Throughput and Delay Characteristics for a Slow-Frequency Hopped Aircraft-to-Aircraft Packet Radio Network

THESIS

Scott Russell Harrison
Captain, USAF

AFIT/GCS/ENG/90M-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GCS/ENG/90M-01

Throughput and Delay Characteristics for a Slow-Frequency
Hopped Aircraft-to-Aircraft Packet Radio Network

THESIS

Scott Russell Harrison
Captain, USAF

AFIT/GCS/ENG/90M-01

Approved for public release; distribution unlimited

AFIT/GCS/ENG/90M-01

Throughput and Delay Characteristics for a Slow-Frequency Hopped Aircraft-to-Aircraft Packet Radio Network

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science (Computer Systems)

Scott Russell Harrison, B.S.
Captain, USAF

March, 1990

Preface

The purpose of this thesis effort was to develop a mathematical representation of a frequency hopping, aircraft-to-aircraft, packet radio network, the goal being to obtain baseline figures for the bandwidth required and delay for such a system. To arrive at these goals a veritable plethora of design choices and analyses were faced.

Several existing models matched the situation at hand fairly closely, and, with minor modifications, they were applied. In each case, however, the assumptions of the original model had to be stretched or violated. Violating these assumptions was due mainly to the small population of the aircraft-to-aircraft network and that message arrivals at each of the aircraft would not be Poisson in nature. Even with these limitations, the results obtained closely match or exceed the performance parameters of currently operational systems. This was somewhat of a surprise since the number of frequency hopped channels was increased to improve on current LPI and anti-jam capabilities. Although this design decision would normally increase the bandwidth required, this wasn't demonstrated by these models.

Throughout the process of developing and writing this thesis, I have received a great deal of support from others. Above all, I'd like to thank my wife, Gay, her patience and understanding were limitless and she often provided the gentle nudging and new perspectives I needed to keep going. My classmates were an invaluable source of good humor and encouragement. Lastly, I'd like to express my thanks to my Thesis Advisor, Lt. Col. C. R. Bisbee, and the AFIT Faculty members who made this effort possible.

Scott Russell Harrison

Table of Contents

List of Figures

Figure | Page

List of Tables

AFIT/GCS/ENG/90M-01

Abstract

The current lack of data transfer capability between tactical aircraft results in decreased mission effectiveness or the need to equip every aircraft with all the sensors and data processing equipment to perform each task expected of them. In the Air Force, these developments established the need for aircraft-to-aircraft data communications so that navigation, threat and targeting information could be shared within a flight and so that digital voice communications could be maintained as well. Data rates within a flight of fighter aircraft to support these requirements will need to be between 100 and 250 Kilobits Per Second (KBPS) and between 2.4 KBPS and 16 KBPS between the flight and a command and control aircraft. Other sources place the former figure as high as 512 KBPS. In order to increase the security of these radio channels, against both jamming and intrusion, anti-jam and low probability of interception (LPI) techniques must be used. One method of achieving these capabilities is to use spread spectrum techniques such as frequency hopping. The Air Force is currently concentrating its efforts around the Joint Tactical Information Distribution System (JTIDS). This system is, however, lacking in data rate capacity, LPI capability, and, while it is jam-resistant, it does not have anti-jam capabilities. The purpose of this thesis effort was to develop a mathematical representation of a frequency hopping, aircraft-to-aircraft, packet radio network, in order to obtain baseline figures for the bandwidth required and delay for such a system. Several existing models closely matched the situation at hand, and, with minor modifications, they were applied. In each case, however, the assumptions of the original model had to be stretched or violated. Even with these limitations, the results obtained closely match or exceed the performance parameters of current operational systems.

Throughput and Delay Characteristics for a Slow-Frequency Hopped Aircraft-to-Aircraft Packet Radio Network

I. Introduction

Background

Since the Vietnam War, the battlefield has become increasingly dependent on computers and digital communications. From smart bombs to computerized navigation systems to digital voice communications, computer systems have been deployed in order to gain an edge in the accuracy, speed, and security of weapon systems. However, to this day, with few exceptions, aircraft are unable to share data gained from increasingly sophisticated systems. Data must be shared through verbal communications and thus is limited to a very low rate and to a content that is subject to interpretation. This lack of data transfer capability results in decreased mission effectiveness or the need to equip every aircraft with all the sensors and data processing equipment to perform each task expected of them. In the Air Force, these developments established the need for aircraft-to-aircraft data communications so that navigation, threat and targeting information could be shared within a flight and so that digital voice communications could be maintained as well. Digital voice communications are of interest because of the ease of encryption and the high quality of voice reproduction. One example, where the need for shared data is demonstrated, is a flight of aircraft involved in a low-level mission using terrain following and terrain avoidance radar (TF/TA). Since present tactical radar cannot carry out TF/TA and aerial search simultaneously, the flight would be oblivious to airborne threats. If, however, data could be shared between aircraft, one aircraft could carry out the TF/TA requirement while another aircraft provided aerial search.

Mr. Walt Hartman, who works for the Air Force at the Air Force Wright Aeronautical Labs, Wright-Patterson AFB, Ohio developing the next generation of aircraft communications equipment,

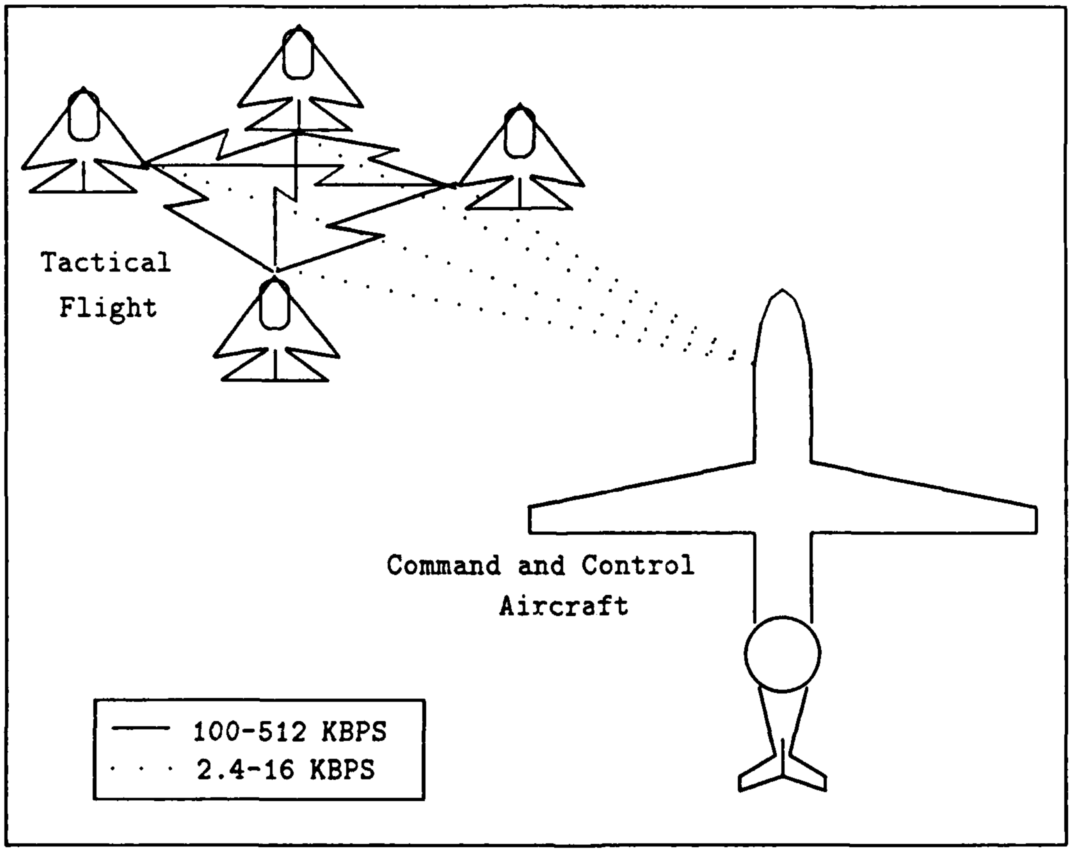

Figure 1. Aircraft-to-Aircraft Communications

estimates that data rates within a flight of fighter aircraft to support these requirements will need to be between 100 and 250 Kilobits Per Second (KBPS) and between 2.4 KBPS and 16 KBPS between the flight and a command and control aircraft (17) . Other sources place the former figure as high as 512 KBPS (39), see Figure 1.

In order to increase the security of these radio channels, against both jamming and intrusion, anti-jam and low probability of interception (LPI) techniques must be used. One method of achieving these capabilities is to use spread spectrum techniques. This thesis will focus on the spread spectrum technique known as slow frequency hopping. This technique involves frequency hopping where the hopping rate is less than the data rate.

Currently, aircraft-to-aircraft data communications systems are rather limited. The Air Force is currently concentrating its efforts around the Joint Tactical Information Distribution System (JTIDS). This system is, however, lacking in data rate capacity, LPI capability, and the ability to efficiently integrate voice communications. The proposals to overcome the shortcomings of JTIDS involve the use of airborne Packet Radio Networks (PRNs). The Air Force is not alone in its interest in PRNs, Michael Pursley of the University of Illinois recently stated:

> Although a number of different spread-spectrum transmission protocols have been discussed in the literature, there are not many performance results available. The design and analysis of these protocols is still an active area of basic research and several of the more important problems remain unsolved. (26:118)

The problems associated with the combination of PRNs and slow frequency hopping/spread spectrum techniques, are the justification for this research.

Problem Statement

The specific problem to be addressed by this research is to propose a viable set of parameters for an airborne PRN that will support the required data rates, anti-jam/LPI capabilities, and voice communications. The projected benefit from this research is a framework for the development of an aircraft-to-aircraft data communications system or simply a starting point for students doing research in this area in the future.

Summary of Current Knowledge

As stated previously by Michael Pursley, very little performance data exists on the type of systems in question. This is undoubtably due to the fact that thus far there have been very few applications of slow frequency hopping PRNs. However, there are a couple of applications that bear mention along with the contributions of current theoretical work.

Applications. Two applications of slow frequency hopping systems that should be mentioned here are the Air Force's *Have Quick* radio system and the Low-Cost Packet Radio (LPR). *Have Quick* is a analog voice system that uses slow frequency hopping to prevent enemy exploitation of aircraft-to- aircraft communications. While analog voice is not sent in packets and thus does not constitute a PRN, this system is important because it was developed specifically to give tactical aircraft anti-jam/LPI capabilities that the U.S.A.F. experience in Vietnam and Israeli experience in the 1973 Mideast War demonstrated were necessary. It is interesting to note that the frequency hopping system employed by *Have Quick* requires the synchronization of all system clocks prior to a mission. As the mission continues, the clocks drift relative to one another and propagation delay varies as the distance between the aircraft varies. This causes parts of the voice signal to drop-out. This is tolerable in a voice system because the human ear tends to integrate out these imperfections, a link carrying digital data is not as forgiving.

The LPR is mentioned here because it is the only system thus far to implement slow frequency hopping for a data network. Although the LPR's primary mode of operation uses a spread spectrum technique known as Direct Sequence Pseudo–Noise, it allows the user to use slow frequency hopping on a packet-by- packet basis (8).

Theory. The current theoretical work in this area is being done mainly in the area of error probabilities for packets transmitted over slow frequency hopped channels and the throughput over these channels. This work will contribute to this thesis effort by allowing basic performance parameters to be derived. Other work includes studies of various channel access protocols to limit packet collisions on slow frequency hopped channels as well as work determining better ways to encode data for transmission over these channels. These methods of encoding data hope to eliminate or minimize the effects of noise, jamming, and self-interference by other users of the net, so that data dropouts can be recovered at the receiver. So, while the direct benefit of other's research might be limited, their work provides a framework from whence to start. A more complete summary of the currently available literature in this field follows in Chapter II, Review of Literature.

Limitations and Scope of Research

This research will be limited to the analysis of a slow frequency hopping PRN, or sub-net, that will exist between a tactical flight consisting of four tactical aircraft and a command and control aircraft. This network must support the channel capacities mentioned before: 512 KBPS within the flight and 16 KBPS between the flight and the command and control aircraft (See Figure 1). Further, the research will be limited to only those issues of design outlined in the next section.

Approach to the Research

The approach to this research will be guided by the design issues outlined by Barry M. Liener, Donald L. Nielson, and Fouad A. Tobagi in their article, *Issues in Packet Radio Network Design* (21). Although these gentlemen outline 36 design issues, or questions, that must be answered in designing a PRN, an effort of that magnitude is not possible here. However, by choosing several issues to address in this research, a foundation for other work in this area can be laid.

The following issues of PRN design will be specifically addressed by this research:

1. Given the type of terrain in which the network is to operate, the user's locations, and their requirements in terms of traffic, mobility, anti-jamming capabilities, etc., select:
   (a) The radio frequency and RF bandwidth.
   (b) The signaling, encoding, and modulation schemes.
   (c) The network topology.
2. How to allocate the bandwidth in time and space to the nodes in the network.
3. What channel access protocol and code assignment algorithm should be used.

These issues cover the basic design decisions that must be made in a PRN for the Physical and Data Link layers, as well as the Medium Access sublayer, of the ISO's OSI Reference Model. Each of these issues will be addressed by applying the techniques and/or arguments presented in recent literature on slow frequency hopping PRNs. While the choice of using slow frequency hopping is actually a design decision that must be weighed against other spread spectrum techniques, it was chosen in this case to limit the scope of this research. It was also selected to make use of the current theoretical work being done with slow frequency hopping.

As pointed out by Liener *et.al.*, the design decisions made in a PRN are highly interdependent. Thus, design decisions made at one level of the ISO OSI model may affect several other levels. Therefore a PRN design may not be complete until a cross examination of the design decisions at all levels is made (21:7).

Organization of Thesis

This thesis is organized in accordance with the *AFIT Style Guide for Theses and Dissertations*. Chapter I is an introduction to the research and includes background, a problem statement, summary of current knowledge, the scope of research, approach to the research, and an outline of the thesis.

Chapter II is a review of literature pertinent to this research. It includes current information on aircraft data communications, PRNs, and frequency hopping spread spectrum radios. The specific area of slow frequency hopping PRNs is also covered with special emphasis given to the throughput characteristics of these networks and the error detection/correction methods and coding.

Chapter III discusses the methodology used in developing or selecting each of the parameters or protocols of interest. The actual analysis of the parameters and protocols is the content of Chapter IV. Chapter IV will also contain a discussion of the design decisions made as they affect other levels of the ISO OSI model. Chapter V summarizes this thesis and recommends areas for further study.

II. Review of Literature

Introduction

Justification of the Search and Review. The purpose of this literature review is to present a sampling of the currently available literature dealing with aircraft data communications, PRNs and frequency hopping techniques. It is intended that this review will build the background information necessary to develop the matematical model to investigate and characterize a data network suitable for use in aircraft-to-aircraft data communications.

Scope of the Research Topic and of the Data Base. In order to limit this review to a manageable level, it will be limited to those topics essential to developing the required results regarding channel capacities and channel access protocols for a slow hopping aircraft-to-aircraft data network. To that end, those topics include aircraft data communications, PRNs, frequency hopping spread spectrum radios, and frequency hopping PRNs.

The data bases queried for information on these topics include the Defense Technical Information Center *Technical Report Summaries*, the NASA *RECON* data base, and *Knowledge Index* data bases, including *INSPEC, Microcomputer Index, The Computer Database, Engineering Literature Index, and NTIS. Books in Print* was also researched. The references cited throughout the work were found through these data bases or through the bibliographies of references found in these data bases.

This literature review will examine the issues of aircraft data communications, PRNs, and frequency hopping spread spectrum radios with an emphasis on the background of those areas and the current state of the art. The area of frequency hopping PRNs will be covered in more depth including currently operational systems as well as the current theoretical work in this area. The issue of error correction/detection and coding will also be addressed.

Aircraft Data Communications

Background. During the early 1970s, the need for aircraft data communications for communications, navigation, and identification (CNI) was identified (3:7). At the same time, USAF experience in South East Asia and the Israeli Air Force's experience during the 1973 Mideast war, established the need for anti-jam aircraft communications. In the case of the Israelis, the Egyptians were able to almost completely jam all voice communications between aircraft. Although the Israelis were able to use channel switching procedures to avoid the jamming, the Egyptians quickly determined the new frequencies being used and were able to jam them. The loss of communications between tactical aircraft had a devastating effect on the effectiveness of Israeli forces (11:1–2).

To meet the need for jam resistant voice communications, the Air Force has progressed through the *Have Quick*, *Seek Talk*, and *Have Clear* programs (11, 28, 34, 19). *Have Quick* was a simple, quick, and inexpensive modification to existing analog voice aircraft radio systems that converted them to slow-frequency hopping systems thus increasing their resistance to jamming (34:125). *Seek Talk* was intended to replace *Have Quick* but was subsequently replaced by the *Have Clear* program because it failed to meet operational requirements (34:126).

In order to meet the need for CNI data communications that were jam resistant, development of the Joint Tactical Information Distribution System (JTIDS) was launched (11:2). JTIDS is the data communication system that is the current state of the art in aircraft data communications.

State of the Art. JTIDS is a spread spectrum, fast frequency hopping communications system which provides secure, jam-resistant, digital communication of CNI information over a wide frequency band. Only JTIDS users, or subscribers, will have access to this broadcast information. Each JTIDS terminal consists of the following subsystems: antenna, transmitter/receive group, signal processor, communications processor, and digital display (11:3). The number of users can theoretically range from two up to a few thousand.

JTIDS operates in the 960 – 1215 MHz frequency range, the same band currently used for air navigation, measuring equipment, and identification friend or foe (IFF). JTIDS has successfully demonstrated its electromagnetic compatibility with the other users of this band (28:58). System power is sufficient for line of sight ranges of up to 500 nm. By using airborne relays, the system range can be extended up to 1200 nm, although for each relay required the system capacity is significantly decreased (3:8–11).

JTIDS gains its jam resistant qualities by using a hybrid of spread spectrum techniques. These techniques are time division multiple access (TDMA, sometimes referred to as time hopping), frequency hopping, and direct sequence pseudo-noise (29:824). The most significant of these is the TDMA, not only does it add to the systems jam resistance, but it is also the key to each user's access to the system. Code division multiple access (CDMA) is superimposed over the TDMA structure to extend the system to multi-net operations (CDMA will be addressed later in this chapter). Users are assigned one or more nets (codes) and allocated time slots during which they may send or receive information on those nets. Each receiver may receive all information transmitted on their assigned nets, therefore some information filtering is required. Users may enter the net at any time and each user maintains system time independently. Network entry is achieved upon receipt of any message from another user that has achieved synchronization (3:7). JTIDS is designed for a theoretical maximum of 128 nets, however this is not physically possible. Testing has indicated that the maximum number of nets in one geographical area will be closer to 20. If that number is exceeded, mutual interference will begin to degrade system performance (3:11).

The TDMA architecture can be described as follows: Each day (24 hours) is divided into 112.5 epochs. An epoch is 12.8 minutes long and is divided into 98,304 time slices. Each time slice is 7.8125 ms long. The 98,304 time slices are divided into three interleaved sets (A, B, and C), each containing 32,768 time slices. The time slices/sets are arranged as such: A1, B1, C1, A2, B2, C2, A3, ..., A32768, B32768, C32768, A1, B1, ...The basic message unit is a time slot which is a subset of the time slice, the details of which are classified (3:9).

JTIDS is capable of distributing data in one of two formats; free text (unformatted messages) and standard message formats. The system can handle free text at 28.9 KBPS and standard message formats at 59.5 KBPS. Digitized voice, encoded at 16 KBPS, is one type of free text that the system will handle (3:9).

The JTIDS program has experienced quite a difficult development process (28, 11, 19, 23, 34, 38) , which explains why, in 1989, a system that began development in 1976, is only partially deployed in two aircraft systems (the E–3A and the F–15). Some of these difficulties arose when the Navy's development effort split from the Air Force's. The Navy wanted to pursue a technique known as distributed time division multiple access then, because of cost over-runs and interoperability problems, was forced to rejoin the Air Force effort.

JTIDS has also been the subject of outward criticism. Some of its shortcomings were outlined in a paper entitled *Challenges and Opportunities in Air Force Tactical Communications.* Those shortcomings are:

> Size, weight and power. The JTIDS terminal planned for fighter aircraft, and the smallest terminal ("manpack" version), would benefit from dimensional and power reductions; to make the "fighter" terminal amenable to smaller tactical aircraft, and to make a true manpack terminal. (It has been shown that newer technology can achieve the desired dimensional reductions.)
>
> Interoperability.
>
> Cost. The costs of installing and retrofitting JTIDS terminals in tactical aircraft are very high.
>
> ECCM. The increasing ECM threat, in both jammer sizes and numbers, may make improved ECCM more desirable in the future. (2:2)

Another problem, especially from the aspect of voice communications, is that while major users such as command and control aircraft may be allotted many time slices every 12 seconds, a tactical fighter may only be allotted one of those slots. This would make voice communications on the part of the tactical fighter impossible (28:58).

JTIDS has another flaw, while it is jam resistant, it is not designed with low probability of interception (LPI) in mind. If this were the case, the power output of the system would be variable so as to minimize the power spectral density (PSD, which is covered later in this chapter) while maintaining an acceptable quality of communications.

Two systems have been proposed thus far to make up for some of the shortcomings mentioned, the first, Enhanced JTIDS System (EJS) as proposed would have the following characteristics:

1. Secure voice only, using TDMA techniques in the Alternate Band which falls just below 1 GHz.
2. Secure voice and limited data, also using TDMA, in the JTIDS band, 960–1215 MHz.
3. The system would be supplemented with a *Have Quick* radio operating in the traditional UHF (225–400 MHz) band.(19)

The Alternate Band coincides with frequencies used by the USSR for military communications, so any Soviet attempt at jamming that band during wartime would also disrupt their own communications. Part of the concept behind EJS is to spread communications over three frequency bands to make the Soviet ECM task more difficult.

The second proposal is known as the Integrated Communications Navigation Identification Avionics (ICNIA) system. ICNIA is a feasibility study intended to provide a single integrated system that can perform, on a time shared basis, the functions that now require 14 different subsystems, including voice radios, TACAN and Navstar navigation, IFF, ILS (Instrument Landing System), and JTIDS (9). Information on ICNIA is very scarce in the open literature.

Packet Radio Networks

Background. PRNs represent the growth of packet switching technology into the realm of broadcast radio. The first system of this type to be fielded was the University of Hawaii's *Aloha* system. This system was developed to allow students spread out over seven campuses on four islands to access one central computer center. radio was selected over the traditional phone lines because of the high cost and low reliability of the phone system in the islands. The system was developed around small FM transmitter/receivers with a range of about 30 km. Two frequencies are used in the system, 407.350 MHz for traffic from the remote stations to the central computer, and 413.475 MHz for acknowledgements and other traffic from the central computer to the remote stations. The channel access protocol known as *Aloha* was developed specifically for this system (37:25–26).

PRNs provide attractive solutions to many troublesome network problems. They provide a reliable medium to networks that have mobile users (such as an aircraft-to-aircraft network), networks that are located in remote or hostile areas where the use of a telephone system is not possible, and networks where the traffic tends to be bursty or have a low data rate (16:41). Although the first two applications are obvious, the third may require some explanation. If a network has traffic that is bursty in nature (it has a high ratio of peak bandwidth to average bandwidth) or when it has a low data rate, the expense of dedicated transmission lines where the user pays for peak bandwidth capacity is not cost effective. Packet radio, on the other hand, offers the capability of sharing a single, high capacity, broadcast channel instead of having a large number of channels with fixed, mostly unused, capacity (36:185).

PRNs are not without their problems. With their unique set of advantages they also bring along a set of design considerations that are not uncomplicated. Among these design complexities are methods for sharing the common radio channel (channel access protocols), methods for determining connectivity and using the connectivity to route messages through the network (routing), and methods for achieving reliable communications in a typically noisy radio environment (21:6–7). The first two of these issues, channel access protocols and routing, will be covered later in this section. The third issue, achieving reliable communications, is usually accomplished by applying spread spectrum techniques. One of those spread spectrum techniques, frequency hopping, will be covered in the next section. Like any other network, the design considerations in a PRN can be organized via the ISO's OSI Reference Model, although in the case of PRNs the various layers are highly interdependent. The design decisions made at one level may affect the design of several other levels (21:7). For example, while applying the spread spectrum techniques mentioned may be one way to achieve a reliable transmission medium at the physical layer, it also affects the design choices made in the Medium Access Sublayer, the Data Link Layer, and the Network Layer.

Channel Access Protocols. In order for a node in the proposed aircraft-to-aircraft data network to transmit data, it must first access the radio channel. However, since there are other aircraft in the network, there must be an orderly way to access the channel, thus channel access protocols. Channel access protocols come in two basic types, controlled access and random access. The controlled access protocols include polling and token passing, while the random access protocols include *Aloha* (and variations thereof) and Carrier Sense Multiple Access (CSMA) protocols (30:403). There are also several hybrid schemes. Brief descriptions taken from *Computer Networks* (36) of each of these protocols follow.

Polling. Polling controls access to the transmission medium via a central controller. As the name implies, the central controller polls each station to see if it has data to send. If the station has data to send, it transmits, if not, it replies negatively to the controller. The next station is then polled. This continues until all the stations have been polled, the central controller then repeats the cycle. This type of polling is known as roll-call polling. Usually the terminals are polled once per cycle,however, high priority or heavy users may be polled several times per cycle. A slight modification to this, known as hub polling, makes more efficient use of time and the line capacity. By passing the poll to the most distant station first and then having that station transmit its data or pass the poll to the next closer station and so on until the poll comes back to the controller, the time it takes for a poll to go out and back to each station is saved.

Token Passing. Token Passing protocols were developed specifically for logical ring networks. Since it is possible to configure a broadcast network in this manner, they bear mentioning here. The token is a special bit pattern that circulates around the logical ring. When a station has data to send it seizes the token and transmits. Stations are not allowed to transmit without the token and the time that they may hold the token is limited. Under conditions of heavy load, transmission of data passes around the ring round-robin fashion and channel utilization approaches 100%.

Aloha. *Aloha* (developed for the University of Hawaii's *Aloha* system mentioned earlier) is an extremely simple method of controlling access, it allows users to transmit whenever they have data to send. Of course, some of the data frames will collide with those transmitted by other users and both frames will be destroyed. If a frame is destroyed, the distant end does not receive it and thus does not acknowledge it. The transmitting station then times-out and "backs-off" a random amount of time and retransmits the frame. The back-off ti... ..ust be random across the network or the same frames would collide again an' again. Obviously this system cannot be very efficient and, in theory, can only attain a ma-imum channel utilization of 18%.

Slotted Aloha. A simple modification to the original *Aloha* protocol is known as Slotted *Aloha*. In Slotted *Aloha*, users are not allowed to transmit whenever they have data, but instead the user must wait for the beginning of the next time slot to transmit. This system requires the additional overhead and synchro-ization required to form the time slots, but returns twice the channel utilization, 37%, that is possible with *Aloha*.

