Simulation of the MAC Portion of IEEE 802.11 and Bursts of Errors for Wireless Data Networks

Farhood Moslehi

Project Report submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Systems Engineering

Dr. S. F. Midkiff, Chair
Dr. N. J. Davis, IV
Dr. H. A. Kurstedt, Jr.

August 12, 1997
Blacksburg, Virginia

Keywords: Wireless Networks, IEEE 802.11, CSMA/CA, Bit Errors

Simulation of the MAC Portion of IEEE 802.11 and Bursts of Errors for Wireless Data Networks

Farhood Moslehi

(ABSTRACT)

The focus of this research is to investigate the effects of bursts of errors and packet collisions on the performance of the medium access control (MAC) portion of the IEEE 802.11 wireless local area network (LAN) protocol.

An important ingredient in rapid expansion of wireless networks is the seamless transition between wired and wireless systems. The IEEE standards group in charge of developing the widely used IEEE 802.3 LAN standard has developed the IEEE 802.11 wireless LAN standard. IEEE 802.11 remains hidden from the upper levels of the network, thus allowing a seamless transition between networks. The foundation protocol for the IEEE 802.11 standard, known as Distributed Foundation Wireless Medium Access Control (DFWMAC), operates at the MAC level of the Data Link Layer. The protocol bases its access control mechanism on a principle called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), which is an adaptation of the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol used by IEEE 802.3 standard. The collision avoidance scheme in CSMA/CA allows data packets to be transferred via the wireless medium with lower probability of packet collision.

In a slotted multi-access wireless system, performance parameters are affected by the bit error rates on the communication channel. These errors occur as a result of noise introduced by the radio channel or data packet collisions.

Collisions occur when two or more stations select the same time slot to transmit their data, thus causing corruption in data packets.

In this research, a simulation model coded in Microsoft's Visual Basic programming environment is utilized to investigate the effects of bit errors and packet collisions on performance in CSMA/CA. Performance parameters used in this study include throughput, medium utilization, collisions and station data queue lengths.

In the simulation model, error bursts in the communication channel are modeled using a simple Gilbert model with two states, good (G) and bad (B). State G is error free, thus errors can only occur while the model is in state B. Collisions are simulated by two or more stations starting to transmit data packets in the same time slot. Therefore, as the number of stations increases, more and more stations compete for the medium, resulting in an increase in the number of collisions. Collisions are also increased by the amount of traffic that each station introduces into the system. Station load is defined here as the number of data packets per unit time that are released by the higher network protocol layers.

The results in Chapter 5 demonstrate that higher network throughput can be achieved when the aggregate load on the network is distributed. For example, 30 stations offering 20 kilobits per second (kbps) of load for a total of 600 kpbs, results in a network throughput of 585 kbps. However, three stations offering 200 kbps of load for a total of 600 kbps offered load, results in a network throughput of 486 kbps. The distributed load is serviced at a 17 percent higher rate. However, once the network becomes saturated at above 40 stations for this model, collisions will more than offset the performance gains produced by the distribution of load.

Furthermore, reducing the packet size by 50 percent under an approximately 19.5 percent packet error rate results in a 12 percent gain in throughput. This is primarily due to higher utilization of the network by shorter packets. However, as the packet error rate is reduced, the performance gap between the two packet sizes is reduced. Once the errors are removed completely from the communications channel, the longer packets produce a higher throughput than the shorter packets.

Table of Contents

List of Tables

List of Figures

1. Introduction

1. 1. Overview

Communication is a process of transforming, interpreting, and processing information among persons or machines. This process involves a sender, receiver and transmission medium over which the information flows. Traditionally, when a small group of computing devices are interconnected together for data communications services, it is referred to as a local area network (LAN). LANs provide connectivity and services for data flow and sharing in an organization. The medium of choice for physical level interconnectivity in a LAN has been copper wire. Copper wire is inexpensive, readily available, and easy to install. As the need for more bandwidth has become more apparent, fiber optic cabling has begun to play an important role in LAN connectivity.

With users becoming more mobile, the wired nature of LANs will be a restriction. Moreover, some buildings, in the case of LANs, and some cities, in the case of wide area networks (WANs), do not lend themselves to being wired or cabled, posing physical restrictions on the wired approach. Therefore, wireless devices where electromagnetic radiation is the medium of connectivity provide new flexibility and services to users in environments that cannot be well served by traditional wired networks.

1.2. Wireless Networks

Wireless networks can be separated into two broad categories: infrastructure and *ad hoc*. An infrastructure wireless network refers to wireless stations connected to a wired network via access units or wireless hubs much like workstations being attached to a backbone network via a hub in a wired local area network. An *ad hoc* network is a collection of wireless mobile hosts forming a temporary network without established centralized administration or other infrastructure.

An important function of wireless networks is to extend access to the present wired infrastructure. This is only practical through seamless integration of wireless and wired networks. Therefore, the IEEE 802 Working Group, responsible for the development of LAN standards such as Ethernet and Token Ring, developed the IEEE 802.11 standard for wireless networks [1]. The 802.11 protocol is part of the IEEE 802 family of standards which allows for seamless integration between various members of the IEEE 802 family such as Ethernet (802.3) and Token Ring (802.5).

The medium access control (MAC) portion of the IEEE 802.11 protocol, called the Distributed Foundation Wireless Medium Access Control (DFWMAC), is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism with a rotating backoff window. Due to the limited bandwidth offered by wireless media and the fact that collision detection cannot be implemented in a wireless radio frequency network, collisions put a heavier burden on a wireless system than a wired network. The Collision Avoidance mechanism allows the transmission of frames with lower probability of collision than the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) mechanism used by Ethernet, but collisions can and do occur.

Networks and their respective protocols provide several functions to transfer information from a source to its destination. One of these functions includes acting as “bit pipes” for data transfers. Digital communication channels are influenced by noise in the channel. The noise in the channel can cause a bit or a series of bits to change their states from a 0 to a 1 or from a 1 to a 0. Upon this change of state the affected frame becomes corrupted; thus the protocol will cause the source to retransmit the same frame. It is impossible to predict at what point during a transmission of a data frame a burst of errors may or may not occur. However, it is possible to model a burst-noise binary channel using a Markov chain with two states, a good state (error free) and a bad state (possibility of errors) [2]. Data frames can only be successfully transmitted, i.e. without errors, between bursts of errors.

In this research, the MAC portion of the IEEE 802.11 wireless LAN protocol along with errors due to burst noise in a digital binary channel have been simulated through a computer simulation model. This simulation model allows us to gain a better understanding of the affect of bit errors on network performance.

Data frames are assumed to be either the maximum allowed size by the MAC portion of IEEE 802.11 (18,704 bits), or 50 percent of the maximum size (9,352 bits). It is further assumed that the higher level layers of the model deliver frames for transmission with a Poisson distribution (exponential interarrival times).

1.3. Report Organization

This report presents the simulation model used in this research and performance results. Chapter 2 presents an overview of the physical architecture of wireless networks and of the wireless medium. This chapter also describes the DFWMAC, the concept of random backoff time, collisions and the queuing model. Chapter 3 presents the burst-noise channel model (Gilbert model) with probability parameters. Chapter 4 presents the simulation model, which is coded in Microsoft's Visual Basic 5.0. Chapter 5 describes various simulation runs and results. Parameters such as load factor, number of stations and probability factors for error generation are varied to understand the behavior of the network. Chapter 6 contains concluding remarks, suggestions for performance enhancements and suggestions for future work. Appendixes A through C provide numerical results. A glossary is provided in Appendix D.

2. IEEE 802.11 Wireless LAN

2.1. Introduction

Wireless LAN technologies offer a wide range of capabilities and operate in different ways and environments. The common denominator among all of these technologies is that they do not require a fixed wire connection, but instead transmit signals to one or more wireless receivers over a wireless channel. This chapter describes the portion of the electromagnetic spectrum used to send information and techniques that are used to utilize radio frequencies as a communications medium. The chapter further focuses on the MAC portion of IEEE 802.11, including CSMA/CA, the backoff mechanism and packet collisions.

2.2. Communications and Electromagnetic Spectrum

A communication process is defined as the exchange of information between two or more entities. The source is the entity that sends the information and the destination is the entity that receives and processes the information. In a wireless network, like any other communication system, there is a source called the transmitter and a destination called the receiver, and the medium is air rather than copper or fiber optic cables used in more conventional communications systems. In wired digital communication systems, digital information is sent through wires as electrical pulses (signals). For example, a 5-volt pulse followed by a 0-volt pulse could represent a 1 and a 0, but in a wireless communication system, digital information is encoded and transmitted using electromagnetic (E-M) radiation.

The basic principle upon which a wireless communication system operates is the same as in a wired system. Instead of electrical pulses through wires, E-M waves are sent through open space. First, the transmitter captures a data stream and then outputs this information in the form of E-M waves using an antenna.

Since, the receiver in this system is tuned to the same frequency as the transmitter, it is able to detect these E-M waves and translate them back into a digital signal or data stream.

Techniques such as amplitude modulation (AM) and frequency modulation (FM) are used to add information to the basic E-M wave. In amplitude modulation, changing the amplitude of a wave in a predetermined fashion encodes digital data. The receiver can detect the amplitude changes of the carrier wave and is then able to recreate the transmitted data. In frequency modulation, which is widely used for wireless data networks, the frequency of the carrier wave is changed in a predetermined fashion. Like electrical pulses in a wired network, waves of various frequencies are transmitted by the transmitter and translated back to their original form by the receiver. Therefore, the amount of information, commonly referred to as bandwidth, is directly proportional to the range of wave frequencies. Figure 2.1 depicts different frequencies associated with the E-M spectrum.

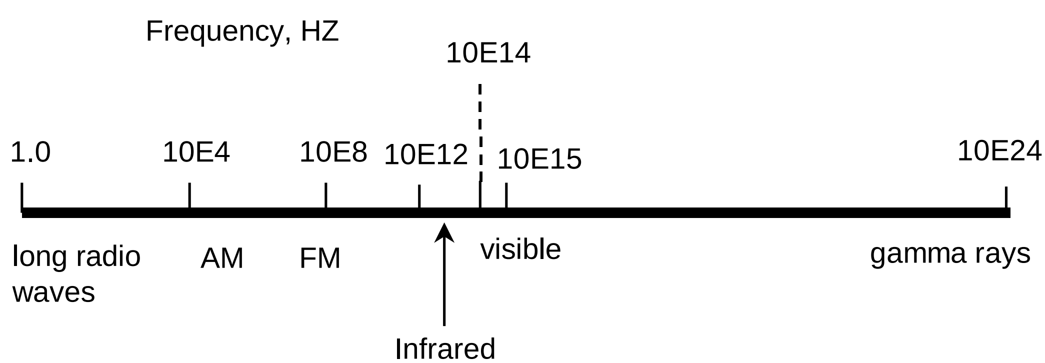

Figure 2.1. E-M spectrum

Wireless data networks fall into two major categories: networks that operate in the infrared portion of the E-M spectrum (just under the visible light) and radio frequency networks. The key difference between the use of infrared and radio frequency is the support of roaming. Infrared is a line-of-sight technology. There has to be a direct line of sight or at least a surface to bounce the waves from the transmitter to the receiver. Radio frequency systems can penetrate through objects such as walls and doors in most office buildings; hence their popularity in present wireless systems.

In the United States, the Federal Communications Commission (FCC) has the authority to regulate the use of different parts of the E-M spectrum. According to the FCC, unlicensed wireless networks can operate in three areas of the radio spectrum, referred to as the Industrial, Scientific and Medical (ISM) bands [3]. The frequencies for these regions are:

- 902 - 928 MHz
- 2.4 - 2.4835 GHz
- 5.725 - 5.825 GHz

Since there are a host of other unlicensed devices that operate in these same regions of the E-M spectrum, techniques are needed to avoid interference. Furthermore, security issues arise from sharing the same frequency among multiple wireless networks. A technique developed by the military to help secure its transmissions offers a way around the interference and security problems. This technique, called spread spectrum, involves spreading transmissions across a range of frequencies, rather than transmitting on one frequency all the time [4].

One approach known as frequency hopping spread spectrum (FHSS) involves dividing a range of the radio spectrum into individual channels, each on a specific frequency. A transmitter hops from one frequency to the next in a predetermined fashion and, since the receiver is aware of the hopping pattern, it can follow the transmitter and receive the information. This method is particularly attractive in the ISM band, where there is a high probability of one or more high-powered narrow-band interference sources. The second method of spread spectrum communication is direct sequence spread spectrum (DSSS). The DSSS system provides a wireless LAN with both 1 Mbps and a 2 Mbps data communication capability. Simulation models in the following sections are based upon a 2 Mbps data communication capability. Here, the source data to be transmitted is first exclusive-ORed with a pseudorandom binary sequence. The bits making up the sequence are random, but the same sequence is made much larger than the source data rate. When this data is modulated and transmitted, due to the fact that it was exclusive-ORed with a longer sequence of random binary bits, it occupies a wider frequency band than the original source data bandwidth. This technique makes the transmitted signal appear as noise to any other device using the same frequency spectrum. Since all the receivers in the same wireless network know the binary sequence being added to the transmitted signal, they are able to extract the original information from the signal. The receivers first search for a preamble sequence signal and, once it has been recognized, they start decoding the bit stream [5].

