MODELING AND SIMULATION OF COMMUNICATIONS SYSTEMS IN OPNET

THESIS

David M. Banker, Second Lieutenant, USAF

AFIT/GE/ENG/00M-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/00M-03

MODELING AND SIMULATION OF COMMUNICATION SYSTEMS IN OPNET

THESIS

Presented to the Faculty

Department of Electrical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

David M. Banker, B.S.E.E

Second Lieutenant, USAF

March 2000

AFIT/GE/ENG/00M-03

MODELING AND SIMULATION OF COMMUNICATION SYSTEMS IN OPNET

David M. Banker, B.S.E.E

Second Lieutenant, USAF

Approved:

Dr. Michael A. Temple (Chairman)	14 Mar 00 date
Maj. Richard A. Raines (Member)	14 Mar 00 date
Maj. Rusty O. Baldwin (Member)	14 Mar 00 date

ACKNOWLEDGEMENTS

I would like to acknowledge the many people without whom I could never have finished this document.

First, I would like to express my thanks to my advisor, Dr. Michael A. Temple for his guidance and support throughout all aspects of the thesis effort. I appreciate the insight, guidance, and patience he exhibited throughout my AFIT experience. I would also like to thank my thesis committee, Major Raines and Major Baldwin, for their insight and advice, as well as their feedback on this research effort. Thank you to my sponsor, Mr. Kenneth Fore, for giving me the opportunity to excel by backing this research.

I also thank my GE-00M classmates, as their insight and help with various tasks made my thesis possible. A special thanks goes to Ron, Steve, Alex, and Brett, as their support made it possible to survive Friday meetings.

The biggest thanks go to my future wife, Michele, as her encouragement and persistent prodding over the past year made it possible to finish this document. Her understanding of late nights and awkward schedules is greatly appreciated.

I would also like to thank my family. From my parents, Michael and Joan, to my siblings, Katie, Christine, Kelly, and John, all have been supportive of my efforts, and helped to encourage me through their constant caring and love. Special thanks goes to Andrew, who has made me see how special my own family will be.

Finally, I would like to thank God, who has made all of this possible.

David M. Banker

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

This research aims to present accurate computer models of a communication link and a Super High Frequency (SHF) radio communication system. Network Warfare Simulation (NETWARS) is a J-6 initiative aimed at modeling all communication traffic in the Department of Defense (DoD) for testing and analysis of specific real world scenarios. The AN/TSC-94 is a SHF radio system with satellite communication capabilities. The AN/TSC-94 incorporates a Direct Sequence Spread Spectrum (DSSS) radio link for certain Anti-Jam (AJ) features. A DSSS 'spreads' signal power over a large bandwidth, reducing power previously concentrated within the original system bandwidth. The simulations were performed using OPNET. Simulation results show DSSS lowered Bit Error Rate (BER) over links not using spread spectrum. Results show that in the presence of multiple jamming forms, the DSSS link performed without bit errors while the normal (non-DSSS) link was disrupted by the jammer, experiencing BER's of up to 0.43. The AN/TSC-94 was able to defeat the jammer using the DSSS link. By performing in normal mode during unjammed scenarios, and switching to AJ mode in the presence of a hostile transmitter, the AN/TSC-94 demonstrated its ability to successfully communicate in multiple access and hostile environments.

MODELING AND SIMULATION OF COMMUNICATION SYSTEMS IN OPNET

1. Introduction

As a new millenium begins, a premium is being placed on information in the technology world. Such a premium is seen in the latest Command, Control, Communications, and Computer (C4) Systems (J-6) initiative, Network Warfare Simulation (NETWARS) [6]. NETWARS is a software package developed to simulate communication traffic between Department of Defense (DoD) resources. During the Gulf War there existed an alarming lack of communications assets, as the existing communications infrastructure was taxed beyond its limit. NETWARS was designed to simulate any scenario, given multiple communication resources, and determine where shortcomings, interferences, or complete failures may occur. NETWARS is being developed because of these possible downfalls.

The Air Force Communications Agency (AFCA), is tasked with modeling all Air Force communication resources. From aircraft radios to small hand-held communicators, all communication asset are to be modeled. One such radio is the AN/TSC-94, a truck-mounted, Super-High Frequency (SHF) system which uses a satellite (AN/TSC-100) relay to transmit signals throughout the world. The system has many advanced features, including Anti-Jam (AJ) capability, encryption, and spread spectrum [5].

The AN/TSC-94 uses Direct Sequence Spread Spectrum (DSSS) to provide AJ capability. DSSS uses a 'spreading code' to significantly increase signal bandwidth, while simultaneously reducing signal power [1]. Effective DSSS techniques can reduce transmitted power below channel noise level. The low-power, high-bandwidth characteristics of the signal combined with the ability ro reduce jamming power in the receiver give the AN/TSC-94 capability to operate in hostile jamming environments.

1.1. Background

This section offers an overview of DSSS, as well as information on NETWARS. Any assumptions made, as well as an overall approach to modeling, are also included.

1.1.1. Direct Sequence Spread Spectrum (DSSS)

The practice of using DSSS dates back as early as World War II, when a dual channel system was used to simultaneously transmit the spreading code and the modulated signal; a primitive implementation of DSSS, as the code is the key to the entire technique. The spreading code is a Random Binary Waveform (RBW) generated by a Linear Feedback Shift Register (LFSR) [2]. The spreading waveform is binary, i.e., alternating between two possible values. Generally the waveform values are represented by a zero for the low and a one for the high. The code is considered pseudo-random, meaning the code will repeat, but appears random. This is accomplished using the LFSR configuration. The pseudo-random nature of the code provides some security, as a code which follows a pattern would be much easier to detect and exploit.

The LFSR generates RBW's having a much higher data rate than the original signal. This accounts for the 'spreading' feature of a spread spectrum signal. Another spread spectrum signal feature is power level reduction. The radiated power spectral density (watts/herts) is reduced by a factor proportional to the spreading code [2]. When a given signal is spread over a larger bandwidth, there must be a corresponding reduction in Power Spectral Density (PSD) [4]. The larger the ratio between the spreading code rate and the original data rate, the lower the transmission power. This lower power, higher bandwidth feature gives the signal a Low Probability of Detection (LPD) characteristic.

1.1.2. NETWARS

U.S. Army Lieutenant General Douglas Bucholz, the Joint Staff's Director for Command, Control, Communications, and Computer (C4) Systems (J-6), is on a mission to achieve information superiority. He believes this advantage will provide significant capabilities in terms of survivability, lethality, and operational tempo [6]. Accordingly, the modeling and simulation tool called NETWARS is being developed.

NETWARS will test and evaluate communications networks through computer modeling and simulation. These simulations will test and evaluate the operability of a communications system under any number of conditions within a given environment. NETWARS is designed be all encompassing in terms of battlefield communications. Multiple radio systems, satellite systems, cellular systems and data transmission networks will be represented in any given scenario [6]. To effectively evaluate communication assets, it is first necessary to model individual communication system components. When the individual components are modeled, NETWARS will be able to accurately depict multiple battlefield communication scenarios.

1.2. Problem Statement

As NETWARS continues to grow, the need for accurate computer models becomes increasingly important. The purpose of this thesis is twofold: 1) to design, model, simulate and test the AN/TSC-94 radio in OPNET for incorporation into NETWARS, and 2) to design, model, simulate and test a direct sequence spread spectrum communication link (pipeline) for use in OPNET.

1.3. Assumptions

This work makes many assumptions to decrease model complexity, without severely affecting simulation accuracy. The AN/TSC-100, a geostionary satellite, is modeled as a "bent pipe" satellite. It retransmits received data for the AN/TSC-94 to ensure frequency matching. Because no information was available on the satellite characteristics, it is modeled after the AN/TSC-94. Another assumption is that radio automatically establishes communication with intended receivers. This assumption eliminates the need for a 'beam-finding' algorithm in the radio model.

1.4. Scope

The scope of this thesis is limited to the DSSS link, and to the AN/TSC-94 radio. Special consideration is given to the affect that the AN/TSC-94 will have on the NETWARS research, e.g. transmission characteristics.

1.5. Approach

The DSSS model is developed by modifying an existing OPNET radio link (Transceiver Pipeline). The model is tested and compared to existing theory for the applied conditions, since no actual link data is available. The AN/TSC-94 radio model was developed based on information contained in its Technical Order (TO). The model was subjected to interference ranging from ambient noise to hostile jammers.

1.6. Equipment

A Sun SparcStation Ultra 1, with a 167 Mhz processor, was used to run various OPNET simulations. OPNET is a network modeling and simulation tool developed by MIL3, Inc., Washington, D.C.