Carrier Sense Multiple Access. The Carrier Sense Multiple Access (CSMA) protocols all share the common ability to sense the carrier for ongoing transmissions and transmit or refrain from transmitting based on that information. CSMA is divided into three groups: 1-persistent CSMA, nonpersistent CSMA, and *p*-persistent CSMA. 1-persistent CSMA senses the carrier continuously

Protocol	Maximum Channel Utilization
1-persistent CSMA	52.9%
0.1-persistent CSMA	79.1%
Nonpersistent CSMA	81.5%
0.01-persistent CSMA	82.7%

Table 1. Comparison of CSMA Channel Utilization.
(20:1411)

and transmits the user's traffic the instant it detects a free channel. Obviously, if there are several users on the network and two or more have traffic to send, a collision will occur when the channel becomes free. In this case the transmission is not acknowledged and the transmitter times-out and must retransmit the traffic. Nonpersistent CSMA, on the other hand, does not sense the carrier continuously for the purpose of transmitting immediately when it detects an idle channel. Rather, when it has traffic to send it waits a random amount of time, senses the channel, and either transmits if it finds the channel idle or backs-off and repeats the random wait if the channel is busy. P-persistent CSMA senses the channel when it is prepared to send and if the channel is idle it transmits with a probability of p. With a probability of $q = 1 - p$ it defers transmission a random amount of time at which time it starts the algorithm again. Channel utilization is improved over both *Aloha* protocols in all cases, and is summarized in Table 1. It is important to note, however, that the delay characteristics of the network get worse as p is decreased.

Carrier Sense Multiple Access with Collision Detect. A modification of the CSMA protocols is Carrier Sense Multiple Access with Collision Detect (CSMA/CD). Quite simply, the system senses the carrier even during transmission and if a collision is detected it stops transmission immediately. Therefore, if two stations sense the carrier is idle at the same time and begin to transmit they do not waste time and bandwidth transmitting the rest of the already garbled frame. Time is also saved because the stations know that the collision occurred and don't have to wait to time-out waiting for the frame acknowledgement to repeat the algorithm for retransmissions. Performance is significantly enhanced and a theoretical channel utilization of $\rho = \frac{1}{1+6.44a}$ is attainable. Where a is the ratio of the propagation delay between the two most distant stations and the message length in time. So by tuning this ratio by either limiting the distance between stations, decreasing the transmission rate, or increasing the message length, a channel utilization of close to one (100%) can be achieved (30:445–446). Again, it is important to note that as the channel utilization improves the delay characteristics of the net become more undesirable so a compromise must be met. A problem with this family of protocols is the difficulty in detecting a collision on a radio channel, however, when dealing with frequency-hopped systems, a collision will only destroy those bits transmitted at the frequency where the transmissions overlap. These bits can be recovered by a robust error correction scheme.

Collision-Free Protocols. A family of Collision-Free protocols exists in which the basic idea is for the stations that have data to transmit make a reservation during a contention period, then send their data according to the order decided during the contention period. All of these protocols result in high channel utilization but with differing delay characteristics and, in some cases, stations receive differing levels of service as a side-effect of the protocol. As before, in general, the delay characteristics of the protocol get worse as the channel utilization improves.

Limited-Contention Protocols. In order to get the best of both worlds, low delay at light loads (as in *Aloha*) and high channel utilization at high loads (as in Collision-Free, Polling, and Token Passing protocols), the Limited-Contention protocols were developed. Basically the two most popular of these methods, the Adaptive Tree Walk Protocol and the Urn Protocol, act as contention protocols (*Aloha*, CSMA) at low loads and as collision free protocols at high loads. These hybrid protocols provide an excellent mix of favorable channel utilization and delay characteristics.

Routing. Routing of packets in a Packet Radio Network is a prime design decision. Even though the specific case discussed in this paper, a tactical flight of aircraft, is an extremely small network, routing is still important. Routing and flow control strategies have three main objectives:

1. Ensuring that messages entered into the net are delivered to their destination (reliability).
2. Ensuring that messages are delivered with the shortest possible time delay (efficiency).
3. Ensuring that the packet overhead and control packets consume as little of the channel capacity as possible.(16:41)

Obviously these are worthy goals in a network that is carrying time critical tactical data as well as voice communications. Without efficient routing and flow control, the same problems that affect conventional computer networks (packet switched) will degrade the performance of a PRN. Packets circulating around the network endlessly and duplicate copies of the same packet circulating the network must both be avoided. Because of the high degree of mobility of the network in question, a fixed routing algorithm will not work, therefore some sort of adaptive algorithm must be adopted. Furthermore, since any node may be removed at any time, the routing algorithm must not reside on any single platform, but rather must be distributed. For a detailed examination of PRN Routing algorithms, see (16).

One type of routing that looks promising for this application, known as flooding, is also the simplest. Flooding methods involve transmitting the packet to every station in the network. Then each station that receives it decides, first whether or not they are an addressee of the packet and second, whether or not they should retransmit, or forward, the packet. The second decision is usually based on whether they have seen the packet before and, if they have, they do not retransmit the packet. The algorithm does not attempt to store routes at all and, in fact, requires no central control. The main advantages of Flooding methods are the lack of required overhead and network management functions. However, the method is somewhat inefficient in that every station in the network must receive every packet at least once (21:12–13). Therefore:

> ...flooding methods tend to be well suited to applications where there is a high need for reliable delivery in the presence of uncertain connectivity and when the connectivity is changing so rapidly that it is difficult for routing information to be determined and disseminated throughout the network in a consistent manner. (21:12)

This method seems to be well suited for an aircraft to aircraft PRN since the topology will be constantly and rapidly changing and high reliability is paramount. The drawback being the extra channel capacity necessary for the retransmissions of packets.

State of the Art. The state of the art in PRNs can be considered to be the Defense Advanced Research Projects Agency (DARPA) PRNET. PRNET is an experimental multihop PRN that has been under development and constant revision since 1972. The network operates in and around the San Francisco Bay Area (37:26–27).

The net consists of a number of packet switching nodes, known as Packet Radio Units (PRUs). These communicate with one another via a broadcast radio channel. Each PRU consists of a radio receiver/transmitter and a digital controller. The radio provides the access to the broadcast medium while the digital controller carries out the packet switching functions. These functions include routing decisions, retransmissions, flow and congestion control, as well as assessing the PRUs connectivity with the network. All PRUs in the network are identical and serve two purposes. First, they serve as receiver/transmitters for host sites. These hosts may include computer centers, mobile terminals, remote terminals, processors or sensors dedicated to a specific mission, or internet gateways. The PRUs also serve as repeaters. In a dense population of users, no additional repeaters are necessary because each PRU acts as a both a host and a repeater. In more sparsely populated nets or areas, standalone PRUs (PRUs without a user or host) may be used as dedicated repeaters.

Early in PRNET's development, Control Stations were used to gather connectivity information, compute routes and provide those routes dynamically to the network as users and connectivity changed. Later a system using multiple, independent Control Stations was developed so that the net would not be dependent on a single Control Station. Now the network control is fully distributed and each PRU is capable of determining routing on its own.

In order to support mobile users and be highly deployable, the network was designed around a single operating frequency and omnidirectional antennas. The most current development in conjunction with DARPA's PRNET is the Low-Cost Packet Radio (LPR) which employs spread spectrum signalling (the LPR is covered later in this chapter).

Frequency Hopping Spread Spectrum Radio

Spread spectrum techniques are used in radio transmissions to gain several advantages over normal fixed carrier modulation. These benefits include Low Probability of Interception (LPI), Low Probability of Position Fix (LPPF), Low Probability of Signal Exploitation (LPSE), selective addressing capability, multiple access capability, anti-jamming capability and high interference rejection (5:6). All of these characteristics are highly desirable if not mandatory in aircraft-to-aircraft data communication system. Spread spectrum is accomplished using one of several methods to spread a radio signal over a wide frequency band. These methods include direct sequence, frequency hopping, time hopping and pulsed frequency modulation (5:3). For the purposes of this paper the focus will be on frequency hopping.

Background. Frequency hopping is accomplished through the use of a series, known as the frequency hopping pattern, that is used to select the transmitting and receiving frequencies in a radio network. All transmitters and receivers in the network must be synchronized and know the frequency hopping pattern in order to transmit and receive, this is why it is sometimes referred to as stored reference frequency hopping (SR/FH) (29:823). There are two types of frequency hopping, fast frequency hopping, in which the hopping rate is faster than the data rate, and slow frequency hopping, where the data rate is faster than the hopping rate.

The first patent for such a system was awarded to actress Hedy Lamarr and composer George Antheil in 1941. It was a system to prevent the jamming of control signals to radio controlled torpedoes. The frequency hops were controlled by means of twin, identically coded slotted player-piano rolls, one at the transmitter, one on-board the torpedo. Antheil had perfected the synchronization required by a system such as this in the 1920s when he created a player-piano duet known as *Ballet*

Mechanique. However, it wasn't until 1963 that the system was finally tried and proven (24:89).

System Quality Factors. There are several parametrics that are used to measure the quality or effectiveness of spread spectrum systems. Among these, the most common are process gain, jamming margin and power spectral density. A new parametric that measures the effectiveness of a spread spectrum system in escaping detection, LPI Quality Factor, has been proposed in recent literature (6).

Process gain refers to the difference between a system's output and input signal-to-noise ratios. When comparing spread spectrum systems to narrow band systems, process gain reflects the advantage gained by spreading the signal. Process gain may be approximated for frequency hopping systems as follows (6:10–12):

$$process\ gain = G_p = \frac{BW_{RF}}{R_{Info}} = number\ of\ frequency\ choices$$

Where the RF bandwidth (BW_{RF}) is the bandwidth that the signal is spread over and the information rate (R_{Info}) is the data rate in the channel. A reasonable process gain does not, however, guarantee that a system can operate in the presence of hostile jamming. A system's ability to operate in the presence of jamming is indicated by its jamming margin.

Jamming margin takes into account the requirement for a useful system output signal-to-noise ratio as well as accounting for system losses. It can be represented as follows (6:10):

$$jamming\ margin = G_p - [L_{Sys} + (\frac{S}{N})_{Out}] = M_j$$

Where L_{Sys} represents the system losses and $(\frac{S}{N})_{Out}$ represents the signal-to-noise ratio required at the information output. Jamming margin indicates how much jamming, in decibels, can be present for a system to perform satisfactorily. While process gain and jamming margin deal with a system's ability to transmit and receive information, power spectral density and LPI Quality Factors deal with a system's ability to escape detection.

Power spectral density is a direct measure of a system's LPI capabilities, this includes the related capabilities of low probability of position fix (LPPF) and low probability of signal exploitation (LPSE). This parametric of a spread spectrum signal measures the relation of the power density to the frequency spectrum that it is spread across. Thus the signal detection problem of, first, monitoring a larger frequency band and, second, being able to detect a lower power density in that band is combined in the parametric of power spectral density. Power spectral density is a function of the system's output power, the information bandwidth, the method of spreading, and the spreading bandwidth (31:18).

LPI Quality Factors, other than PSD, have been expressed in a recent paper, *System Quality Factors for LPI Communications* by Lawrence L. Gutman and Glenn E. Prescott (15). The authors propose that the quality of a LPI system, Q_{LPI} is a function of quality factors relating to the antenna (Q_{ANT}), atmospheric conditions (Q_{ATM}), type of modulation (Q_{MOD}), and interference (Q_{ADA}). The latter is referred to as the quality of adaptive technologies, thus the ADA subscript. The actual relation of these quality factors is as follows:

$$Q_{LPI} = Q_{ANT} + Q_{ATM} + Q_{MOD} + Q_{ADA}$$

Where each of the components can be expressed as follows:

$$Q_{ANT} = 10\log\left(\frac{G_{tc}G_{ct}}{G_{ti}G_{it}}\right)$$

$$Q_{ATM} = \xi_i R_i - \xi_c R_c$$

$$Q_{MOD} = 10\log\left(\frac{S_i/N_{oi}}{S_c/N_{oc}}\right)$$

$$Q_{ADA} = 10\log\left[\frac{kT_{ei}+T_o(F_o-1)+\sum_{n=1}^{N}\sum_{m=1}^{M} g_{in}g_{im}J_{nmi}/B_i}{kT_{ec}+T_o(F_c-1)+\sum_{n=1}^{N}\sum_{m=1}^{M} g_{cn}g_{cm}J_{nmc}/B_c}\right]$$

Where G_{tc}, G_{ct}, G_{ti}, and G_{it} are the gain of the transmitter antenna in the direction of the intended receiver, the gain of the intended receiver's antenna in the direction of the transmitter, the gain of the transmitter antenna in the direction of the intercepting receiver, and the gain of the intercepting receiver's antenna in the direction of the transmitter; ξ_i and ξ_c are the generalized average loss factor due to atmosphere at the intercepting receiver and the intended receiver; R_i and R_c are the range from the transmitter to the intercepting receiver and the intended receiver; S_i/N_{oi} and S_c/N_{oc} are the signal- to-noise density ratios required at the intercepting receiver and the intended receiver to achieve some minimum error performance; k is Boltzman's constant; T_{ai} and T_{ac} are the temperature of the intercepting and intended receivers' antennae; F_i and F_c are the noise figures of the intercepting and intended receivers; and $\sum_{n=1}^{N}\sum_{m=1}^{M} g_{in}g_{im}J_{nmi}/B_i$ and $\sum_{n=1}^{N}\sum_{m=1}^{M} g_{cn}g_{cm}J_{nmc}/B_c$ represent the effect of jammers from n directions in m frequencies with a jammer power spectral density of J_{nmx} and bandwidth of B_x on the intercepting and intended receivers.

In the above argument the effect of a transmitter *jamming* an intercepting receiver is created by null steering antennae (i.e. the transmitter reduces his power in the direction of the hostile receiver). The system Quality Factor can also be expressed as:

$$Q_{LPI} = 20\log(\frac{R_c}{R_i})$$

Where, again, R_i and R_c are the range from the transmitter to the intercepting receiver and the intended receiver. This indicates that any improvement in the overall LPI Quality Factor will allow the system to operate undetected over longer distances or force the interceptor to move closer to the transmitter. The overall system quality can be improved by improving any of the separate quality factors identified with the obvious exception of Q_{ATM} which we have little or no control over.

Multi-user Nets. Two methods of multiple access to frequency hopping channels are currently used, code division multiple access (CDMA) and time division multiple access (TDMA). By assigning different users in the network different frequency- hopping sequences, or spreading codes, both multiple access and addressing capabilities may be gained, this is what is known as CDMA

(26:117). If the spreading codes are carefully chosen to be orthogonal and robust error correction coding is used, several users, or subnets, may occupy the same bandwidth without interfering with one another. Likewise, if a user in one subnet knows, or can reproduce, the spreading code of another subnet, he can *address* packets to that subnet.

In a TDMA system users, or subnets, are assigned time slots during which they may communicate. This system is easier to implement than CDMA even though it requires a system clock. It is, however, inherently less efficient than CDMA in that it does not make effective use of the channel bandwidth. For example; a very inactive user's time slot may go empty several cycles in a row, while heavier users fill their time slot(s) each cycle and may still have excess data to send. TDMA is implemented in the JTIDS system.

Frequency Hopping Packet Radio Networks

Background. The bulk of the literature available on frequency-hopping packet radios is mostly theoretical, very little in the way of performance parameters for actual systems exists. Much work has been done in proposing channel access protocols for such systems as well as establishing error probabilities, delay characteristics, and throughput for such systems (26, 7, 12, 14, 13, 40, 10, 1, 27, 4). The vast amount of theoretical interest in this area can be tied to its potential for use in satellite networks, tactical data networks, and cellular phone/radio networks (26, 25, 8, 37, 21). The combination of spread spectrum techniques and PRNs holds great promise for those users that demand the LPI, anti-jam, and multi-access capabilities of spread spectrum and the mobility of data communications provided by PRNs. The multi-access capabilities are especially attractive to those who are adversely affected by the ever decreasing amount of RF spectrum available for use.

Channel Access Protocols. Channel access in a frequency hopping PRN may be vastly simplified over that in a non-spread spectrum system. This is due to the *capture* properties of spread spectrum systems. Capture refers to a receiver's ability to successfully receive a packet in the presence of time overlapping packets. Because transmission in PRNs is asynchronous, a preamble is used preceding a data packet to synchronize the receiver to the transmitted packet. Having acquired synchronization by processing the preamble, the receiver then switches to the frequency hopping sequence. Once in the frequency hopping sequence, the receiver has captured the incoming packet and can ignore all other transmissions. Therefore the network is susceptible to collisions only when the preamble is being processed and for very short periods of time during transmission when two frequency hopping patterns may overlap on the same frequency. If the former source of collisions is significant, standard channel access protocols could be implemented during the period of time the preamble is being processed. In the latter case, two measures can be taken, first, the frequency hopping patterns can be carefully chosen such that they are orthogonal and, secondly, error correction codes and data interleaving that will correct for the number of bits lost in one frequency hop may be implemented. One further measure may be taken, once a receiver has locked on to an incoming packet it inhibits its transmitter from sending any packets. This is based on the assumption that any receiver that can "hear" that transmitter is probably already locked on to the first packet. Any attempt to transmit may just cause a collision. This is what is referred to as carrier sensing in spread spectrum systems.

Forward Error Correction. The high cost, measured in channel utilization, of any retransmissions scheme of correcting packets that are received in error in a broadcast network makes forward error correction (FEC) a necessity in the network proposed herein. Although error detection codes such as the cyclic redundancy code may be used as a backup, they would most likely be found inside a robust FEC scheme. One such family of FEC codes are known as Reed-Solomon codes.

Reed-Solomon (RS) codes are made up of blocks of $2^Q - 1$, Q-bit characters. There are also RS code block lengths of 2^Q and $2^Q + 1$ but their performance is basically the same as the $2^Q - 1$ codes. An RS code using $Q = 8$ bit characters and 64-character redundancy would be referred to as a (255,191) RS code ($2^8 - 1 = 255$ and $255 - 64 = 191$). This code would be capable of correcting up to 32 character errors or, equivalently, 256 bit errors. Thus in a block of 255 characters, 191 of them are carrying information and 64 of them are redundant (32:160–161). This works out to be about 34% overhead. Performance of these codes is extremely robust, so robust in fact, that the manufacturers of the JTIDS system claim that "[t]his code permits reconstruction of the information content of a message even if up to fifty percent of the pulses are lost." (33)

The performance of RS codes, or any other codes for that matter, can be improved by interleaving. This method is particularly successful in environments were burst errors are likely to occur (like those due to self-interference and jamming). Using this approach, packets are interleaved prior to transmission and deinterleaved prior to decoding. An interleaver rearranges the ordering of a sequence of symbols or bits in a deterministic manner and the associated deinterleaver reverses this process. The process makes the error correcting code's job easier by splitting up burst errors that occur during transmission among several symbols. For example, a burst error 8 bits long might take out a whole character but using interleaving this burst error could be spread out so that each of 8 characters had 1 bit error a piece. Thus the job of the FEC code is greatly simplified.

Throughput and Delay Characteristics. The throughput and delay characteristics of spread spectrum PRNs are addressed by (4, 7, 10, 12, 13, 14, 18, 25, 26, 27, 30, 37, 40). Each of these references selects a spread spectrum technique, a channel access method, and, in some cases, a routing technique, and analyzes the throughput and delay characteristics of the proposed PRN. In each case the interplay of the network design decisions and the assumptions that must be made are the crux of the work. The models that are most readily applicable to the problem at hand, as well as any modifications or further assumptions, are discussed at length in Chapters III and IV of this thesis.

State of the Art. Numerous spread spectrum transmission protocols for PRNs or data transmission have been proposed in literature but, at the time, there is very little performance data available (26:118). However, one current PRN, the Low-Cost Packet Radio (LPR), uses hybrid spread spectrum techniques and a proposed PRN, SINCGARS/SPRNET (SINCGARS Packet Radio Network), will use slow frequency hopping. The implementation and performance of the LPR follows, as do implementation details of SINCGARS/SPRNET.

Low-Cost Packet Radio. (8:34–41) The Low-Cost Packet Radio (AN/PRC-118) is the current test bed for the development and testing of packet radio concepts, techniques and protocols. The LPR implements spread spectrum coding continuously by direct sequence coding and slow frequency hopping can be implemented on a packet-by-packet basis. The LPR uses a direct sequence pseudonoise (DSPN) spread spectrum waveform. To ensure security, the DSPN code required is supplied by the National Bureau of Standards Data Encryption Standard (DES). The bit stream to be transmitted is added modulo 2 to the DSPN spreading code. The resulting signal is used to modulate the radio's 281.6 MHz carrier. The result is a minimum shift keying spread spectrum signal at an intermediate frequency of 284.8 MHz with phase shift keying. This wave form successfully counters the effects of multipath interference and narrow band interference. The signal has the ability to separate out all multipath signals separated by at least 78ns, meeting the requirements set for the LPR by DARPA who is supporting LPR research. The LPR also demonstrated an anti-jam margin of 9.4db at a transmission rate of 100kSPS (kilo symbols per second) and 3.4db at 400kSPS. Additional spread spectrum coding can be specified by upper level protocols, in which case the LPR changes intermediate frequencies on a packet-by-packet basis. This feature provides additional processing gain against multipath fading and narrow band interference.

SINCGARS/SPRNET. (22:449–454) The SINCGARS/SPRNET project is an attempt to make the US Army's SINCGARS VHF-FM tactical radio work in a packet switched radio network. The radio is capable of handling voice and data communications in single channel and slow frequency hopping modes. It is hoped that the channel efficiencies in the frequency hopping mode will exceed 1 (determined in the single channel mode) due to the capture effect of spread spectrum techniques.

The SPRNET project has made numerous error control design choices, such as interleaving, Reed-Solomon coding and the utilization of side information that, when combined with the SINCGARS frequency hopping, will provide a highly reliable, anti- jam data link for the modern battlefield.

Conclusion

The need exists for Low Probability of Interception, anti- jam, high data rate communication channels between the tactical aircraft of tomorrow. One way that those needs can be met is with an aircraft-to-aircraft PRN that employs frequency hopping techniques. In order to develop these networks, basic research must be done to determine the radio frequency and RF bandwidth; the signaling, encoding, and modulation schemes; how to allocate the bandwidth in time and space to the nodes in the network; what channel access protocol and code assignment algorithm should be used; and how to combine forward error correction (FEC) and acknowledgement mechanisms (such as ARQ) so as to achieve an adequate level of link performance. Once this basic research is conducted it can be used as a foundation on which to build experimental systems and begin to gather performance data.

III. Development of System Model

In order to develop the models necessary to analyze the throughput, delay and required bandwidth for the proposed aircraft-to-aircraft packet radio net, the Carrier Sense Multiple Access protocol for frequency hopping radios was selected. Recall that this system 'sensed' the channel by recognizing if its receiver had locked onto a packet. If the receiver was locked on, transmissions were suppressed. The model will be developed for two cases, first for the network under LPI conditions and, secondly, for the network under jamming conditions. An equation for the required bandwidth under both conditions will then be presented.

LPI Environment Model

The low probability of interception model is based on the assumption that each flight of aircraft is operating as a subnet, that is they are assigned their own frequency hopping code, and that the network has the capability to adjust its output power. The power would be adjusted such that it is a minimum but still maintains flight connectivity. This could be accomplished by exchanging information in the header of network packets regarding the quality of each aircraft's last transmission. If, for example, an aircraft's transmission fell below a predetermined gain or was riddled with errors, the other aircraft in the flight would return that information in their next transmission and the offending aircraft's transmit power would be increased. If, on the other hand, an aircraft was exceeding a predetermined gain and thus endangering the flight's LPI status, the flight would inform the transmitter of this and that aircraft's transmit power would be reduced.

Since it is unreasonable to assume that there would be one Command and Control aircraft per flight or that a Command and Control aircraft would always be within the flight's LPI range, Command and Control aircraft would be required to gain access to each tactical subnet by using higher gain, directional antennae. The burden of directional antennae would be much easier for the larger Command and Control aircraft to assume than the tactical aircraft in the flight.

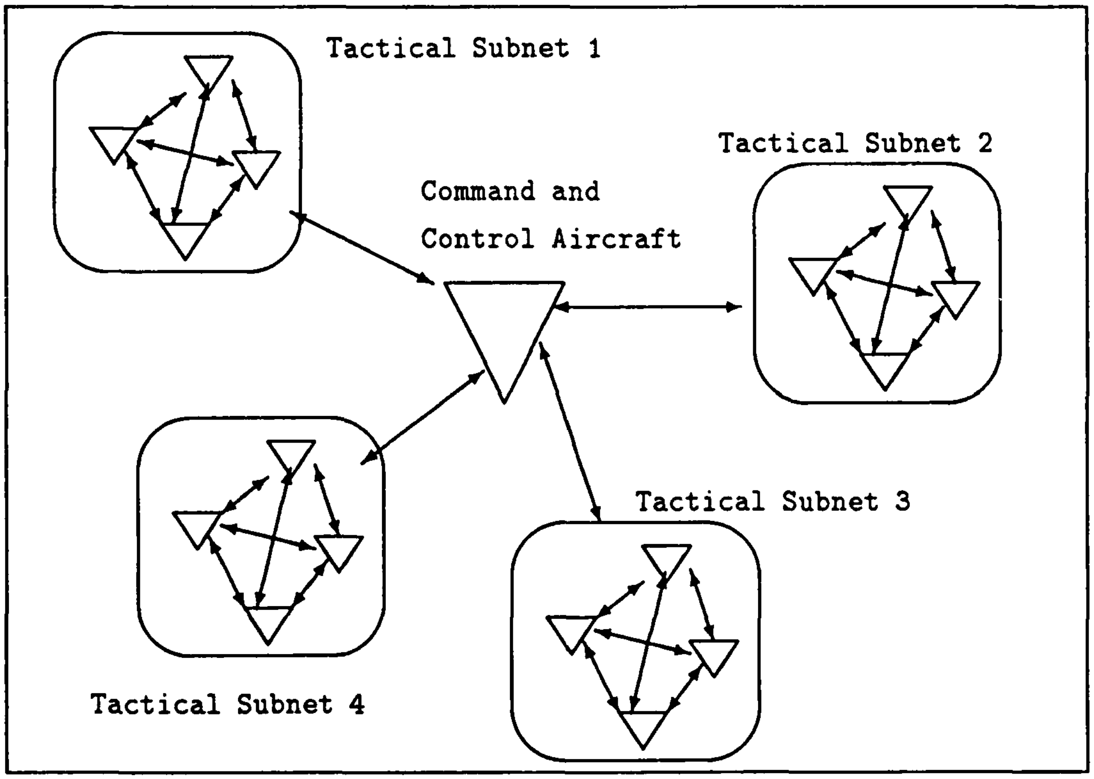

Figure 2. Tactical Subnets

Assumptions. In order for this analysis to be tractable, independent poisson arrivals at each of the aircraft transmitters must be assumed. The arrival rate at each of the aircraft will be referred to as λ_t while the arrival rate at the Command and Control aircraft (destined for the flight being analyzed) will be referred to as λ_c. Based on the required data rates mentioned in Chapter I, $\lambda_c = .125\lambda_t$. The arrival rate at each aircraft is assumed to include new arrivals, retransmissions and exponential backoff of those packets that arrive while transmission is suppressed. In previous analyses, the exponential backoff has simply been considered a retransmission. In order to follow previous analyses, each aircraft is assumed to have a single message transmission buffer. If more messages are generated by the aircraft's computer systems to send when the buffer is full, they must either be held by the computer system or lost. This is not an entirely unreasonable assumption considering the volatility of the data to be shared among aircraft, if a message doesn't make it on its first try, it will probably be updated by the next message that is sent.

Because this model concerns a broadcast net, there is the potential for the two outermost aircraft to transmit at the same time, that is, within τ_{max} of each other and thus synch up with their closest neighbor (see Figure 3). This would make the calculation of throughput and delay an intractable problem as well as making the routing and retransmission schemes a much more complex problem.