DSSS combines a data signal at sending station with higher data rate bit sequence (processing gain). A high processing gain increases the signal’s resistance to interference [1]. FCC allows a processing gain of 10 for DSSS.

2.3. Wireless LAN Model and Slotted Multi-Access Systems

To illustrate the integration of wireless systems into the present wired data network systems the Open Systems Interconnect (OSI) model developed by International Organization for Standards (ISO) is followed. Figure 2.2 depicts the OSI model and the corresponding layers in a wireless system. The physical level is the lowest level. In a wired network information such as cabling types, network interface card (NIC) connections, and voltage levels is addressed by physical layer standards. In a wireless system, information such as frequency levels and modulation techniques is specified.

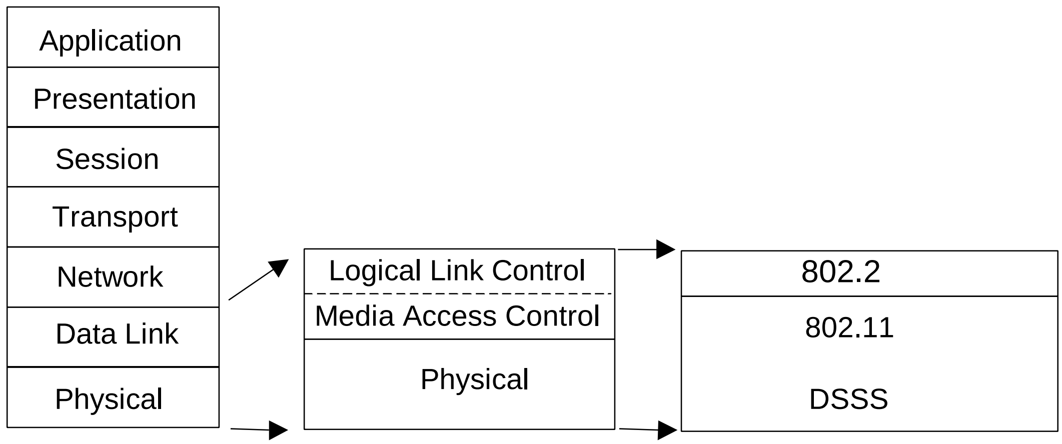

Figure 2.2. Affected layers of the OSI model.

The layer above the physical layer is the data link layer. Here, the protocol defines the rules for accessing the network. The data link layer is usually separated into two separate sub-layers. The logical link control (LLC) sub-layer provides a logical link between network nodes. The second sub-layer (beneath the LLC) is the medium access control (MAC) sub-layer, which implements a distributed access control protocol.

In slotted multi-access protocols, such as Ethernet (IEEE 802.3), data packets, or frames, are sent during time slots that are defined by the protocol. In these networks nodes with data packets ready for transmission compete for access to the medium. The most widely used MAC protocol, used in Ethernet, is based on a discipline called carrier-sense multiple-access with collision detection (CSMA/CD) to access the medium. CSMA/CD can only be utilized in logical bus networks. To transmit a frame using CSMA/CD, a node includes a destination address in the frame. The frame is then broadcast over the medium (cable). All nodes connected to the network detect when a frame is transmitted. The node with the correct address contained in the frame header opens the data packet to receive the data. To access the medium, a node with data frames to send “listens” to the medium to detect whether a frame is presently being transmitted over the medium. As mentioned above, frames are broadcast over the network, thus enabling every node to be aware of frame transmissions. If a carrier signal (CS) is detected, the node defers its transmission until the current transmission over the medium has ended. Once the medium is free, the node will access the medium and commence its transmission. It is highly likely that under heavy load conditions two or more nodes detect the free medium and simultaneously access the medium to transmit their frames. A collision is then said to occur that causes all frames to have corrupted data bits. Since data frames are broadcast over the network, a source also monitors the data frame being sent over the network. In the event of a collision the transmitted signal differs from the monitored signal and a collision is detected. To notify the network of the collision, a jamming signal is sent and the process is repeated to re-transmit the collided frame.

2.4. IEEE 802.11 MAC

The IEEE 802.11 wireless LAN standard supports operation in two separate modes, a distributed coordination (DCF) and a centralized point-coordination mode (PCF). The IEEE 802.11 MAC is called Distributed Foundation Wireless MAC (DFWMAC) and the access mechanism is based on a modified version of the CSMA/CD access protocol (discussed in the previous section) called carrier sense multiple access with collision avoidance (CSMA/CA). Lower bandwidths in

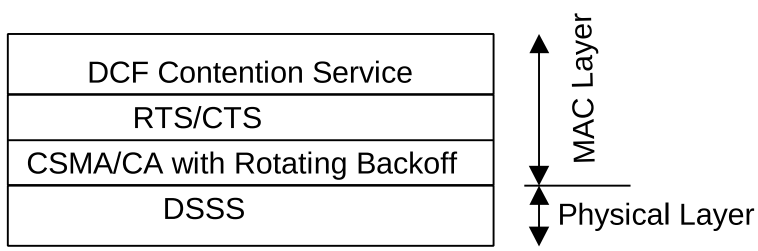

Figure 2.3. Structure of DFWMAC.

wireless networks and the fact that collision detection cannot be implemented in a wireless radio frequency network, make collisions a highly taxing proposition, thus the CSMA/CA scheme has a built-in mechanism to avoid collisions and provide fair access to the medium. Figure 2.3 depicts the structure of the DFWMAC used in this study.

One of the issues that arises in a wireless network is that there is no guarantee that the destination is within range of the source at the beginning of the transmission or perhaps it moves out of range during transmission. The DFWMAC handles a four-way handshake procedure shown in Figure 2.4 to insure connection integrity. For contention-free service, the Point Coordination

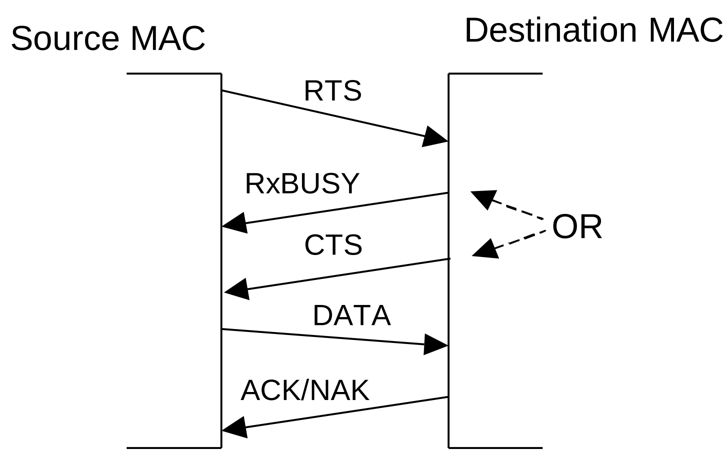

Figure 2.4. Four-way handshake.

Function (PCF) portion of the DFWMAC is used. Whenever a node intends to transmit data using PCF, it first sends a short request-to-send (RTS) message over the network. The message includes the source and destination addresses. If the destination is ready to receive a frame, a clear-to-send (CTS) message is broadcast over the network. If the destination is busy, it broadcasts a receiver-busy (RxBUSY) message. If a CTS signal is detected by the source then the source sends the data. If the data frame is successfully received by the destination, it returns an ACK to the source. However, if the data frame is corrupted a NAK is sent to the source. In the case of a station being out of range, it is built into the protocol to repeat the above sequence for a pre-determined number of times. If the destination repeatedly does not send an ACK or a NAK, it is assumed that the node is out of range.

Under the CSMA/CA technique, all stations listen to the medium as in CSMA/CD. A station that is ready to transmit a frame will sense the medium and, if the medium is busy, it will wait until the end of the current transmission. It will then wait for an additional predetermined time period, denoted as DIFS (DCF Inter-frame Space), and then pick a random time slot within a contention window to transmit its frame. If there are no other transmissions before this time slot arrives, it will start transmitting its frame. If there are transmissions by other stations during this time period known as the backoff time, the station will freeze its counter and will resume the count where it left off after the other station has completed its frame transmission and after DIFS [6]. The collisions can now occur only when two or more stations select the same time slot in which to

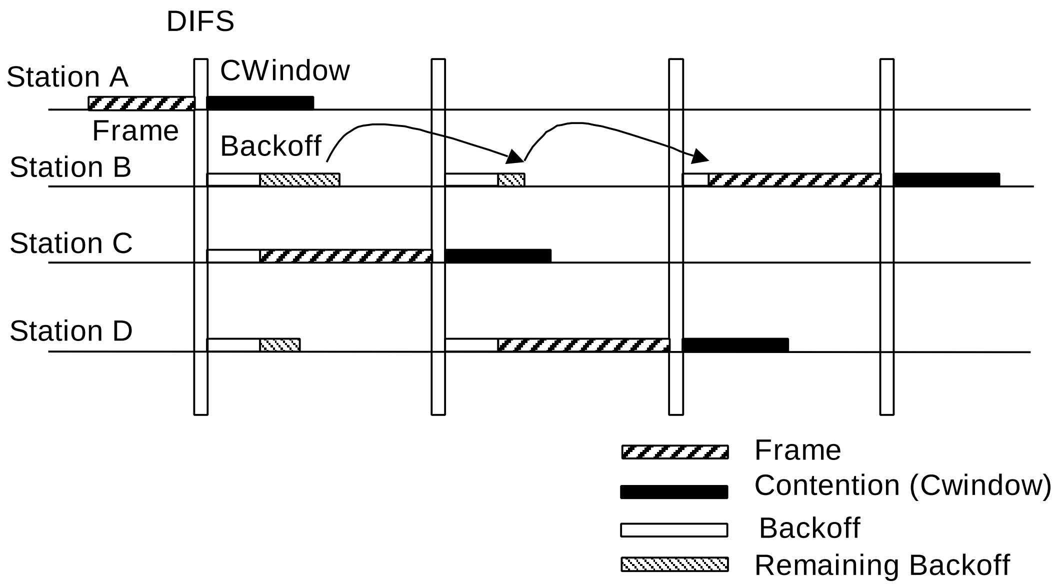

Figure 2.5. Transmission and backoff procedure.

transmit their data frames. In the event of a collision an explicit NAK is sent to the source by the destination notifying it of the collision. The source will then backoff within a larger contention window to lessen the possibility of a subsequent collision. Figure 2.5 illustrates the DFWMAC medium access technique.

2.5. Random Backoff Time

As noted above, if the medium is busy, the protocol dictates that stations follow the random backoff procedure. The backoff procedure selects a random slot from the slots that are available in the contention window according to the following equation.

$$BackoffTime = \mathrm{INT}(CW * \mathrm{Random}()) * \mathrm{Slot\text{-}Time} \qquad (2.1)$$

CW is an integer between *CWmin* and *CWmax*. *Random()* is a random number between 0 and 1. *Slot-Time* is the constant value of time for each slot and is fixed for a given physical transmission scheme.

The backoff timer will decrement by amount *Slot-Time* after every slot, while the medium is free. The timer freezes while the medium is busy. The counter will resume counting when the medium becomes free. Transmission occurs when the counter reaches zero. A station that has just transmitted a frame and has another

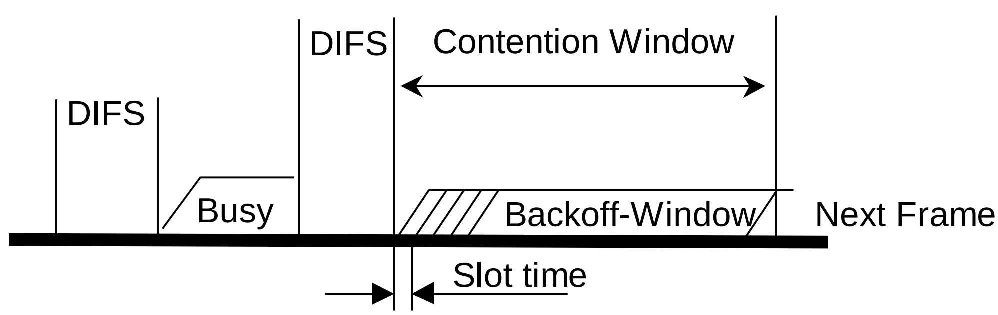

Figure 2.6. Basic access and backoff procedure.

frame to transmit will enter the contention competition and follows the backoff procedure for a new time slot [6]. Figure 2.6 shows the basic access and backoff procedure.