1.7. Thesis Organization

Chapter 2 presents a background on the effects of a spread spectrum system on an ordinary communication link, to validate the assumptions made about the link. An overview of OPNET is also presented, to give an understanding of the model development. NETWARS is also discussed, to inform the reader on how the research fits with its desired end use. Chapter 3 defines methodology for building the radio model, and the spread spectrum link. Model validation is presented as well. Chapter 4 explains the tests used to simulate the model, as well as simulations results. Chapter 5 provides thesis conclusions, and provides recommendations for future research.

2. Spread Spectrum Technology, OPNET, and the AN/TSC-94 Radio

This chapter discusses important aspects of a DSSS system. A DSSS system involves many complicated equations with a high degree of difficulty. However, it is possible to consider the spread spectrum link while asking a most important modeling question, "What can be left out?" By examining the DSSS effects on a communication link, the "effects" of spread spectrum modulation can be effectively modeled, rather than the entire link.

OPNET is the main simulation tool used by NETWARS. It is a network modeling tool designed and developed with little emphasis on wireless communications. Since spread spectrum is only useful over broadcast links, a modified radio link is created. While NETWARS does contain basic tools for radio links, updating is required to model the effects discussed earlier, i.e., higher bandwidth and lower power spectral density..

The final aspect of this thesis is modeling the AN/TSC-94 SHF radio. This radio employs satellites to achieve global communications. The radio system also includes spread spectrum features and employs convolutional encoding to reduce the bit error rate – a necessity for satellite communications [3]. The radio is also subject to the modeling question, as a high-fidelity model serves to increase complexity and computational intensity without providinga significant increase in accuracy.

2.1. Spread Spectrum Technology

There are several types of spread spectrum systems including DSSS and frequency hopping, i.e., the center frequency of the transmitted signal is changed according to preset spreading code values. However, DSSS has distinct characteristics which are easily modeled, with the key being the spreading code as generated by a LFSR.

2.1.1. Spreading Code Generation and Characteristics

The spreading code generation is the most important step in spread spectrum communications. These codes are binary waveforms with very distinct and important characteristics. Binary waveforms have only one of two possible values, usually a one and a zero. Spread spectrum is characterized by a signal occupying a bandwidth much in excess of the minimum bandwidth needed to transmit the signal [1]. The higher bandwidth is achieved by having a spreading code at a much higher frequency than that of the original data signal. The spreading code must also have other distinct properties, which include *balance*, *run*, and *correlation* [1].

Balance simply says that a code will differ in the number of ones and zeros by no more than one. For instance, if a signal is of length fifteen, there will be eight ones, and seven zeroes. *Runs* in a code are sequences of similar binary digits which occur right next to each other. In a pseudonoise (PN) sequence, about half the runs will be of length one, one-quarter of the runs will be of length two, one-eighth of the runs will be of length three, and so on [2].

These properties are functions of the LFSR, but do not necessarily have a large impact on the performance of the system. Correlation, however, has the largest impact on how the signal interacts with other signals. The correlation property states that, "if a period of the sequence is compared term by term with any cyclic shift of itself. It is best if the number of agreements differs from the number of disagreements by not more than one count."[1] That is, the highest correlation you will have when trying to synchronize these signals is when the original signal is correlated with itself. A low correlation enables a number of multiple access (MA) schemes to take place, as multiple users can employ different instances of the same code, and still not have a high degree of interference.

2.1.2. Processing Gain

As mentioned before, a large signal bandwidth is used by the spreading code. This bandwidth must be much larger than the data rate for the original signal. An important measurement of the performance of the spreading signal is the processing gain. Processing gain is defined as the advantage that spreading gives a signal when it is transmitted in a hostile environment. The processing gain is approximately given as the ratio between the spreading code chip rate R_p, and the data rate R [1].

2.2. OPNET

OPNET is a powerful modeling and simulation tool, and has been designated for use in the NETWARS project. OPNET uses three layers for modeling. The first is the component level, where a user may choose any number of pre-constructed devices for a simulation. These components can be satellites, network routers, internet workstations, or any number of computer network devices. The next level is the node level, where a user may construct his own device for use in the component level. Through a series of processors, transmitters and receivers, a user can construct a device to the exact specifications needed for a simulation. This level will be used to construct the AN/TSC-

94A, along with the next level, the process level. The process level is where node behavior is defined. Using a modified C/C++ code, OPNET allows the user to program the processor to do nearly anything relating to a packet, whether it be sending it, appending data to it, or even having it destroyed. Through the use of *Kernel Procedures* a user may execute many functions designed to manipulate a packet in any way imaginable. Kernel Procedures are OPNET specific C++ functions for reading or writing values in simulations. These three layers make OPNET not only a very powerful tool, but a very complete tool for those who are able to use it.

2.2.1. Transceiver Pipeline

The Transceiver Pipeline is a 14-stage process executed as a packet goes from transmitter to receiver. The Transceiver Pipeline stages are divided into these two components, with six stages in the transmitter, and eight stages in the receiver. The Transceiver Pipeline takes the packet from the transmitters in the node level, and through its stages, delivers it to the receivers. Each of these stages is a series of C/C++ code designed to simulate exactly what may happen over the course of a transmission, whether it be a simple point-to-point link, a bus tap, or even a radio broadcast. The radio broadcast link is the most complete, as the earlier links simply omit unnecessary stages.

2.2.1.1. Stage 1 – Receiver Group

The first stage of the Transceiver Pipeline is called receiver group. Receiver Group simply decides which receivers may receive the signal about to be transmitted. The default criterion for receivers being accepted is that all receivers are considered valid. That is, the stage is implemented for the user to enter his own code for eligible receivers. One may put power or distance limitations on which receiver may receive certain signals, but by default all receivers are considered valid.

2.2.1.2. Stage 2 – Transmission Delay

Transmission Delay is the second stage for the transceiver pipeline. It determines how long a packet may take to be processed through the transmitter. It is based on the data rate and the length of the packet being transmitted. The delay may be used as a metric for any system, to determine whether or not it is performing to specifications.

2.2.1.3. Stage 3 – Link Closure

Link closure is implemented directly after the transmission delay stage, and serves to eliminate some of the receivers by performing a line-of-sight calculation. This enables OPNET to decide if the transmitter can possibly see the receiver, or if by the curvature of the earth, transmission is impossible. By using the positions on the Earth, and assuming a perfect sphere (i.e. no mountains) for the Earth, OPNET calculates whether or not transmission can be achieved. If there is no way to reach the receiver, the packet is destroyed and the link dropped. If the transmitter can communicate with the receiver, the packet proceeds to the next stage. Three modes of closure are evaluated: 1) ground to satellite, 2) satellite to ground, and 3) satellite cross links. The satellite cross links evaluation encompasses any ground to ground communications.

2.2.1.4. Stage 4 – Channel Match

After it is determined that the transmitter may reach the receiver, it is necessary to decide if the receiver can decode the transmission. Channel match compares a number of characteristics from both the transmitter and receiver. These include the transmission frequency, bandwidth, data rate, spreading code, and modulation. If the transmission frequency is not matched exactly, the stage decides if there is any overlap between the bandwidth of the receiver and the transmitted signal. For a condition with no overlap, the transmission is tagged as invalid, and is ignored. The signal is considered interference when there is some overlap, or when there is full frequency and bandwidth match but some mismatch between the other values. This interference is used later in the noise stages. If the frequency, bandwidth, data rate, spreading code, and modulation all compare, the transmission is marked as valid, and proceeds to the next stage.

2.2.1.5. Stage 5 – Transmitter Antenna Gain

The next stage is called the transmitter antenna gain stage, and does exactly what its name implies. By examining the antenna model (editable by the user), OPNET determines the gain in the direction of the receiver. The gain is calculated by examining the vector between the transmitter and receiver. This stage may be executed multiple times for a single packet, once for each receiver still within line-of-sight of the transmitter.

2.2.1.6. Stage 6 – Propagation Delay

Propagation delay is exactly what it seems. The packet is tracked from the time it leaves the transmitter until the time it reaches the receiver. This stage is executed much like the transmission delay, in that the time is calculated on the distance between the transmitter and receiver and the speed at which it propagates (the default value is the speed of light). There are two different values for the propagation delay, as the time is calculated from the first bit of the packet, and at the last bit of the packet. This is the last stage of the Transceiver Pipeline associated with the transmitter.

2.2.1.7. Stage 7 – Receiver Antenna Gain

The received antenna gain stage is called the same as the transmitter gain, except for the values, which are taken from the receiver attributes. The receiver is also denoted as the origin for the calculation, instead of the transmitter. The receiver antenna gain may be applied to any incoming signal, valid or interference.