In order to eliminate this problem, it is assumed that the synchronization time, t_s, is at a minimum τ_{max} and that the synchronization signal is designed such that if any two synch signals overlap, the synch will fail and other aircraft will be allowed to transmit. This implementation would require a single transmitter/multiple receiver system, not unlike JTIDS. JTIDS uses a set of receivers that listen to a series of frequencies in order to synch up with transmissions intended for a station. If the series of frequencies is incorrect or not complete, the receiver breaks off and begins listening to the series again with the hopes of synching up with a transmission. For simplicity, the worst case scenario, $t_s = \tau_{max}$, will be analyzed.

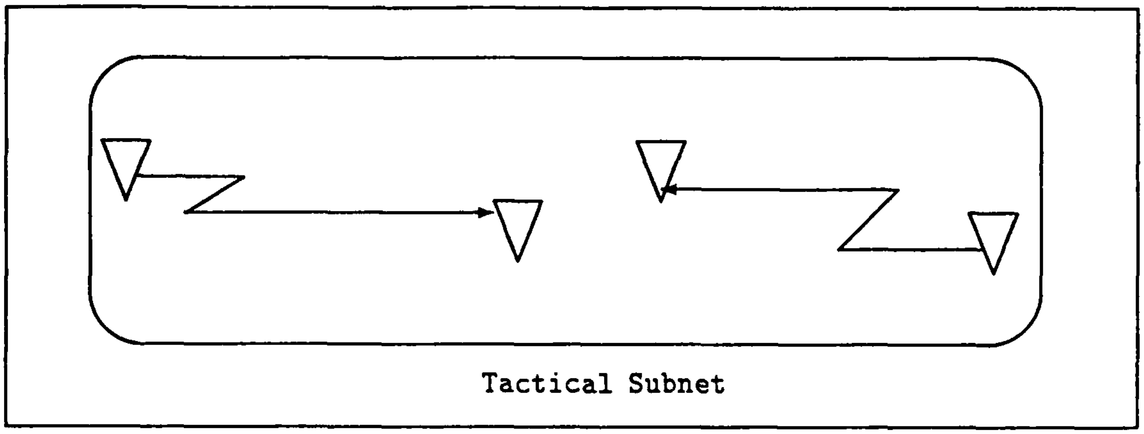

Figure 3. Simultaneous Transmission Conflict

Throughput Analysis. Following an analysis first put forth by Klienrock and Tobagi (20) and repeated by Hayes (18), throughput can be expressed as follows:

$$\rho = \frac{\bar{T}}{\bar{B} + \bar{I}} \tag{1}$$

Where ρ is the channel utilization, $\bar{T}$ is the average time spent in sending useful information, $\bar{B}$ is the average busy time of the channel, and $\bar{I}$ is the average idle time of the channel.

In order to evaluate each of these parameters, it is useful to illustrate a successful transmission cycle (Figure 4) as well as an unsuccessful cycle (Figure 5).

Note that m is defined as the packet size in time and that τ_{max} is the propagation delay between the two aircraft that are the farthest apart.

$\bar{T}$ can then be expressed as the time spent sending usable information given a successful cycle plus the time spent sending usable information given an unsuccessful cycle. However, since no usable information is derived from an unsuccessful cycle, the second term goes to zero and thus $\bar{T}$

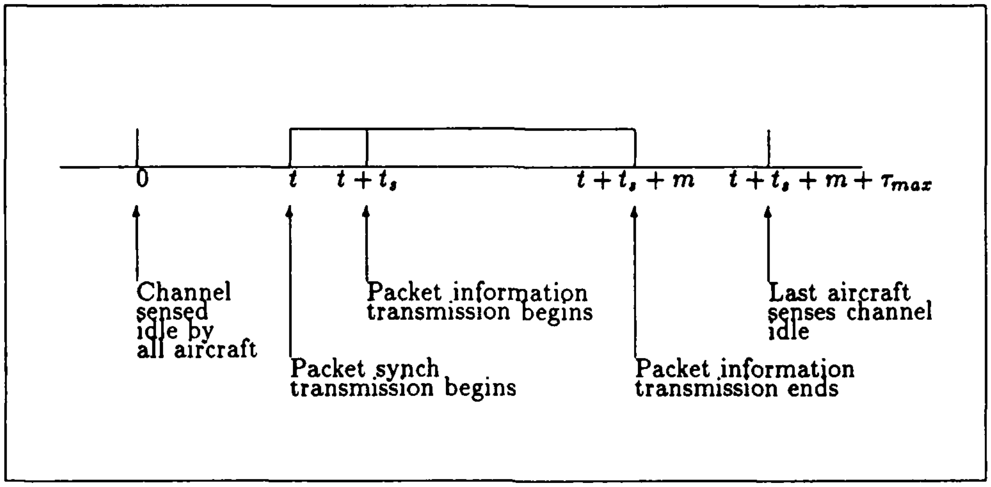

Figure 4. Successful Transmission Cycle

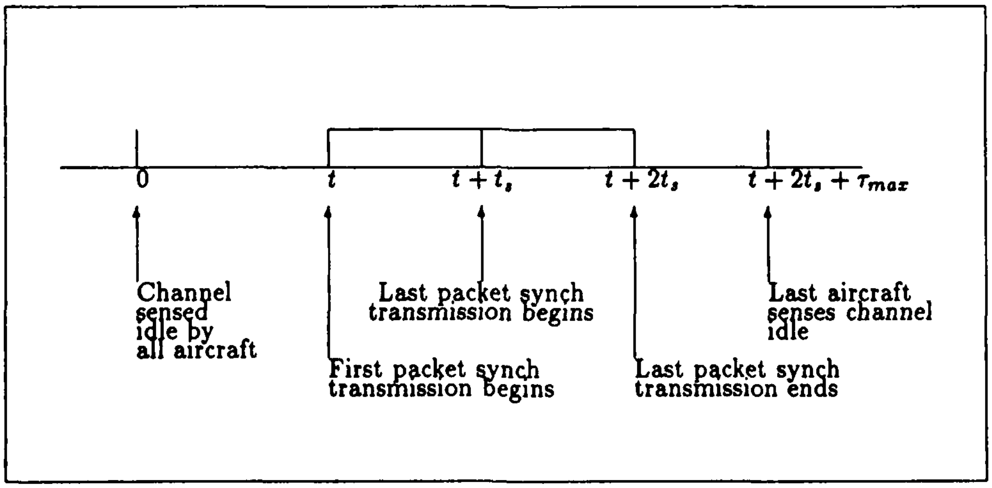

Figure 5. Unsuccessful Transmission Cycle

can be expressed as follows:

$$\bar{T} = (m)P_s \tag{2}$$

Where P_s is the probability of a successful transmission.

Likewise, the average channel busy period can be expressed as the average time that the channel is occupied by any transmissions. This can be shown to be the total time of a successful cycle given that a successful cycle occurred, plus the total time of an unsuccessful cycle given that an unsuccessful cycle occurred. Or:

$$\bar{B} = [(t_s + m + \tau_{max})P_s + (2t_s + \tau_{max})(1 - P_s)] \tag{3}$$

The average channel idle time is merely the reciprocal of the channel arrival rate, Λ. Because the arrival rates at each of the aircraft have been assumed to be independent poisson functions, the sum of these arrival rates will have a poisson distribution as well. Therefore the channel arrival rate is equal to the sum of the arrival rates at each of the stations.

$$\Lambda = 4\lambda_t + \lambda_c \tag{4}$$

And the channel idle time may be expressed as:

$$\bar{I} = \frac{1}{\Lambda} \tag{5}$$

The probability of a successful transmission is based on the probability that there is not an arrival and subsequent transmission during the synch time of the packet in question from any of the other aircraft. Thus, from the standpoint of tactical aircraft #1, it is the probability that tactical aircraft #2 does not transmit *and* tactical aircraft #3 does not transmit *and* tactical aircraft #4 does not transmit *and* the Command and Control aircraft does not transmit. The probability that there is no transmission in time t is $e^{-\lambda t}$. Therefore, the probability that no other aircraft transmits during the period $t_s + \tau_{max}$, P_s, is:

$$P_s = e^{-\lambda_t(t_s+\tau_{max})}e^{-\lambda_t(t_s+\tau_{max})}e^{-\lambda_t(t_s+\tau_{max})}e^{-\lambda_t(t_s+\tau_{max})}e^{-\lambda_c(t_s+\tau_{max})} \tag{6}$$

Or:

$$P_s = e^{-3.125\lambda_t(t_s+\tau_{max})} = e^{-\Lambda(t_s+\tau_{max})} \tag{7}$$

Combining the equations 1,2,3, and 5 results in the following expression for channel utilization:

$$\rho = \frac{mP_s}{[(t_s + m + \tau_{max})P_s + (2t_s + \tau_{max})(1 - P_s)] + \frac{1}{\Lambda}} \tag{8}$$

Substituting for P_s and t_s and normalizing the equation so that m and Λ are represented in terms of contention slots results in:

$$\rho = \frac{me^{-2\Lambda\tau_{max}}}{(m - \tau_{max})e^{-2\Lambda\tau_{max}} + 3\tau_{max} + \frac{1}{\Lambda}} \tag{9}$$

Where a contention slot equals $2t_s + \tau_{max}$ or $3\tau_{max}$.

Delay Analysis. In order to analyze the average delay of the system, an imbedded Markov chain is used as in (18)[215-217]. The chain is imbedded immediately after the transmission point for the duration of the contention period. Since the system can, at most, have five messages in it (a single message buffer at each aircraft is assumed), the Markov chain can be illustrated as in Figure 6. A "state" in this Markov chain is defined as the number of messages in the system at a specific point in time. By solving for the steady state probabilities that a system is in each "state", the average number of messages in the system can be determined.

Thus if $t_{i,j}$ is the probability that the state transitions from state i to state j and P_i is the

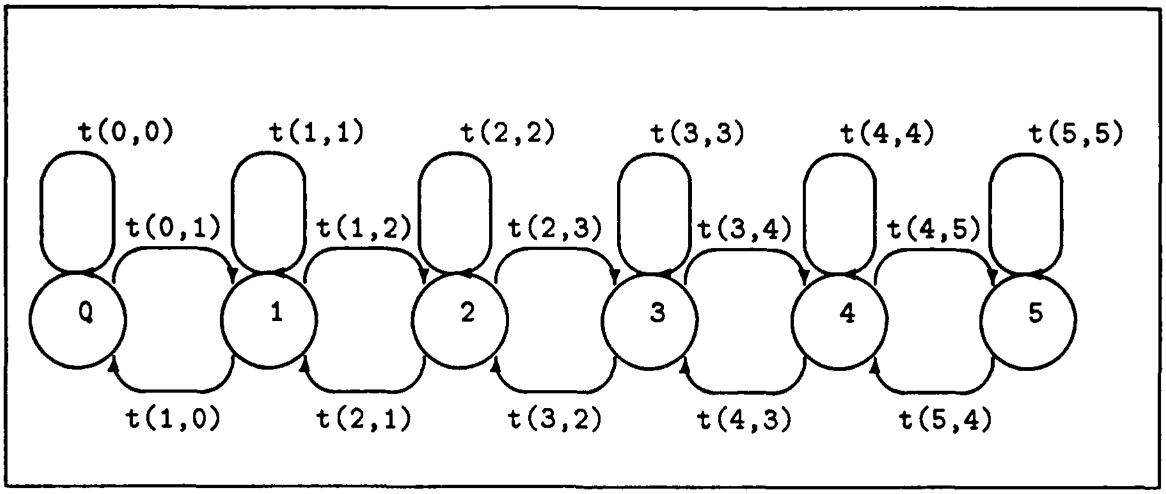

Figure 6. Imbedded Markov Chain

probability that the system is in state i. Therefore, for the purposes of the system in question, the next state probabilities can be enumerated as follows:

$$P_{j=0} = t_{0,0}P_0 + t_{0,1}P_1 \tag{10}$$

$$P_{j=1} = t_{0,1}P_0 + t_{1,1}P_1 + t_{2,1}P_2 \tag{11}$$

$$P_{j=2} = t_{0,2}P_0 + t_{1,2}P_1 + t_{2,2}P_2 + t_{3,2}P_3 \tag{12}$$

$$P_{j=3} = t_{0,3}P_0 + t_{1,3}P_1 + t_{2,3}P_2 + t_{3,3}P_3 + t_{4,3}P_4 \tag{13}$$

$$P_{j=4} = t_{0,4}P_0 + t_{1,4}P_1 + t_{2,4}P_2 + t_{3,4}P_3 + t_{4,4}P_4 + t_{5,4}P_5 \tag{14}$$

$$P_{j=5} = t_{0,5}P_0 + t_{1,5}P_1 + t_{2,5}P_2 + t_{3,5}P_3 + t_{4,5}P_4 + t_{5,5}P_5 \tag{15}$$

Because the imbedded Markov chain represents all the states the system can be in, the sum of the probabilities that the system is in each state must be one.

$$P_0 + P_1 + P_2 + P_3 + P_4 + P_5 = 1 \tag{16}$$

In order to carry out this analysis from Hayes (18) , two new parameters must be introduced, α and σ. Where α represents the probability that a backlogged transmission transmits during the next contention period. The parameter σ is the probability that there are one or more arrivals during a contention period and is defined as follows:

$$\sigma = 1 - e^{\Lambda(2t_s + \tau_{max})} \tag{17}$$

Then using the knowledge that the probability that only one backlogged packet is transmitted is $i\alpha(1-\alpha)^{i-1}$ and the probability of two or more being transmitted is $1 - i\alpha(1-\alpha)^{i-1}$. Herein lies one of the weaknesses of this analysis, Λ is assumed to be poisson in nature and, with the small number of stations in this network, that is a fairly weak assumption. If, however, the arrivals at each station to be transmitted are poisson in nature, the sum of these poisson processes will also be a poisson process. Applying this and the probability that there are one or more arrivals to the system during a contention period, σ, the transition probabilities can be represented as follows (from (18)[216]):

$$t_{i,j} = 0 \qquad j \le i-2 \tag{18}$$

$$t_{i,j} = i\alpha(1-\alpha)^{i-1}(1-\sigma)^{N-i} \qquad j = i-1 \tag{19}$$

$$\begin{aligned} t_{i,j} = {} & (1-\alpha)^{i}(N-i)\sigma(1-\sigma)^{N-i-1} \\ & +[1 - i\alpha(1-\alpha)^{i-1}](1-\sigma)^{N-i} \qquad i = j \end{aligned} \tag{20}$$

$$t_{i,j} = (N-i)\sigma(1-\sigma)^{N-i-1}[1-(1-\alpha)^{i}] \qquad j = i+1 \tag{21}$$

$$t_{i,j} = \begin{pmatrix} N-i \\ j-i \end{pmatrix} \sigma^{j-i}(1-\sigma)^{N-j} \qquad j > i+1 \tag{22}$$

It is important to note that because this analysis does not use the large number of messages assumptions that most analysis make, i.e. N does not approach ∞, thus Hayes' derivation cannot be followed completely.

Equations 18-22 can then be substituted into Equations 10-15 and Equations 10-16 can then be solved simultaneously (see Appendix A). The average number of messages in the system, $\bar{n}$, can then be solved for using the following equation:

$$\bar{n} = 0P_{j=0} + 1P_{j=1} + 2P_{j=2} + 3P_{j=3} + 4P_{j=4} + 5P_{j=5} \tag{23}$$

The average delay, $\bar{d}$, in the system can then be gained by applying Little's Law.

$$\bar{d} = \frac{\bar{n}}{\Lambda} \tag{24}$$

Note that the average delay will be given in the number of contention periods the average packet must wait through. In order to convert that to something more useful, like time, this can be multiplied by the length of the average busy period of the system plus the idle period of the system. Since the analysis has been done looking at the system as a whole, the system arrival rate, Λ, must be used.

Jamming Environment Model

The model of the jamming environment is based on Pursley's work in (25). It supposes that the subnet has been subjected to jamming and as a result has turned its transmit power up in order to 'burn through' the jamming. Likewise, the other tactical subnets in the area have turned their transmit power up. This subjects each subnet not only to the jamming interference but to the interference of the other subnets as well, *Assumptions.* Assumptions from (25) as well as the assumption that the new parameter, the average interference level, includes interference from the other subnets as well as the jammers.

Throughput Analysis. The throughput analysis here follows the same logic as in the LPI case above, however, $\bar{T}$, the time of a successful transmission is further conditioned that given a successful transmission it can be successfully decoded. In other words, that the interference traffic has not destroyed more bits that the Reed-Solomon code can correct for. Therefore the throughput can be expressed as Equation 9 conditioned by the probability that the packet can be correctly decoded. Thus:

$$\rho = \frac{me^{-2\Lambda\tau_{max}}P_c}{(m-\tau_{max})e^{-2\Lambda\tau_{max}}+3\tau_{max}+\frac{1}{\Lambda}} \tag{25}$$

Where P_c is the probability that a packet can be correctly decoded. From (25) P_c can be represented as:

$$P_c = \sum_{j=0}^{K-1} f(j)P_c(j) \tag{26}$$

$f(j)$ denotes the probability that there are j interfering packets (or jammers) and $P_c(j)$ is the probability that the packet can be correctly decoded in the presence of j interfering packets. For a finite population of interfering subnets/jammers, $f(j)$ is based on the binomial distribution and is represented as:

$$f(j) = \begin{pmatrix} K-1 \\ j \end{pmatrix} d^j(1-d)^{K-1-j} \tag{27}$$

Where K is the local population of subnets and d is the duty cycle of each of the subnets. $P_c(j)$ can be represented as:

$$P_c(J) = \sum_{i=0}^{e} \begin{pmatrix} n \\ i \end{pmatrix} [1-(1-P_h)^j]^i(1-P_h)^{j(n-i)} \tag{28}$$

Where e is the number of bit errors the Reed-Solomon code can correct, n is the number of bits in a code block and P_h is the probability of hit in a frequency. For memoryless frequency hopping patterns, $P_h \leq 2/q$. Therefore, to do a worst case analysis, let $P_h = 2/q$. This P_h would be representative of interference from other subnets or spot/sweep jammers, but not representative of barrage or full bandwidth jammers if the enemy has that capability.

Delay Analysis. Since in this analysis, actual retransmissions of traffic that is incorrectly received must be requested, the markov chain analysis in the LPI case will not hold (a transmission that experiences a collision is backlogged where in the jamming case a transmission that can't be correctly decoded must be retransmitted). Therefore a less rigorous (mathematically), more intuitive analysis is used (20). Consider that if the throughput of a channel is ρ, then, on the average, $1/\rho - 1$ packets will require retransmission. The time to retransmit a packet (assume the next packet received carries the retransmission request) can be represented as:

$$r = \frac{1}{\Lambda} + t_s + m + \tau_{max} + t_s + m + \tau_{max} \tag{29}$$

The average delay of a packet can then be represented as:

$$\bar{d} = \frac{1}{\Lambda} + t_s + m + \tau_{max} + (\frac{1}{\rho} - 1)r \tag{30}$$

Substituting for r and normalizing to represent Λ and m in terms of contention slots, this becomes:

$$\bar{d} = \frac{1}{\Lambda} + m + \frac{2}{3} + (\frac{1}{\rho} - 1)(\frac{1}{\Lambda} + 2m + \frac{4}{3}) \tag{31}$$

Required Bandwidth Analysis

In order to evaluate the total bandwidth required for a system as described, the following parameters must be accounted for:

1. R_b, the information transmission rate.

2. R_r, the required information transmission rate.

3. ρ, the channel utilization.

4. q, the number of frequency hopped channels.

5. n, the Reed-Solomon Code block length.

6. k, the number of information bits in the Reed-Solomon Code block.

If binary signaling is used, the bandwidth required to transmit a certain data rate and that data rate are equivalent. Applying this fact, the bandwidth, B, of a system can be expressed as (25:658):

$$B = \frac{qnR_b}{k\rho} \tag{32}$$

Note that the required information transmission rate, R_r, can be expressed as:

$$R_r = \frac{R_b}{\rho} \tag{33}$$

No distinction is made between the LPI and the jamming models for the required bandwidth because once the design decision to use Reed-Solomon Coding is made, it cannot be *turned on or off* depending on the conditions the net encounters. In other words, during LPI conditions the net must bear the overhead cost of FEC that must be in place for the jamming environment.

IV. Results and Discussion of System Models

LPI Environment

Throughput. The results presented in this section were arrived at using the analysis discussed in the LPI throughput analysis section of the last chapter. Figure 7 was plotted using Equation 9 and selecting a range of system input, Λ, that showed the major features of the curve. The graph demonstrates the effect that varying the message length, m, has on system throughput. Recall from the last chapter that m is expressed in multiples of contention slots so that all time related factors would be expressed in common units.

Another major feature of the graph to take note of is that at $\Lambda < 1$, in all cases, the curve peaks. This is due to the onset of system instability, which means the system input is exceeding the system throughput by too much and stations are backlogging messages. For the system described herein, this cannot be allowed to occur. Since each station is assumed to have a single message buffer, if a station backlogs and another message arrives, one of the messages must be lost.

Intuitively this instability will occur when system input exceeds one message per contention slot. Thus since we have five stations in the net, if Λ exceeds .2, the system will go into a region of instability. With extremely long message lengths this tends to be mitigated by the extra data carried by each message so the unstable region is pushed further to the right. A more rigorous analysis of system instability could be carried out using the method found in (30:210-215). Taking this instability into consideration, Figure 8 is a more accurate depiction of the operating throughput of the system.

Note that due to this instability, only the portion of the graphs for which system input is less than .242 should be used when making design decisions. This consideration applies to all graphs that are based on system input.

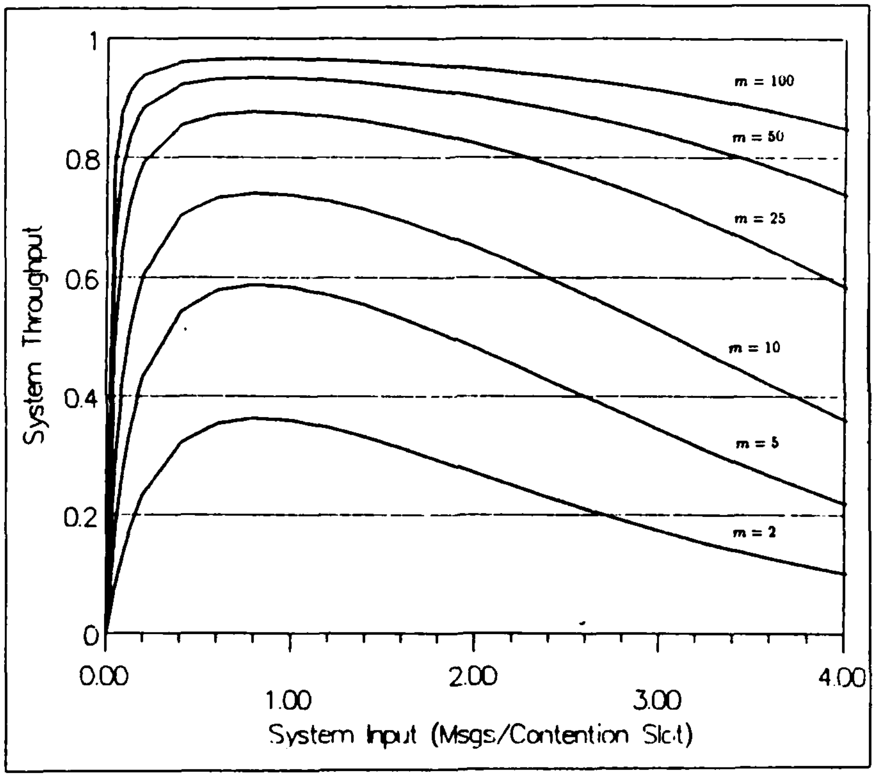

Figure 7. System Throughput for LPI Environment (m expressed in contention periods)

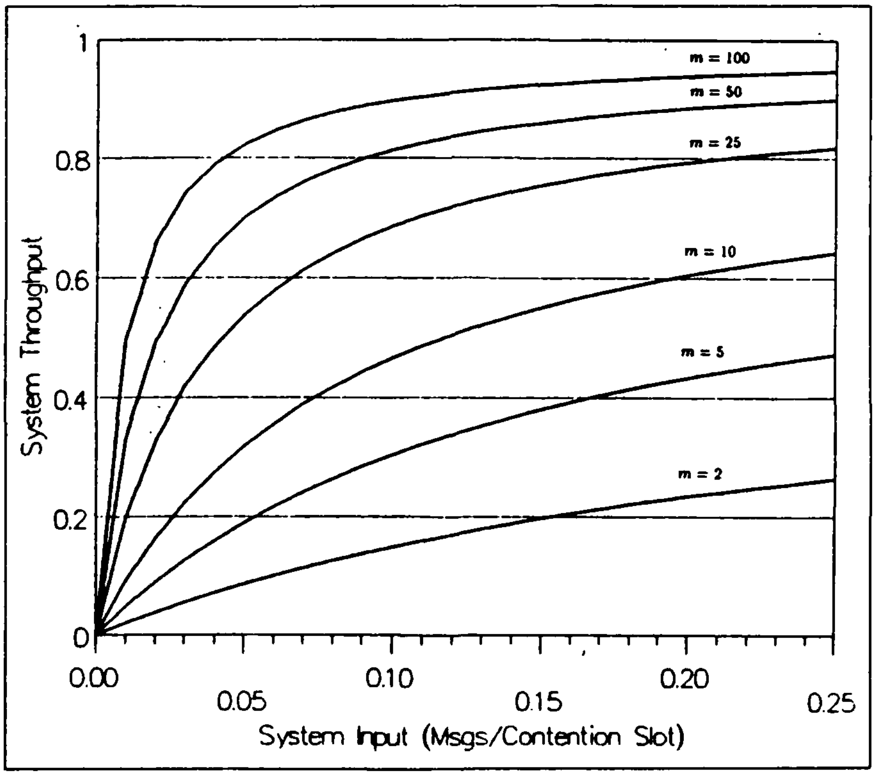

Figure 8. Stable Region of System Throughput for LPI Environment (m expressed in contention periods)

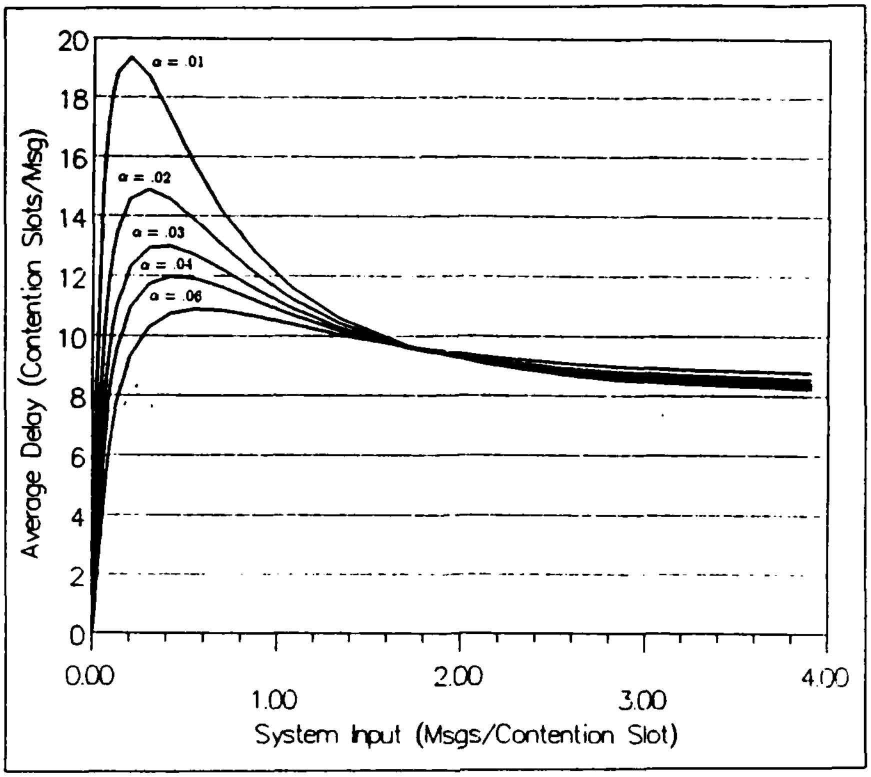

Figure 9. System Delay for LPI Environment, $\alpha = .01 \ldots .06$

Delay. Obtaining results for the LPI delay of the proposed system proved to be extremely difficult and time consuming. ***Macsyma*** was used to first substitute Equations 18-22 into Equations 10-15 and then solve Equations 10-16 simultaneously (See Appendix A). The results generated were in the form of FORTRAN code which was then used to create data files for various values of α and σ.