This procedure results in multiple stations deferring and entering into a rotating backoff procedure; the station selecting the lowest time slot will transmit first. This method tends toward being fair across the network on a first come first served basis.

The contention window (*CW*) in the above equation controls the time slot competition among the stations. Let us assume that station A is ready to transmit a frame at time X_1, station B is ready to transmit a frame at time X_2 ($X_2 > X_1$) and that *CW* is constant at 31. Each station will then calculate a backoff time using Equation 2.1. Station A selects a time slot from the first contention window and starts its counter. This shrinks the portion of the contention window that overlaps the contention window for station B. By time X_2, when station B enters

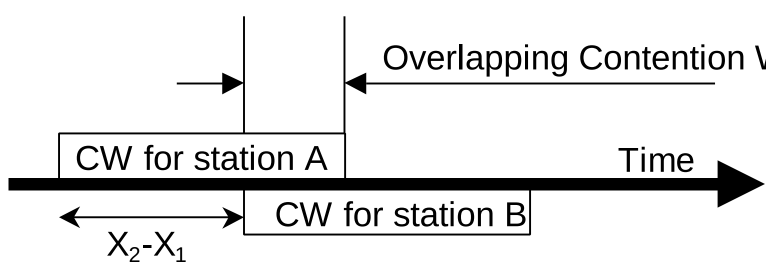

Figure 2.7. Contention window.

the competition, station A has shrunk its window by X_2-X_1, thus giving station A higher probability for transmission than station B with its newly selected time slot. Figure 2.7 depicts this scenario.

2.6. Collisions

As mentioned in the previous sections, collisions can and do occur with DFWMAC when two or more stations select the same time slot from overlapping contention windows. Unlike IEEE 802.3 a collision cannot be detected and transmission stopped. The corrupted frames are transmitted in full, resulting in a reduction in throughput. After each collision, stations must reenter the competition with exponentially increasing *CW* values. Initial transmissions occur with 31 time slots per contention window. The first retransmission occurs with 63

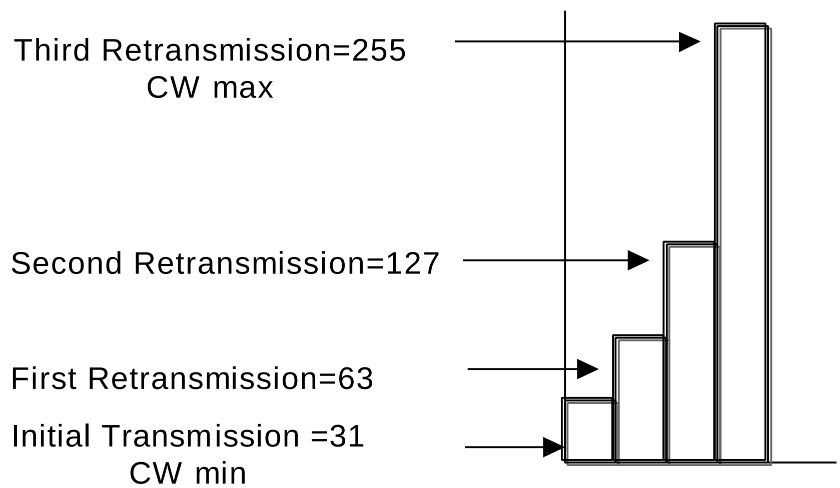

Figure 2.8. Exponential increase of contention window.

time slots per contention window, the second occurs with 127 time slots and finally the third and subsequent occur with 255 time slots per contention window. As the contention window becomes larger the probability of selecting a unique time slot increases, thus enabling a particular collided frame to be successfully transmitted without further delay. Figure 2.8 shows the contention window progression for the model used in this study.

The performance of an IEEE 802.11 wireless LAN is affected by the behavior of the CSMA/CA scheme for given network and traffic conditions. In addition, the performance of a real network will be affected by bit errors, as discussed in the next chapter.

3. Burst-Noise Error Model (Gilbert Model)

3.1. Introduction

The bit error rate (BER) for a communication channel is defined as the probability of a single bit being corrupted in a defined number of transmitted bits. For example, a BER of 10^{-4} would mean that, on average, 1 bit in every 10,000 bits would be corrupted. Data packets consist of a fixed number of bits to be transmitted.

Modeling the exact structure of bursts of noise in real digital communications channels is a complex problem. In general, there are no set procedures nor are there exact parameters that can be used to accurately predict the occurrence of such clusters of errors. As discussed previously, data packets are rendered useless if one or more data bits that make up a particular frame change state. The use of an approximate model to estimate error performance allows the complex statistics of errors to be reduced to a manageable set of parameters. Gilbert, in a paper published in March of 1960, showed that a simple Markov model with two states and three parameters approximates the behavior of these errors clusters [2]. These probability parameters are typically derived from measured experimental statistics.

The Gilbert model is used to model errors in this research, as described in this chapter. Section 3.2 focuses on the theory and the structure of the Gilbert Model. Section 3.3 deals with simulation of a wireless data network using the Gilbert model.

3.2. Gilbert Model

A symmetric binary channel models a noisy binary communications channel. A channel is symmetric whenever the probability of a digit changing its state is a function of only binary noise digits. The channel adds noise digits Z_n, which are added to input digits X_n to produce output digits Y_n. The addition is a modulo-2 operation. The channel is also said to be memoryless; that is a series of independent trials produces the noise digit [2].

The Gilbert model utilizes a Markov chain with two states to generate bursts of errors. A Markov chain is defined as a discrete-state Markov process. A Markov process is a process where the future states are independent of past states and depend only on the present state. The two states of the Gilbert model

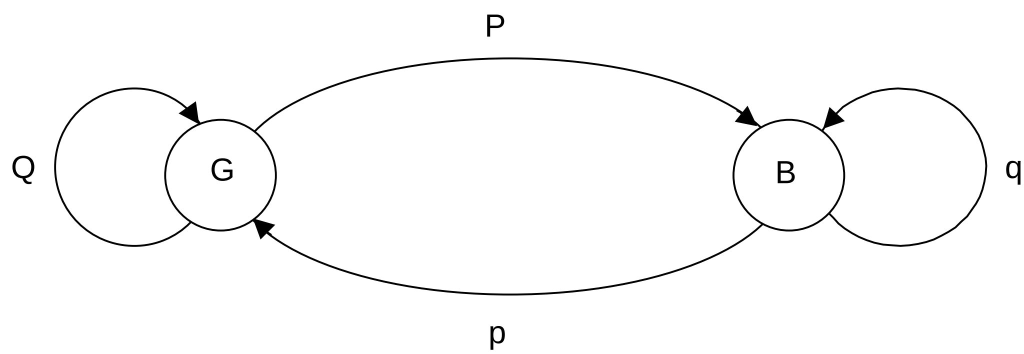

Figure 3.1 . Transition diagram.

are called G (good) and B (bad). In state G the process is error free (Z_n =0), and in state B there is a probability of an error (Z_n =0 or 1). Figure 3.1 depicts the transition diagram for the Markov chain.

P is defined as the probability of the state changing from G to B. Similarly p is defined as the probability of the state changing from B to G. Q and q are the probabilities of remaining in state G and B, respectively. Both states G and B tend to persist; therefore, transition probabilities P and p are small compared to probabilities Q and q of remaining in the G and B states, respectively. Mathematically these quantities are defined as shown.

P = probability (G$\rightarrow$B)

p = probability (B$\rightarrow$G)

Q = 1 - P

q = 1 - p

State G is error-free, thus errors can only occur, with probability 1-h, in state B. h is the probability of making no errors in state B. Runs in state G are followed by runs in state B with clusters of errors occurring during runs in state B. Furthermore, since the process is memoryless and errors in state B are independent, it is assumed that the number of bits transmitted while in G and B is geometrically distributed with mean 1/P and 1/p, respectively.

Values for parameters P, p and h are not easily obtained. However, there are methods of deducing them from statistical measurements for a particular channel. These values are not the same for all binary channels nor they are the same for different physical environments. These constants can be estimated by fitting a curve to measured results. By knowing the three parameters, P, p and h, it is possible to simulate an error distribution model for clusters of errors that occur in a binary channel.

However, it is not always possible to approximate the values of P, p and h by fitting a curve to measured results. In such cases, ranges of values are selected for each parameter. Mathematically or through computer simulations unlikely values are discarded and each parameter is narrowed down to an acceptable and a smaller range of new values. Based on a particular issue that is being investigated, different values of P, p and h are used to show best and worst case scenarios for a particular performance parameter.

3.3. Simulation of Gilbert Model

To gain a better understanding of whether or not data packets can be transmitted error free in a communications channel, and the resulting effect on performance, the Gilbert model is simulated through a computer code. Simulation of the Gilbert model is a discrete-event simulation model. Discrete-event systems change state at discrete points in time as opposed to continuous systems that are a function of time. The transition diagram in Figure 3.1 along with the three parameters P, p and h are used to simulate the Gilbert model.

Looking at each bit individually is one way of simulating the Gilbert model. To determine whether or not a particular bit is in error using this method, the simulation model is required to make individual calculations for each bit. While the model is in state G, a single calculation is made to determine if a transition occurs. When the model switches to state B, then an additional calculation needs to be made to determine whether or not a bit is in error. Thus, each packet is responsible for at least 18,704 calculations and comparisons. A small error in any calculation would affect the results of the calculations greatly, since the small error would be repeated a large number of times. Furthermore, the computer code would be inefficient for large simulations.

In this research, the Gilbert model is implemented by looking at bit positions where changes occur. A change is defined as either a state change or an error occurrence. The number of bit positions between such changes can be represented by a geometrically distributed random variable. The geometric distribution is used to model the number of attempts between successive failures (or successes) [7]. For a general geometric distribution:

$$p\,(k) = (1-p)^{k-1}\,p\ ,\ 0 \le k \le \infty\ ,\ 0 \le p \le 1 \qquad (3.1)$$

p (k) is the probability mass function and p is the probability of a state change. Geometric variate k, which is the number of attempts between successive failures or successes, can be easily generated using inverse transformation [7]:

$$k = \ln(r) / \ln(1\text{-}p) \tag{3.2}$$

Variable r denotes a uniformly distributed random number between 0 and 1.

Simulating the Gilbert model using the geometric distribution function requires multiple applications of Equation 3.1. The geometric distribution of bits in state G and B are derived from equation 3.1 and shown as Equation 3.3 and 3.4, respectively.

$$p(k_g) = (1\text{-}P)^{k_g-1}P \quad , \; 0 \le k_g \le \infty\,, 0 < P < 1 \tag{3.3}$$

$$p(k_b) = (1\text{-}p)^{k_b-1}p \quad , \; 0 \le k_b \le \infty\,, 0 < p < 1 \tag{3.4}$$

Given Equations 3.3 and 3.4, the inverse transformation function in Equation 3.2 is applied to determine bits between a change from state G to B and state B to G as represented by Equations 3.5 and 3.6.

$$k_g = \frac{\ln(r)}{\ln(1\text{-}P)}$$

$$k_b = \frac{\ln(r)}{\ln(1\text{-}p)} \tag{3.6}$$

Equations 3.1 and 3.2 are also applied to calculate the position of error bits while in state B.

$$p(k_{1\text{-}h}) = (h)^{\,k_{1\text{-}h}-1}\,(1\text{-}h)\,, 0 \le k_{1\text{-}h} \le \infty, 0 < (1-h) < 1 \tag{3.7}$$

$$k_{1\text{-}h} = \frac{\ln(r)}{\ln(1\text{-}h)} \tag{3.8}$$

Figure 3.2 depicts the application of geometric distribution to the Gilbert model.