2.2.1.8. Stage 8 – Received Power

After being received by the antenna, the received power is calculated. The received power stage has two functions: 1) calculating power for incoming signals, and 2) deciding which incoming signals are valid. This stage determines which transmitter is 'valid' in the case of two incoming, identical signals. A signal is marked valid based on the power calculation. The receiver then locks onto the valid signal and marks every other signal as interference. The received power then becomes the signal power in the signal-to-noise ratio calculation.

2.2.1.9. Stage 9 – Background Noise

The background noise stage is the first of two stages designed to calculate noise in the communications system. Background noise takes into effect galactic, urban, or thermal noise. The noise is determined after three assumptions as to the source of noise: a constant ambient noise, constant background noise source, and constant thermal noise at the receiver. The noise is calculated through the power spectral density in the transmission band. Aggregate thermal noise is calculated through Boltzmann's constant, held at 290° K. This noise is multiplied over the entire bandwidth, and is subject to receiver antenna gain. The other portion of the total system noise comes from interference.

2.2.1.10. Stage 10 – Interference Noise

Interference noise is that noise which is attributed to signals in the transmission band that cannot be decoded by the receiver. When the channel match stage is executed, a transmitted signal is deemed as invalid, valid, or interference. A receiver then ignores the signal if it is invalid, 'decodes' the signal if it is valid, or calculates the amount of noise the signal contributes to the valid signal. The interference noise stage can be executed for as many signals which interfere with a valid signal at every time interval during a simulation.

2.2.1.11. Stage 11 – Signal to Noise Ratio (SNR)

The SNR stage simply determines the ratio of signal to noise by dividing the previously calculated powers. The SNR divides the valid received signal power by the sum of the interference noise and the background noise. This number is then converted into decibels. The SNR stage is executed each time the interference noise and background noise stages are executed.

2.2.1.12. Stage 12 – Bit Error Rate (BER)

The BER stage finds a bit-error rate for the signal based on the SNR and the modulation of the signal. Predetermined modulation curves are used as look-up tables, as the SNR is used to find the BER at the correct simulation time. The SNR used by the stage is called the 'effective SNR', and is the sum of the calculated SNR and a processing gain associated with the bandwidth and data rate of the signal.

2.2.1.13. Stage 13 – Error Allocation

The Error Allocation stage uses the BER and the packet length to determine how many, if any, errors there are in the packet. The error allocation stage may be calculated many times during the transmission of a single packet, as there can be multiple SNR's, which leads to a dynamic BER. The accumulation of bit-errors is given as the number of errors in a packet.

2.2.1.14. Stage 14 – Error Correction

The last stage, error correction, simply decides if the encoding power of the signal has enough error correcting capabilities to overcome the number of packet errors. Users choose how many bit errors a packet may have, while still being considered valid. The stage compares a user-defined number to the number of bit-errors, and decides if a packet is correct or incorrect.

2.3. AN/TSC-94A

The radio communication system modeled is the Satellite Communications Terminal AN/TSC-94A (TSC-94). The TSC-94 is designed to provide super high frequency (SHF) radio links to a communications satellite for up to twenty-four voice and/or data subscribers. It also has the capability to handle secure voice transmissions. The TSC-94 is broken up into five main subsystems: The Antenna group, the Receiver group, the Transmitter group, the Communications subsystem (to include the multiplexer/demultiplexer group and modem group) and the Power group. The three main groups are the receiver group, the transmitter group, and the communications subsystem. These three groups contain the bulk of the information required to accurately model the TSC-94.

2.3.1. Antenna Group

The main function of the antenna group focuses on the different modes of operation which are possible using the TSC-94. There are three possible ways to send a signal from one radio to the next. Figure 1 shows how direct broadcast to another TSC-94 is one possible form for communication.

Figure 1. Direct Broadcast

In Figure 2, a signal is sent to its destination via the AN/TSC-100A satellite. The signal is transmitted from one radio to a satellite, and then relayed to another radio by the satellite.

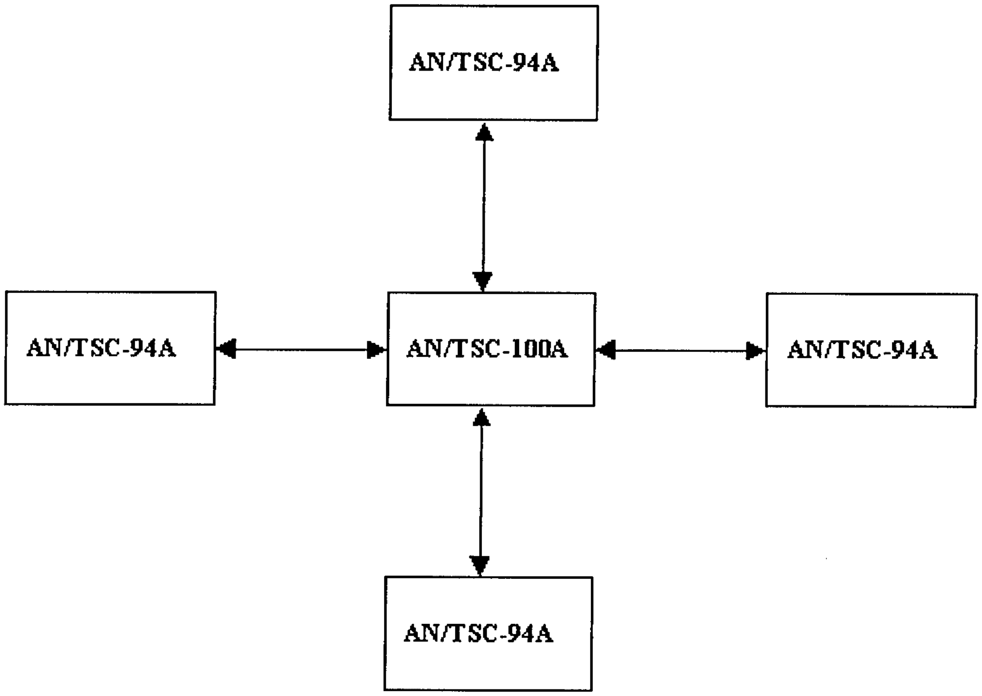


Figure 2. Nodal Communications

Figure 3 shows the mesh method of communication. The signal originates in one TSC-94, and is transmitted to the closest satellite. The satellite relays it to another satellite, which is able to find the destination radio, and transmit the signal. These are the three possible methods of signal propagation using the TSC-94 and its associated satellites.

The antenna group takes the received signal (on a carrier of 7.25 GHz to 7.75 GHz) to the receiver group. It also broadcasts signals from the transmitter group (on a carrier of 7.9 GHz to 8.4 GHz) [5].

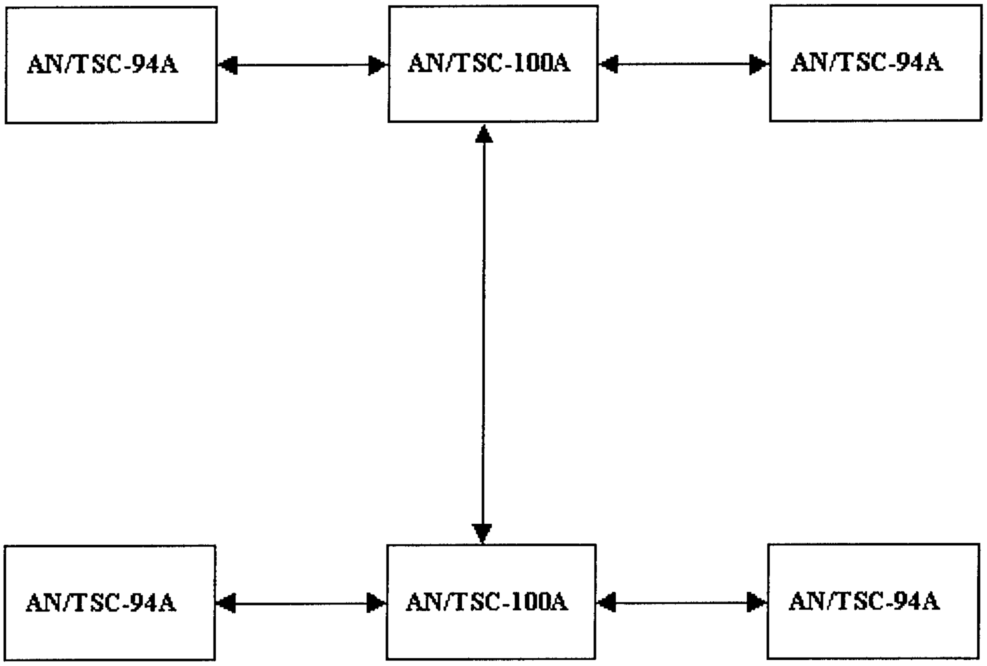


Figure 3. Mesh Communications

2.3.2. Receiver Group

The receiver group amplifies the SHF signal, as well as downconverting it to an Intermediate Frequency (IF) signal at 70 MHz. The antenna-mounted electronics located in the receiver group ensure the antenna is directed correctly to receive signals from the satellite. These signals are in the frequency band of 7.25 GHz to 7.75 GHz. The first process through which a signal goes is the downconversion. Signals which are received in this 500 MHz band are attenuated by the SHF input Bandpass Filter (BPF) by 0.4 dB. All other signals are attenuated by approximately 20 dB. The 70 MHz signal is sent to the front panel as a modem interface, and the second is amplified and sent to built-in test equipment (BITE) which checks the signal for errors [5].