The delay curves, Figures 9 and 10, were plotted for a range of values for α that are representative of those used in the current literature and demonstrate the characteristics of the family of curves. Recall that α represents the probability that a station transmits in the next contention slot it senses to be free and that σ represents the probability that there a one or more messages transmitted in a slot, which is a function of Λ. Again, note that the data was normalized so that the delay would be given in the number of contention slots a message would be delayed.

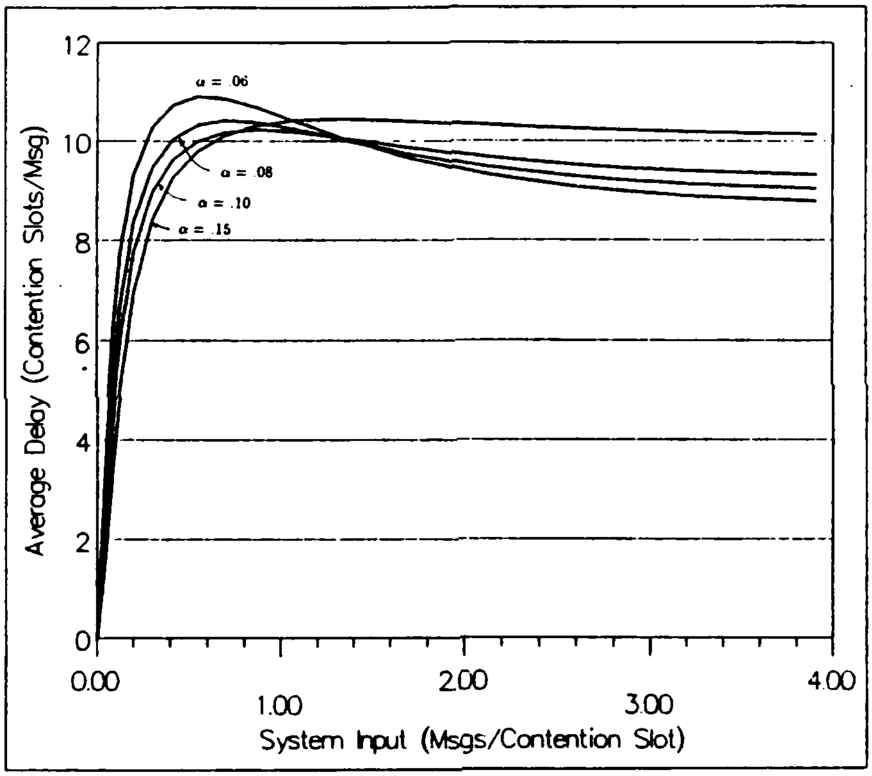

Figure 10. System Delay for LPI Environment, $\alpha = .06 \ldots .15$

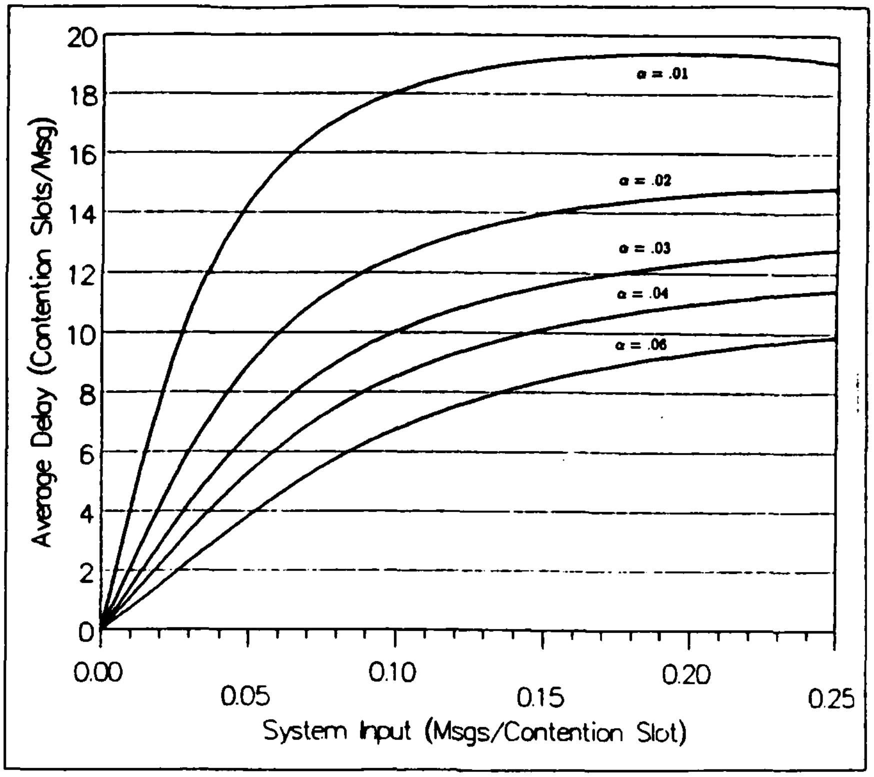

Figure 11. Stable System Delay for LPI Environment, $\alpha = .01 \ldots .06$

The same system instability that affected throughput affects delay, however, in the case of delay, it appears to make system performance increase (i.e. for $\Lambda > 1$ delay is decreasing). System performance does not in actuality increase, the decreasing delay is due to the increasing number of messages that are being backlogged and lost. Thus, the operational (stable) portion of these curves is represented by Figures 11 and 12.

In order to convert the delay represented in the curves into time, an operating point in terms of system input must be selected. The time of delay would be the number of contention slots a message is delayed times the average length of a message/collision. This can be represented as:

$$t = d[\rho(t_s + m + \tau_{max}) + (1 - \rho)(2t_s + \tau_{max})] \tag{34}$$

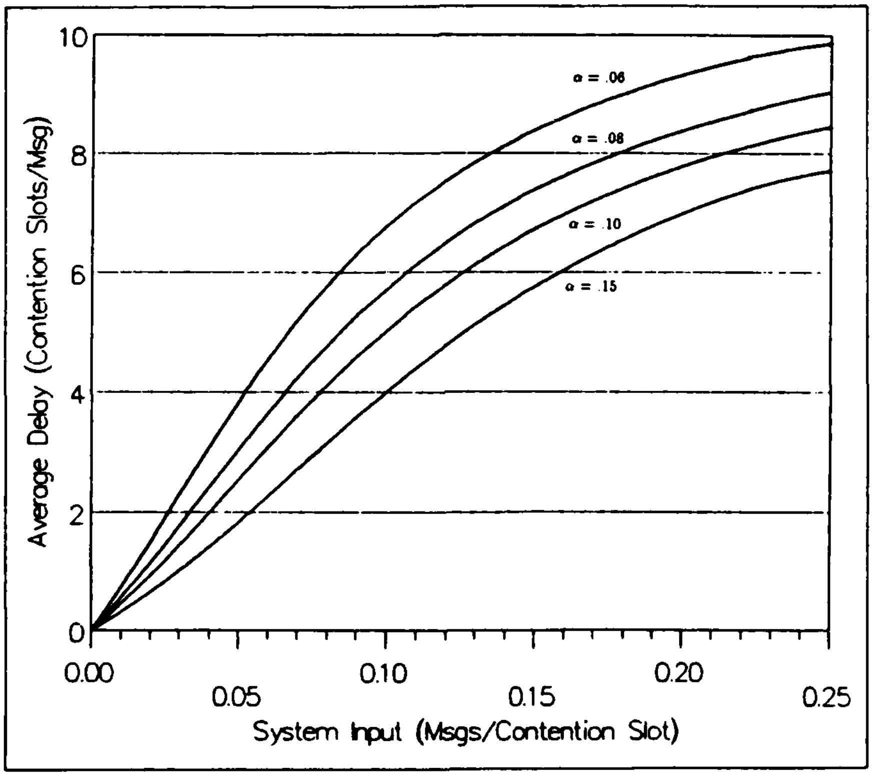

Figure 12. Stable System Delay for LPI Environment, $\alpha = .06 \ldots .15$

Where d is the delay in contention slots at the operating point and ρ is the system throughput at the operating point.

Jamming Environment

Throughput. Equations 27, 28, and 26 were solved for P_c, the probability of correct reception and decoding of a message, for various Reed-Solomon Codes and numbers of frequency hopped channels (see Appendix C). Because of the tremendous number of possible combinations of R-S Codes and parameters q, K, and d that could be analyzed at this point, the field had to be narrowed and three sets of parameters were selected to be graphed. A R-S Code of (64,40) was selected in all three cases because it has the least coding overhead and also provides the least protection from errors. In each case two of the parameters were held constant and the third varied to study the effects.

In the first graph, Figure 13, d is held constant at .6 and K is held constant at 30. Remember that d represents the duty cycle of the interfering nets, K represents the number of interfering nets, and q is the number of frequency hopped channels. As is expected, as q increases the system's ability to overcome the outside interference. In fact, with 150 frequency hopped channels, the throughput of the system is the same in the jamming environment as it is in the LPI environment. Note that the gain in performance from 125 channels to 150 channels is only marginal, while the penalty in bandwidth would be 20%. The performance of the 50 frequency hopped channel system is such that while it only would take 40% of the bandwidth of a 125 channel system it could only sustain a throughput equal to 7.1% of the 125 channel system.

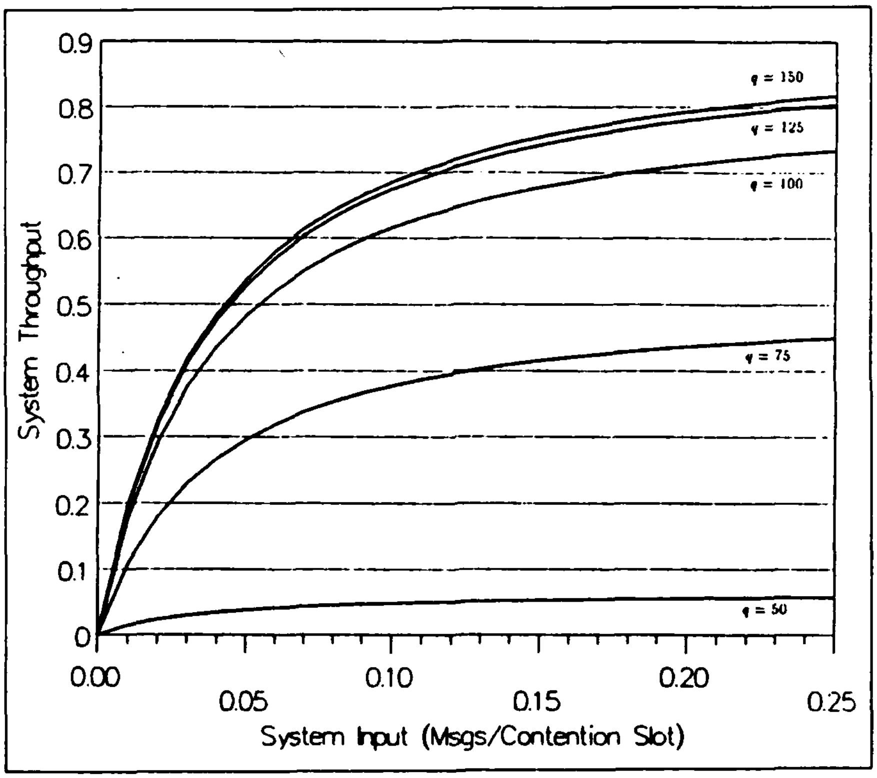

Figure 13. System Throughput for Jamming Environment, $d = .6$, $K = 30$, (64,40) R-S Code

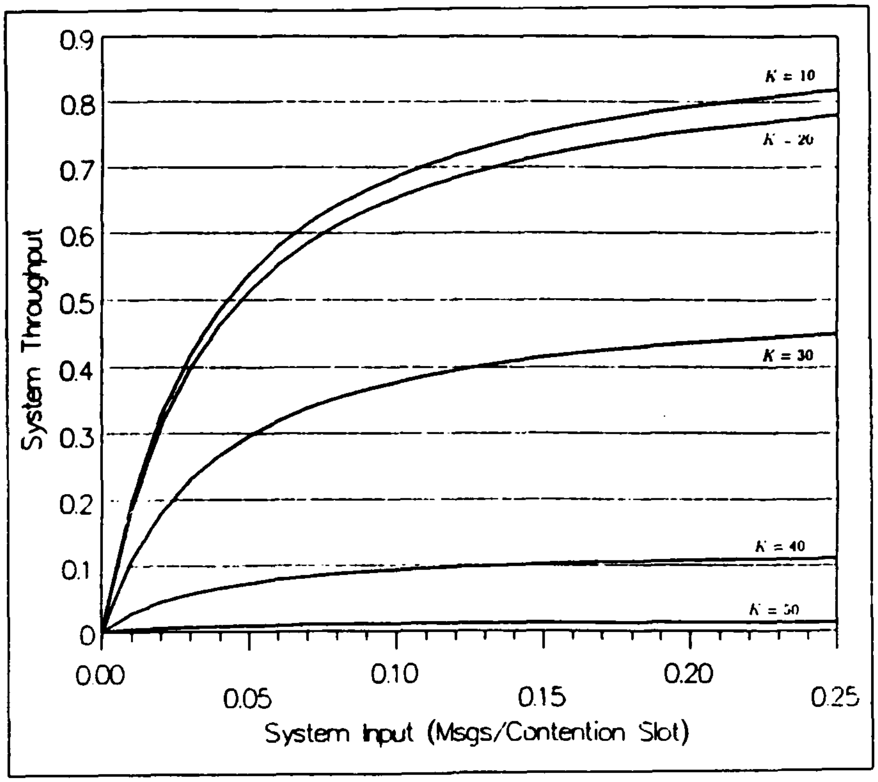

Figure 14. System Throughput for Jamming Environment, $d = .6$, $q = 75$, (64,40) R-S Code

In Figure 14 q is held constant at 75 channels and the duty cycle of the interfering nets is held constant at .6. The performance of the system holds up well as K goes from 10 to 20 interfering nets/jammers, only losing 4.7% of its throughput. However, as K increases beyond 20 the system performance rapidly goes down such that throughput at $K = 50$ is only 1.3% and if there are 75 interfering nets/jammers the throughput is 0%. Thus a system with the parameters as described by this graph could hardly be considered robust in the face of increasing interference.

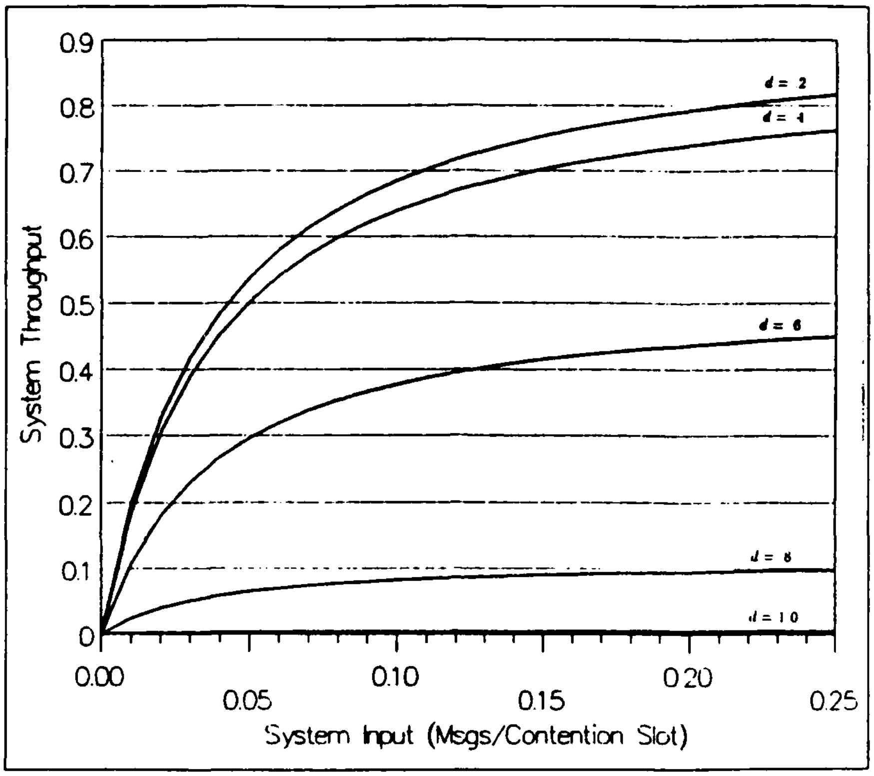

Figure 15. System Throughput for Jamming Environment, $K = 30$, $q = 75$, (64,40) R-S Code

Figure 15 demonstrates the effect of varying duty cycle on the part of the interfering nets/jammers. In this case q and K were held constant at 75 and 30 respectively. This graph again demonstrates the system's sensitivity to the level of interference. As one might suspect, if the duty cycle of the nets/jammers is fairly low, it has little effect on the throughput of the system. When the duty cycle of the nets/jammers exceeds .5 however, the performance rapidly degrades to a low of 0.4% when the duty cycle of the 30 interfering nets/jammers is 1.0. Again, a system with the parameters that were selected for this graph could not seriously be considered for an aircraft-to-aircraft network because of its lack of resiliance.

The parameters that were varied in this section an other design criteria will be covered in greater depth in the discussion section of this Chapter.

Delay. The delay of the system under jamming conditions was calculated using Equation 31 for typical values of ρ. Figure 16 depicts the delay, in contention slots for a system with a throughput of .6. Note that the delay in the jamming environment is roughly 10 times the value of delay in the LPI environment for long message lengths and the same or slightly higher for shorter message lengths. The system is also highly sensitive to changes in system throughput (See Appendix D). For example, if $m = 25$ and $\rho = .2$ delay is asymtotic to 232 contention slots, where if $m = 25$ and $\rho = 1.0$ delay is asymtotic to 26 contention slots, a difference of almost a factor of 10. It is also of interest that as the system input, Λ, approaches 0, the system delay goes to infinity, this is due to the systems dependence on retransmissions. If the system has to wait forever to get a retransmission request then the delay will approach infinity.

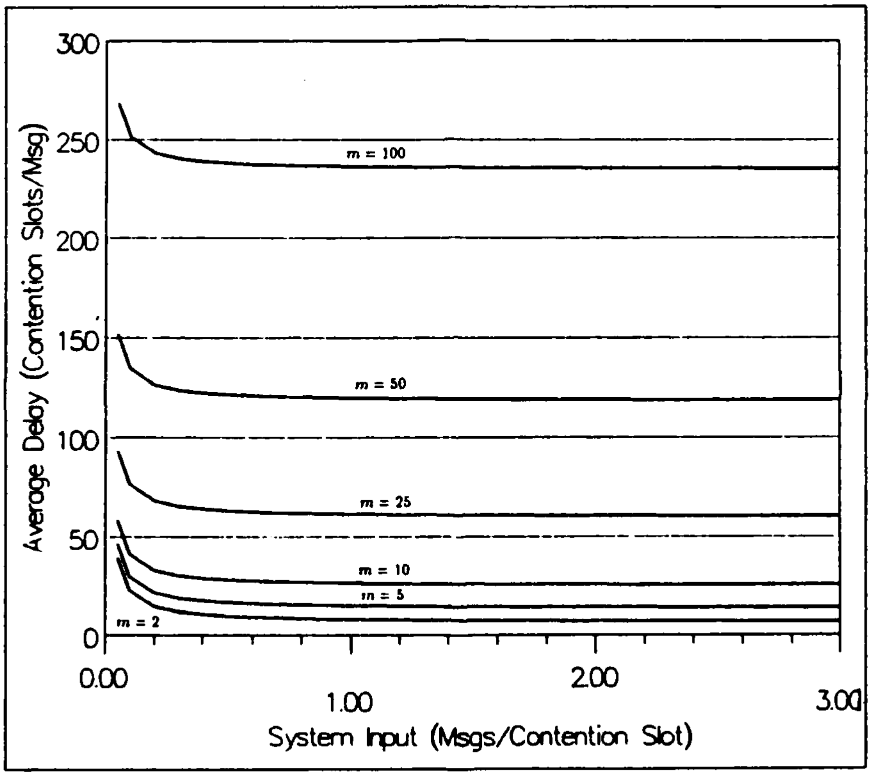

Figure 16. System Delay for Jamming Environment, $\rho = .6$

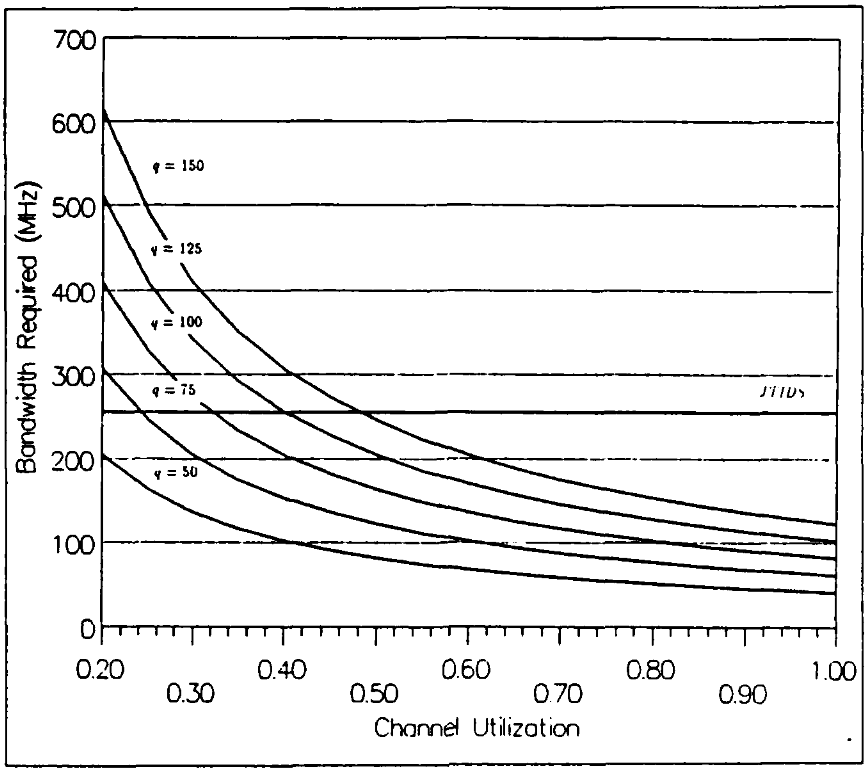

Figure 17. Required System Bandwidth, (64,40) Reed-Solomon Coding

Required Bandwidth

The calculation of the required bandwidth for the systems is very straightforward. The bandwidth of the JTIDS system was superimposed on each graph as a baseline. The required bandwidth can be designed to be less than that of JTIDS in almost all cases, by carefully selecting the R-S coding such that system throughput will remain high even in the event of interference. A discussion of the design trade-offs involved with this will be the subject of the next section of this thesis. The plots in this section were developed using Equation 32.