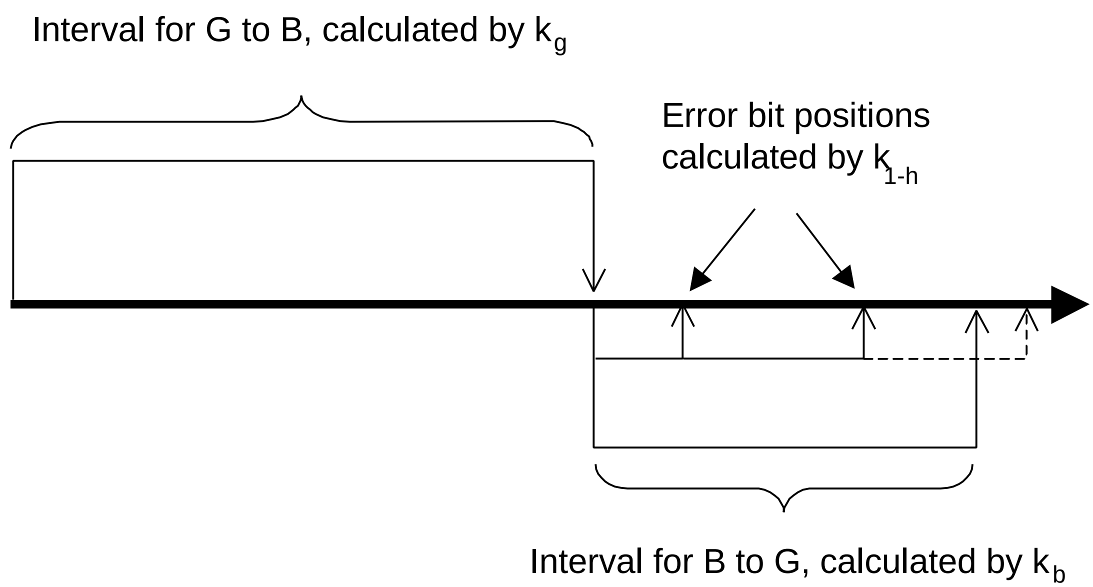

Figure 3.2. Application of geometric distribution to Gilbert model.

Given that the Gilbert model starts at state G, the simulation model calculates the bit position for transition from state G to B by using Equation 3.5. Similarly, the simulation model utilizes Equation 3.6 to calculate the bit position for transition from state B to G. Upon entering state B, Equation 3.8 is used to calculate the value of k_{1-h}. The parameter k_{1-h} points to the position of the next error bit, thus if the value of k_{1-h} is greater than k_b, the transition to state G occurs prior to the next error position and the error does not occur. A new k_g value is calculated and the cycle repeats itself. However, if the value of k_b is greater than k_{1-h}, an error occurs at position calculated by k_{1-h}. Equation 3.8 is used again to calculate new error positions. Every time a new value for k_{1-h} is calculated, it is added to the total value of previous k_{1-h} values. The sum of all k_{1-h} positions are then compared with the value of k_b to determine whether or not the next error position occurs prior to the state changing from B to G . This process is repeated until the sum of k_{1-h} positions exceeds $k_{b.}$ Upon reaching the bit position pointed to by k_b the model reverts back to state G, and all values are reset.

The following chapter discusses the queuing model used by the simulation and how the implementation of the Gilbert model is incorporated into the simulation model for the CSMA/CA portion of the IEEE 802.11 protocol. Furthermore, the logic flow through the simulation model is discussed.

4. Simulation Model

4.1. Introduction

The previous two chapters described the access mechanism used by IEEE 802.11 and the error clusters in a communications channel from a theoretical point of view. This chapter focuses on the development of the complete simulation model and its implementation. This entails understanding the behavior of CSMA/CA and burst errors, along with the queuing model used in this simulation. The model is implemented through a computer code written in the Visual Basic 5.0 programming environment using the approach of [8].

4.2. Assumptions

There are two different groups of parameters that are used by the simulation model. The first group includes parameters that have constant values. These parameters are discussed in this section. The second group includes parameters that are input to the simulation model to observe their effects on performance measures. These parameters include Gilbert model parameters, the number of stations in the network and offered load. The second group of parameters is described in Chapter 5.

Fixed parameters are defined below.

- Packet lengths are set to 18704 bits. The simulation model also utilizes a packet length of 9532, as discussed in Section 5.5.
- The collision zone is defined to be 1 μs. This is the time needed for all other stations to become aware of an ongoing transmission. During this period if other stations commence packet transmissions, collisions occur.
- The slot-time is defined as the fixed unit of time which comprise a contention window. The value is set at 50 μs. In a slotted multi-access system events are processed in terms of slot-times.

- DIFS is the waiting period before and after packet transmissions. Positive and negative acknowledgements are sent during this time interval which is set at 150 μs.

Data packets are assumed to arrive randomly to be served by the medium. The interarrival times between packet arrivals are independent and exponentially distributed, hence, the packet arrivals are Poisson arrivals. Equation 4.1 is used to generate interarrival times t, according to an exponential distribution.

$$t = -T * \ln(r) \tag{4.1}$$

T is the mean interarrival time, and r is a uniformly distributed random number between 0 and 1.

4.3. Simulation Model

This is an event-driven simulation model. This means that rather than being based on pre-determined units of time such as “ticks” of a clock, the flow through the program is controlled by events, specifically by the next event, whatever that event may be. The code is also designed modularly; different modules control different events using an object oriented paradigm.

There are two macro-level operations that are performed on each data packet by the program. A data packet is first processed through the CSMA/CA modules that determine whether or not a packet is transmitted collision free. Secondly, the packet is processed by the Gilbert model routines to determine whether the packet has been corrupted by bit errors in the channel.

Let us look at two different hypothetical runs by the simulation code. The first example consists of a single packet being delivered to the system for delivery and, while the packet is being processed, no other events occur. This is considered to be the simplest simulation run. Figure 4.1 shows an event diagram of this example. Note that numbers in parenthesis indicate the progression of events. For example, “End DIFS” (2) occurs prior to “End of each slot” (3).

The first event is a packet arrival event, and is calculated by the event-arrival module in the program. The event arrival module considers the load variable specified as program input. The module uses the following relationship

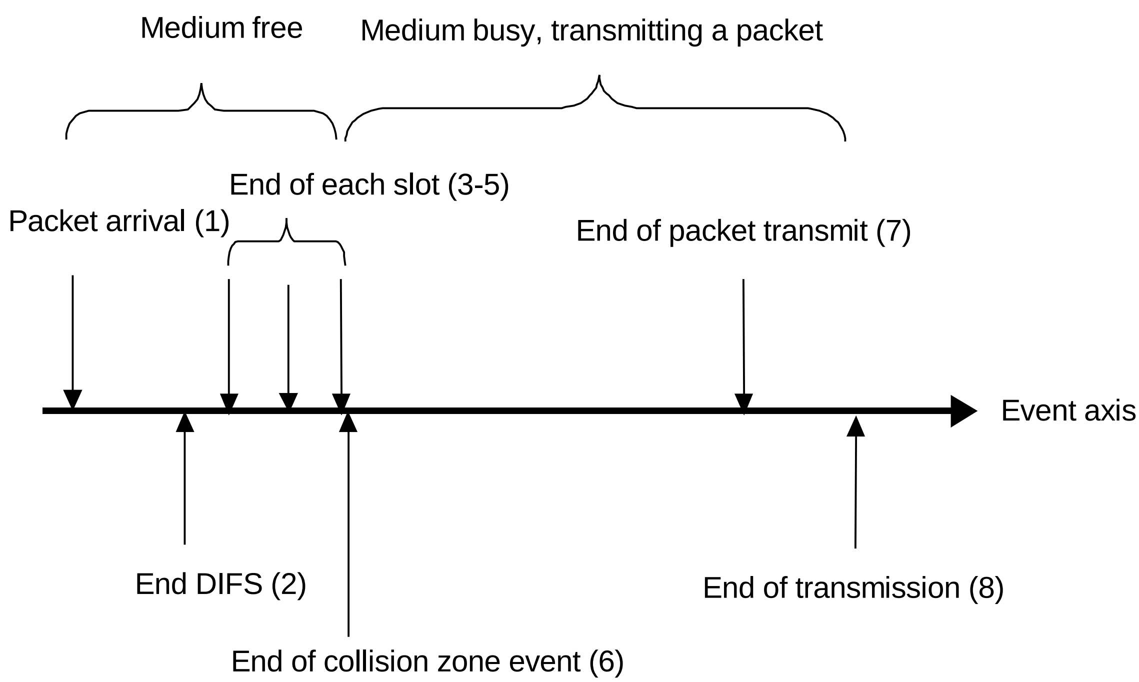

Figure 4.1. Event flow diagram for a single packet.

to calculate exponentially distributed interarrival times for packets.

$$\text{arrivals} = -\text{load} * \log(r) \tag{4.2}$$

Variable r is a uniformly distributed random variable between 0 and 1.

After the packet arrives, the system looks for the next event. Since the medium is free and there are no other packets in the queue, the next event is the “End of DIFS” event. At this event the backoff counter is calculated from the time slots available within the contention window as described in Chapter 2. Events 3 through 5 are all end-of-slot events. Each time an end-of-slot event occurs, one unit is subtracted from the backoff counter until it reaches zero. When the backoff counter reaches zero, the packet begins transmission. Based on this implementation of IEEE 802.11 at 2 Mpbs, the elapsed time for packet transmission is 9.35 ms. The 6th event is the "End-of-collision" event which occurs 1µs after the start of transmission. During the collision zone, two or more stations can commence their frame transmissions, thus causing packet collisions. After this event, the system condition changes from free to busy. The 7th event is the actual completion of packet transmission. This event is followed by the 8th event, which is the "End-of-transmission" event, at which point ACKs or NAKs complete transmission back to the packet source.

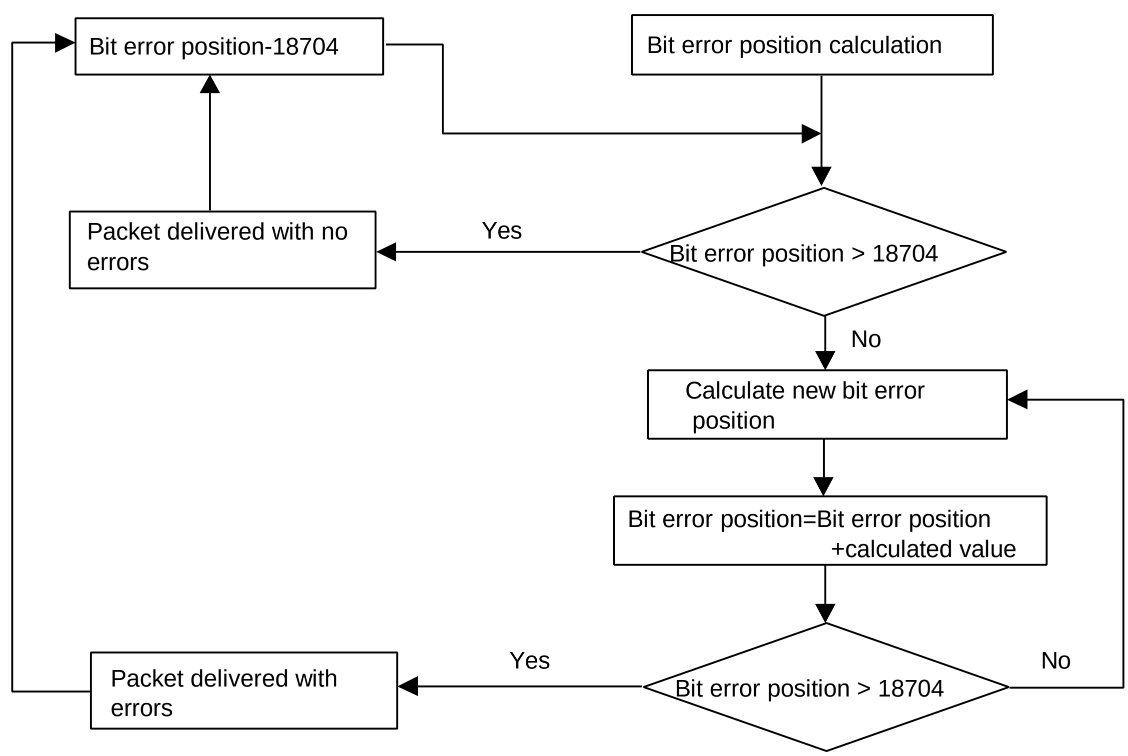

Figure 4.2. Gilbert Module logic flow.

To account for bit errors affecting a transmitted packet, a separate module (Gilbert module) keeps track of the next error bit as described in Section 3.3. Figure 4.2 shows how the Gilbert module processes packets. This module is executed at the end of the packet-transmit module.