2.3.3. Transmitter Group

Generally, the transmitter group acts like the receiver group in reverse. The signals are upconverted to the assigned carrier for transmission in the antenna group. The transmitter group is also responsible for providing the necessary power to the signal for its journey into space. The transmitter group, like the receiver group, also has controls for antenna positioning.

This group will produce a signal in the SHF frequency band that ison a carrier between 7900 MHz and 8400 MHz. The SHF signal is passed through a filter to ensure the signal is clean, and an intermediate power amplifier (IPA) module supplies a gain of 29 dB to the output SHF over the 500 MHz band. The RF amplifier takes the SHF signal from the upconverter at +21.0 dB (minimum) and applies up to 20 dB of amplification [5].

2.3.4. Communications Group

The communications subsystem processes both transmit and receive signals between the VF/DATA entry panel and the modem patch panel. The terminal contains two multiplexer/demultiplexer TD-1389s. Each can process four types of signals: 1) Digital, 2) Analog, 3) Frequency Shift Keyed (FSK), and 4) Conditioned diphase (secure). The modem group combines all the signals from the multiplexer/demultiplexer group into composite 70 MHz IF signals. During non-jammed operation, the signal is produced by the MD-945 modem. If jamming is present, the anti-jam (AJ) modems are used to produce a spectrum of signals 40 MHz wide.

The TD-1389, commonly called the low rate multiplexer (LRM) multiplexes and demultiplexes 12 full duplex channels on the user input side. It has five main modes of operation: 1) LRM, where information is multiplexed into a composite signal consisting of up to 12 user channels, not to exceed a composite data rate of 256 kbps, 2) Loop Group Modem (LDM), provides seven full duplex user channel uimputs at a data rate of 16 or 32 kbps with composite outputs of 128 kbps and 256 kbps, respectively, 3) GSC-24, emulates the AN/GSC-24 multiplex format and operating characteristics, 4) Group Modem, emulates the MD-1026 group modem, converting CDI to NRZ (Tx) and NRZ to CDI (Rx), 5) Rate Pass, overrides all normal functions of the LRM and process only channel one at the 75 baud data rate. The maximum composite data rate for the TD-1389 is 256 kbps.

The KG-81 Trunk Encryption Device (TED) performs digital encryption/decryption in full-duplex synchronous operation, at data rates of 9.6 kbps to 20 Mbps [5]. The TD-1337 provides digital combining and decombining functions necessary to accommodate the interfaces of the MD-945 modems and the AJ modems with the baseband equipment (LRMs, KG-81s, KG-84) and the TRI-TAC signals. The output of the TD-1337 consists of a single digital signal, produced by combining the various inputs.

The modem group consists of two MD-945 modems, one AJ modem, and various instruments for modem and channel control. The MD-945 modem is a BPSK/QPSK modem capable of transmitting and receiving data traffic at any rate between 6 kbps and

4.99999 Mbps. This 70 MHz IF signal is demodulated and differentially decoded, if error correction encoding is not used. Otherwise, the data is Viterbi decoded prior to sending the data to the TD-1337. The received binary NRZ data is supplied directly to the TD-1337. The AJ modem transmits and receives spread-spectrum code division multiple access (CDMA) signals at any rate between 75 bps and 32 kbps.

3. Methodology

This chapter presents implementation details for concepts and techniques presented in Chapter 2. Modeling a Direct Sequence Spread Spectrum (DSSS) link required an evaluation of the important effects of DSSS. Actual spread spectrum communication systems include spreading code generation hardware, but spreading code application is not the important effect of DSSS. Spreading code properties such as processing gain, low transmission power, and cross-correlation provide the most important characteristics of a DSSS system.

The basic OPNET Transceiver Pipeline lacked the capability to effectively model a DSSS link. Using the OPNET link, a high transmit power radiated from the transmitters to receivers in close proximity to the source. With a high signal power, the low probablitly of intercept (LPI) characteristic associated with low transmit power (in DSSS) was removed the from the signal. Processing gain not only contributes to transmission power reduction, but to noise power reduction at the reciever. By reducing the amount of jammer noise at the receiver, the jammer is defeated. OPNET makes no attempt to remove jamming power from received signals. By modifying the OPNET Transceiver Pipeline, these effects can help modify the basic OPNET radio link for DSSS modeling.

The AN/TSC-94 is a complex radio model. The Technical Order (TO) revealed many internal components such as multiplexers, encryption devices, and control devices which contributed to the overall radio performance. Evaluation of the AN/TSC-94 TO presented opportunities to reduce the model complexity while retaining overall system performance parameters. Many AN/TSC-94 performance parameters are modeled through simple modifications to transmitted packets.

During normal (non-DSSS) mode, the AN-TSC-94 operates through the MD-945 modem, which allows up to twelve full-duplex communications channels. The composite data rate for this modem is 256 kbps. Instead of dividing the radio traffic between twelve channels, a single channel is implemented to model the full capacity of all channels. Through this implementation, the "worst-case" scenario is realized, as the MD-945 is operating at full capacity, over the full transmission bandwidth of the AN/TSC-94 radio.

3.1. OPNET Implementation of DSSS System

Specific DSSS system characteristics include low transmission power, high signal bandwidth, and interference noise reduction. To model these effects, modifications were made to various Transceiver Pipeline stages. The transmitter and receiver modules include parameter fields necessary for system modeling, and include signal bandwidth, spreading code, and processing gain. In OPNET, DSSS processing gain defaults to a calculated value equal to the signal bandwidth divided by the data rate. Despite OPNET's predefined fields, several stage modifications were required to successfully implement DSSS capability. Specific code changes are listed in Appendix A.

3.1.1. Transceiver Pipeline Modifications

Modifications in the Transceiver Pipeline were only required in a few stages. The stages changed to simulate DSSS performance include the Channel Match, Transmitter Antenna Gain, Receiver Antenna Gain, Signal – to – Noise Ratio, and Bit Error Rate stages. There were modifications required in both the transmitter and receiver themselves, before the link could be used. Addition of the attributes listed in Table 1 enables the program to obtain necessary values for successful simulation.

Attribute	Variable Type	Default Value
Family	Integer	0 (disabled)
Cross Correlation	Double	1 (disabled)
Processing Gain	Double	BW/R_d

Table 1. Receiver Extended Attributes

3.1.1.1. Channel Match Stage Modifications

The Channel Match stage provides a check to ensure all relevant signal information exists to allow the transmitter and receiver to communicate. Certain spread spectrum spreading codes possess unique auto – and cross – correlation properties. For this research, a cross-correlation value is assigned between zero and one to represent the degree to which one signal "matches" another signal. The actual cross-correlation value is based on spreading code length and other code properties. Here, a 'family' of codes posess the unique qualities of minimal cross – correlation values between families, and reduced intra-family cross – correlation values.

The first Channel Match stage modification adds the 'family' attribute to the list of channel matched properties. If the code's family attribute does match, the signal is tagged as interference and its signal power spectral density is used in noise calculation. If the family attribute matches, along with all other attributes, the received signal is tagged as valid, and can be processed in later stages. The family attribute is another channel property, which must be synchronized for proper communication.

3.1.1.2. Transmitter / Receiver Antenna Gain Modifications

The Transmitter Antenna Gain and Receiver Antenna Gain stages account for the low power feature of a DSSS link. As the Transceiver Pipeline is executed, information is passed through the stages using Transmission Data Attributes (TDA's). TDA's contain information for transmitters and receivers and can be used to obtain any information about the link. One specific TDA is the *signal transmitted power.* Transmitted power is taken from the transmitter channel attributes, and its value is not modifiable by a Kernel Procedure, i.e., it is a read-only value. Since spread spectrum (SS) techniques reduce signal power ($^{\text{Watts}}/_{\text{Hertz}}$), the TDA for transmitted power must be adjusted. The transmitted signal must not interfere with other signals; radiated power is drastically reduced by the SS processing gain. In an effective system, the SS processing gain can reduce the transmitted power below the ambient noise level.