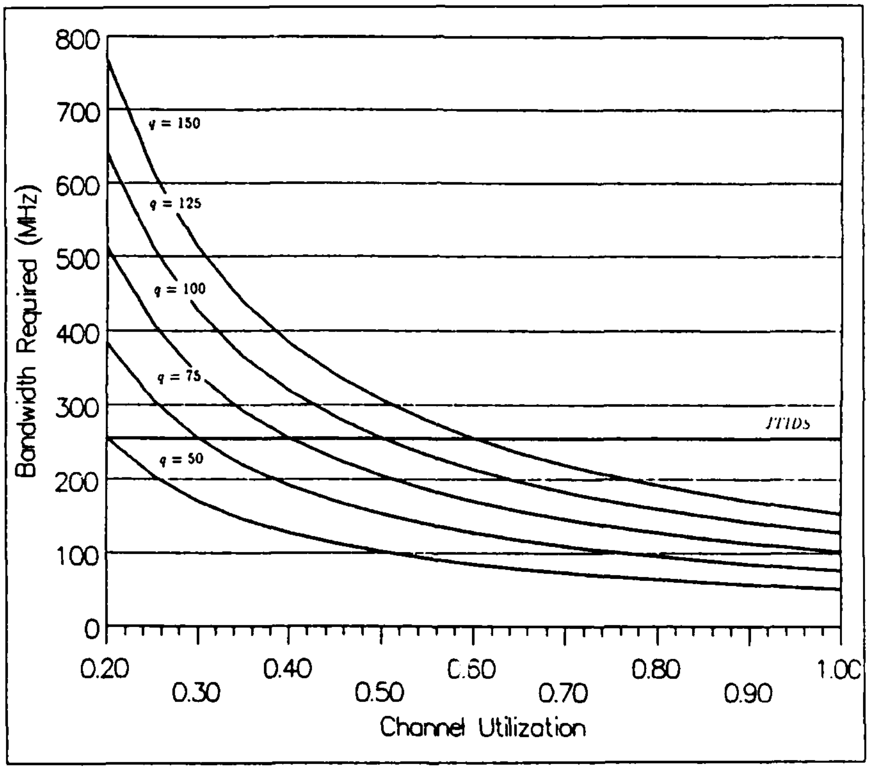

Figure 18. Required System Bandwidth, (64,32) Reed-Solomon Coding

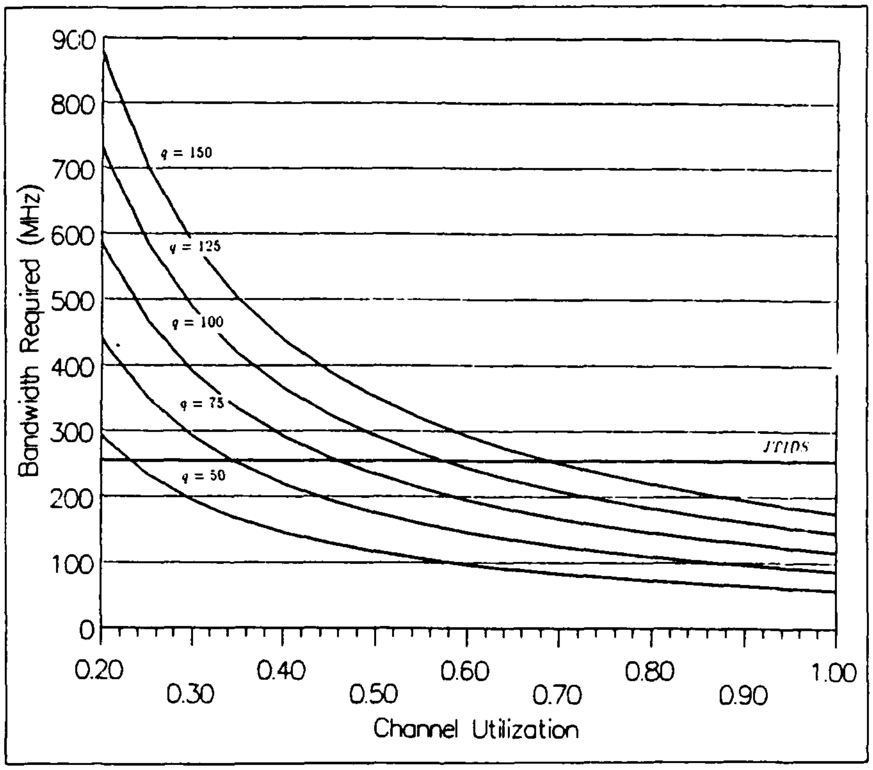

Figure 19. Required System Bandwidth, (64,28) Reed-Solomon Coding

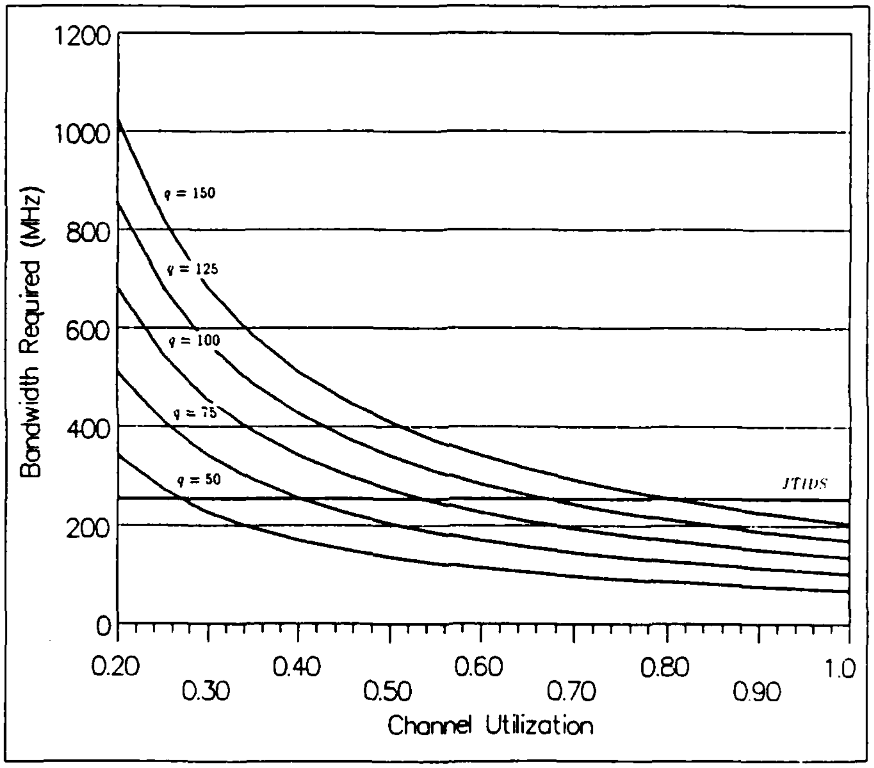

Figure 20. Required System Bandwidth, (64,24) Reed-Solomon Coding

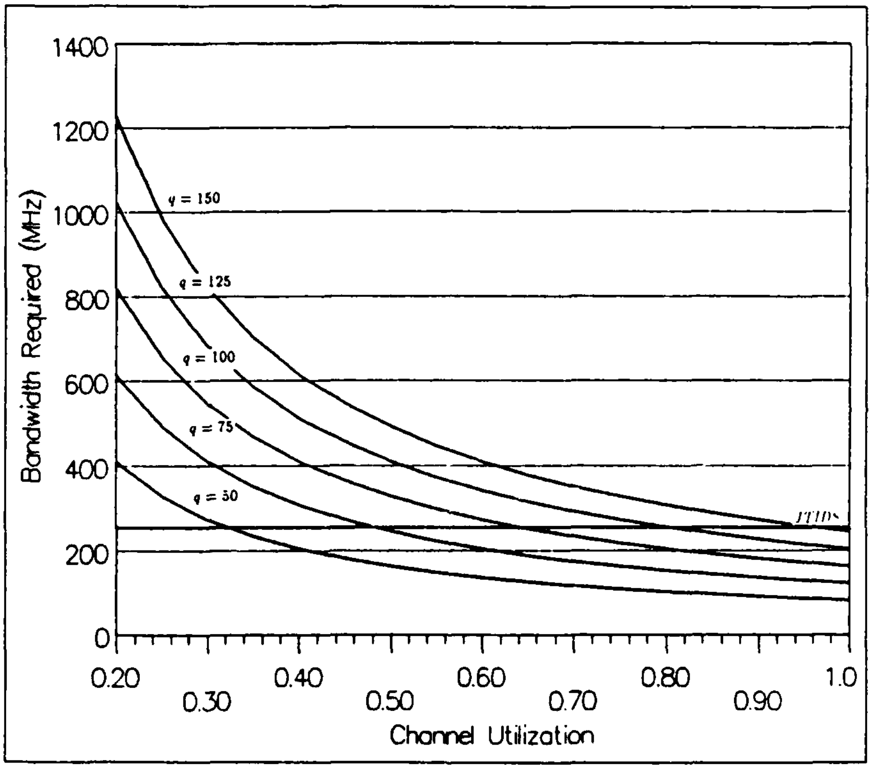

Figure 21. Required System Bandwidth, (64,20) Reed-Solomon Coding

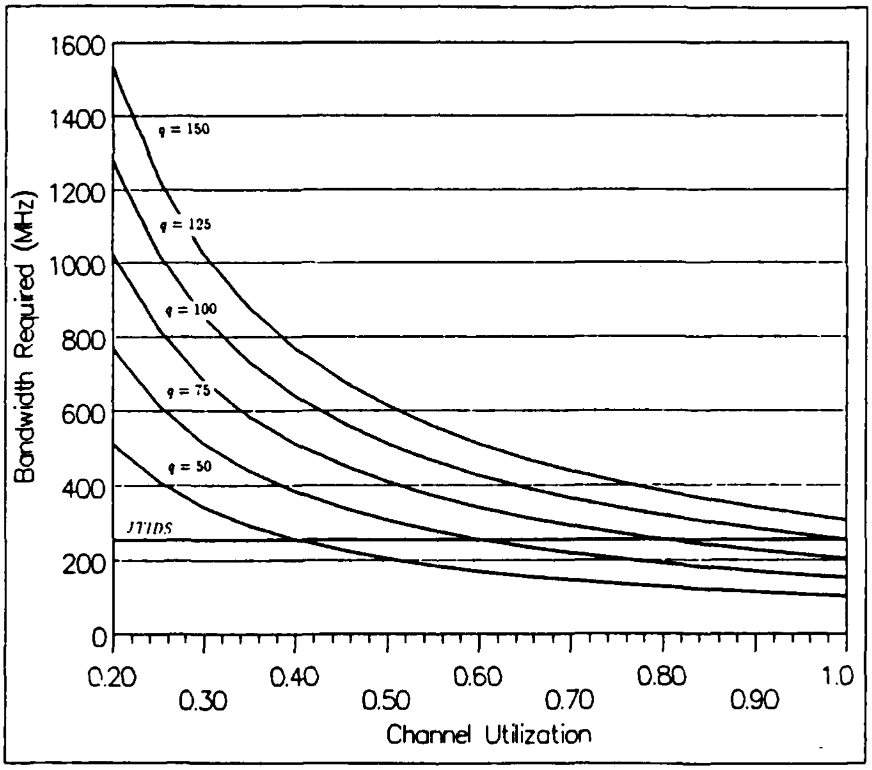

Figure 22. Required System Bandwidth, (64,16) Reed-Solomon Coding

Discussion

At this point it is necessary to limit the number of system parameters to a manageable number based on design criteria that can either be assumed from the problem statement of this thesis or made on the basis of the information presented thus far in this chapter. Based on the limited set of parameters, baselines for the throughput and delay of the aircraft-to-aircraft data network proposed by this thesis can be given.

Interference Levels. The actual determination of interference levels that could be expected in peacetime and wartime operating scenarios of the aircraft-to-aircraft data network are not a design descision, but rather a design consideration. Real world measures of these parameters are not available in the open literature. Taking this into consideration, to account for jammers in addition to interfering nets, the number of interfering nets/jammers, K, was increased from 20 to 87.5% over the values found in (25). In order to proceed with the analysis without having real-world data to draw on, a psuedo-worst case (i.e. worse than any case in the open literature) of $K = 50$ and $d = .6$ is selected.

Frequency Hopped Channels. The determination of an appropriate number of frequency hopped channels, q, is a balance between the added bandwidth that more channels add to the system vice the added LPI and Anti-Jam capability as well as the resistance to interference that those added channels give. Since JTIDS has been the baseline to which other parameters have been compared throughout this work, it seems a likely place to begin to pick a required number of frequency hopped channels. JTIDS utilizes 85 frequency hopped channels and thus any system that is proposed should exceed that. In this work three values of q have exceeded 85, those being 100, 125, and 150. These numbers of frequency hopped channels would increase the LPI quality factor, Q_{LPI}, by 17.6%, 47.1%, and 76.5%, respectively. A similar increase in the system's Anti-Jam capability would also be realized. A second consideration in selecting the number of frequency hopped channels is the increase in resistance to interference that is gained by adding more channels. Based on the information in Figure 13 as well as the information in Appendix C, the gain in capability to resist interference between 125 channels and 150 channels is marginal compared to the 20% gain in bandwidth (unless, of course, a low overhead R-S Code is selected or extremely high interference is encountered).

Reed-Solomon Code. The selection of an appropriate R-S Code is again a trade-off between required bandwidth and resistance to interference. However, by comparing Figures 17-22 and the data in Appendix C, a reasonable code selection can be made. The goals being, to use less bandwidth than JTIDS on one hand and to maintain a high enough system utilization on the other. Taking into account the psuedo-worst case scenario for interference and the already selected number of frequency hopped channels, all of the codes except the (64,40) code meet these requirements. Thus in order to conserve bandwidth, the (64,32) R-S Code is selected. If interference much in excess of the psuedo-worst case were expected, a higher overhead coding would need to be selected to insure adequate throughput levels (P_s drops to very low levels for increased interference with this code).

Throughput and Delay. The throughput and delay of the system now remain dependent on but a single variable, m, the message length. At this point, the actual operational requirements of the system described herein would have to be further defined to include a maximum acceptable message delay, a minimum required message size, and the operational range of the system. These parameters can be related to one another with the relationship of τ_{max} to the size of the contention slot and the free space propagation of radio waves. Doing this for representative operational ranges and message sizes results in the data found in Tables 2 and 3.

m : (contention slots)	$\tau_{max} = 10$ (kilometers)	$\tau_{max} = 25$ (kilometers)	$\tau_{max} = 50$ (kilometers)
2	.0003 sec.	.0008 sec.	.0015 sec.
5	.0006 sec.	.0015 sec.	.0030 sec.
10	.0011 sec.	.0028 sec.	.0055 sec.
25	.0026 sec.	.0065 sec.	.0130 sec.
50	.0051 sec.	.0128 sec.	.0255 sec.
100	.0101 sec.	.0253 sec.	.0505 sec.

Table 2. Seconds of Delay per Contention Slot of Delay

m : (contention slots)	$\tau_{max} = 10$ (kilometers)	$\tau_{max} = 25$ (kilometers)	$\tau_{max} = 50$ (kilometers)
2	102 bits	256 bits	512 bits
5	256 bits	640 bits	1281 bits
10	512 bits	1281 bits	2562 bits
25	1281 bits	3202 bits	6405 bits
50	2562 bits	6405 bits	12810 bits
100	5124 bits	12810 bits	25620 bits

Table 3. Bits per Message

Using this data, the minimum message size required, the maximum allowable message delay, and the worst case, that is jamming environment, delay (see Figure 16 and Appendix D) an optimum message length can be determined. Once that optimal message length is determined, a maximum throughput could be determined using the data in Appendix C and Equation 25.

If, for example, the network were to operate over a 50 kilometer range, have a maximum message delay of .5 second, and each packet carry a minimum of 512 bytes of information, the design tradeoffs would be as follows: In order to transmit a message that contains a minimum of 512 bytes a message length of between $m = 10$ and $m = 25$ will be required. Since the parameter of q and the R-S Code were selected to provide a minimum channel throughput of .8 at the chosen interference level, maximum delay can be calculated using $\rho = .8$ (see Figure 42) and Table 3. At this throughput, the delay in contention slots and seconds is shown in Table 4.

If, however, the system throughput were allowed to fall to $\rho = .6$, the delays experienced by the system would increase to those found in Table 5.

Note that in both cases, a message length of between $m = 10$ and $m = 25$ is required to meet the maximum delay requirements. Remember also that the same message length was required to meet the minimum message size requirement. Keeping the message size above $m = 10$ also serves the purpose of keeping channel utilization high. Note from Appendix C and Figures 7 and 8 that channel utilization will not fall below 70% for the system parameters and interference selected.

m : (contention slots)	Delay: (contention slots)	Delay: (seconds)
10	16.25	.0894
25	38.75	.504
50	76.25	1.944
100	151.25	7.638

Table 4. Delay per Message, $\rho = .8$

m : (contention slots)	Delay: (contention slots)	Delay: (seconds)
10	25.22	.139
25	60.22	.783
50	118.55	3.023
100	235.22	11.879

Table 5. Delay per Message, $\rho = .6$

q :	Reed-Solomon Coding				
	(64,32)	(64,28)	(64,24)	(64,20)	(64,16)
100	.6533	.7817	.8136	.8177	.8177
125	.7924	.8152	.8177	.8177	.8177
150	.8144	.8177	.8177	.8177	.8177

Table 6. Maximum System Throughput

Bandwidth Required. Reviewing quickly the design decisions that have been made, the following values were selected for system parameters:

- $q = 100$, 125, or 150.
- Reed-Solomon Coding of (64,32) or better.
- $m \approx 25$.

With these selections, maximum system throughput is presented in Table 6.

Given the values for throughput found in Table 6 and the system parameters, the bandwidth required for this system can be calculated using Equation 32. These values are enumerated in Table 7. Recall that the JTIDS system uses 255 MHz of bandwidth.

q :	Reed-Solomon Coding				
	(64,32)	(64,28)	(64,24)	(64,20)	(64,16)
100	157	150	168	200	250
125	162	179	209	250	313
150	189	215	250	301	376

Table 7. System Bandwidth Required (MHz)

V. Conclusions and Recommendations for Further Research

This research has proposed several mathematical models that could be used to analyze an aircraft-to-aircraft packet radio network. Although some of the assumptions that these models are based on were violated, the results obtained are comparable to the performance parameters of the currently operational JTIDS system. In fact, the proposed slow-frequency hopping radio channel uses much less bandwidth than the JTIDS system in most cases, while the number of frequency hopped channels and thus the LPI/anti-jam capabilities are increased over that system.

The bottom line of this investigation is that the proposed network could be implemented with between 150 and 376 MHz of bandwidth used. And that the delay, which inturn determines the message size and the throughput, can be limited, depending on the system requirements, to a half a second. If data compression techniques were to be used and the minimum required message size reduced, this figure could be specified to be even lower.

The first area of further work in this area is that of reducing the number of parameters involved. At this point there are simply too many variations to cover adequately. Excellent candidate variables that should be combined are K the number of interfering nets or jammers, and d the duty cycle of those nets or jammers. These variables both are directly related to the interference level and an attempt should be made to combine them into a single interference parameter. Likewise, the parameters of m and τ_{max} seem to be related and if an ideal message length for a given network size could be found, these two parameters could be combined as well.

An ideal area for further work in this area is the verification of the models presented herein. The nature of the results obtained by this research is questionable due to the assumptions that were made to account for the small number of stations in the network and the good possibility that message arrivals in this situation would not be Poisson in nature. Thus an attempt to verify the results of these models via simulation would seem to be in order. Care should be taken however to avoid tainting the simulation with assumptions that are similar to the models, to do this would only serve to verify the results of the models and not the models themselves.

Another area of interest may be to actually look at the data that is to be shared among the flight of aircraft to determine if other channel access methods or protocols may be better suited for aircraft-to-aircraft communication. For example, if each aircraft's contribution to the flight comes at very regular intervals, a slotted ring may be the ideal solution. The integration of voice communications into the regular flow of data may be another consideration meriting more attention than this research allowed.

The design decisions of higher levels of the ISO's OSI model also deserve attention. Issues such as routing and acknowledgement schemes and how they affect and are affected by the channel access methods will play a considerable part in any aircraft-to-aircraft PRN. Aside from the network issues, the radio communications issues such as different spread spectrum schemes and hybrid spread spectrum schemes will also have a bearing on the design of future systems.

Appendix A. *Delay Calculations for the LPI Environment*

In order to calculate the delay for the LPI environment, Equations 10-16 and 18-22 had to be solved simultaneously. Solving these equations was done in two steps, first, the equations were solved and FORTRAN code was generated for them using *Macsyma*, and secondly, a FORTRAN program that would generate files of data points to plot was written around the generated code. A sample input to *Macsyma* makes up the first section of this appendix while the second part includes the actual FORTRAN program used to generate data. Note that for each different value of a (known as α in the text) a different *Macsyma* run was made, this was done to keep the generated FORTRAN code of a reasonable size.

Macsyma Input

```
writefile("solution.01");

t00:5*s*(1-s)^4+(1-s)^5$

t01:0$

t02:10*s^2*(1-s)^3$

t03:10*s^3*(1-s)^2$

t04:5*s^4*(1-s)$

t05:s^5$

t10:a*(1-s)^4$

t11:(1-a)*4*s*(1-s)^3+(1-a)*(1-s)^4$

t12:4*s*(1-s)^3*a$

t13:6*s^2*(1-s)^2$

t14:4*s^3*(1-s)$

t15:s^4$

t21:2*a*(1-a)*(1-s)^3$
```

```
t22:(1-a)^2*3*s*(1-s)^2+(1-2*a*(1-a))*(1-s)^3$

t23:3*s*(1-s)^2*(1-(1-a)^2)$

t24:3*s^2*(1-s)$

t25:s^3$

t32:3*a*(1-a)^2*(1-s)^2$

t33:(1-a)^3*2*s*(1-s)+(1-3*a*(1-a)^2)*(1-s)^2$

t34:2*s*(1-s)*(1-(1-a)^3)$

t35:s^2$

t43:4*a*(1-a)^3*(1-s)$

t44:(1-a)^4*s+(1-4*a*(1-a)^3)*(1-s)$

t45:s*(1-(1-a)^4)$

t54:5*a*(1-a)^4$

t55:1-5*a*(1-a)^4$

p0+p1+p2+p3+p4+p5=1$

p0=t00*p0+t10*p1$

p1=t01*p0+t11*p1+t21*p2$

p2=t02*p0+t12*p1+t22*p2+t32*p3$

p3=t03*p0+t13*p1+t23*p2+t33*p3+t43*p4$

p4=t04*p0+t14*p1+t24*p2+t34*p3+t44*p4+t54*p5$

p5=t05*p0+t15*p1+t25*p2+t35*p3+t45*p4+t55*p5$

programmode:false$

solve([d28,d29,d30,d31,d32,d33,d34],[p0,p1,p2,p3,p4,p5])$

temp1=e38$

subst(.01,a,%)$

fortran(%);
```

```
temp2=e39$

subst(.01,a,%)$

fortran(%);

temp3=e40$

subst(.01,a,%)$

fortran(%);

temp4=e41$

subst(.01,a,%)$

fortran(%);

temp5=e36$

subst(.01,a,%)$

fortran(%);

closefile();

quit();
```