Error bit positions are stored in an error bit position counter. Every time a packet is transmitted, the counter is compared with the total number of bits ready to be transmitted (18704). If the next error bit position is greater than 18704, then the frame is transmitted error free and the total number of transmitted bits (18704) are subtracted from the error position counter. If the error position counter is less than 18704, then the transmitted packet contains error bits. New error position bits are calculated and added to the counter until the value of the counter

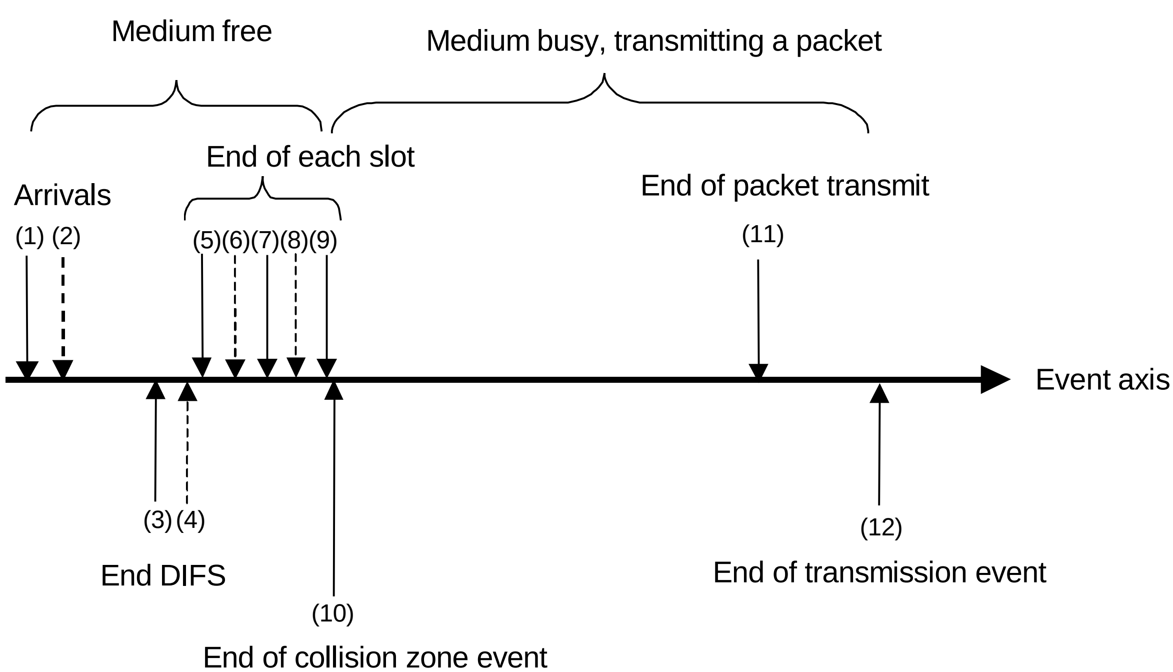

Figure 4.3. Events for two stations with data packets.

exceeds 18704. When the counter surpasses this threshold a corrupted packet is transmitted, and the above cycle repeats itself by calculating new error positions.

Now, let us look at a slightly more complicated sequence of events where two stations receive packets for transmission. Figure 4.3 shows the event diagram for such a system. The events for station A are represented by solid arrows, the events for station B are represented by dashed arrows, and the numbers in the parentheses represent the order of event occurrences.

Event 1 is a packet arrival at station A. Event 2 is a packet arrival at station B. At event 2, DIFS for station B begins, while DIFS for station A had already begun at event 1. Now, let us calculate the time remaining from event 2 to event 3 using Figure 4.3. At event 2, the elapsed time of DIFS for station A is calculated as T_2-T_1, with T_1 representing time at event 1 and T_2 representing time at event 2. Therefore, the DIFS counter for station A, which starts at 150 μs at event 1, is reduced by T_2-T_1. At event 2 the time that remains to reach event 3 is calculated by:

$$T_3\,(\mu s) = 150-(T_2 - T_1) \qquad (4.3)$$

Once, event 3 is reached, station A selects a backoff counter, while station B is still within its DIFS zone. Let us assume station A selects a backoff value of 3. This means that station A needs to complete 3 slot events before the backoff counter reaches zero. Let us further assume that station B selects a backoff value of 5. At event 4 Equation 4.3 is applied using the slot-time interval constant (50 μs) to calculate the remaining time to event 5 (the completion of first slot event by station A) as shown below.

$$T_5\,(\mu s) = 50-(T_4 - T_3) \qquad (4.4)$$

Every time an end-of-slot event occurs for any station, the backoff value counter for that station is reduced by 1. When the backoff counter of station A in

Figure 4.3 reaches 0, station A enters a collision zone interval. At the end of the collision zone interval, the status of the medium changes from “free” to “busy.” Upon occurrence of this event station B freezes its backoff counter and the counter reverts back to the previous integer value which would be 3 in our example. Events 11 and 12 are reached by station A while station B remains at event 8. The system status flag changes from “busy” to “free” at event 12, thus the next event (barring any new arrivals for either station) is for station B to complete an end-of-slot event, thus reducing the value of the backoff counter by 1. At this point the backoff value for station B is at 2. Therefore, station B completes 2 more successive end-of-slot events. This allows station B to reach event 9, continue on to event 12 of Figure 4.3, and transmit its packet.

The above example involves only two stations, but the same logic is applicable to any number of stations. The only difference between a multiple station system versus a two-station system lies in what event for which station is occurring next. Every event for each station needs to be calculated individually and then all the next-events are compared with one another. The next event for the entire system is the smallest next-event value. The value of next-event is then subtracted from every active station’s event counters, and the process is repeated.

For the two-station example above, let us assume that station A has an arrival at time T, and station B has an arrival at T+x, where $0<x<1$ μs. Furthermore, stations A and B select the same random slot-time to transmit their respective packets. Station A starts its packet transmission and before the end of collision zone event in Figure 4.3 is reached which notifies other stations that the medium is busy, station B starts its transmission. The simulation model will advance to the next two events, although a collision has occurred, and both stations must re-transmit the collided packets.

4.3 Verification and validation

Verification for this simulation model is done through a comparison between the model description and the various modules used in the simulation program. A systematic step-by-step walkthrough of the program code along with the examination of the intermediate results for the computer-generated parameters verifies the result produced by this simulation model.

The validation of the simulation model is achieved by a comparison between the generated results and expected results. The model is further validated by comparing the results generated by this model to the results of other simulation models that are not exact, but are based on the same fundamentals. Results from Weinmiller, *et al*. [6] are used for comparison of results.

In the following chapter, the simulation model discussed in this chapter is utilized to investigate the effects of load, number of stations and bit error parameters.

5. Results

5.1. Introduction

This chapter presents the results of the simulation model. As described previously there are two major parts in this simulation model. First, the CSMA/CA portion of the simulation model is used to understand the behavior of the network under different loads and numbers of stations. Secondly, the Gilbert model portion is used to understand the behavior of the network under different error conditions. Although, the two parts are discussed separately, it is important to note that the simulation program used to produce the results of this chapter fully integrates the two parts into a single network model.

The duration for simulation runs in this chapter is 300 seconds. The accuracy gained as a result of runs that are beyond 300 seconds is insignificant. Moreover, simulation runs of greater than 300 seconds for 50 stations under high loads are extremely time consuming, and are difficult to achieve using a microcomputer.

5.2. Parameters for the Simulation Model

5.2.1. Network Parameters

There are two sets of parameters that are utilized by the simulation model. The first group includes constants that are coded directly into the program. The second group consists of parameters that can be specified for each simulation run. These parameters have been defined through pilot studies, as described in Section 5.2.2. For example, three levels of load are considered, high, medium and low. Based on pilot studies, network utilization values of 60, 40 and 20 percent constitute high, medium and low load factors, respectively. Tables 5.1 and 5.2 summarize the parameters. Note that in Table 5.2, load values are offered load per station.

Table 5.1. Constant Parameters

Slot time	50	µs
DIFS	150	µs
Collision zone	1	µs
DSSS capacity	2	Mbps
CW1,CW2,CW3,CW4	31,63,127,255	slots

Table 5.2. Variable Parameters

High load	36	Kbps
Medium load	20	Kbps
Low load	8	Kpbs
High number of stations	50	stations
Medium number of stations	20	stations
Low number of stations	2	stations

5.2.2. Gilbert Model Parameters

As discussed in Section 3.3, Gilbert model parameters can be selected from a range of values for each parameter. Based on previous studies [9] and typical BERs for wireless digital communications channels, this range is selected and summarized in Table 5.3.

Table 5.3. Range of Values for Gilbert Model Parameters

Parameter	Minimum value	Maximum value
P	0.000001	0.01
p	0.001	0.1
H	0.2	0.8

Based on the ranges in Table 5.3 and the performance parameters considered in this study, a pilot study was conducted to select applicable values of P,p, and h for further simulation. The selection process considers the percentage of error free packets that are transmitted by the simulation model under different error conditions. For example, it is quite obvious that an error rate of 98.9 percent of packets renders this simulation model useless. The results of this pilot study are given in Table 5.4.

Table 5.4. Results of Pilot Study for Gilbert Model Parameters

p	P	h	Packet errors (%)
0.001	0.000001	0.2	2.8
0.001	0.000001	0.8	2.3
0.001	0.0001	0.2	98.9
0.001	0.0001	0.8	97.4
0.001	0.01	0.2	100
0.001	0.01	0.8	100
0.01	0.000001	0.2	0.463
0.01	0.000001	0.8	1.7
0.01	0.0001	0.2	96.6
0.01	0.0001	0.8	98.1
0.01	0.01	0.2	100
0.01	0.01	0.8	100
0.1	0.000001	0.2	1.76
0.1	0.000001	0.8	1.25
0.1	0.0001	0.2	94.2
0.1	0.0001	0.8	89.5
0.1	0.01	0.2	100
0.1	0.01	0.8	100
0.001	0.00001	0.2	21.1
0.001	0.00001	0.8	20.3
0.01	0.00001	0.2	19.7
0.01	0.00001	0.8	19.8
0.1	0.00001	0.2	18
0.1	0.00001	0.8	13.3

From the pilot study, four values for each parameter are selected. These variables are selected from four different regions of the above table. The selection process is also based on error percentages that allow the simulation model to function properly. Error percentages above 50 percent are not considered as functional values for this simulation model. The four values chosen are listed in Table 5.5 below.

Table 5.5. Selected Gilbert Model Parameters

p	P	h	Error percent
0.001	0.000001	0.2	2.8
0.1	0.000001	0.8	1.25
0.001	0.00001	0.8	20.3
0.1	0.00001	0.8	97.4

5.3. Performance Results

This section describes the results of performance experiments using the simulation model. Appendix A contains tabular data associated with the figures presented in this chapter.

To gain a better understanding on how the system behaves under different loads and numbers of station, the error-free model is considered first. The graph

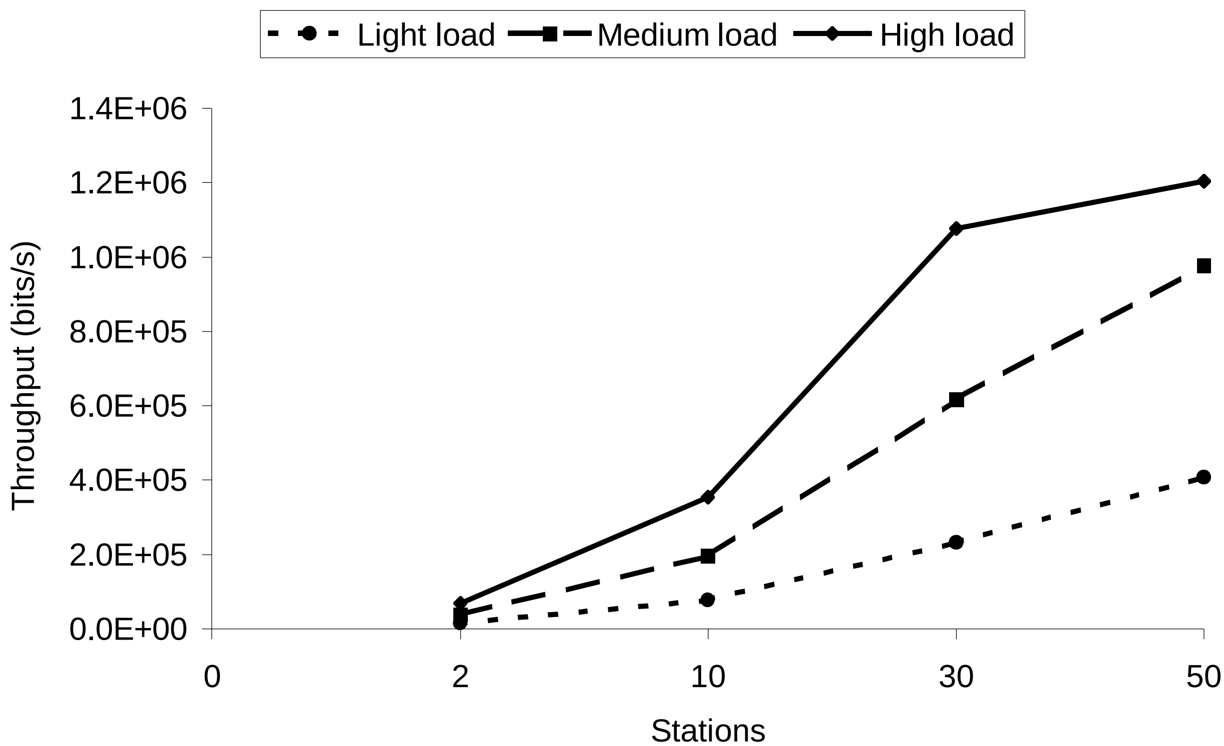

Figure 5.1. Throughput vs. number of stations.

in Figure 5.1 gives the throughput for a varying number of stations. Throughput is defined as the rate at which requests are serviced by the system. The duration for this simulation is 300 seconds.