The transmitted power may not be modified in the pipeline stages. Instead, the processing gain is incorporated into the Transmitter Antenna Gain stage (transmitter gain may be modified). Given the look-up table for antenna gain is in dB, converting processing gain into decibels (dB) enables the user to apply the processing gain directly to transmitter antenna gain. The processing gain is subtracted from the antenna gain, yielding a negative gain which is the desired result. The effective transmitted power is then less than the original signal power. The Transmitter Antenna Gain stage reduces the radiated signal power in all directions.

Receiver Antenna Gain is able to reverse the effects that spreading has on a transmitted signal. However, this only applies if the receiver and transmitter can match spreading codes. By modifying this stage, the user is able to radiate a low power signal and reconstruct the original power characteristics by re-applying the processing gain at the receiver antenna; processing gain (dB) is taken from the receiver attributes and added to the receiving antenna gain. This process is not performed for signals with non – matching spreading code criteria. Given the transmitted SS signal is below the noise level, it provides minimal interference to other receivers and is less detectable by hostile devices.

3.1.1.3. Signal –To – Noise Ratio Modifications

The Signal-to-Noise Ratio (SNR) stage is where signal power is divided by noise power, and stored for later use by the Bit Error Rate (BER) stage. It is in this stage that the processing gain is applied to the signal. The processing gain is inserted as part of the SNR calculation, multiplied by the original SNR.

DSSS signals using spreading codes with family cross-correlation values are accounted for in the SNR stage. The cross-correlation value is read from the receiver attributes and multiplied by the interfering signal's received power. This accounts for cross correlation interference between two codes of the same family. DSSS signals that arrive simultaneously with codes from different families are treated as interference.

3.2. Radio Models

The two systems modeled in OPNET are the AN/TSC-94 SHF radio and the AN/TSC-100 satellite system. The satellite is modeled as a "bent pipe", i.e., signals are received and re-transmitted without processing. The AN/TSC-100 receives signals on the

SHF radio transmit band (7.9 GHz to 8.4 GHz) and broacasts on the SHF receive band (7.25 GHz to 7.75 GHz).

3.2.1. AN/TSC-94 SHF Radio Model

The AN/TSC-94 radio uses multiple channels and must be able to switch to Anti-Jam (AJ) mode when an undesired channel environment exists. This situation can be characterized as yielding high BER. An OPNET node model for the AN/TSC-94 is shown in Figure 4, where traffic generators are shown for each channel supported. The traffic generators send packets into a processor which deliver or destroy packets based on switch criteria.

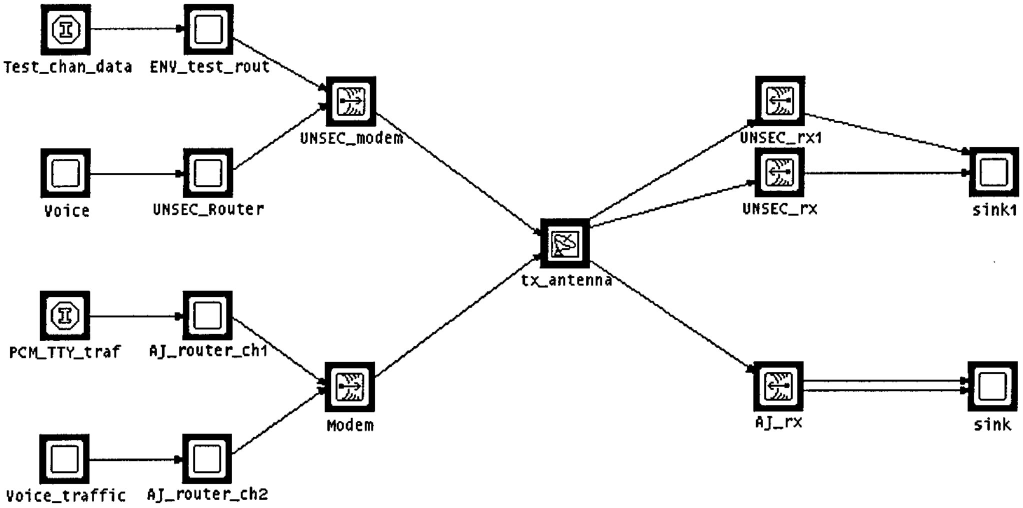


Figure 4. OPNET Node Model of AN/TSC-94

If BER becomes too high, packets transmitted over the DSSS link are passed to the transmitter and the spread spectrum link is enabled. When the BER remains below a specific threshold value, the DSSS channels remain disabled at this processor, and the radio continues normal operation.

The traffic generation for the normal mode include the test channel, and a voice channel. The voice channel is a single channel representation of the composite 256 kbps capacity, 12-channel modem. This represents the highest loading case for the radio. The AJ mode presents Pulse Code Modulated (PCM), Teletype (TTY), and voice traffic.

The switching processor is a three state device as shown in Figure 5.

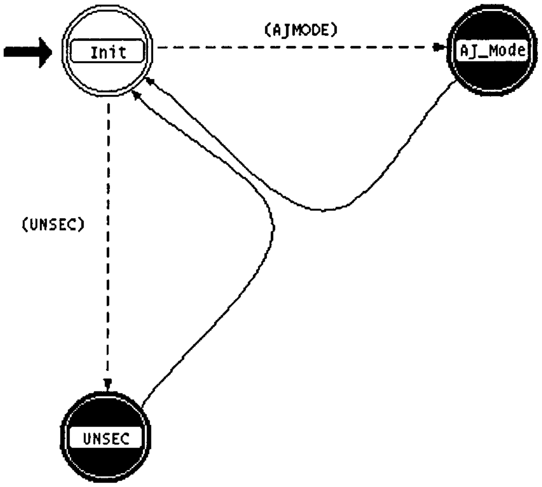


Figure 5. Generic Process Model for Routers

The initial (Init) state takes packets from the incoming streamand checks to see if a specific transmitter attribute has been set. This attribute (Bit Errors) acts as the switching criteria. If the Bit Errors attribute is below the threshold value, or has not been set, the radio defaults to an unsecure (UNSEC) mode of communication. However, if the BER is set and below the threshold, the radio sends packets via the AJ modem.

The second and third states decide how to deal with the packet. These two states represent normal (UNSEC) and spread spectrum (AJ) channels. Normal channels operate regularly over the link, i.e., with no spread spectrum characteristics. At least one normal channel is broadcasting at all times to provide an accurate, uniform BER calculation for estimating system performance. All other unsecure channels may be disabled if the radio converts to AJ mode which is normally disabled. Although it generally operates mainly below the ambient noise level, the spread spectrum signal is disabled during normal operation to simulate switching the primary transmitted signal between the MD-945 (UNSEC) and AJ Modems.

Once a packet is designated for transmissison, it is passed to the OPNET transmitter, where the user is permitted to input channel characteristics, i.e. the user may set channel bandwidth, center frequency, data rate, and transmission power. The user also sets the spreading code value for the system at this point. When a packet arrives at a transmitter module, a preliminary set of TDA's are defined. While some TDA's can be modified, signal information such as data rate, transmission power, and frequency may not be changed. Because these values are transmitter specific, they cannot be modified anywhere else in the simulation, e.g. at the receiver. The transmitter also executes the first six stages of the Transceiver Pipeline.

Packets are sent from the transmitter module to the antenna module. The antenna module contains power gain information, as well as directional attributes designed for antenna steering. By assuming perfect beam lock, the antenna direction attributes may be ignored. For this research, identical antenna characteristics are used for transmitting and receiving.

Receiver modules accept packets from antenna modules, activating the final eight Transceiver Pipeline stages. The remaining TDA's are set throughout these stages. The TDA's written by the transmitter stage serve to provide information about channel and signal characteristics, while the receiver written TDA's relay statistical information concerning system performance. The receiver sends the packets to the final processor as shown in Figure 6.