FORTRAN Code

```
 INTEGER I

 REAL s,N,temp1,temp2,temp3,temp4,temp5

 OPEN(UNIT=15,FILE="data.01")

 DO 10 I = 0,50

 s = FLOAT(50-I)*0.02

 temp1 = -(0.0004341033960042261*s**11-0.004232508111041205*s
1   **10+0.01844939433017961*s**9-0.0472087443154596*s**8+0.0781386
2   112807607l*s**7-0.0867121533518441 8*s**6+0.0644643543066275 9*s*
3   *5-0.0309298669653011 1*s**4+0.0086820679200845 23*s**3-0.0010852
4   58490010565*s**2)/(196.8954093372043*s**14-1559.377450872575*s*
5   *13+5255.197144158542*s**12-9696.215120779925*s**11+10426.84046
6   598844*s**10-6361.421807391454*s**9+1957.349859947388*s**8-284.
7   551003940435 5*s**7+76.19268852339109*s**6-2.833984360583399*s**
8   5+0.6755768686434346*s**4-0.0100708667781646*s**3+0.00168220547
9   0551225*s**2-1.085258490010565e-05*s+1.085258490010565e-06)

 temp2 = (0.06511550940063392*s**12-0.6131710468559694*s**11+
1   2.555783743974881*s**10-6.158841930809958*s**9+9.38227889028073
2   4*s**8-9.247882232830941*s**7+5.756090446739371*s**6-2.06851364
3   4167107*s**5+0.3313327056623166*s**4-0.002740551742450923*s**3+
4   0.0005481103484901846*s**2)/(196.8954093372043*s**14-1559.37745
5   0872575*s**13+5255.197144158542*s**12-9696.215120779925*s**11+1
6   0426.84046598844*s**10-6361.421807391454*s**9+1957.349859947388
7   *s**8-284.5510039404355*s**7+76.19268852339109*s**6-2.833984360
8   583399*s**5+0.6755768686434346*s**4-0.0100708667781646*s**3+0.0
9   01682205470551225*s**2-1.085258490010565e-05*s+1.08525849001056
1   5e-06)

 temp3 = -(4.341033960042262*s**13-38.70753435548294*s**12+14
1   9.9463265777331*s**11-327.6100398317982*s**10+436.611249243984*
2   s**9-355.6314835610329*s**8+164.7824178465126*s**7-34.603536171
3   02374*s**6+1.141360062540296*s**5-0.2683117875712894*s**4-0.001
4   481983903172114*s**3)/(196.8954093372043*s**14-1559.37745087257
5   5*s**13+5255.197144158542*s**12-9696.215120779925*s**11+10426.8
6   4046598844*s**10-6361.421807391454*s**9+1957.349859947388*s**8-
7   284.5510039404355*s**7+76.19268852339109*s**6-2.833984360583399
8   *s**5+0.6755768686434346*s**4-0.0100708667781646*s**3+0.0016822
9   05470551225*s**2-1.085258490010565e-05*s+1.085258490010565e-06)

 temp4 = (108.5258490010565*s**14-859.1569688992213*s**13+289
1   2.75254438261*s**12-5326.688394566419*s**11+5701.725830440954*s
2   **10-3435.736550748948*s**9+1011.002450156312*s**8-117.92594682
```

```
3     21879*s**7+29.89778870420217*s**6+0.3616037974802357*s**5+0.044
4     77460416162534*s**4)/(196.8954093372043*s**14-1559.377450872575
5     *s**13+5255.197144158542*s**12-9696.215120779925*s**11+10426.84
6     046598844*s**10-6361.421807391454*s**9+1957.349859947388*s**8-2
7     84.5510039404355*s**7+76.19268852339109*s**6-2.833984360583399*
8     s**5+0.6755768686434346*s**4-0.0100708667781646*s**3+0.00168220
9     5470551225*s**2-1.085258490010565e-05*s+1.085258490010565e-06)

      temp5 = -(-88.36956033614775*s**14+695.8794480133113*s**13-2
1     323.671949911049*s**12+4218.96679448552*s**11-4394.94457837834*
2     s**10+2482.896705220797*s**9-581.2863897588147*s**8-7.483511803
3     395519*s**7-5.848333143791055*s**6-0.07902338808987875*s**5)/(1
4     96.8954093372043*s**14-1559.377450872575*s**13+5255.19714415854
5     2*s**12-9696.215120779925*s**11+10426.84046598844*s**10-6361.42
6     1807391454*s**9+1957.349859947388*s**8-284.5510039404355*s**7+7
7     6.19268852339109*s**6-2.833984360583399*s**5+0.6755768686434346
8     *s**4-0.0100708667781646*s**3+0.001682205470551225*s**2-1.08525
9     8490010565e-05*s+1.085258490010565e-06)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

10    CONTINUE

      CLOSE(UNIT=15)

      OPEN(UNIT=15,FILE="data.02")

      DO 20 I = 0,50

      s = FLOAT(50-I)*0.02

      temp1 = -(0.003137559578448l8*s**11-0.03059120588986976*s**1
1     0+0.1333462820840476*s**9-0.3412096041562396*s**8+0.56476072412
2     06724*s**7-0.626727525795024*s**6+0.4659275973995547*s**5-0.223
3     5511199644328*s**4+0.0627511915689636*s**3-0.00784389894612045*
4     s**2)/(179.3419442424815*s**14-1420.652774707165*s**13+4796.463
5     172628691*s**12-8900.270528159428*s**11+9721.968809659856*s**10
6     -6208.617936744554*s**9+2232.097930001407*s**8-534.505447291886
7     9*s**7+150.6315918387017*s**6-10.48312019449705*s**5+2.58105144
8     433204*s**4-0.0736705668669l756*s**3+0.01255183910949601*s**2-0
9     .000156877978922409*s+1.56877978922409e-05)

      temp2 = (0.2353169683836135*s**12-2.215901452279027*s**11+9.
1     23619100905683*s**10-22.25706325961678*s**9+33.90685399030894*s
2     **8-33.42581488147032*s**7+20.81714752450752*s**6-7.49932767098
3     7302*s**5+1.218605729129427*s**4-0.0200099462911236*s**3+0.0040
4     01989258224719*s**2)/(179.3419442424815*s**14-1420.652774707165
5     *s**13+4796.463172628691*s**12-8900.270528159428*s**11+9721.968
6     809659856*s**10-6208.617936744554*s**9+2232.097930001407*s**8-5
```

```
     7    34.5054472918869*s**7+150.6315918387017*s**6-10.48312019449705*
     8    s**5+2.58105144433204*s**4-0.07367056686691756*s**3+0.012551839
     9    10949601*s**2-0.000156877978922409*s+1.56877978922409e-05)

      temp3 = -(7.84389894612045*s**13-69.9412961474904*s**12+270.
     1    9397092989072*s**11-592.0402702871528*s**10+789.5340731618603*s
     2    **9-644.6343542251059*s**8+301.398370060161*s**7-66.26165509526
     3    829*s**6+4.130463503140115*s**5-0.9579661084313566*s**4-0.01097
     4    310674023854*s**3)/(179.3419442424815*s**14-1420.652774707165*s
     5    **13+4796.463172628691*s**12-8900.270528159428*s**11+9721.96880
     6    9659856*s**10-6208.617936744554*s**9+2232.097930001407*s**8-534
     7    .5054472918869*s**7+150.6315918387017*s**6-10.48312019449705*s*
     8    *5+2.58105144433204*s**4-0.07367056686691756*s**3+0.01255183910
     9    949601*s**2-0.000156877978922409*s+1.56877978922409e-05)

      temp4 = (98.04873682650563*s**14-776.1970141200455*s**13+261
     1    6.414959061011*s**12-4837.122184678948*s**11+5237.697734931068*
     2    s**10-3267.719494199949*s**9+1090.713433595947*s**8-210.2789525
     3    666011*s**7+51.57246803995146*s**6+1.3045198618097*s**5+0.1776
     4    340492494526*s**4)/(179.3419442424815*s**14-1420.652774707165*s
     5    **13+4796.463172628691*s**12-8900.270528159428*s**11+9721.96880
     6    9659856*s**10-6208.617936744554*s**9+2232.097930001407*s**8-534
     7    .5054472918869*s**7+150.6315918387017*s**6-10.48312019449705*s*
     8    *5+2.58105144433204*s**4-0.07367056686691756*s**3+0.01255183910
     9    949601*s**2-0.000156877978922409*s+1.56877978922409e-05)

      temp5 = -(-81.29320741597591*s**14+636.6118616409989*s**13-2
     1    109.871600451806*s**12+3789.989595169716*s**11-3882.96400653889
     2    *s**10+2128.973802963065*s**9-462.5013726349833*s**8-11.1643334
     3    7621328*s**7-11.35029921562198*s**6-0.3120320402890004*s**5)/(1
     4    79.3419442424815*s**14-1420.652774707165*s**13+4796.46317262669
     5    1*s**12-8900.270528159428*s**11+9721.968809659856*s**10-6208.61
     6    7936744554*s**9+2232.097930001407*s**8-534.5054472918869*s**7+1
     7    50.6315918387017*s**6-10.48312019449705*s**5+2.58105144433204*s
     8    **4-0.07367056686691756*s**3+0.01255183910949601*s**2-0.0001568
     9    77978922409*s+1.56877978922409e-05)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

 20   CONTINUE

      CLOSE(UNIT=15)

      OPEN(UNIT=15,FILE="data.03")

      DO 30 I = 0,50

      s = FLOAT(50-I)*0.02
```

```
 temp1 = -(0.00955701668455827*s**11-0.0931809126744431 4*s**1
1    0+0.4061732090937265*s**9-1.039325564445712*s**8+1.720263003220
2    489*s**7-1.909014082740515*s**6+1.419216977656903*s**5-0.680937
3    4387747768*s**4+0.1911403336911654*s**3-0.02389254171139568*s**
4    2)/(163.2258699824595*s**14-1292.998817649775*s**13+4372.476731
5    157357*s**12-8157.564074035253*s**11+9046.377790554444*s**10-60
6    27.614954249354*s**9+2450.552588089235*s**8-751.6269974142396*s
7    **7+221.8605222965722*s**6-21.77999958770691*s**5+5.52965919856
8    1607*s**4-0.2270699716680155*s**3+0.03943377799263805*s**2-0.00
9    07167762513418703*s+7.167762513418703e-05)

 temp2 = (0.4778508342279135*s**12-4.499762022312852*s**11+18
1    .7556452434456 1*s**10-45.19672473739015*s**9+68.85526732893195*
2    s**8-67.88807096686927*s**7+42.30498824572226*s**6-15.278500427
3    37136*s**5+2.518569474216709*s**4-0.06157871575101978*s**3+0.01
4    231574315020396*s**2)/(163.2258699824595*s**14-1292.9988176497 7
5    5*s**13+4372.476731157357*s**12-8157.564074035253*s**11+9046.37
6    7790554444*s**10-6027.614954249354*s**9+2450.552588089235*s**8-
7    751.6269974142396*s**7+221.8605222965722*s**6-21.77999958770691
8    *s**5+5.529659198561607*s**4-0.2270699716680155*s**3+0.03943377
9    799263805*s**2-0.0007167762513418703*s+7.167762513418703e-05)

 temp3 = -(10.61890742728697*s**13-94.68483467189469*s**12+36
1    6.7896820605682*s**11-801.5906221450066*s**10+1069.688262778524
2    *s**9-875.4834996622065*s**8+413.0326901481456*s**7-94.81389635
3    723668*s**6+8.397415315956954*s**5-1.919854501347201*s**4-0.034
4    2503927902721*s**3)/(163.2258699824595*s**14-1292.998817649775*
5    s**13+4372.476731157357*s**12-8157.564074035253*s**11+9046.3777
6    90554444*s**10-6027.614954249354*s**9+2450.552588089235*s**8-75
7    1.6269974142396*s**7+221.8605222965722*s**6-21.77999958770691*s
8    **5+5.529659198561607*s**4-0.2270699716680155*s**3+0.0394337779
9    9263805*s**2-0.0007167762513418703*s+7.167762513418703e-05)

 temp4 = (88.49089522739139*s**14-700.5069727109167*s**13+236
1    3.999984187559*s**12-4388.113285437404*s**11+4806.013506541333*
2    s**10-3097.810689682654*s**9+1143.569562553632*s**8-280.6207988
3    823049*s**7+66.36843776074378*s**6+2.640579009675593*s**5+0.395
4    2454829447402*s**4)/(163.2258699824595*s**14-1292.998817649775*
5    s**13+4372.476731157357*s**12-8157.564074035253*s**11+9046.3777
6    90554444*s**10-6027.614954249354*s**9+2450.552588089235*s**8-75
7    1.6269974142396*s**7+221.8605222965722*s**6-21.77999958770691*s
8    **5+5.529659198561607*s**4-0.2270699716680155*s**3+0.0394337779
9    9263805*s**2-0.0007167762513418703*s+7.167762513418703e-05)

 temp5 = -(-74.73497475506814*s**14+581.8729375115709*s**13-1
1    913.314061463676*s**12+3398.151787498283*s**11-3419.92476403436
2    *s**10+1814.51238706544*s**9-361.6017074868873*s**8-11.64342690
3    131672*s**7-16.44913354885077*s**6-0.6926168851365258*s**5)/(16
4    3.2258699824595*s**14-1292.998817649775*s**13+4372.476731157357
5    *s**12-8157.564074035253*s**11+9046.377790554444*s**10-6027.614
6    954249354*s**9+2450.552588089235*s**8-751.6269974142396*s**7+22
```

```
     7  1.8605222965722*s**6-21.77999958770691*s**5+5.529659198561607*s
     8  **4-0.2270699716680155*s**3+0.03943377799263805*s**2-0.00071677
     9  62513418703*s+7.167762513418703e-05)

        N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

        WRITE(15,15) N

30      CONTINUE

        CLOSE(UNIT=15)

        OPEN(UNIT=15,FILE="data.04")

        DO 40 I = 0,50

        s = FLOAT(50-I)*0.02

        temp1 = -(0.02042365857765891*s**11-0.1991306711321744*s**10
     1  +0.8680054895505038*s**9-2.221072870320407*s**8+3.6762585439786
     2  04*s**7-4.079625800887368*s**6+3.032913298782349*s**5-1.4551856
     3  73658198*s**4+0.4084731715531782*s**3-0.05105914644414728*s**2)
     4  /(148.4408446413032*s**14-1175.637636270728*s**13+3981.02945365
     5  2654*s**12-7465.530526431051*s**11+8400.756239945742*s**10-5823
     6  .844358694556*s**9+2618.637633970416*s**8-937.7959398255656*s**
     7  7+288.7300264703615*s**6-35.69976756209872*s**5+9.3330175073819
     8  56*s**4-0.4909235728303505*s**3+0.08684309824375384*s**2-0.0020
     9  42365857765891*s+0.0002042365857765891)

        temp2 = (0.7658871966622092*s**12-7.212104435235803*s**11+30
     1  .06107246899171*s**10-72.44016401763395*s**9+110.3622175745892*
     2  s**8-108.8277838507208*s**7+67.85813749038053*s**6-24.569554895
     3  70191*s**5+4.108665690427476*s**4-0.1329665271983002*s**3+0.026
     4  59330543966004*s**2)/(148.4408446413032*s**14-1175.637636270728
     5  *s**13+3981.029453652654*s**12-7465.530526431051*s**11+8400.756
     6  239945742*s**10-5823.844358694556*s**9+2618.637633970416*s**8-9
     7  37.7959398255656*s**7+288.7300264703615*s**6-35.69976756209872*
     8  s**5+9.333017507381956*s**4-0.4909235728303505*s**3+0.086843098
     9  24375384*s**2-0.002042365857765891*s+0.0002042365857765891)

        temp3 = -(12.76478661103682*s**13-113.8184239030839*s**12+44
     1  0.9059236456303*s**11-963.6915266855806*s**10+1286.858206736656
     2  *s**9-1055.797355432799*s**8+502.5946974434658*s**7-120.1795794
     3  99397*s**6+13.47159450199746*s**5-3.033298901711223*s**4-0.0750
     4  245162143187*s**3)/(148.4408446413032*s**14-1175.637636270728*s
     5  **13+3981.029453652654*s**12-7465.530526431051*s**11+8400.75623
     6  9945742*s**10-5823.844358694556*s**9+2618.637633970416*s**8-937
     7  .7959398255656*s**7+288.7300264703615*s**6-35.69976756209872*s*
     8  *5+9.333017507381956*s**4-0.4909235728303505*s**3+0.08684309824
     9  375384*s**2-0.002042365857765891*s+0.0002042365857765891)
```

```
      temp4 = (79.77991631898013*s**14-631.5157696878886*s**13+213
     1   3.654594403487*s**12-3976.676462839009*s**11+4404.845246480746*
     2   s**10-2927.626542682631*s**9+1173.547243488921*s**8-332.1382449
     3   434497*s**7+75.47201292851793*s**6+4.211761773754397*s**5+0.692
     4   9775585719747*s**4)/(148.4408446413032*s**14-1175.637636270728*
     5   s**13+3981.029453652654*s**12-7465.530526431051*s**11+8400.7562
     6   39945742*s**10-5823.844358694556*s**9+2618.637633970416*s**8-93
     7   7.7959398255656*s**7+288.7300264703615*s**6-35.69976756209872*s
     8   **5+9.333017507381956*s**4-0.4909235728303505*s**3+0.0868430982
     9   4375384*s**2-0.002042365857765891*s+0.0002042365857765891)

      temp5 = -(-68.66092832232309*s**14+531.3570799718026*s**13-1
     1   732.790548149421*s**12+3040.715611852598*s**11-3001.95905940270
     2   6*s**10+1536.049397402227*s**9-276.7005539574258*s**8-9.4655533
     3   46342421*s**7-21.09778106816555*s**6-1.2140009802443*s**5)/(148
     4   .4408446413032*s**14-1175.637636270728*s**13+3981.029453652654*
     5   s**12-7465.530526431051*s**11+8400.756239945742*s**10-5823.8443
     6   58694556*s**9+2618.637633970416*s**8-937.7959398255656*s**7+288
     7   .7300264703615*s**6-35.69976756209872*s**5+9.333017507381956*s*
     8   *4-0.4909235728303505*s**3+0.08684309824375384*s**2-0.002042365
     9   857765891*s+0.0002042365857765891)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

40    CONTINUE

      CLOSE(UNIT=15)

      OPEN(UNIT=15,FILE="data.06")

      DO 50 I = 0,50

      s = FLOAT(50-I)*0.02

      temp1 = -(0.0558436150293592*s**11-0.5444752465362522*s**10+
     1   2.373353638747766*s**9-6.072993134442813*s**8+10.05185070528466
     2   *s**7-11.1547621021145*s**6+8.292776831859841*s**5-3.9788575708
     3   41843*s**4+1.116872300587184*s**3-0.139609037573398*s**2)/(122.
     4   473056216509*s**14-968.9135939128334*s**13+3287.463503593649*s*
     5   *12-6223.449272246091*s**11+7200.842247561491*s**10-5366.976774
     6   654185*s**9+2825.495093131818*s**8-1225.16654334336*s**7+406.11
     7   64943442993*s**6-67.96936775839752*s**5+18.77204300359257*s**4-
     8   1.372981852551934*s**3+0.2515636040870442*s**2-0.00837654225440
     9   388*s+0.0008376542254403881)

      temp2 = (1.39609037573398*s**12-13.14651770482831*s**11+54.7
     1   9654724755872*s**10-132.0468813715056*s**9+201.1830590208851*s*
     2   *8-198.445335695421*s**7+123.8931194962082*s**6-45.093224068698
     3   43*s**5+7.760183205542602*s**4-0.3713006318441436*s**3+0.074260
```

```
     4   12636882872*s**2)/(122.473056216509*s**14-968.9135939128334*s**
     5   13+3287.463503593649*s**12-6223.449272246091*s**11+7200.8422475
     6   61491*s**10-5366.976774654185*s**9+2825.495093131818*s**8-1225.
     7   16654334336*s**7+406.1164943442993*s**6-67.96936775839752*s**5+
     8   18.77204300359257*s**4-1.372981852551934*s**3+0.251563604087044
     9   2*s**2-0.00837654225440388*s+0.000837654225440388l)

      temp3 = -(15.51211528593311*s**13-138.3137279617557*s**12+53
     1   5.7827154478323*s**11-1171.366300727304*s**10+1566.284959681197
     2   *s**9-1291.433979593962*s**8+625.8716175311336*s**7-161.3518355
     3   66134*s**6+24.58198438447735*s**5-5.351838894485076*s**4-0.2157
     4   095869322398*s**3)/(122.473056216509*s**14-968.9135939128334*s*
     5   *13+3287.463503593649*s**12-6223.449272246091*s**11+7200.842247
     6   561491*s**10-5366.976774654185*s**9+2825.495093131818*s**8-1225
     7   .16654334336*s**7+406.1164943442993*s**6-67.96936775839752*s**5
     8   +18.77204300359257*s**4-1.372981852551934*s**3+0.25156360408704
     9   42*s**2-0.00837654225440388*s+0.000837654225440388l)

      temp4 = (64.63381369138796*s**14-511.5411558668387*s**13+173
     1   2.385083148622*s**12-3255.52674573199*s**11+3687.111242527674*s
     2   **10-2591.80118376642*s**9+1178.693775674317*s**8-389.756234886
     3   8313*s**7+80.64078912913665*s**6+7.559104935562498*s**5+1.50525
     4   5945380563*s**4)/(122.473056216509*s**14-968.9135939128334*s**1
     5   3+3287.463503593649*s**12-6223.449272246091*s**11+7200.84224756
     6   1491*s**10-5366.976774654185*s**9+2825.495093131818*s**8-1225.1
     7   6654334336*s**7+406.1164943442993*s**6-67.96936775839752*s**5+1
     8   8.77204300359257*s**4-1.372981852551934*s**3+0.2515636040870442
     9   *s**2-0.00837654225440388*s+0.000837654225440388l)

      temp5 = -(-57.83924252512103*s**14+441.8603227600616*s**13-1
     1   415.368602107538*s**12+2418.937449746411*s**11-2287.02284415819
     2   3*s**10+1074.46201965406*s**9-148.073591268067*s**8+0.940986017
     3   6362444*s**7-28.90008066336345*s**6-2.650601455886583*s**5)/(12
     4   2.473056216509*s**14-968.9135939128334*s**13+3287.463503593649*
     5   s**12-6223.449272246091*s**11+7200.842247561491*s**10-5366.9767
     6   74654185*s**9+2825.495093131818*s**8-1225.16654334336*s**7+406.
     7   1164943442993*s**6-67.96936775839752*s**5+18.77204300359257*s**
     8   4-1.372981852551934*s**3+0.2515636040870442*s**2-0.008376542254
     9   40388*s+0.000837654225440388l)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

 50   CONTINUE

      CLOSE(UNIT=15)

      OPEN(UNIT=15,FILE="data.08")

      DO 60 I = 0,50
```

```
 s = FLOAT(50-I)*0.02

 temp1 = -(0.1067553065099998*s**11-1.040864238472498*s**10+4
1    .537100526674993*s**9-11.60963958296248*s**8+19.21595517179997*
2    s**7-21.32437247537247*s**6+15.85316301673498*s**5-7.6063155888
3    37488*s**4+2.135106130199997*s**3-0.2668882662749996*s**2)/(100
4    .7203511772893*s**14-795.152448585691*s**13+2700.228647215133*s
5    **12-5154.502443055329*s**11+6122.794469685788*s**10-4870.97507
6    8335072*s**9+2892.559804140923*s**8-1412.715724572736*s**7+498.
7    4448686306824*s**6-101.6095870765713*s**5+29.52781369870659*s**
8    4-2.682420180885869*s**3+0.5080160129356296*s**2-0.021351061301
9    99997*s+0.002135106130199997)

 temp2 = (2.001661997062497*s**12-18.84898380567184*s**11+78.
1    565233384703*s**10-189.3238638888278*s**9+288.4641519317421*s**
2    8-284.6276331040389*s**7+177.9303457823598*s**6-65.112034092851
3    8*s**5+11.5313136787296*s**4-0.7252398540081511*s**3+0.14504797
4    08016302*s**2)/(100.7203511772893*s**14-795.152448585691*s**13+
5    2700.228647215133*s**12-5154.502443055329*s**11+6122.7944696857
6    88*s**10-4870.975078335072*s**9+2892.559804140923*s**8-1412.715
7    724572736*s**7+498.4448686306824*s**6-101.6095870765713*s**5+29
8    .52781369870659*s**4-2.682420180885869*s**3+0.5080160129356296*
9    s**2-0.02135106130199997*s+0.002135106130199997)

 temp3 = -(16.68051664218747*s**13-148.729351364910 7*s**12+57
1    6.1097813470945*s**11-1259.850675158323*s**10+1686.921838554999
2    *s**9-1397.953971671096*s**8+689.6739681808487*s**7-190.2745815
3    903266*s**6+35.23811123039794*s**5-7.381497631926061*s**4-0.434
4    1385389462782*s**3)/(100.7203511772893*s**14-795.152448585691*s
5    **13+2700.228647215133*s**12-5154.502443055329*s**11+6122.79446
6    9685788*s**10-4870.975078335072*s**9+2892.559804140923*s**8-141
7    2.715724572736*s**7+498.4448686306824*s**6-101.6095870765713*s*
8    *5+29.52781369870659*s**4-2.682420180885869*s**3+0.508016012935
9    6296*s**2-0.02135106130199997*s+0.002135106130199997)

 temp4 = (52.1266145068358 6*s**14-412.4544877168306*s**13+140
1    0.179066784573*s**12-2653.441698833513*s**11+3071.32236727576 2*
2    s**10-2269.065620883163*s**9+1130.121833408476*s**8-402.2572042
3    121212*s**7+73.92160855908103*s**6+10.5691713990283*s**5+2.5603
4    14511871443*s**4)/(100.7203511772893*s**14-795.152448585691*s**
5    13+2700.228647215133*s**12-5154.502443055329*s**11+6122.7944696
6    85788*s**10-4870.975078335072*s**9+2892.559804140923*s**8-1412.
7    715724572736*s**7+498.4448686306824*s**6-101.6095870765713*s**5
8    +29.52781369870659*s**4-2.682420180885869*s**3+0.5080160129356 2
9    96*s**2-0.02135106130199997*s+0.002135106130199997)

 temp5 = -(-48.59373667045343*s**14+366.0174442266729*s**13-1
1    149.318567068586*s**12+1905.99522376254*s**11-1712.013194522398
2    *s**10+721.1053034201049*s**9-64.31412777078752*s**8+16.6847511
3    6830327*s**7-34.54558793620049*s**6-4.562596609195508*s**5)/(10
```

```
     4    0.7203511772893*s**14-795.152448585691*s**13+2700.228647215133*
     5    s**12-5154.502443055329*s**11+6122.794469685788*s**10-4870.9750
     6    78335072*s**9+2892.559804140923*s**8-1412.715724572736*s**7+498
     7    .4448686306824*s**6-101.6095870765713*s**5+29.5278136987O659*s*
     8    *4-2.682420180885869*s**3+0.5080160129356296*s**2-0.02135106130
     9    199997*s+0.002135106130199997)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

60    CONTINUE

      CLOSE(UNIT=15)

      OPEN(UNIT=15,FILE="data.10")

      DO 70 I = 0,50

      s = FLOAT(50-I)*0.02

      temp1 = -(0.167365651248*s**11-1.631815099668*s**10+7.113040
     1    17804*s**9-18.20101457322*s**8+30.12581722464*s**7-33.431288836
     2    788*s**6+24.853799210328*s**5-11.92480265142*s**4+3.34731302496
     3    *s**3-0.41841412812*s**2)/(82.558106352*s**14-649.6689682548*s*
     4    *13+2205.37663665732*s**12-4240.142173476428*s**11+5164.1133136
     5    74809*s**10-4361.579859156952*s**9+2854.006516252774*s**8-1516.
     6    500565880404*s**7+564.26064702702*s**6-132.6437468490504*s**5+4
     7    0.432324241772*s**4-4.294226095704*s**3+0.839152779174*s**2-0.0
     8    41841412812*s+0.0041841412812)

      temp2 = (2.51048476872*s**12-23.64039823878*s**11+98.5365271
     1    7226*s**10-237.4500177081*s**9+361.8119946771*s**8-357.11645835
     2    042*s**7+223.54937056278*s**6-82.26486663426*s**5+14.9931729243
     3    *s**4-1.162261467*s**3+0.2324522934*s**2)/(82.558106352*s**14-6
     4    49.6689682548*s**13+2205.37663665732*s**12-4240.142173476428*s*
     5    *11+5164.113313674809*s**10-4361.579859156952*s**9+2854.0065162
     6    52774*s**8-1516.500565880404*s**7+564.26064702702*s**6-132.6437
     7    468490504*s**5+40.432324241772*s**4-4.294226095704*s**3+0.83915
     8    2779174*s**2-0.041841412812*s+0.0041841412812)

      temp3 = -(16.7365651248*s**13-149.2257630186*s**12+578.00554
     1    155[illegible]s**11-1264.3080238026*s**10+1695.270749391*s**9-1412.1438
     2    560298*s**8+709.1296800966*s**7-208.0333234674*s**6+44.12480221
     3    26*s**5-8.838926712*s**4-0.71744535*s**3)/(82.558106352*s**14-6
     4    49.6689682548*s**13+2205.37663665732*s**12-4240.142173476428*s*
     5    *11+5164.113313674809*s**10-4361.579859156952*s**9+2854.0065162
     6    52774*s**8-1516.500565880404*s**7+564.26064702702*s**6-132.6437
     7    468490504*s**5+40.432324241772*s**4-4.294226095704*s**3+0.83915
     8    2779174*s**2-0.041841412812*s+0.0041841412812)
```

```
      temp4 = (41.841412812*s**14-330.96314961*s**13+1126.2947211*
     1  s**12-2152.780393725*s**11+2545.425197073*s**10-1965.575721897*
     2  s**9+1046.590544925*s**8-384.551265285*s**7+60.436815525*s**6+1
     3  2.765586797*s**5+3.796752285*s**4)/(82.558106352*s**14-649.6689
     4  682548*s**13+2205.37663665732*s**12-4240.142173476428*s**11+516
     5  4.113313674809*s**10-4361.579859156952*s**9+2854.006516252774*s
     6  **8-1516.500565880404*s**7+564.26064702702*s**6-132.64374684905
     7  04*s**5+40.432324241772*s**4-4.294226095704*s**3+0.839152779174
     8  *s**2-0.041841412812*s+0.0041841412812)

      temp5 = -(-40.71669354*s**14+301.96925352*s**13-927.34566777
     1  *s**12+1485.548474306*s**11-1254.207566386*s**10+456.12848857*s
     2  **9-15.07081969*s**8+35.07524797*s**7-37.931178966*s**6-6.88853
     3  8014*s**5)/(82.558106352*s**14-649.6689682548*s**13+2205.376636
     4  65732*s**12-4240.142173476428*s**11+5164.113313674809*s**10-436
     5  1.579859156952*s**9+2854.006516252774*s**8-1516.500565880404*s*
     6  *7+564.26064702702*s**6-132.6437468490504*s**5+40.432324241772*
     7  s**4-4.294226095704*s**3+0.839152779174*s**2-0.041841412812*s+0
     8  .0041841412812)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

70    CONTINUE

      CLOSE(UNIT=15)

      OPEN(UNIT=15,FILE="data.15")

      DO 80 I = 0,50

      s = FLOAT(50-I)*0.02

      temp1 = -(0.3189365350319707*s**11-3.109631216561714*s**10+1
     1  3.55480273885875*s**9-34.68434818472681*s**8+57.40857630575473*
     2  s**7-63.70757287263615*s**6+47.36207545224765*s**5-22.724228121
     3  02791*s**4+6.378730700639414*s**3-0.7973413375799268*s**2)/(49.
     4  48398338459766*s**14-383.9868886313308*s**13+1293.928183078489*
     5  s**12-2518.980605204396*s**11+3253.843422340348*s**10-3145.2066
     6  29549999*s**9+2472.702687141621*s**8-1507.679236535979*s**7+620
     7  .4218638552675*s**6-184.7537691200348*s**5+62.91695807894455*s*
     8  *4-8.590242563624346*s**3+1.804571056669864*s**2-0.119601200636
     9  989*s+0.0119601200636989)

      temp2 = (3.189365350319707*s**12-30.03319038217724*s**11+125
     1  .1825900000485*s**10-301.6608060510723*s**9+459.7220006183872*s
     2  **8-454.1562453992018*s**7+285.338759846495*s**6-106.5779587898
     3  502*s**5+20.87158207194514*s**4-2.345121581117432*s**3+0.469024
     4  3162234863*s**2)/(49.48398338459766*s**14-383.9868886313308*s**
     5  13+1293.928183078489*s**12-2518.980605204396*s**11+3253.8434223
```

```
     6    40348*s**10-3145.206629549999*s**9+2472.702687141621*s**8-1507.
     7    679236535979*s**7+620.4218638552675*s**6-184.7537691200348*s**5
     8    +62.91695807894455*s**4-8.590242563624346*s**3+1.80457105666986
     9    4*s**2-0.119601200636989*s+0.0119601200636989)

      temp3 = -(14.17495711253203*s**13-126.3749744763038*s**12+48
     1    9.4059262700956*s**11-1071.126873839237*s**10+1441.47347493188*
     2    s**9-1216.893836890626*s**8+638.436764801548*s**7-213.380638111
     3    372*s**6+55.39680280267295*s**5-9.542778475412109*s**4-1.568824
     4    125776367*s**3)/(49.48398338459766*s**14-383.9868886313308*s**1
     5    3+1293.928183078489*s**12-2518.980605204396*s**11+3253.84342234
     6    0348*s**10-3145.206629549999*s**9+2472.702687141621*s**8-1507.6
     7    79236535979*s**7+620.4218638552675*s**6-184.7537691200348*s**5+
     8    62.91695807894455*s**4-8.590242563624346*s**3+1.804571056669864
     9    *s**2-0.119601200636989*s+0.0119601200636989)

      temp4 = (23.62492852088672*s**14-186.6297894409473*s**13+639
     1    .1783706717286*s**12-1249.185009727056*s**11+1554.424533371706*
     2    s**10-1313.162846763548*s**9+770.9171128610889*s**8-276.2115988
     3    775537*s**7+18.89836069732909*s**6+13.48922573918262*s**5+7.266
     4    744197182617*s**4)/(49.48398338459766*s**14-383.9868886313308*s
     5    **13+1293.928183078489*s**12-2518.980605204396*s**11+3253.84342
     6    2340348*s**10-3145.206629549999*s**9+2472.702687141621*s**8-150
     7    7.679236535979*s**7+620.4218638552675*s**6-184.7537691200348*s*
     8    *5+62.91695807894455*s**4-8.590242563624346*s**3+1.804571056669
     9    864*s**2-0.119601200636989*s+0.0119601200636989)

      temp5 = -(-25.85905486371094*s**14+183.1821420778516*s**13-5
     1    25.1854725801367*s**12+750.0375422900352*s**11-499.987833792730
     2    4*s**10+75.23509786400379*s**9+10.05281681607417*s**8+80.030836
     3    74427734*s**7-36.58490711405859*s**6-14.10779244160547*s**5)/(4
     4    9.48398338459766*s**14-383.9868886313308*s**13+1293.92818307848
     5    9*s**12-2518.980605204396*s**11+3253.843422340348*s**10-3145.20
     6    6629549999*s**9+2472.702687141621*s**8-1507.679236535979*s**7+6
     7    20.4218638552675*s**6-184.7537691200348*s**5+62.91695807894455*
     8    s**4-8.590242563624346*s**3+1.804571056669864*s**2-0.1196012006
     9    36989*s+0.0119601200636989)

      N = temp1*1.0+temp2*2.0+temp3*3.0+temp4*4.0+temp5*5.0

      WRITE(15,15) N

 80   CONTINUE

      CLOSE(UNIT=15)

 15   FORMAT(F9.7,',')

      STOP

      END
```

Appendix B. *Results of Delay Calculations for the LPI Environment*

This appendix contains graphical results of the calculations in the previous appendix.