The graph shows throughput for three separate load conditions. The solid line indicates throughput under high load conditions of 36 kbps per station, the dashed line is for a medium load condition of 20 kbps per station, and the dotted line is for low load conditions of 8 kbps per station. It is clear that under very low utilization of the medium, which is represented by 2 stations, there is not a large difference between different load factors. At 10 stations, a positive slope change can be observed for all three curves, which is primarily due to higher load factors on the network. It is also clear that for low and medium load factors the network provides as much throughput as needed to accommodate the added stations. However, it should be noted that at 30 stations the high load curve undergoes a negative slope change, indicating network saturation. This is primarily due to a high utilization of the medium by the stations. The network cannot easily accommodate the high demand for medium access by a large number of stations, and station queues begin to grow.

Figure 5.2 specifies the average queue lengths vs. number of stations. The model is simulated for 300 seconds with the same load factors as discussed previously. The average queue length for the model is defined as the sum of all individual queue lengths for each station. Because of the relatively large queue lengths under heavy load conditions, Figure 5.2 is plotted using a logarithmic scale. In this Figure, at 30 stations the curve for heavy load experiences a positive slope change. This is primarily due to medium congestion caused by a larger number of stations with heavier load factors attempting to access the medium. As a result of these constraints, and more collisions caused by heavier network traffic, the queue length for the network increases exponentially.

Figure 5.3 depicts the effect of the number of stations on packet collisions for different loads. The numbers of collisions are represented by the highest number of collisions observed in any individual station for a particular number of stations and load factors. The duration for this simulation model is 300 seconds.

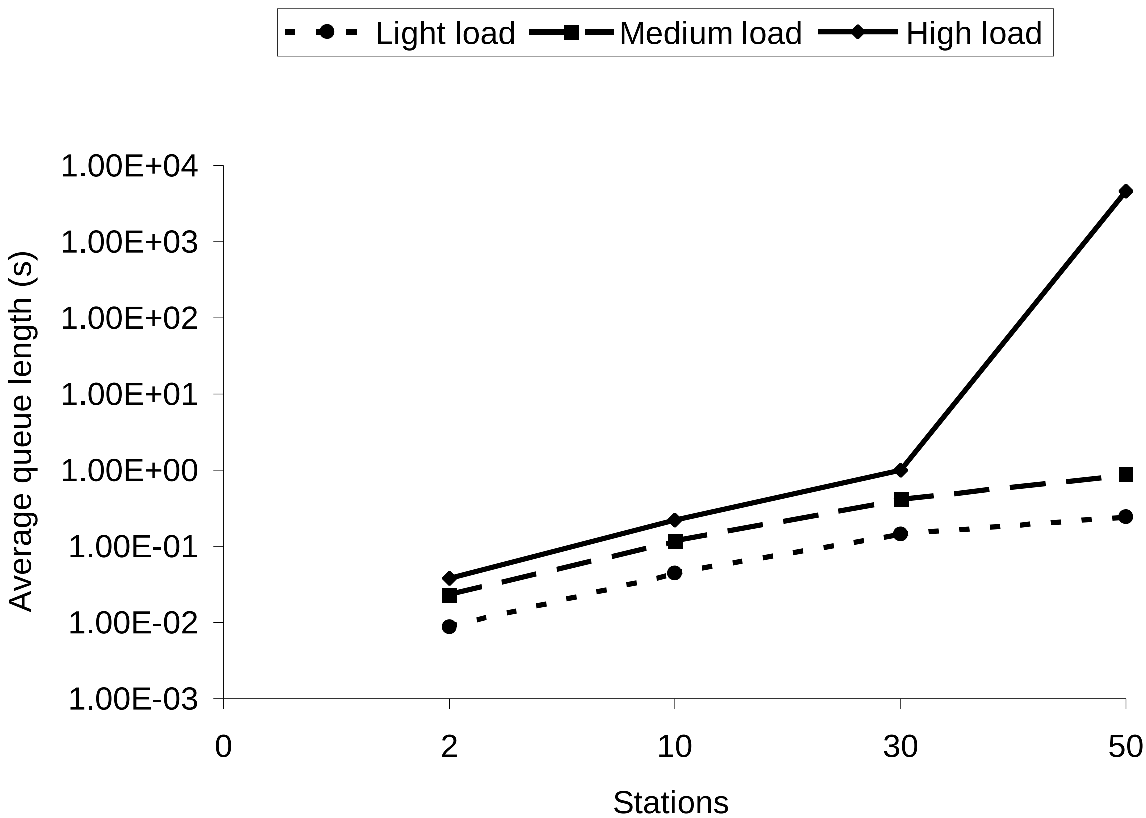

Figure 5.2. Average queue length vs. number of stations.

As demonstrated previously by other performance measures, the network accommodates low and medium load factors with a minimum number of collisions. However, there are several sharp positive inflection points at 10 and 30 stations for the high load curve. The main culprit for these is the heavy demand that is put on the network by a larger number of stations with heavier loads. Furthermore, collisions create a snowball since they lead to the retransmission of the same packet, thus adding to the load. The maximum number of collisions observed in Figure 5.3 was 619. However, the vertical axis

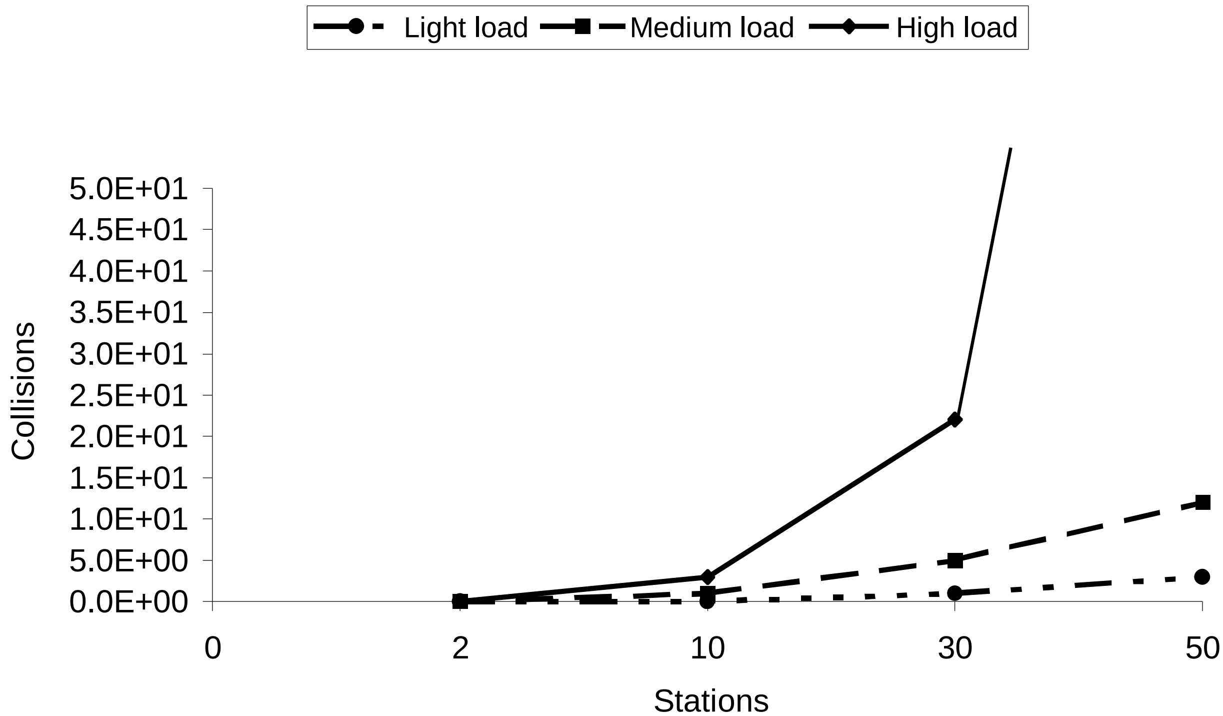

Figure 5.3. Collisions vs. Number of stations

has been adjusted to show the number of collisions for low and medium loads.

5.4. Effect of Errors

The error free model demonstrated how particular performance measures vary under different conditions. In this section, the effects of channel errors are added to the simulation model to understand how the same particular performance measures are affected. Appendix B contains tabular results for the following two simulation experiments. The duration for the simulations is 300 seconds.

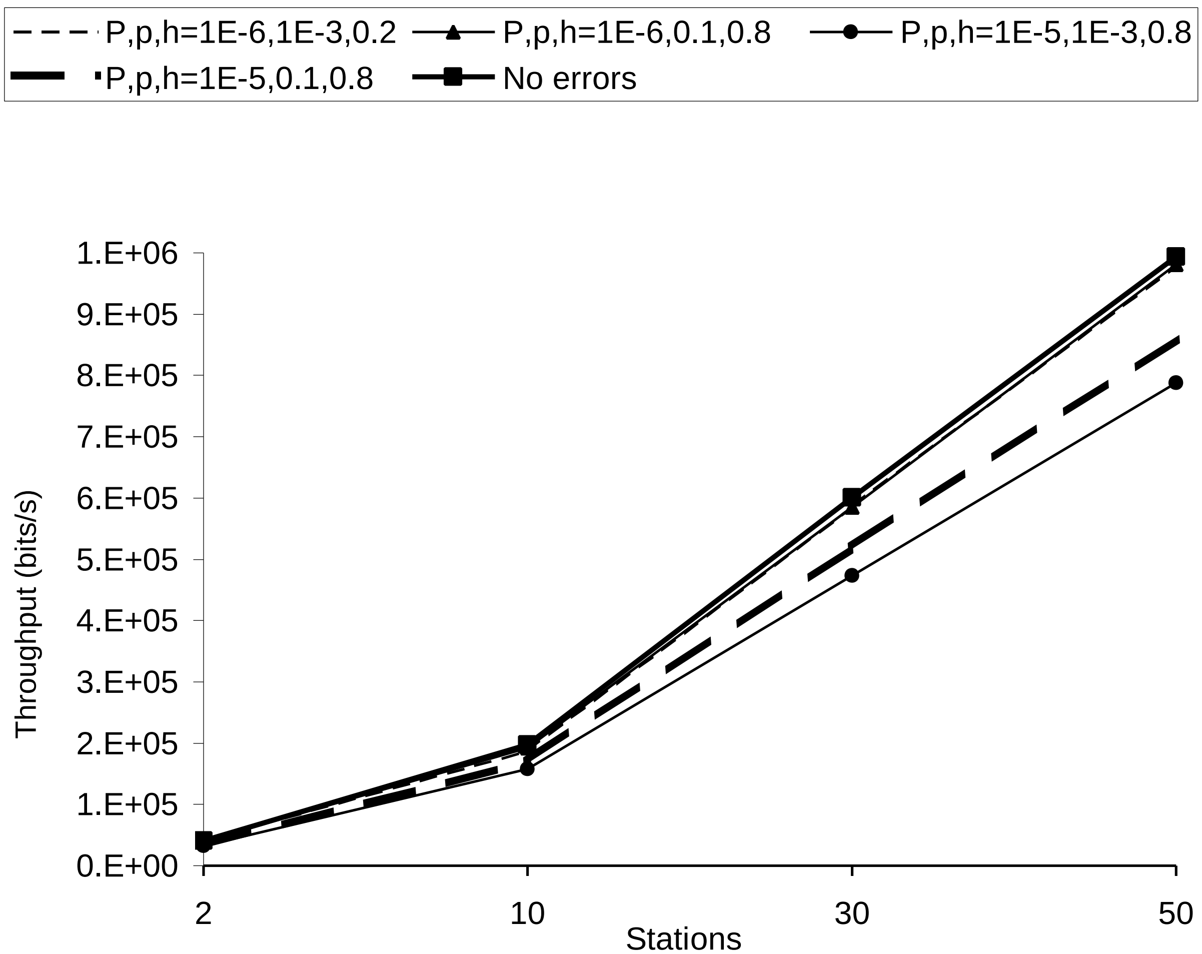

Figure 5.4. BER effect on throughput for varying numbers of stations.

Figure 5.4 depicts the effect of Gilbert model error parameters on throughput for a varying number of stations under a constant load of 20 kbps per station. For a small number of stations, throughput is slightly reduced by errors. However, as the number of stations increases, the effects become more pronounced. It is also important to note the effect of varying the value of p under low BER conditions and high BER conditions as represented by middle two curves and bottom two curves, respectively. Under low BER conditions (P=1E-6, .001<p<0.1, h=0.8), the two curves are almost superimposed. However, under slightly higher BER conditions represented by (P=1E-5, .001<p<0.1,h=0.8), parameter p has a much larger affect on system throughput.