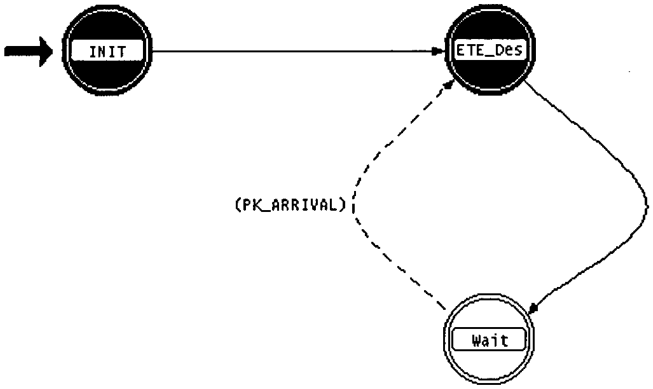


Figure 6. Generic Process Model for data sink

This module retreives bit errors information about the packet and passes it to the transmitter module. Using the OPNET Kernel Procedures *op_ima_obj_attr_get()* and *op_ima_obj_attr_set()*, the transmitter and receiver can access (get) and write (set) values used for channel switching. The receiver 1) gets newly acquired information about the channel, 2) reads the bit errors, and 3) sends it. When the transmitter sees a large (or small) BER, it knows to transmit in DSSS (or normal) mode.

There is one channel that will broadcast information in normal mode at all times. If a high SNR drives a system into AJ mode, packets in the normal channel are destroyed before they reach the transmitter modules. Since the environment SNR should be taken from only one source, a uniform, steady transmission is best suited for SNR measurement. If AJ mode destroys packets before they are transmitted, packets cannot reach the receiver, and SNR estimation becomes impossible. Estimation of a single, continuous transmission allows accurate bit error calculation.

3.2.2. AN/TSC-100

The satellite model is designed to retransmit received signals at a new carrier frequency, permitting intended receivers to receive signals in the correct frequency band. The AN/TSC-100's "bent-pipe" design gives rise to the OPNET model configuration in Figure 7.

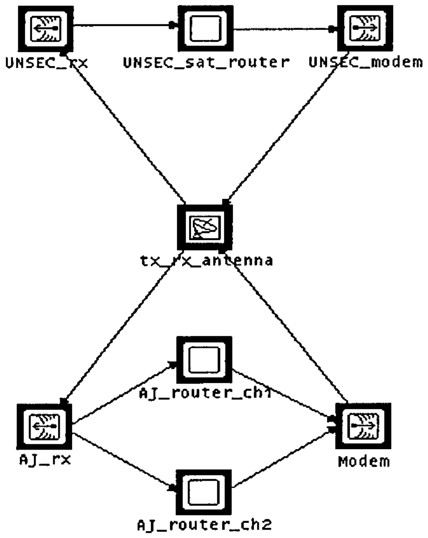


Figure 7. AN/TSC-100 Satellite Node Model

As in the AN/TSC-94, the antenna module serves both the transmitter and receiver modules. The signal is first received by either the AJ or normal mode receiver. Differing channel characteristics ensure the packet is delivered to the correct module. After the Transceiver Pipeline stages execute, packets are sent to the satellite processor. This processor reads the bit errors from the transmitter and makes a hard decision about which channel is best for re-broadcast. The packet TDA's are assigned and it is sent back to the antenna for transmission. The packet is then retransmitted over a new channel. The AN/TSC-94 radio transmits and receives on mutually exclusive frequency bands [5].

4. Results

Spread spectrum systems are characterized by three general attributes: jam resistance, transmit power reduction, and increased Bit Error Rate (BER) perfomance. The Direct Sequence Spread Spectrum (DSSS) model accounts for all three metrics. In the presence of three different jammers, a DSSS system can operate with minimal bit error performance degredation. The advantage gained using DSSS is demonstrated through the following tests. Also, BER and Signal to Noise Ratio (SNR) values collected during OPNET simulations are shown as time averages, and thus represent steady state conditions.

4.1. OPNET Link Characterization

It is necessary to examine how a communications link performs in OPNET. The standard OPNET Tranceiver Pipeline is used without modification to provide a basis for BER performance increases. Factors for the following tests are based on specific information for the AN/TSC-94, so that the link may more easily be incorporated into the radio model. A bursty traffic source is used to simulate voice traffic.

The first test includes a transmitter (Tx) and receiver (Rx) pair separated by 1000 km. The link is characterized as either normal or DSSS. The *normal link* has no modifications to the OPNET Transceiver Pipeline stages, i.e. the standard OPNET communications link. The *DSSS link* employs several techniques to reduce transmission power and increase BER performance, especially in a hostile environment.

The first test configuration serves to demonstrate communication performance over a single communications link. Factors for the test include transmit power, altitude, modulation, simulation time, and jammer power. The test configuration is given in Table 2. The transmit power, modulation, and other parameters were chosen based on specific AN/TSC-94 performance characteristics. The simulation time was chosen to ensure the average BER and SNR values reach a steady state value.

Factor	Value
Transmit Power	500 W
Altitude	200 M
Modulation	BPSK
Simulation Time	2 hr
Jammer Power	100 W

Table 2. Test Factor Settings

4.2. Cross Channel Interference

This test is used to show how the channel differs when two Tx – Rx pairs are simulated. One set is a normal pair and the other has DSSS capabilities. This provides a basis for later tests, which incorporate different jamming techniques are used. The test configuration is shown in Figure 8.

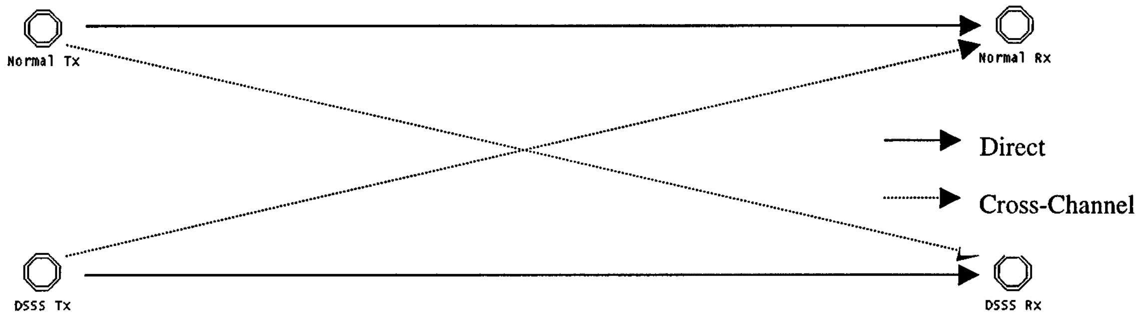


Figure 8. Test Configuration for Normal Mode

Figures 9 and 10 show SNR performance of the normal and DSSS links, respectively.

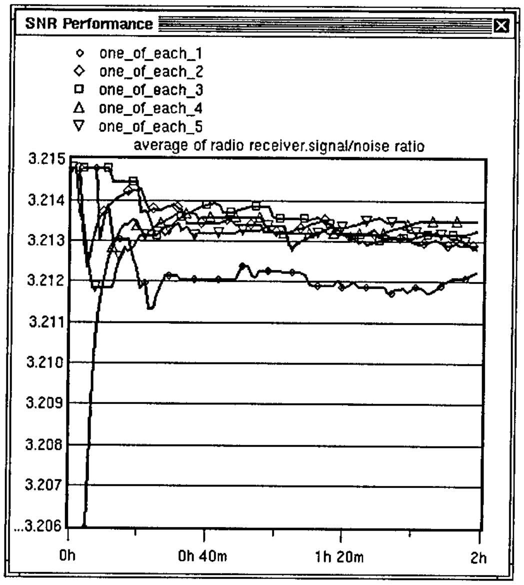


Figure 9. SNR of Normal Link

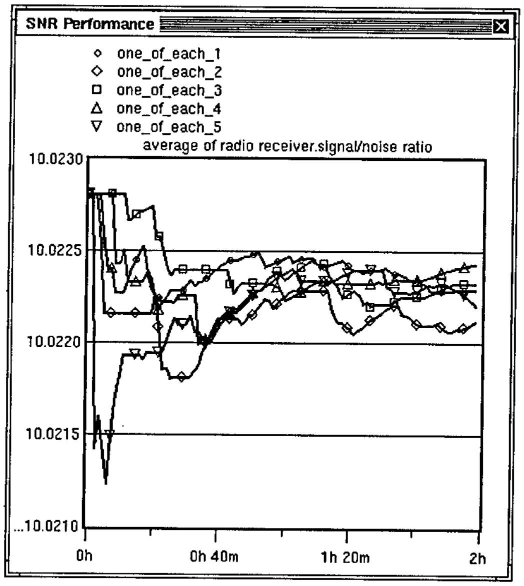


Figure 10. SNR of DSSS Channel

The plot's key represents scenario name (one_of_each_#) and the number of the run. The approximate 7 dB increase in average SNR for the DSSS translates to increased BER performance, the desired result for the link. This increase comes from the amount of noise power the DSSS link is able to "spread". By spreading the noise's power spectral density over a larger bandwidth, a smaller portion of the input noise power remains in the desired signal bandwidth and the SNR is effectively increased.