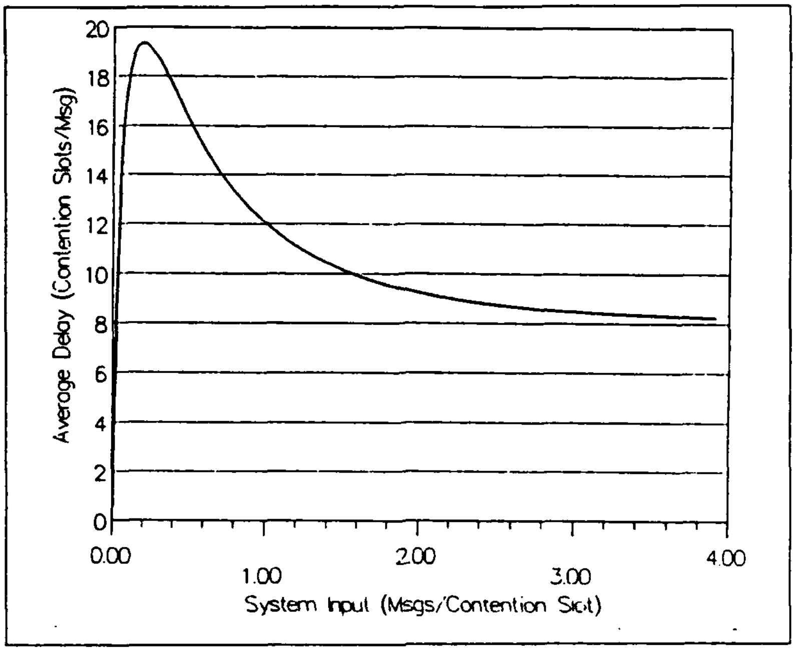

Figure 23. System Delay for LPI Environment, $\alpha = .01$

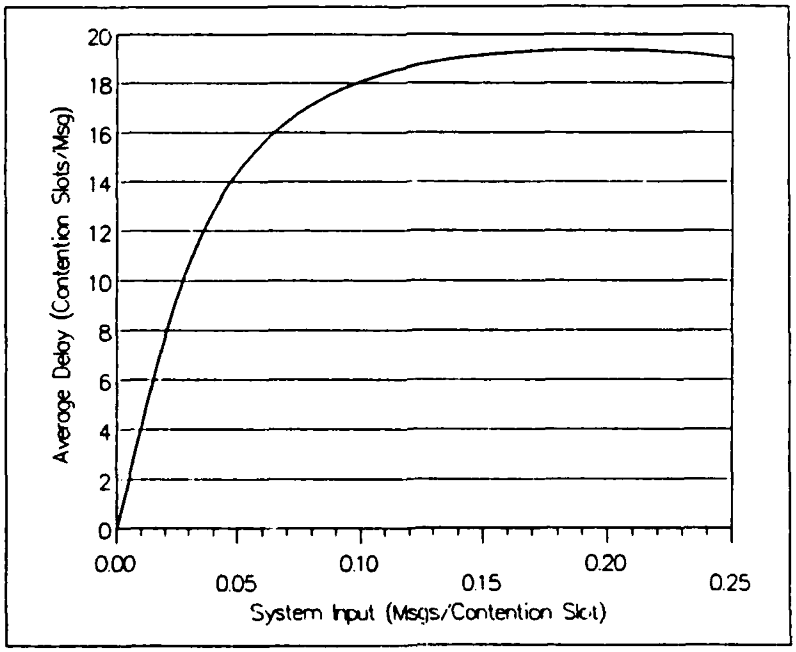

Figure 24. System Delay in Stable Region for LPI Environment, $\alpha = .01$

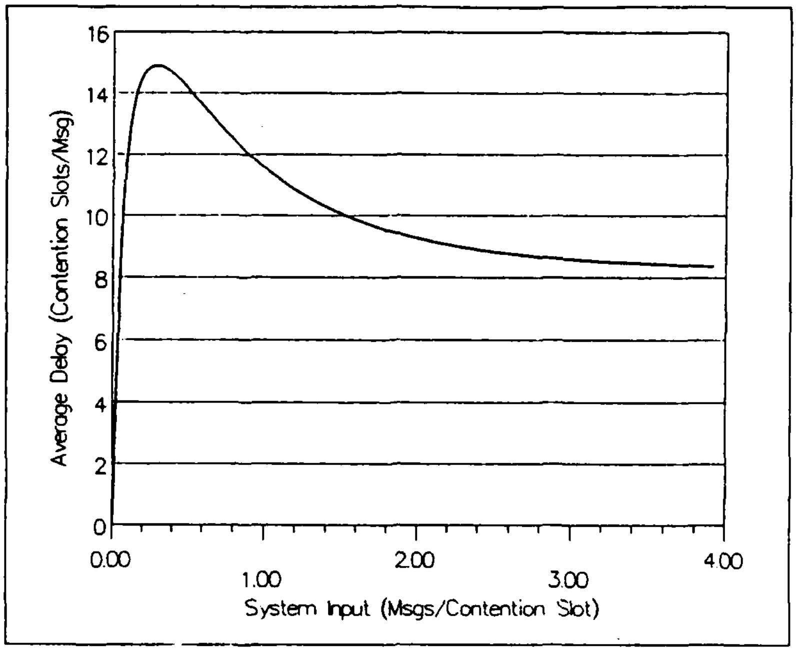

Figure 25. System Delay for LPI Environment, $\alpha = .02$

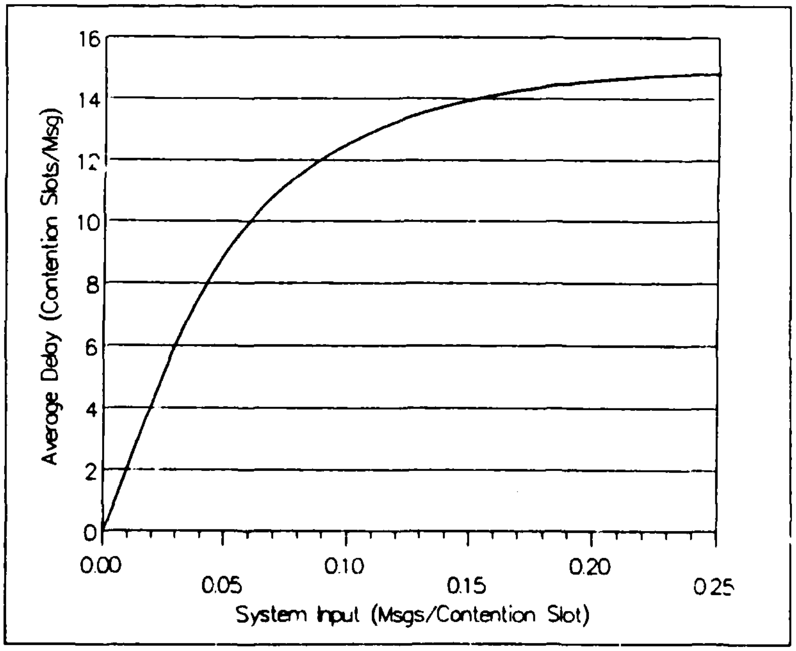

Figure 26. System Delay in Stable Region for LPI Environment, $\alpha = .02$

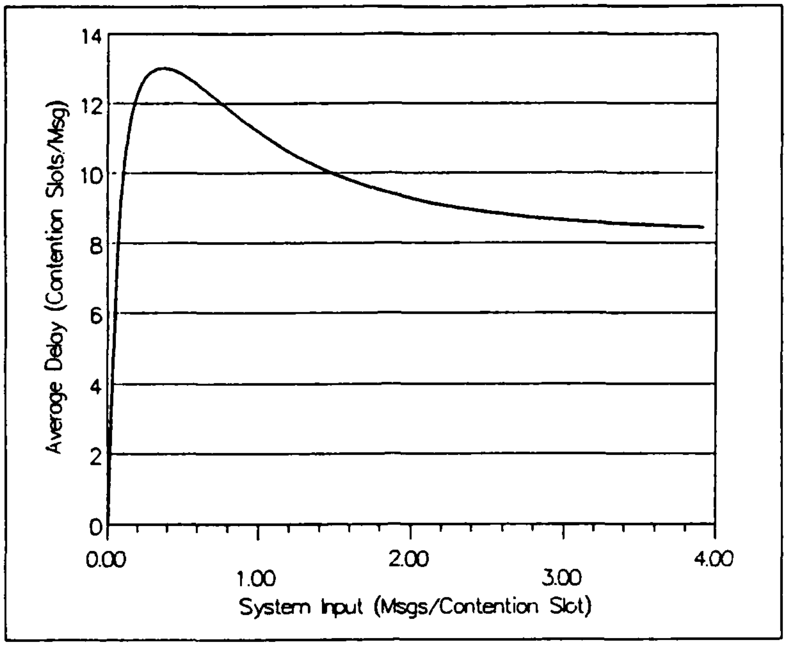

Figure 27. System Delay for LPI Environment, α = .03

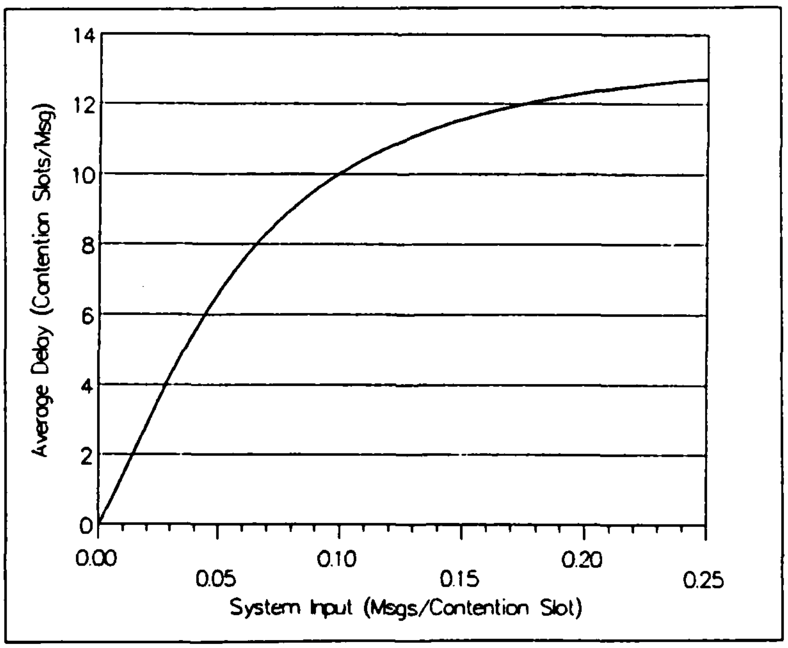

Figure 28. System Delay in Stable Region for LPI Environment, α = .03

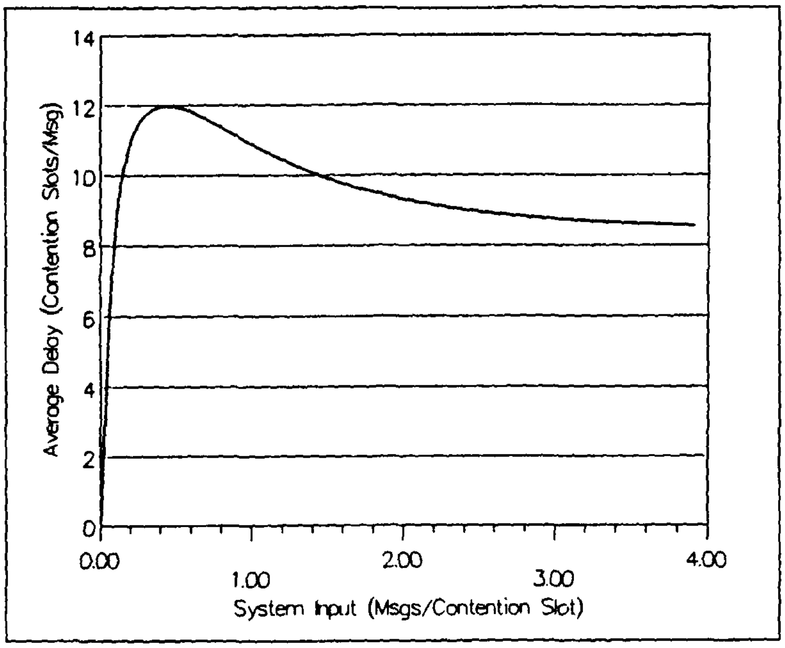

Figure 29. System Delay for LPI Environment, $\alpha = .04$

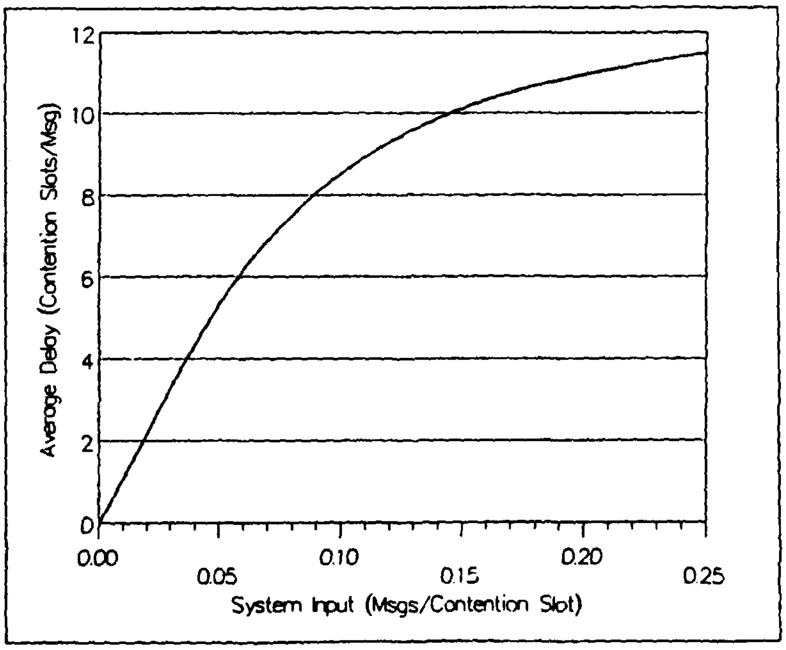

Figure 30. System Delay in Stable Region for LPI Environment, $\alpha = .04$

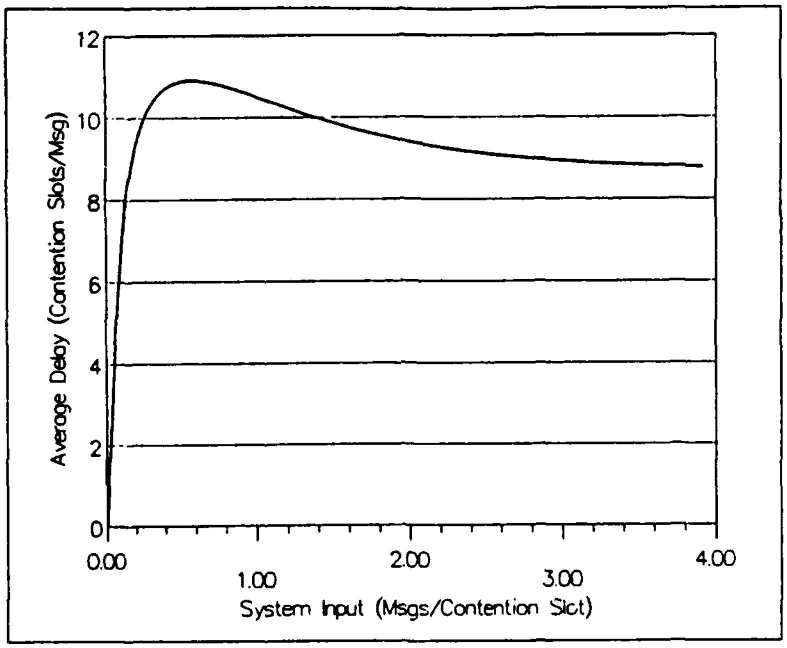

Figure 31. System Delay for LPI Environment, $\alpha = .06$

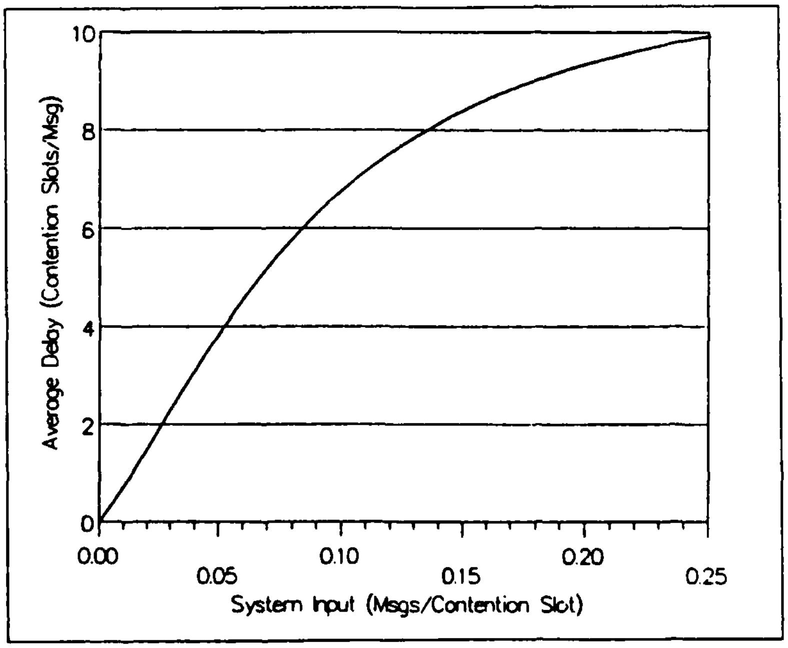

Figure 32. System Delay in Stable Region for LPI Environment, $\alpha = .06$

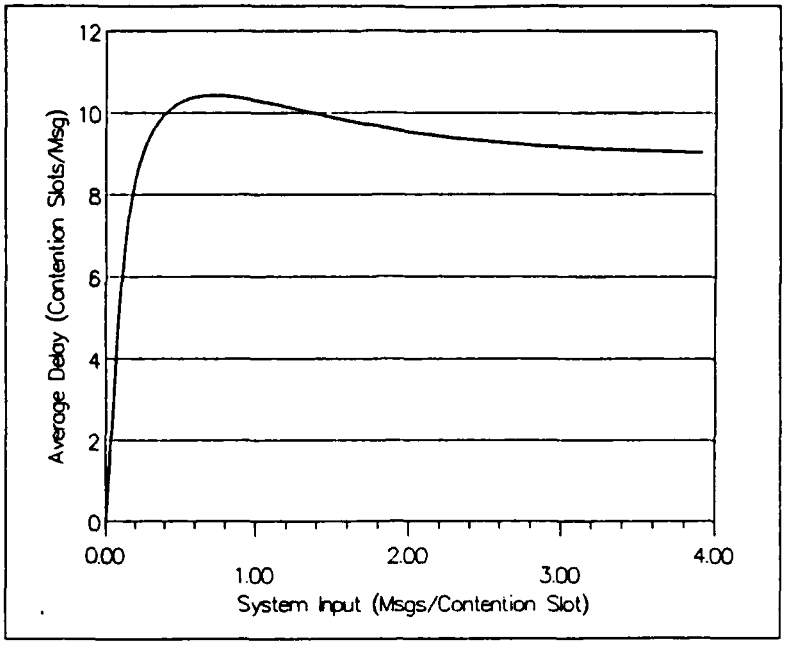

Figure 33. System Delay for LPI Environment, $\alpha = .08$

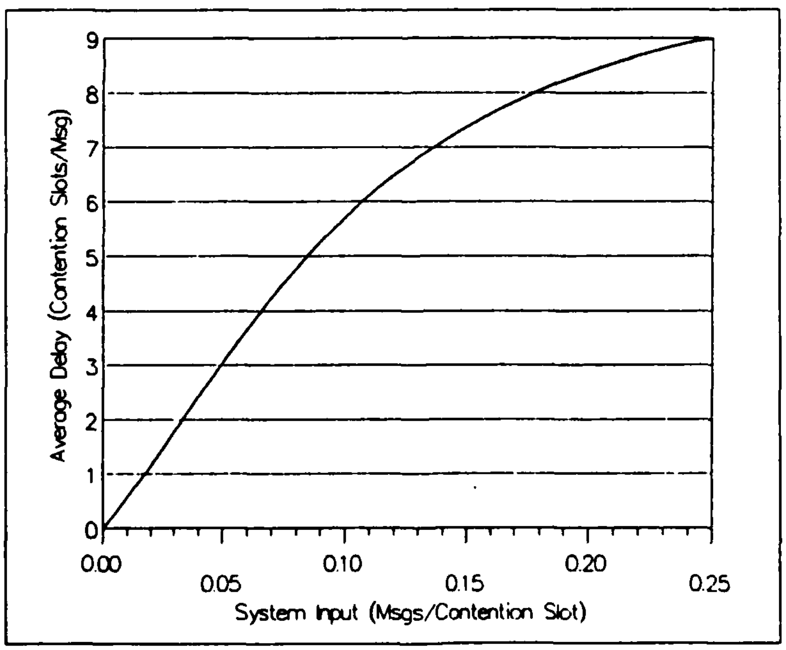

Figure 34. System Delay in Stable Region for LPI Environment, $\alpha = .08$

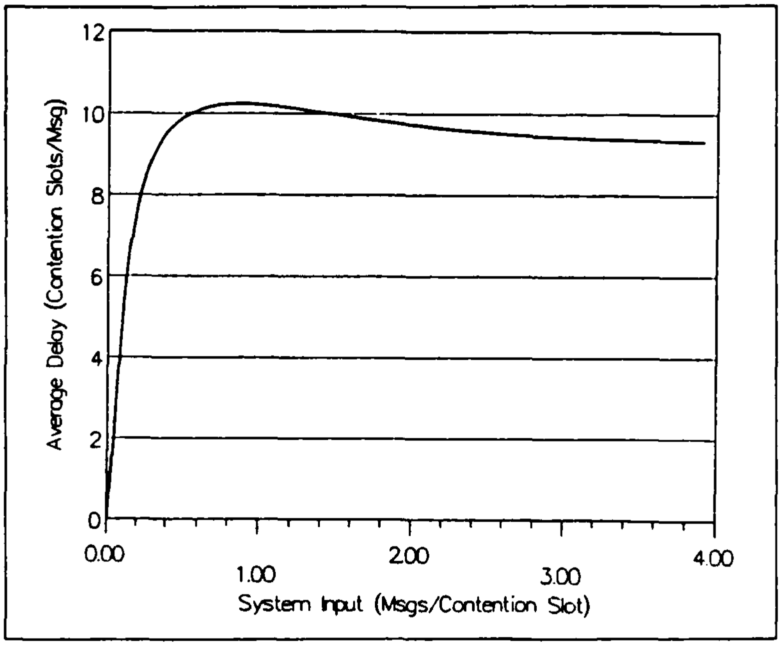

Figure 35. System Delay for LPI Environment, $\alpha = .10$

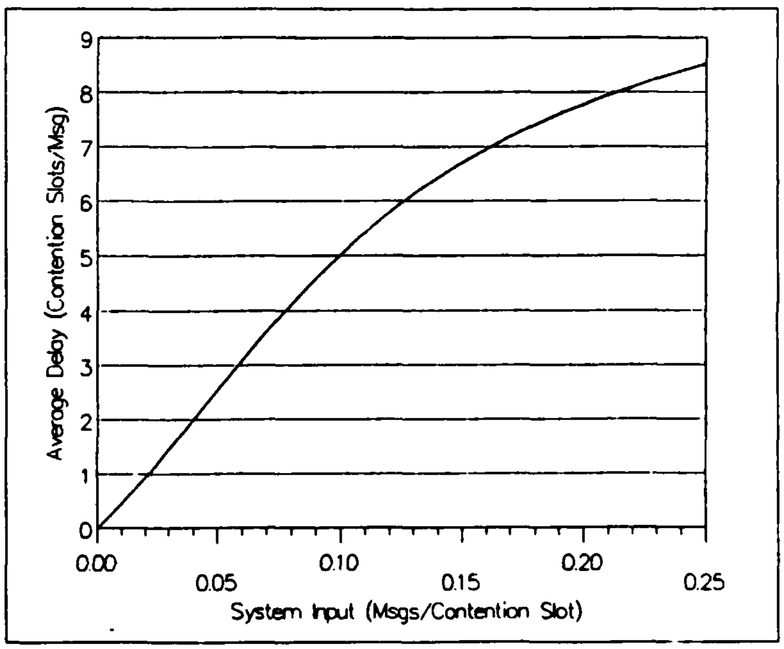

Figure 36. System Delay in Stable Region for LPI Environment, $\alpha = .10$

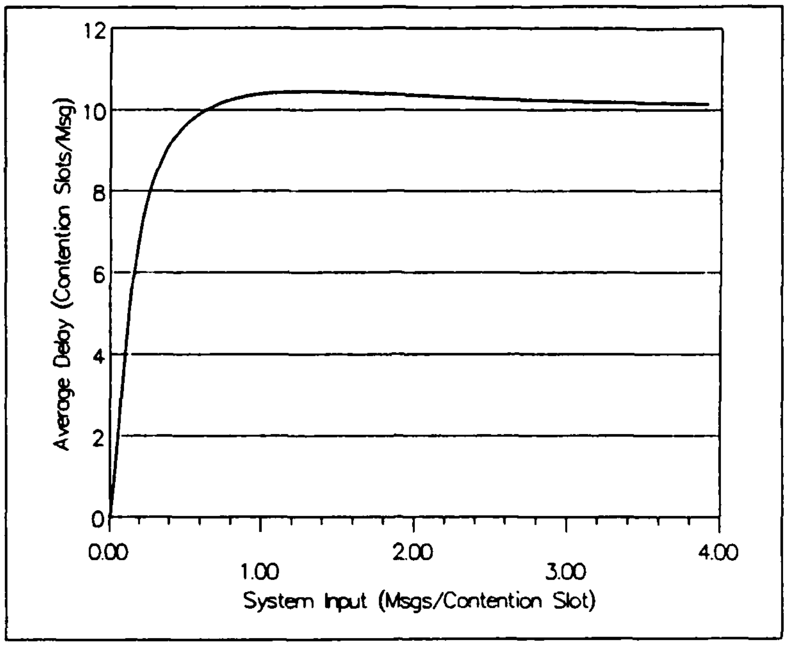

Figure 37. System Delay for LPI Environment, $\alpha = .15$

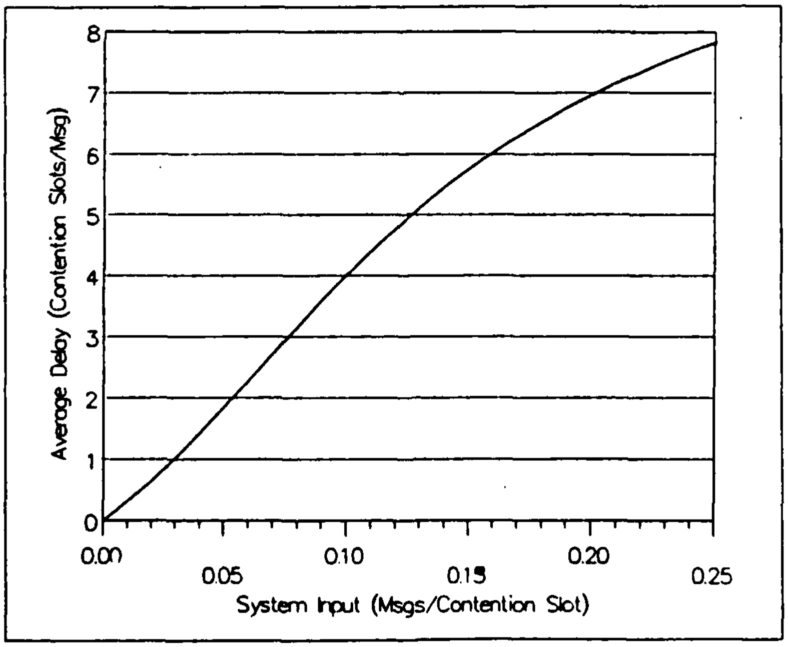

Figure 38. System Delay in Stable Region for LPI Environment, $\alpha = .15$

Appendix C. *Probability of Correct Reception and Decoding of a Message in the Jamming Environment*

This appendix presents P_c for a variety of Reed- Solomon Codes and number of frequency hopped channels, q. These values were calculated using Equations 26, 27, and 28. As a refresher, K represents the number of interfering subnets and/or jammers, d represents the duty cycle of those subnets/jammers, and q is the number of frequency hopped channels. The Reed-Solomon Code is given as (n,k), where n is the number of bits in the code block and k is the number of bits in the block minus the number of redundant bits in the block.