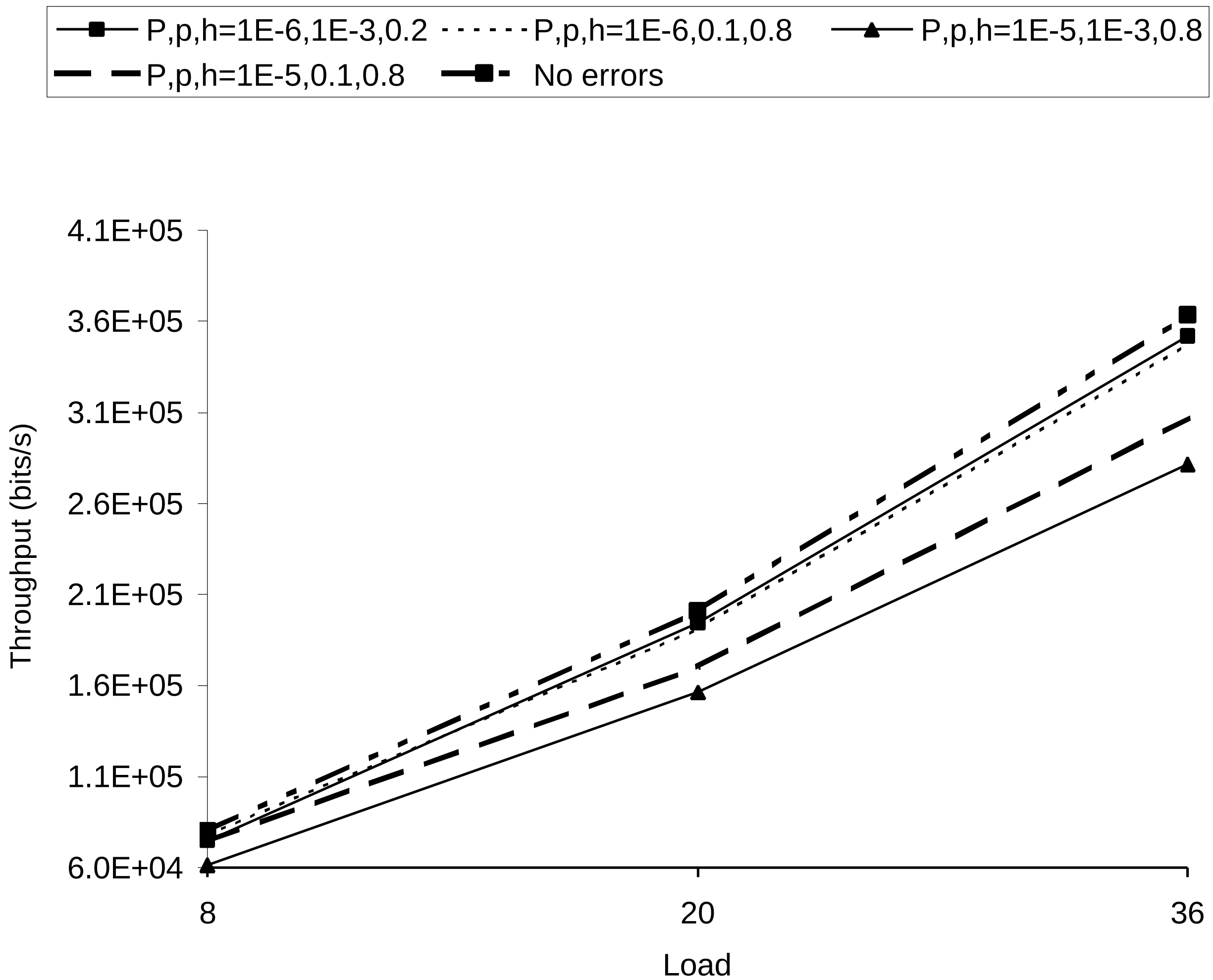

Figure 5.5. BER effect on throughput vs. load

In Figure 5.5, load varies while the number of stations for the simulation model remains constant at 10 stations. The three load levels investigated are low, medium and high at 8, 20 and 36 kbps, respectively. The Gilbert model parameters are the same as in Figure 5.4.

There is not a great difference that can be observed between the error-free curve and the low BER curves represented by the middle two curves.

5.5. Effect of Packet Size

The maximum length of an IEEE 802.11 packet size is 18,704 bits, which has been utilized in the results of previous sections. However, higher level applications on each station do not necessarily release data in maximum packet size segments. In this section, the size of IEEE 802.11 packets is reduced by 50 percent, to 9,352 bits. Appendix C contains the tabular data for the following two experiments. The simulations were for 300 seconds.

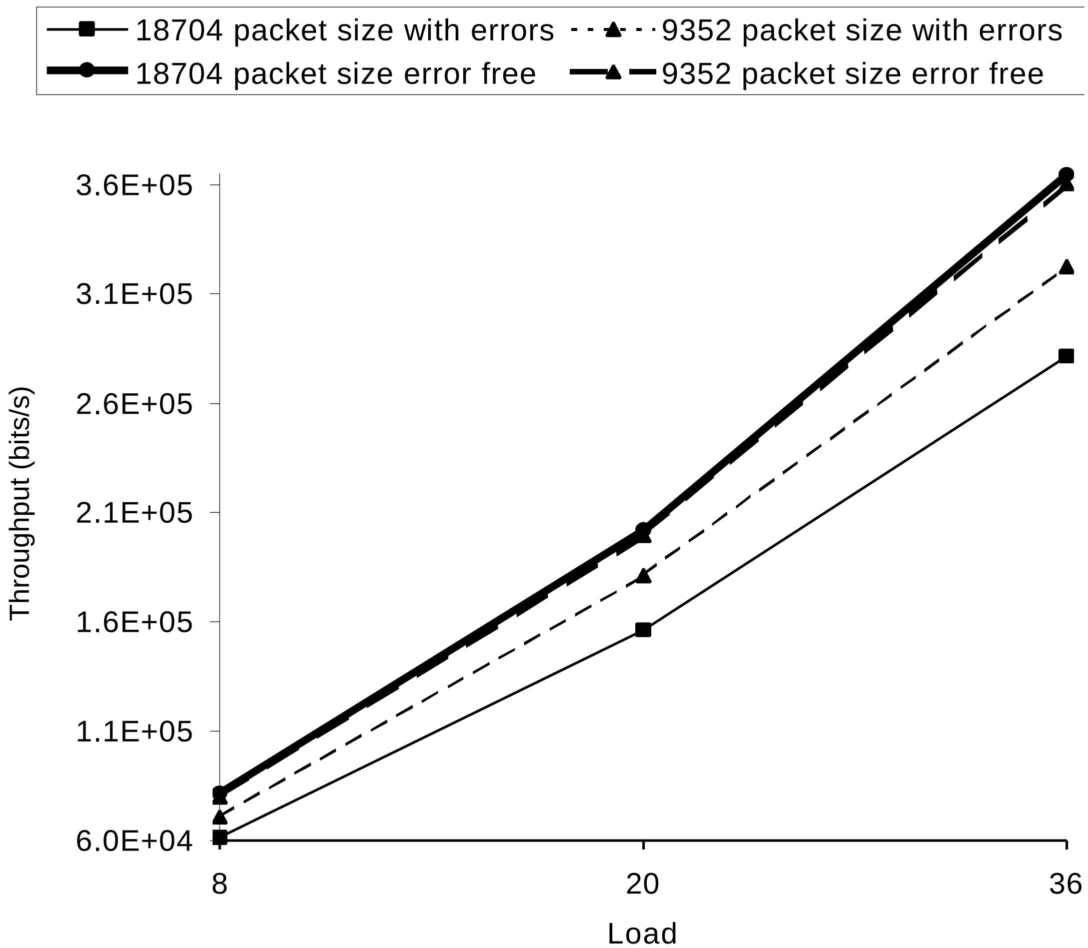

Figure 5.6. Throughput for varying packet size vs. load.

Figure 5.6 depicts the effect of two different packet sizes under different load conditions. The Gilbert model parameters P,p and h are set to achieve a high BER at 1E-5, 1E-3 and 0.8, respectively for the lower two curves. The higher two curves are error-free simulation runs. The number of stations is a constant 10 stations.

In Figure 5.6 the curves with BERs clearly show the increase in throughput as a result of reduction in packet size. The increase in throughput is not linear. As the offered load is increased, smaller packet sizes are more responsive to system demands, hence increased throughput. Once the effects of errors are removed, the longer packet sizes show higher throughputs than the shorter

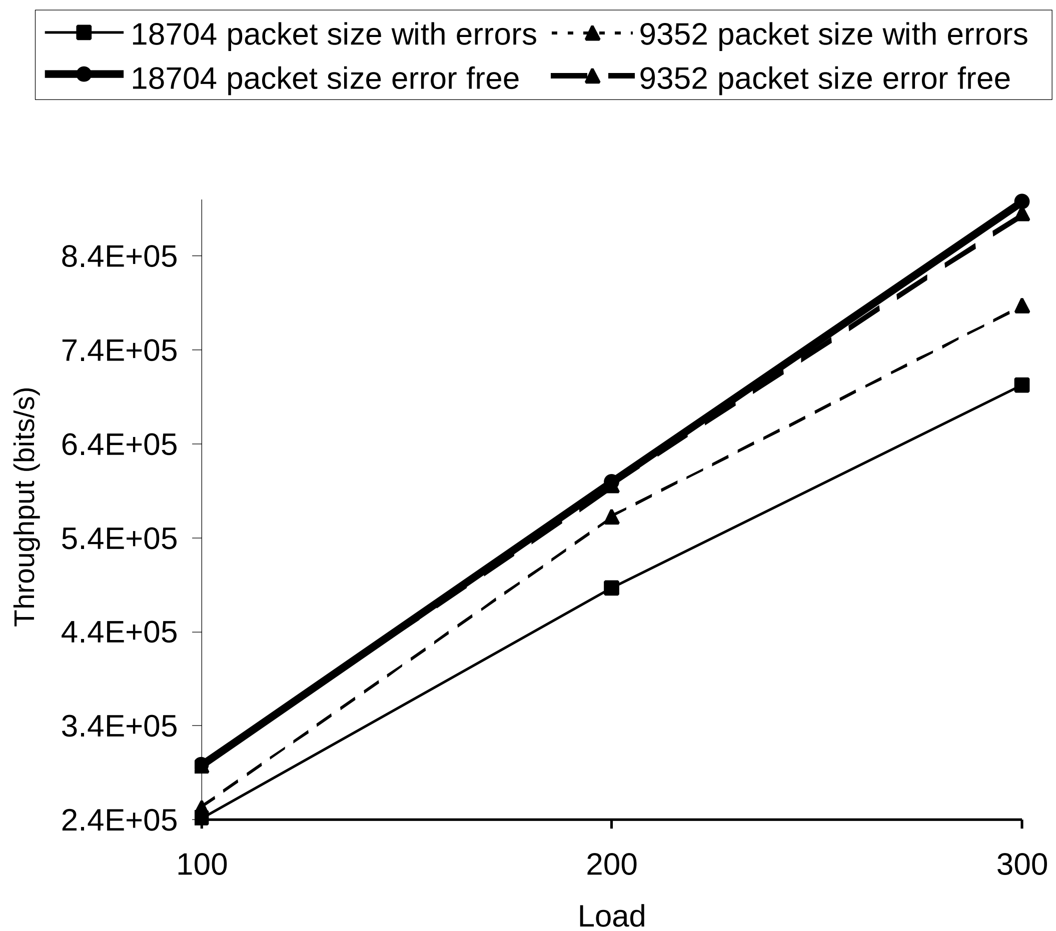

Figure 5.7. Throughput for varying packet size vs. load.

packets as expected. The longer data frames have more data and less overhead bits, thus every packet carries more information causing a higher network throughput.

The responsiveness of the system can be simulated when a few stations have bursts of data packets for transmission. Figure 5.7 shows results for 3 stations with very heavy offered load. The Gilbert model parameters P,p and h are set at 1E-5, 1E-3 and 0.8, respectively, to achieve a high BER. The error-free case is also included in Figure 5.7.

The offered load for this simulation is raised to 100, 200 and 300 kbps, which is representative of bursts of data transmissions by a few stations. The lower dashed line in Figure 5.7 represents smaller packets with the possibility of errors. It is again clear that system throughput is increased for these packets. However, once the possibility of errors is removed, illustrated by the higher two curves in Figure 5.7, the longer packets outperform the shorter ones as measured by throughput. As the load increases, the throughput gap widens between the longer and shorter error-free packets. This is again the result of longer packets carrying more data bits as a percentage of total bits transmitted versus the shorter error-free packets.