4.3. Anti-Jam Performance

A major feature for DSSS is its ability to operate in a hostile environment. The next three test scenarios show how DSSS offers increased SNR while maintaining a relatively constant / low BER. A multiple-mode jammer was placed directly between the receivers used for normal mode tests as illustrated in Figure 11. The DSSS Anti-Jam (AJ) feature enabled the spread spectrum (SS) link to continue operation, while the normal link suffered data loss in all three cases, evident in degraded BER performance.

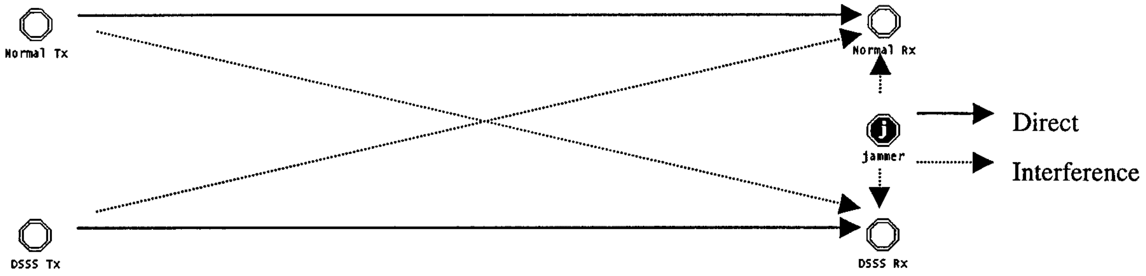


Figure 11. Anti-Jam Test Configuration

The three jammer types, single band, pulsed, and swept band, were centered at the Tx-Rx pair center frequency and represent a worst case scenario, e.g. an enemy is able to pinpoint and strategically attack at the transmission frequency. The single band jammer operates continuously over the entire jamming bandwidth, the pulsed jammer sends repeated pulses in the jamming bandwidth, and a swept band jammer switches on all jamming bandwidth frequencies. All jamming methods were simulated to show link effectiveness against multiple jamming forms.

Normal and AJ SNR plots (single band jammer) are shown in Figures 12 and 13.

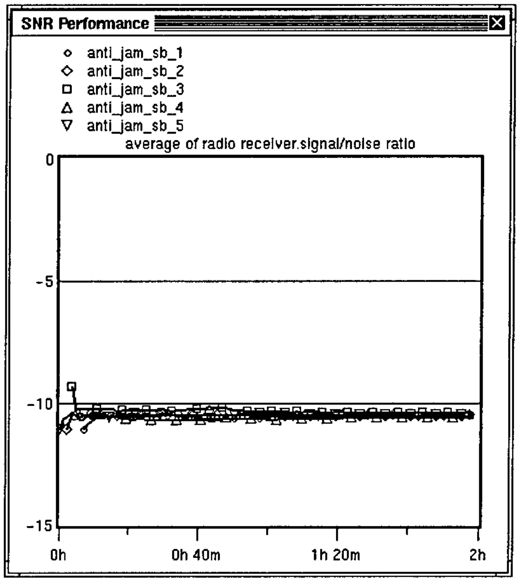


Figure 12. Single Band Jammer SNR (Normal)

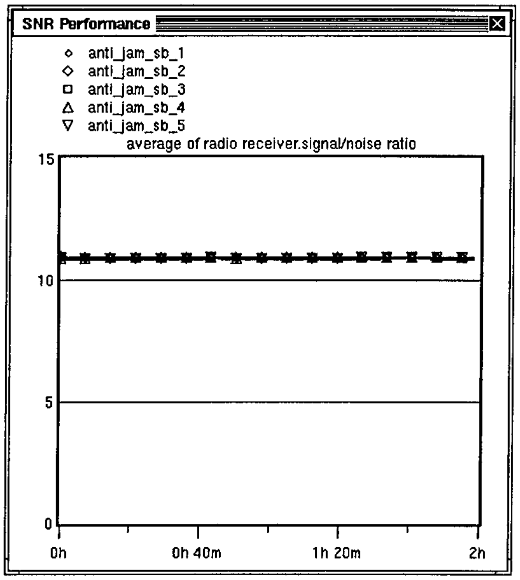


Figure 13. Single Band Jammer SNR (AJ)

The same test setup was run with five separate random number seeds. The x-axis represents time (0 to 2 hours), the y-axis is SNR in dB. The jammer increases the noise seen at the receiver, decreasing the SNR by 22 dB. The lower SNR causes the BER performance shown in Figure 14.

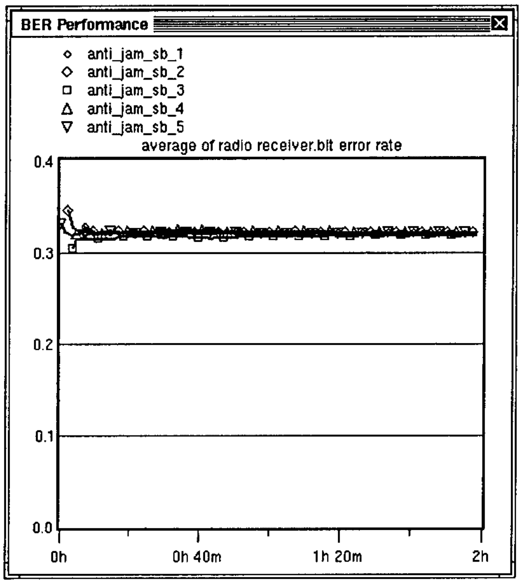


Figure 14. Normal Receiver BER for Single Band Jammer

Processing gain for a single pulse jammer (single band) is given in Equation 1 [1]. The processing gain is the code chip rate over the data rate. In AJ mode, the code chip rate is 20 MHz, and the maximum data rate is 32 kbps, which yeilds a G_p of 625, or 27.9 dB.

$$G_p = \frac{R_c}{R_d} \tag{1}$$

The low SNR in Figure 12 causes high BER, as expected per Equation 2 [1] for BPSK modulation

$$P_B \approx Q\left(\sqrt{2\frac{E_b}{N_0}}\right) = Q\left(\sqrt{2(SNR)}\right) \tag{2}$$

A SNR of −10.5 dB, from Figure 13, yields a theoretical BER of 0.4325 per Equation 2. This BER is higher than shown in Figure 14 (0.32), but both result in a severely degraded signal. BER and SNR plots for the swept band jammer case are located in Appendix B.

4.4. Cross Correlation

The cross correlation value represents the degree to which two signals in the same DSSS family are alike. Receivers may be able to demodulate parts of an incorrect signal if the cross correlation is too high. This test shows how a receiver in the correct family with a different spreading code is able to detect the information in another DSSS Tx-Rx pair. The transmitter and receivers were placed equidistant apart as shown in Figure 15.

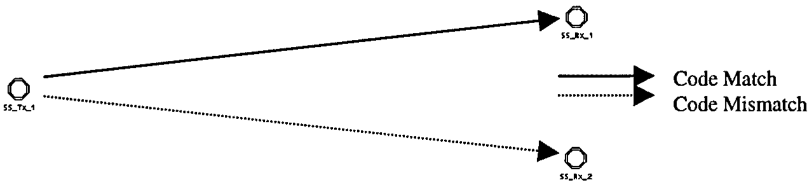


Figure 15. Cross Correlation Test Setup

The cross correlation values were set at three levels, to represent a low, medium and high correlation. The lowest value (0.25) returned no bit errors, as the receiver was able to distinguish the incorrect signal. The medium value (0.55) produced higher throughput, as the received signal was more closely related to the receiver spreading code. The average throughput for the medium and high (0.8) cross correlation values are shown in Figures 16 and 17. Note that as the cross correlation increases, the throughput is increasing as well. As the second receiver's spreading code is more correlated with the original spreading code, what should be noise in the system is being demodulated as a valid signal. The second receiver is demodulating noise, because the cross correlation values are so large. Figures 16 and 17 show how noise is demodulated as valid signal.

The mean traffic in Figure 16 is 219.77 bps, with an upper 2-σ value of 269.37. The lower 2-σ value for Figure 17 is 341.71, with a mean traffic rate of 446.63 bps. With 95% confidence, the throughput is higher for the larger cross correlation value.