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	.998	.974	.888	.725	.254
.4	1	.918	.54	.178	.037	0
.6	.995	.563	.07	.003	0	0
.8	.974	.144	.001	0	0	0
1.0	.903	.006	0	0	0	0

Table 8. Probability of Correct Reception and Decoding of a Message ((64,40)Reed-Soloman Code, $q = 50$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	.999	.994	.97	.723
.4	1	.998	.932	.668	.325	.012
.6	1	.995	.503	.093	.007	0
.8	1	.737	.074	.001	0	0
1.0	1	.304	0	0	0	0

Table 9. Probability of Correct Reception and Decoding of a Message ((64,32)Reed-Soloman Code, $q = 50$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	.999	.995	.893
.4	1	1	.986	.873	.596	.058
.6	1	.994	.782	.288	.046	0
.8	1	.934	.277	.013	0	0
1.0	1	.688	.018	0	0	0

Table 10. Probability of Correct Reception and Decoding of a Message ((64,28)Reed-Soloman Code, $q = 50$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	.999	.973
.4	1	1	.998	.969	.834	.2
.6	1	1	.943	.593	.187	0
.8	1	.992	.616	.091	.003	0
1.0	1	.934	.145	.001	0	0

Table 11. Probability of Correct Reception and Decoding of a Message ((64,24)Reed-Soloman Code, $q = 50$)

d :	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	.996
.4	1	1	1	.996	.959	.473
.6	1	1	.992	.858	.48	.011
.8	1	1	.889	.345	.04	0
1.0	1	.994	.503	.026	0	0

Table 12. Probability of Correct Reception and Decoding of a Message ((64,20)Reed-Soloman Code, $q=50$)

d :	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	.995	.779
.6	1	1	1	.975	.804	.087
.8	1	1	.986	.725	.234	0
1.0	1	1	.867	.217	.01	0

Table 13. Probability of Correct Reception and Decoding of a Message ((64,16)Reed-Soloman Code, $q=50$)

d :	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	.999	.993	.969	.745
.4	1	.997	.932	.697	.377	.023
.6	1	.953	.55	.136	.016	0
.8	1	.758	.12	.004	0	0
1.0	.999	.382	.005	0	0	0

Table 14. Probability of Correct Reception and Decoding of a Message ((64,40)Reed-Soloman Code, $q = 75$)

d :	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	.982
.4	1	1	.999	.98	.886	.306
.6	1	1	.963	.702	.301	.004
.8	1	.994	.73	.186	.016	0
1.0	1	.957	.283	.009	0	0

Table 15. Probability of Correct Reception and Decoding of a Message ((64,32)Reed-Soloman Code, $q = 75$)

d :	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	.998
.4	1	1	1	.998	.976	.603
.6	1	1	.996	.913	.62	.034
.8	1	1	.935	.504	.102	0
1.0	1	.997	.666	.088	.002	0

Table 16. Probability of Correct Reception and Decoding of a Message ((64,28)Reed-Soloman Code, $q = 75$)

d :	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	.997	.854
.6	1	1	1	.987	.876	.169
.8	1	1	.993	.825	.375	.002
1.0	1	1	.925	.372	.037	0

Table 17. Probability of Correct Reception and Decoding of a Message ((64,24)Reed-Soloman Code, $q = 75$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.97
.6	1	1	1	.999	.979	.478
.8	1	1	1	.971	.741	.035
1.0	1	1	.993	.767	.238	0

Table 18. Probability of Correct Reception and Decoding of a Message ((64,20)Reed-Soloman Code, $q = 75$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.997
.6	1	1	1	1	.999	.817
.8	1	1	1	.998	.952	.226
1.0	1	1	1	.966	.66	.01

Table 19. Probability of Correct Reception and Decoding of a Message ((64,16)Reed-Soloman Code, $q = 75$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	.997	.949
.4	1	1	.994	.941	.78	.201
.6	1	.997	.898	.543	.192	.002
.8	1	.972	.558	.108	.008	0
1.0	1	.864	.165	.005	0	0

Table 20. Probability of Correct Reception and Decoding of a Message ((64,40)Reed-Soloman Code, $q = 100$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	.999
.4	1	1	1	.999	.992	.778
.6	1	1	.999	.967	.799	.121
.8	1	1	.978	.726	.282	.002
1.0	1	.999	.85	.273	.024	0

Table 21. Probability of Correct Reception and Decoding of a Message ((64,32)Reed-Soloman Code, $q = 100$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	.999	.942
.6	1	1	1	.997	.956	.381
.8	1	1	.998	.936	.636	.022
1.0	1	1	.979	.655	.168	0

Table 22. Probability of Correct Reception and Decoding of a Message ((64,28)Reed-Soloman Code, $q = 100$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.992
.6	1	1	1	1	.995	.72
.8	1	1	1	.993	.902	.148
1.0	1	1	.999	.921	.525	.005

Table 23. Probability of Correct Reception and Decoding of a Message ((64,24)Reed-Soloman Code, $q = 100$)

d:	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.999
.6	1	1	1	1	1	.934
.8	1	1	1	1	.988	.481
1.0	1	1	1	.993	.867	.065

Table 24. Probability of Correct Reception and Decoding of a Message ((64,20)Reed-Soloman Code, $q = 100$)

d:	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	.993
.8	1	1	1	1	.999	.84
1.0	1	1	1	1	.986	.356

Table 25. Probability of Correct Reception and Decoding of a Message ((64,16)Reed-Soloman Code, $q = 100$)

d:	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	.992
.4	1	1	.999	.991	.946	.529
.6	1	1	.984	.848	.539	.036
.8	1	.997	.871	.43	.101	0
1.0	1	.982	.565	.088	.004	0

Table 26. Probability of Correct Reception and Decoding of a Message ((64,40)Reed-Soloman Code, $q = 125$)

d:	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.959
.6	1	1	1	.998	.969	.484
.8	1	1	.999	.956	.724	.051
1.0	1	1	.986	.744	.267	.001

Table 27. Probability of Correct Reception and Decoding of a Message ((64,32)Reed-Soloman Code, $q = 125$)

d:	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.995
.6	1	1	1	1	.997	.801
.8	1	1	1	.996	.937	.244
1.0	1	1	.999	.951	.649	.016

Table 28. Probability of Correct Reception and Decoding of a Message ((64,28)Reed-Soloman Code, $q = 125$)

d:	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	.96
.8	1	1	1	1	.993	.607
1.0	1	1	1	.996	.919	.133

Table 29. Probability of Correct Reception and Decoding of a Message ((64,24)Reed-Soloman Code, $q = 125$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	.996
.8	1	1	1	1	1	.896
1.0	1	1	1	1	.992	.482

Table 30. Probability of Correct Reception and Decoding of a Message ((64,20)Reed-Soloman Code, $q = 125$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	1
.8	1	1	1	1	1	.989
1.0	1	1	1	1	1	.859

Table 31. Probability of Correct Reception and Decoding of a Message ((64,16)Reed-Soloman Code, $q = 125$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	.999
.4	1	1	1	.999	.989	.79
.6	1	1	.998	.961	.808	.174
.8	1	1	.971	.744	.348	.006
1.0	1	.998	.849	.346	.056	0

Table 32. Probability of Correct Reception and Decoding of a Message ((64,40)Reed-Soloman Code, $q = 150$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	.994
.6	1	1	1	1	.996	.805
.8	1	1	1	.995	.934	.27
1.0	1	1	.999	.948	.661	.023

Table 33. Probability of Correct Reception and Decoding of a Message ((64,32)Reed-Soloman Code, $q = 150$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	.961
.8	1	1	1	1	.993	.632
1.0	1	1	1	.996	.921	.163

Table 34. Probability of Correct Reception and Decoding of a Message ((64,28)Reed-Soloman Code, $q = 150$)

$d:$	$K=10$	$K=20$	$K=30$	$K=40$	$K=50$	$K=75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	.996
.8	1	1	1	1	1	.903
1.0	1	1	1	1	.992	.516

Table 35. Probability of Correct Reception and Decoding of a Message ((64,24)Reed-Soloman Code, $q = 150$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	1
.8	1	1	1	1	1	.989
1.0	1	1	1	1	1	.862

Table 36. Probability of Correct Reception and Decoding of a Message ((64,20)Reed-Soloman Code, $q = 150$)

d :	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$	$K = 75$
.2	1	1	1	1	1	1
.4	1	1	1	1	1	1
.6	1	1	1	1	1	1
.8	1	1	1	1	1	1
1.0	1	1	1	1	1	.985

Table 37. Probability of Correct Reception and Decoding of a Message ((64,16)Reed-Soloman Code, $q = 150$)

Appendix D. *Delay in Jamming Environment*

The results presented in this appendix were delevloped using Equation 31.

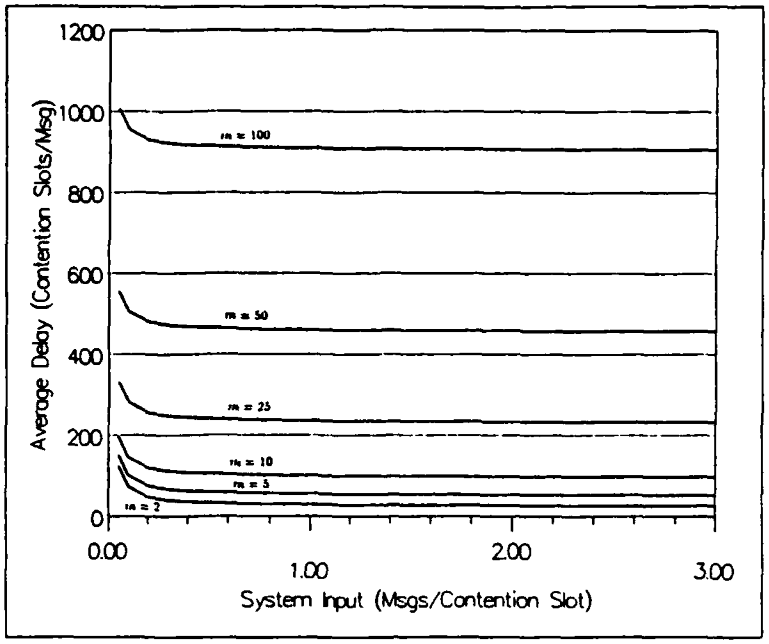

Figure 39. System Delay for Jamming Environment, $\rho = .2$

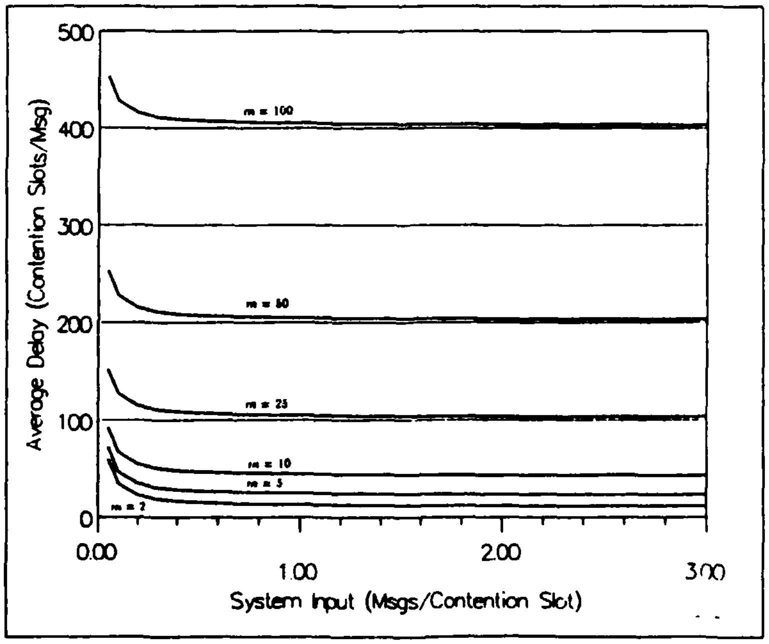

Figure 40. System Delay for Jamming Environment, $\rho = .4$

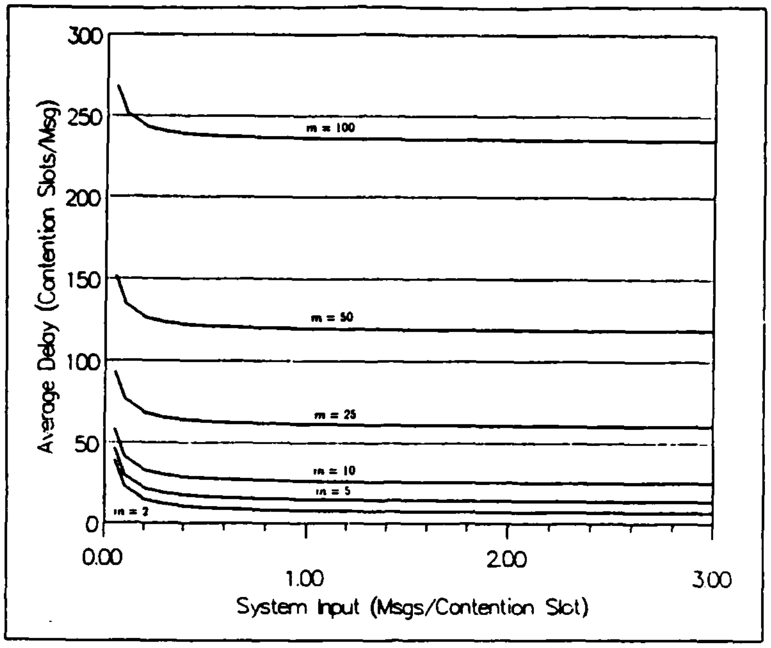

Figure 41. System Delay for Jamming Environment, $\rho = .6$

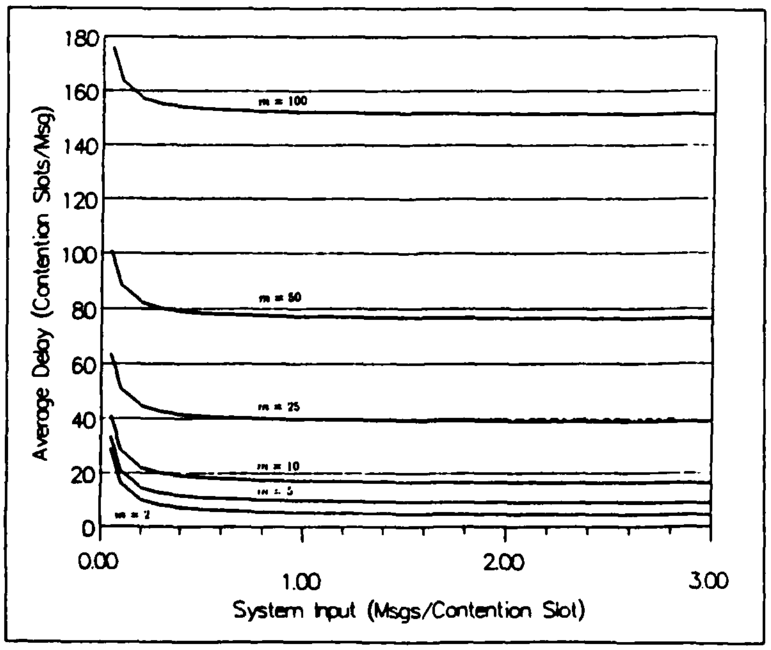

Figure 42. System Delay for Jamming Environment, $\rho = .8$

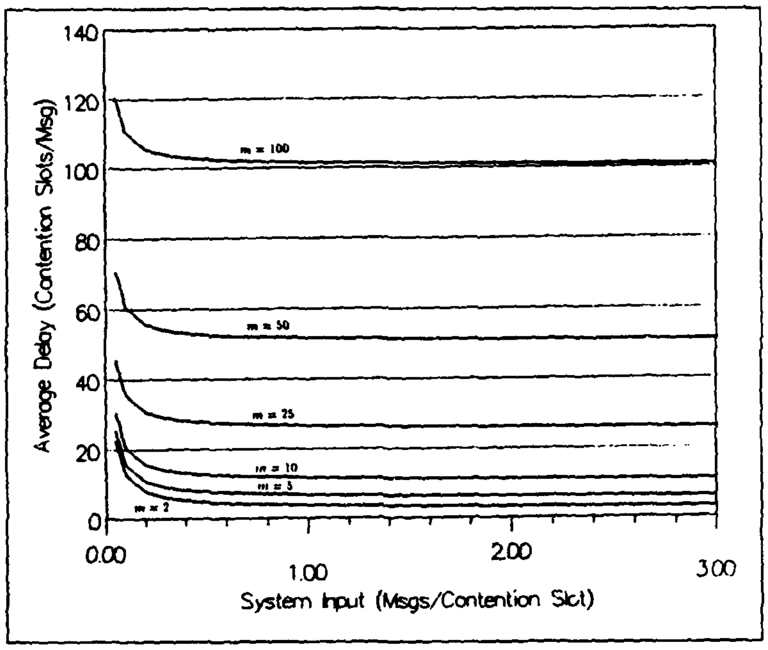

Figure 43. System Delay for Jamming Environment, $\rho = 1.0$

Bibliography

1. Abdelmonem, Ahmed H. and Tarek N. Saadawi. "Performance Analysis Of An Asynchronous Spread Spectrum Packet Radio Network," *MILCOM '88, 3:* 0823–0829 (October 1988).
2. Brick, Donald B. and Fred W. Ellersick. *Challenges and Opportunities in Air Force Tactical Communications,* February 1981. Electronic Systems Division, Hanscom AFB, MA (AD-A097677/9).
3. Department of the Army. *An Analysis of the Joint Tactical Information Distribution System (JTIDS) as a Mobile Subscriber Equipment (MSE) Alternative,* 20 September 1979. Joint Tactical Communications Office, Fort Monmouth, NJ (AD-A079593/0).
4. deSouza, Osmund and others. "Performance Analysis of Spread Spectrum Packet Radio Networks," *MILCOM '88, 2:* 0599–0604 (October 88).
5. Dixon, Robert C. *Spread Spectrum Systems.* New York: John Wiley & Sons, 1976.
6. Dixon, Robert C. *Spread Spectrum Systems,* Second Edition. New York: John Wiley & Sons, 1984.
7. Ephremides, Anthony and others. "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceedings of the IEEE, Volume 75, 1:* 56–73 (January 1987).
8. Fifer, William C. and Frederick J. Bruno. "The Low–Cost Packet Radio," *Proceedings of the IEEE, Volume 75, 1:* 33–41, (January 1987).
9. "Filter Center," *Aviation Week and Space Technology, 118:* 67 (8 August 1983).
10. Frank, Colin D. and Michael B. Pursley. "Bounds on the Throughput in Frequency-Hop Packet Radio Networks with Convulutional Coding," *MILCOM '88, 1:* 0121–0125 (8 August 1988).
11. General Accounting Office. *The Joint Tactical Information Distribution System – How Important Is It.*

    Report No. GAO/PSAD-80-22. General Accounting Office, Procurement and Systems Acquisition Division, Washington, D.C., 30 January 1980 (AD-A082222/1).
12. Gerakoulis, D. P. and others. "A Channel Access Protocol for Imbedding CSMA on Spread-Spectrum Packet Radio Networks," *IEEE International Conference on Communications, 1:* 0199–0203 (June 1988).
13. Geraniotis, Evaggelos and Rajan Mani. "Throughput Analysis of a Random Access Tree Protocol for Frequency-Hopped Spread-Spectrum Packet Radio Networks," *MILCOM '87, 2:* 0528–0533 (1987).
14. Geraniotis, Evaggelos A. and Michael B. Pursley. "Error Probabilities for Slow-Frequency-Hopped Spread- Spectrum Multiple-Access Communications Over Fading Channels," *IEEE Transactions on Communications, COM-30:* 996-1009 (May 1982).
15. Gutman, Lawrence L. and Glenn E. Prescott. "System Quality Factors for LPI Communications." Wright Research and Development Center/Air Force Institute of Technology, Wright-Patterson AFB, OH (Spring 1989).
16. Hahn, Jonathan J. and David M. Stolle. "Packet Radio Network Routing Algorithms, A Survey," *IEEE Communications Magazine, Vol. 22, 11:* 41–47 (November 1984).
17. Hartman, Walter, WRDC–AAAI–2. Personal Interview. Air Force Wright Aeronautical Labs, Wright-Patterson AFB Ohio, 20 April 1989.

18. Hayes, Jeremiah F. *Modeling and Analysis of Computer Communications Networks.* New York: Plenum Press, 1986.

19. Klass, Phillip J. "Plans Change on F-16 Communications," *Aviation Week and Space Technology, 121:* 163 (10 September 1984).

20. Klienrock, Leonard and Fouad A. Tobagi "Packet Switching in Radio Channels: Part I – Carrier Sense Multiple- Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications, Vol. COM-23, 12:* 1400–1416 (December 1975).

21. Leiner, Barry M. and others. "Issues in Packet Radio Network Design," *Proceedings of the IEEE, Vol. 75, 1:* 6–20 (January 1987).

22. Lewis, Mark G. and J.J. Garcia-Luna-Aceves. "Packet Switching Applique for Tactical VHF Radios," *IEEE International Military Communications Conference, 2:* 449–455 (1987).

23. Mordoff, Keith F. "Navy Drops Own Communications System to Join USAF/Army JTIDS," *Aviation Week and Space Technology, 123:* 20 (28 October 1985).

24. Price, Robert "Further Notes and Anecdotes on Spread Spectrum Origins," *IEEE Transactions on Communications, COM-31:* 85–97, (January 1983).

25. Pursley, Michael B. "Frequency-Hop Transmission for Satellite Packet-Switching and Terrestrial Packet Radio Networks," *IEEE Transactions on Information Theory, Vol. IT-32, NO. 5:* 652–667, (September 1986).

26. Pursley, Michael B. "The Role of Spread Spectrum in Packet Radio Networks," *Proceeding of the IEEE, 75:* 116–134, (January 1987).

27. Pursley, Michael B. and D. J. Taipale. "Error Probabilities for Spread-Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding," *IEEE Transactions on Communications, Vol. COM-35, 1:* 1–12 (January 1987).

28. Robinson, Clarence A., Jr. "Management: Common Armed Forces Links Sought," *Aviation Week and Space Technology, 117:* 51–58 (15 November 1982).

29. Scholtz, Robert A. "The Origins of Spread Spectrum Communications," *IEEE Transactions on Communications, COM-30:* 822-854, (May 1982).

30. Schwartz, Mischa. *Telecommunication Networks: Protocols, Modeling and Analysis.* Reading, Massachusetts: Addison–Wesley Publishing Company, 1987.

31. Simon, Marvin K. and others. *Spread Spectrum Communications, Volume 1.* Rockville, Maryland: Computer Science Press, Inc., 1985.

32. Simon, Marvin K. and others. *Spread Spectrum Communications, Volume 2.* Rockville, Maryland: Computer Science Press, Inc., 1985.

33. Singer-Rockwell. *JTIDS Class 2 Terminal.* System Description/Specifications. Collins Government Avionics Division, Rockwell International, Cedar Rapids, IA, November 1983.

34. Stein, Kenneth J. "Avionics: Military Seeks Systems Interoperability," *Aviation Week and Space Technology, 118:* 123–129 (13 June 1983).

35. Stein, Kenneth J. "Avionics: F-15C/D Color Display Nears Test Stage," *Aviation Week and Space Technology, 120:* 77–81 (20 February 1984).

36. Tanenbaum, Andrew S. *Computer Networks, Second Edition.* Englewood Cliffs, New Jersey: Prentice–Hall, Inc., 1988.

37. Tobagi, Fouad A. and others. "Packet Radio and Satellite Networks." *IEEE Communications Magazine, 22:* 24–40, (November 1984).

38. "U.S., Britain Sign Memorandum For JTIDS System Participation," *Aviation Week and Space Technology, 119:* 28 (28 November 1983).

39. Worsley, William H., Deputy Program Manager for ICAAS, Air Force Wright Aeronautical Labs. Personal interview. Wright-Patterson AFB Ohio, 7 April 1989.

40. Xie, Hai and Lazaros Merakos. "Performance Analysis of a Receiver Directed CDMA Packet Radio Network," *IEEE International Conference on Communications, 1:* 0184–0188 (June 1988).

Vita

Captain Scott R. Harrison [redacted] He graduated from Evergreen Senior High School, Evergreen, Colorado, in May 1980. On 30 May 1984, he received a Bachelor of Science Degree in Engineering Sciences from the United States Air Force Academy and was commissioned as a Second Lieutenant. After graduation, he attended the Communication-Electronics Officer Basic Course at Keesler AFB, Mississippi and was assigned to the 390th Communications Operations Squadron, Offutt AFB, Nebraska. While assigned to the 390th, he served as Chief, Weather Communications; Chief, Dedicated Systems Branch; and Chief, Telecommunications Operations Branch. He entered the School of Engineering, Air Force Institute of Technology, in June 1988. He is married [redacted], USAF, of Niceville, Florida.

[redacted]

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
UNLIMITED

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/90M-01

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL (If applicable)
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Institute of Technology
WPAFB, OH 45433-6583

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
THROUGHPUT AND DELAY CHARACTERISTICS FOR A SLOW-FREQUENCY HOPPED AIRCRAFT-TO-AIRCRAFT PACKET RADIO NETWORK

12. PERSONAL AUTHOR(S)
Scott Russell Harrison, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)
March 1990

15. PAGE COUNT
122

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	07	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Packet Radio Networks, Frequency Hopping, Aircraft Communications, Spread Spectrum Communications

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
Thesis Chairman: Charles R. Bisbee, Lt Col, USAF
Professor of Electrical Engineering and Computer Science

Abstract: (see reverse)

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Lt Col Charles R. Bisbee

22b. TELEPHONE (Include Area Code)
(513) 255-3576

22c. OFFICE SYMBOL
AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE