locked. */
rx_ch_obid = op_td_get_int (pkptr, OPC_TDA_RA_RX_CH_OBJID);
op_ima_obj_attr_set (rx_ch_obid, "signal lock", OPC_BOOLINT_DISABLED);

FOUT;
}
```

Bibliography

[Ano96] Anonymous. The technology of CDMA. *America's Network. Vol. 100. Number 17.*, September 1996.

[BM88] D. R. Boggs and J. Mogul. Measured capacities of an ethernet: Myths and reality. *Western Research Laboratory*, pages 1–31, September 1988.

[Chr96] K. Christensen. A performance evaluation of binary logarithmic access method (BLAM). In *Proceedings of the IEEE 21st Conference on Computer Networks*, pages 396–403, October 1996.

[Chr98] K. J. Christensen. A simulation study of enhanced arbitration methods for improving ethernet performance. *Computer Communications*, 21(1):24–36, February 1998.

[EA98] E. Erkip and B. Aazhang. Multiple access schemes over multipath fading channels. In *IEEE International Symposium on Information Theory*, page 216, MIT, Cambridge, MA, USA, August 1998.

[Gal94] R. Gallagher. An inequality on the capacity region of multi–access fading channels. In *Communications and Cryptography– Two Sides of One Tapestry*, pages 129–139. Kluwer Academic Publishers, 1994.

[Gon87] T. A. Gonsalves. Measured performance of the ethernet. In *Advances in Local Area Networks*, pages 383–410, 1987.

[Hay94] S. Haykin. *Communication Systems. 3rd ed.* John Wiley & Sons, Inc., New York, NY, 1994.

[HM95] W. Hayes and M. L. Molle. Solving capture switched two-node ethernets by changing only one node. In *Proceedings of the 20th Conference on Local Computer Networks*, pages 387–395, October 1995.

[Hol92] J. Holtzman. A simple, accurrate method to calculate spread-spectrum multiple-access error probabilities. *IEEE Transactions on Communications. Vol. 40. No. 3.*, March 1992.

[IEE85] IEEE. Carrier sense multiple access with collision detection access method and physical layer specifications. ANSI/IEEE Standard 802.3, 1985.

[IEE97] IEEE. Carrier sense multiple access with collision detection access method and physical layer specifications. ANSI/IEEE Standard 802.11: Wireless LAN Standard, 1997.

[Jai91] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling.* John Wiley and Sons, Inc., New York, NY, 1991.

[JJ99] D. G. Jeong and W. S. Jeon. Capacity of CDMA/TDD systems for mobile multimedia services. In *IEEE VTS 50th Vehicular Technology Conference. Vol. 5*, pages 2606–2610, September 1999.

[Kas88] K. T. Kasengulu. About the capacity of a DS spread cellular system. IEEE Select Areas on Communications, 1988.

[Kil98] P Killelea. *Web Performance Tuning: Speeding Up The Web.* O'Reilly and Associates, New York, NY, 1998.

[Kle92] L. Kleinrock. The latency/bandwidth tradeoff in gigabit networks. *IEEE Communications Magazine*, pages 36–40, April 1992.

[KT75] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part 1-carrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transactions on Communications. COM-23*, pages 1400–1416, December 1975.

[Lab99] Cable Television Laboratories. Data over cable service interface specification. Version 1.2, 1999.

[Let94] K. B. Letaief. Efficient simulation of direct-sequence spread spectrum multiple-access communications systems. *IEEE ICCS*, pages 374–378, 1994.

[LP87] J.S. Lehnert and M.B. Pursely. Error probabilities for binary direct-sequence spread spectrum communications with random signatures. *IEEE Transactions on Communications. COM-35*, pages 87–89, 1987.

[MB76] R. M. Metcalfe and D. R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM. Vol. 19*, pages 395–404, July 1976.

[Meh96] H. Mehling. Tools to help you beat the world wide wait. *Information Week.*, 49(597), September 1996.

[Mil86] R.G Miller. *Beyond ANOVA, Basics of Applied Statistics.* John Wiley & Sons, New York, NY, 1986.

[ML83] J. S. Meditch and C. A. Lea. Stability and optimization of the CSMA and CSMA/CD channels. *IEEE Transactions on Communications. COM-31*, pages 763–774, June 1983.

[O'R83] P. J. O'Reilly. A new technique for performance studies of CSMA/CD local networks. Master's thesis, Georgia Institute Of Technology, May 1983. PhD. Dissertation.

[PD96] L.L. Peterson and B.S. Davie. *Computer Networks: A System Approach.* Morgan Kauffman Publishers, Inc., San Francisco, CA, 1996.

[Pic86] R. L. Pickholtz. *Local Area and Multiple Access Networks.* Computer Science Press, Rockville, MD, 1986.

[PRAS99] J. Perez-Romero, R. Augusti, and O. Sallent. Performance analysis of an ISMA CDMA packet data network. In *IEEE VTS 50th Vehicular Technology Conference. Vol. 5*, pages 2865–2869, September 1999.

[PSW92] M.J. Parham, M.C. Smythe, and B.L. Weiss. Code division multiple access techniques for use in optical fibre local area networks. *Electronics and Communication Engineering Journal*, pages 203–212, August 1992.

[PZB95] R.L. Peterson, R.E. Ziemer, and D.E. Borth. *Introduction to Spread Spectrum Communications.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[Rap96] T.S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, Upper Saddle River, NJ, 1996.