5.6. Summary

The results of the previous sections of this chapter indicate that the load offered by stations and the number of stations have a large effect on the performance of the simulation model. Network throughput increases with an increase in load and number of stations until the network nears its saturation point. Once saturation occurs, network throughput levels off. Queue lengths and collisions remain at reasonable levels until network saturation occurs. At that point both of these performance measures start to increase at a rapid rate. It is important to know at what point network saturation occurs. Because of the rapid rise of the queue lengths near the saturation point, some stations may discard data packets due to an overflow of their queues. If this situation persists, it can lead into an interruption of the services provided by the wireless network. Errors in the communications channel lower the network throughput by anywhere from 1.25 to 20 percent. The lowering of the throughput by the errors is not a uniform phenomenon, thus indicating a coupling effect between error bursts and the access scheme used by the IEEE 802.11 protocol.

Varying the size of data packets further influences network throughput. Short packets result in higher network throughput when the network is operating with errors in the communications channel. This is the result of less information being re-transmitted due to corrupted bits. Although, with no or few errors, longer frames generate higher network throughput. This result can be used to optimize the performance of any given wireless network based on the error characteristics of the communication channel. The maximum size of the packets can be varied dynamically based on the error characteristics of the communication channel to produce the best performance results for any given BER condition.

6. Conclusions

6.1. Summary

In the previous sections, this research described the MAC portion of the IEEE 802.11 wireless protocol and bit errors in a digital communications channel. A computer simulation model was developed to understand the behavior of the protocol through performance measures, including throughput, queue lengths, and collisions.

The CSMA/CA scheme used by the IEEE 802.11 protocol uses a rotating backoff window in order to avoid collisions between packets. Once a station has access to the medium, it is cleared to transmit its packet. Once a packet is transmitted, it can be affected by the binary noise in a digital communications channel. The Gilbert model utilizes a Markov chain with two states to generate bursts of errors in the digital communications channel. Parameters P,p and h define the operation of the Gilbert model.

It is clear from the simulation results that, the offered load and the number of stations in a network do not equally influence the protocol. As the number of stations is increased, it presents a larger burden on the network as opposed to an increase in just offered load by fewer stations. This can be explained by the increased number of collisions when there are a large number of stations. It is impossible for two packets from the same station to collide with one another, however it is probable that two packets from two different stations collide with one another if there are a large number of stations.

The bursts of errors in a binary channel have a negative effect on performance. Throughput is reduced linearly by the errors as the offered load increases. The reduction in throughput is more pronounced for a large number of stations and constant BER than in the error-free case. Therefore, bit errors do have an effect on the way the CSMA/CA portion of the protocol operates.

Finally, the results of the simulation model indicate how packet sizes influence the throughput of the network. Smaller packets are more efficient and they utilize the medium more often than larger packets (Appendix C contains the utilization results for small and large packet sizes). Furthermore, once a collision occurs, the medium is occupied less by the transmission of a corrupted packet, thus raising the responsiveness of the CSMA/CA portion of the protocol. The packet error rate is also reduced slightly, since shorter packet sizes have a lower probability of containing corrupted bits. However, once the errors are removed, performance as measured by throughput is higher for longer packets than shorter packets. Longer packets contain more data bits as a percentage of total packet size than shorter packets. Therefore, barring errors in the communications channel, stations can transmit more information using longer packets without suffering a penalty for corrupted data bits than shorter frames.

6.2. Future Work

There are several important areas for future work. One of the assumptions made in this research is that all stations are notified within 1-μs of a packet transmission. If another transmission begins within a 1-μs window, then a collision occurs. A variable time unit can be assigned to each station based on its proximity, thus reducing the possibility of unnecessary collisions. At low offered load and numbers of stations, a variable proximity factor may not enhance the performance of the system, although under heavy load conditions offered by a large number of stations, collisions play a large role in reducing network performance.

Another area for future work is to extract the error parameters P,p and h from existing IEEE 802.11 trace data. The CSMA/CA portion of the protocol can be simulated accurately, thus the difference between the error-free simulation and actual trace data is attributed to bit errors. A separate Gilbert model simulation model can be devised to approximate the values of P,p and h.

Appendix A

The following three tables contain results for the error-free simulation model described in Section 5.3.

Table A.1 Light Load at 8 kbps

Number of stations	Throughput Kbps	Avg. queue Length (s)	Utilization (%)	Collisions
2	15719	0.0086	0.786	0
10	79228	0.0445	3.96	0
30	244785	0.1456	12.2	1
50	392229	0.2444	19.6	3

Table A.2 Medium Load at 20 kbps

Number of stations	Throughput Kbps	Avg. queue Length (s)	Utilization (%)	Collisions
2	41272	0.0229	2.06	0
10	197406	0.1159	9.87	1
30	600583	0.411	30	5
50	993736	0.876	49.6	12

Table A.3 High Load at 36 kbps

Number of stations	Throughput kbps	Avg. queue Length (s)	Utilization (%)	Collisions
2	68137	0.038	3.4	0
10	358298	0.222	17.9	3
30	1057940	1	52.8	22
50	1196851	4569	60	619

Appendix B

The following tables contain results for the simulation described in Section 5.4.

Table B.1 Constant Load at 20 kbps (BER parameters for this table are: P=1E-6, p=1E-3, h=0.2)

Number of stations	Throughput kbps	Avg. queue Length (s)	Utilization (%)
2	40293	0.907	2.01
10	186923	0.1108	9.34
30	585791	0.407	29.2
50	978696	4569	48.93

Table B.2 Constant Load at 20 kbps (BER parameters for this table are: P=1E-6, p=0.1, h=0.8)

Number of stations	Throughput Kbps	Avg. queue Length (s)	Utilization (%)
2	40009	0.0226	2
10	192085	0.111	9.6
30	584992	0.405	29.24
50	981706	0.897	49.8

Table B.3 Constant Load at 20 kbps (BER parameters for this table are: P=1E-5, p=1E-3, h=0.8)

Number of stations	Throughput Kbps	Avg. queue Length (s)	Utilization (%)
2	32649	0.023	1.6
10	157350	0.116	7.86
30	473770	0.404	23.68
50	787492	0.86	39.37

Table B.4 Constant Load at 20 kbps (BER parameters for this table are: P=1E-5, p=0.1, h=0.8)

Number of stations	Throughput Kbps	Avg. queue Length (s)	Utilization (%)
2	33768	0.022	1.69
10	169894	0.16	8.49
30	520837	0.406	26.04
50	860315	0.964	43

For error-free results, refer to Table A.2.

Table B.5 Constant Number of Stations at 10 (BER parameters for this table are: P=1E-6, p=1E-3, h=0.2)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	74867	0.043	3.74
20	194307	0.1216	10.05
36	352067	0.223	17.6

Table B.6 Constant Number of Stations at 10 (BER parameters for this table are: P=1E-6, p=0.1, h=0.8)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	77892	0.044	3.9
20	191276	0.1127	9.56
36	347496	0.218	17.37

Table B.7 Constant Number of Stations at 10 (BER parameters for this table are: P=1E-5, p=1E-3, h=0.8)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	61329	0.045	3.1
20	156362	0.116	7.8
36	281427	0.220	14.07

Table B.8 Constant Number of Stations at 10 (BER parameters for this table are: P=1E-5, p=0.1, h=0.8)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	74181	0.048	3.7
20	170146	0.115	8.5
36	307110	0.217	15.35

Table B.9 Constant Number of Stations at 10 (Error-free)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	79906	0.044	3.99
20	201041	0.113	9.72
36	363599	0.224	18.18

Appendix C

The following tables contain results for the simulation described in Section 5.5.

Table C.1 Packet size for this table is 18,704 bits. (BER parameters for this table are: P=1E-5, p=1E-3, h=0.8)

Offered load (kbps)	Throughput Kbps	Avg. queue Length (s)	Utilization (%)
8	61329	0.045	3.1
20	156362	0.116	7.8
36	281427	0.220	14.07

Table C.2 Packet size for this table is 9,352 bits. (BER parameters for this table are: P=1E-5, p=1E-3, h=0.8)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	70762	0.048	3.54
20	181302	0.128	9.06
36	322424	0.244	16.12

Table C.3 Packet size for this table is 18,704 bits. (Error-free)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	81588	0.046	4.11
20	201966	0.118	10.20
36	364476	0.225	18.38

Table C.4 Packet size for this table is 9,352 bits. (Error-free)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
8	79985	0.049	4.04
20	199633	0.013	10.06
36	360360	0.241	17.9

Table C.5 Packet size for this table is 18,704 bits. (BER parameters for this table are: P=1E-5, p=1E-3, h=0.8)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
100	241722	0.186	12.08
200	486412	0.413	24.3
300	702506	0.710	35.12

Table C.6 Packet size for this table is 9,352 bits. (BER parameters for this table are: P=1E-5, p=1E-3, h=0.8)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
100	252547	0.183	12.67
200	562316	0.433	26.3
300	787312	0.800	39.36

Table C.7 Packet size for this table is 18,704 bits. (Error-free)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
100	298466	0.179	14.92
200	598947	0.406	29.90
300	897892	0.734	44.89

Table C.8 Packet size for this table is 9,352 bits. (Error-free)

Offered load (kbps)	Throughput kbps	Avg. queue Length (s)	Utilization (%)
100	297588	0.197	14.87
200	595689	0.449	29.78
300	885852	0.855	44.29

Appendix D

Glossary

ACK A positive response returned from a receiver to the sender indicating success.

***ad hoc* network** A stand-alone wireless network without a base station controller. Formed by a small group.

exclusive-OR (XOR) A binary operation between two binary digits. The operation is the same as binary addition without a carry bit.

Bit 1	Bit 2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Frame The unit of information transferred across a data link. Usually, there are control frames for link management and information frames for the transfer of data.

Local Area Network (LAN) A computer network system confined to a local area such as a building. LANs are usually used to provide all computing services such as peripheral sharing, file sharing and Internet access for a limited number of computing devices.

Medium A link that provides a basic building block to support the transmission of information signals.

Medium Access Control (MAC) A data link control function that controls the use of a network medium.

Modulation The process of translating the baseband digital signal to a suitable analog form.

NAK A negative response returned from a receiver to the sender indicating failure.

OSI (Open System Interconnect) A standard specifying an open system capable of enabling the communications between different systems. OSI has seven distinct layers. These layers provide the functions necessary for two applications processes to communicate.

Packet A group of bits transmitted as a whole on a network.

Wide Area Network (WAN) A type of network that provide information transport between LANs and users over a wide geographical area.

Wireless LAN A LAN that uses either radio or infrared as the transmission medium.

References

[1] J. Geier, *Wireless Networking Handbook*, New Riders Publishing, Indianapolis, IN, 1996.

[2] E. N. Gilbert, "Capacity of a burst-noise channel*," Bell Systems Technical Journal*, vol. 39, pp. 1253-1266, Sept. 1960.

[3] R. P. Wenig, *Wireless LANs*, Academic Press, Inc., Chestnut Hill, MA, 1996.

[4] P. T. Davis and C. R. McGuffin, *Wireless Local Area Networks: Technology, Issues, and Strategies*, McGraw-Hill, New York, NY, 1994.

[5] F. Halsall, *Data Communications, Computer Networks and Open Systems*, Fourth Edition, Addison-Wesley, Harlow, England, 1996.

[6] J. Weinmiller, H. Woesner and A. Wolisz, "Analyzing and Improving the IEEE 802.11-MAC Protocol for Wireless LANs", Electrical Engineering Department, Technical University Berlin, Berlin, Germany.

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

[8] M. H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, The MIT Press, Cambridge, Mass, 1987.

[9] B. E. Mullins , “CATER: An Opportunistic Medium Access Control Protocol for Wireless LANs”, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Va, June 1997.

VITA

FARHOOD MOSLEHI

Date of Birth:

June 7, 1963

Education:

B.S. Engineering Science & Mechanics (ESM), Virginia Tech. (January 1993)

M.S. Systems Engineering (Data Networks), Virginia Tech. (August 1997)

Employment:

1993-Present.
Computer Systems Engineer. College of Agriculture. Virginia Tech, Blacksburg, VA. 24061

1988-1993
Electrical Engineer. Air Pollution Laboratory. Virginia Tech, Blacksburg, VA. 24061

1986-1988
Laboratory Specialist. Air Pollution Laboratory. Virginia Tech, Blacksburg, VA. 24061

Publications and Presentations:

Wireless Connectivity (Book Chapter) as part of the " World Wide Web Beyond the Basics " book by Prentice Hall. (Fall 1997)

Automation of Continuous Stirred Reactors Using a PC-Based System. US. Forest Service 1993. (Presentation)

Research Interests:

Computer languages, specifically Object Oriented programming and design. Computer and Data Networks, design and analysis. Computer and Network performance models.