[RH98] S. Ramakrishna and J. M. Holtzman. A comparison between single code and multiple code transmission schemes in a CDMA system. In *IEEE VTS 49th Vehicular Technology Conference. Vol. 2*, pages 791–795, 1998.

[RY94] K. Ramakrishnan and H. Yang. The ethernet capture effect: Analysis and solution. In *Proceedings of the 19th Conference on Local Computer Networks*, pages 228–240, October 1994.

[Sch82] R. A. Scholtz. The origins of spread spectrum communications. *IEEE Transactions on Communications. COM-30, Vol. 5.*, pages 852–884, May 1982.

[SH80] J. F. Shoch and J. A. Hupp. Measured performance of an ethernet local network. *Communications of the ACM. Vol. 23*, pages 711–720, December 1980.

[She82] M. M. Sheppard. The performance of a PN spread spectrum receiver preceded by an adaptive interference suppression filter. Master's thesis, Air Force Institute of Technology, December 1982. AFIT/GE/EE/82-62.

[SK91] W. R. Smith and R. Y. Kain. Ethernet performance under actual and simulated loads. In *Proceedings of the 16th Conference on Local Computer Networks*, pages 569–581, 1991.

[SK98] S. Sun and W. A. Krzymien. Call admission policies and capacity analysis of a multi-service CDMA personal communication system with continuous and discontinuous transmission. In *IEEE VTS 49th Vehicular Technology Conference. Vol. 1*, pages 218–223, 1998.

[Skl88] B. Sklar. *Digital Communications: Fundamentals and Applications.* Prentice Hall, Upper Saddle River, NJ, 1988.

[SSM99] H. Sari, H. Steendam, and M. Moneclaey. On the downlink capacity of cellular CDMA and TDMA over nondispersive channels. In *IEEE VTS 50th Vehicular Technology Conference. Vol. 2*, pages 1165–1169, September 1999.

[Str96] D. Strassberg. Powerline communication: Wireless technology. *EDN*, page 71, June 1996.

[Tan96] A. S. Tannenbaum. *Computer Networks. 3rd ed.* Prentice Hall, Upper Saddle River, NJ, 1996.

[Tec97] OPNET Technologies. OPNET modeling manual. Version 6.0, 1997.

[TH80] F. A. Tobagi and B. Hunt. Performance analysis of carrier sense multiple access with collision detection. *Computer Networks. Vol. 4*, pages 245–259, October/November 1980.

[TK85] H. Takagi and L. Kleinrock. Throughput analysis for persistent CSMA system. *IEEE Transactions on Communications. COM-33. No. 7*, pages 627–638, July 1985.

[Tob80] F. A. Tobagi. Multiaccess protocols in packet communications systems. *IEEE Transactions on Communications. COM–28*, pages 468–488, April 1980.

[Vit95] A. J. Viterbi. *CDMA: Principles of Spread Spectrum Communications.* Addison Wesley Publications, New York, NY, 1995.

[Wan96] J. Wang. Efficient and accurrate ethernet simulation. *IEEE Transactions on Communications*, December 1996.

[Xu99] H. Xu. CDMA (IS-95A) access channel performance analysis. In *IEEE VTS 50th Vehicular Technology Conference. Vol. 1*, pages 526–530, September 1999.

[Zim00] Christine Zimmerman. Ethernet to get supercharged to 10 gigabits. *Internetweek. Issue 801*, page 16, February 2000.

[ZN99] L. M. Zeger and M. E. Newbury. CDMA capacity with added carriers in a cellular network. *Bell Labs Technical Journal. Vol. 4. Number 3.*, pages 104–120, July/September 1999.

Vita

First Lieutenant Robert J. Bonner entered undergraduate studies at the United States Air Force Academy in Colorado Springs, Colorado where he graduated with a Bachelor of Science in Electrical Engineering degree in May 1997. He was commissioned at that same time.

His first assignment was as a student at Keesler AFB, Mississippi in the Basic Communications and Information Officer Training course where he was recognized as a member of the distinguished graduating class for the alpha shreds. In November 1997, he was assigned to the Air Force Communications Agency at Scott AFB, Illinois where he served as a lead engineer for commercial technology product evaluations. In August 1999, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the 333rd Training Squardon, Keesler AFB, Mississippi.

Permanent address: Air Force Institute Of Technology
Wrigth-Patterson AFB, OH 45433-7765

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
March 2001

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
USING DIRECT-SEQUENCED SPREAD SPECTRUM IN A WIRED LOCAL AREA NETWORK

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Bonner, Robert J., 1st. Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/01M-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Attn: Mr. Kenneth Fore
AFCA/ITAI
203 W. Losey Street
Scott AFB, IL 62221
email: Kenneth.Fore@scott.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES
AFIT Technical POC: Rusty O. Baldwin, AFIT/ENG
Rusty.Baldwin@afit.edu

14. ABSTRACT

Code division multiple access provides an ability to share channel bandwidth amongst users at the same time. Individual user performance is not degraded with the addition of more users, unlike traditional Ethernet. Using direct sequenced spread spectrum in a wired local area network, network performance is improved. For a network in overload conditions, individual station throughput is increased by nearly 212% while mean end-to-end delay was reduced by 800%. The vast improvement demonstrated by this research has the capability to extend legacy-cabling infrastructures for many years to come while easily accommodating new bandwidth intensive multimedia applications.

15. SUBJECT TERMS
DSSS, CDMA, Ethernet, CSMA/CD, network, LAN, wired, wireless

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	174	Maj.Rusty O. Baldwin, AFIT/ENG
					19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4582