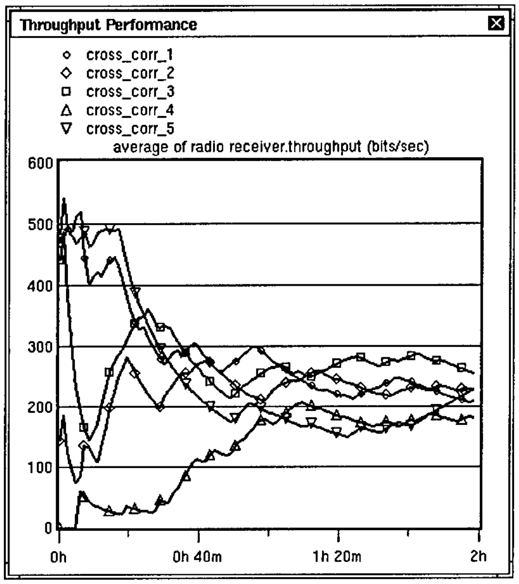


Figure 16. Throughput, Cross Correlation, 0.55

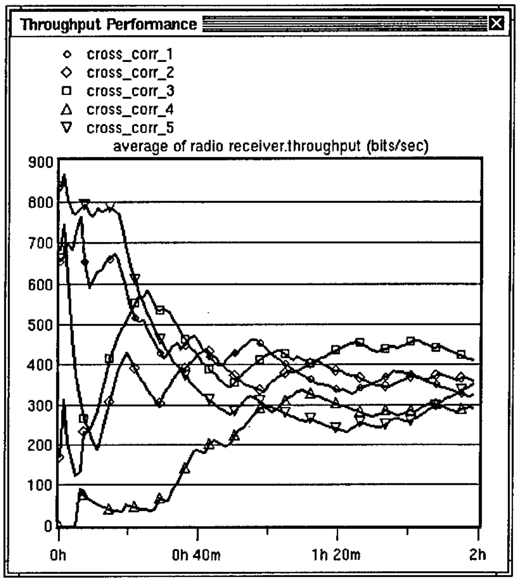


Figure 17. Throughput, Cross Correlation, 0.8

4.5. AN/TSC-94 Radio Performance

The most important consideration throughout the development of the model radio remained the impact it would have on NETWARS. Questions arose about internal model complexity, but the important parts of the simulation are the external effects of radio use. These effects include reduced transmit power, the correct frequencies and data rates, and the ability to switch to AJ mode. The model was tested to determine if AJ mode 1) was attainable through a simple switching mechanism and 2) helped to defeat jammers in the environment.

The switching criteria used was the number of bit errors in a transmitted packet of length 1,024. The 'UNSEC_router' processor reads in the most recent bit error calculation for a packet. A running average of bit errors for the past 30 packets is calculated, and used as switching criteria. If a packet has a large number of errors, 300, the average immediately increases, and the system is driven into AJ mode. The transmitter remains in AJ mode until 30 correct packets have been transmitted and the average bit errors drops below 10. This process yields a BER of 0.09765.

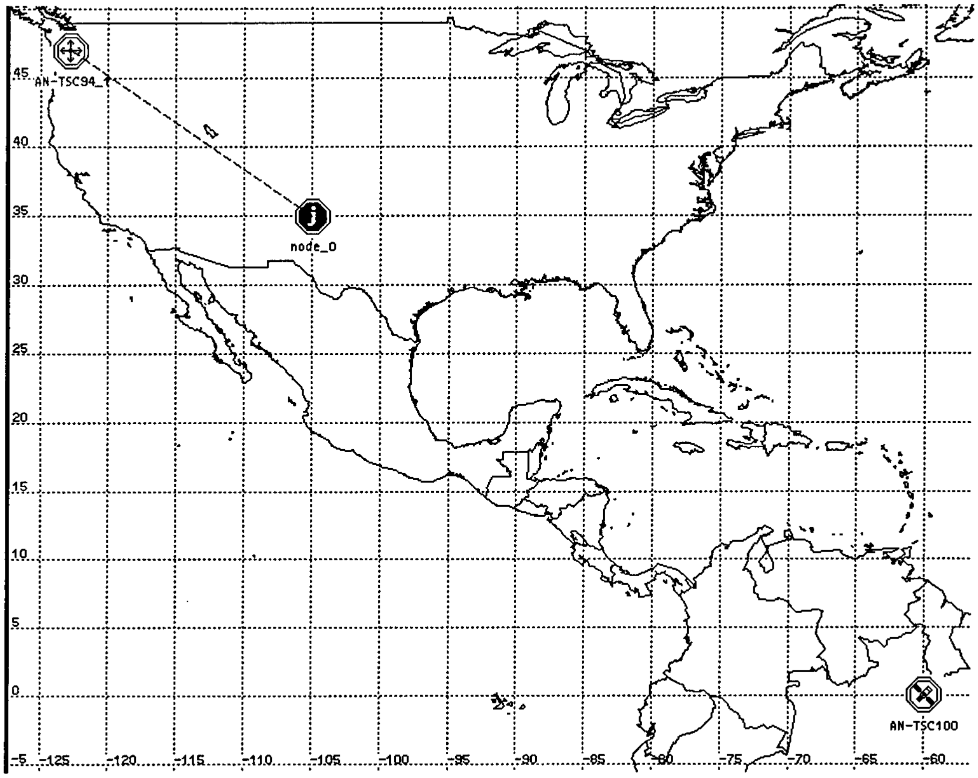


Figure 18. Test Configuration for AN/TSC-94 AJ Switching

The test setup for AJ mode switching is shown in Figure 18. A single AN/TSC-94 is moved toward a jamming source for 40 minutes, waits 'atop' the jammer for 40 minutes, and is returned to its original position for the final 40 minutes. The AN/TSC-100 satellite is in a geostationary orbit. Figures 19 and 20 show the throughput of the system when the jamming source starts to affect packet reception As the radio moves towards the jammer, the radio is in normal transmission mode. Because SNR, shown in Figure 21, remains at an acceptable level, packet reception continues. When the normal mode SNR drops, at about the 40 minute mark, packet errors are reported to the transmitter, and the AJ transmitter becomes the primary transmission source. The normal mode transmitter discontinues sending voice traffic over the unsecure channel. At this 40 minute mark, the radio switches to AJ mode and voice traffic is destroyed before it reaches the transmitter, but test channel data is continuously transmitted. During the on-times for normal mode, traffic is bursty in nature, accounting for the bursty nature of voice communication. The normal mode transmission and reception during non-jammed times differ only by the test signal. That is, there are no bit errors during the normal transmission time.

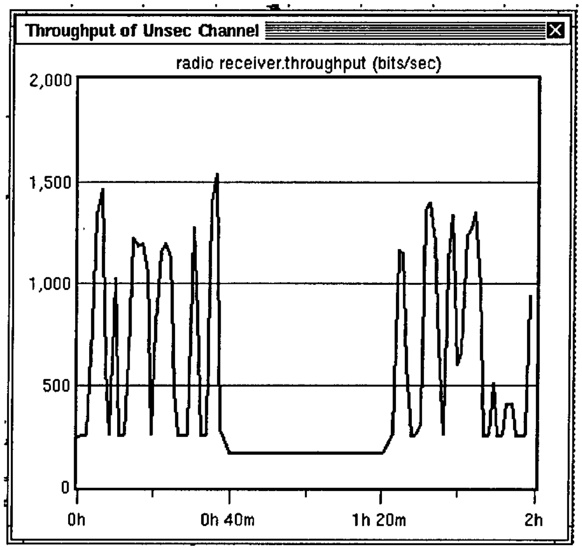


Figure 19. Throughput, Normal Receiver

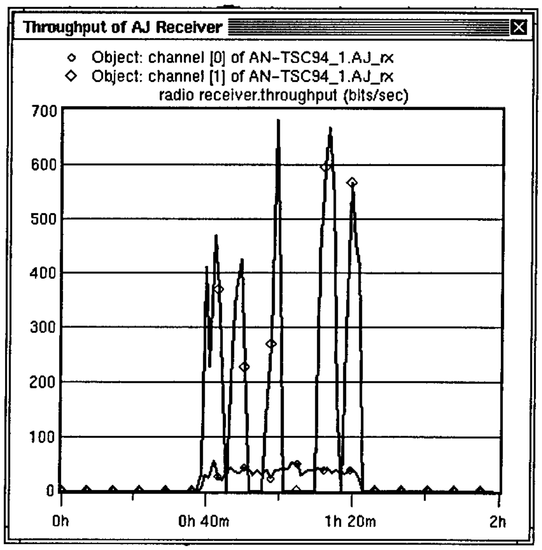


Figure 20. Throughput, AJ Receiver

The SNR is increased by as much as 24 dB in the AJ mode over the normal mode. The SNR can only be calculated for the AJ receiver during the period when packets are received. The continuous SNR represents the tracking channel used to continuously estimate the system. There are brief periods when the normal SNR increases, but the system is still in AJ mode. This is due to the lag on the 30 packet average, so the system does not return to normal mode sporadically because a single packet is received correctly.

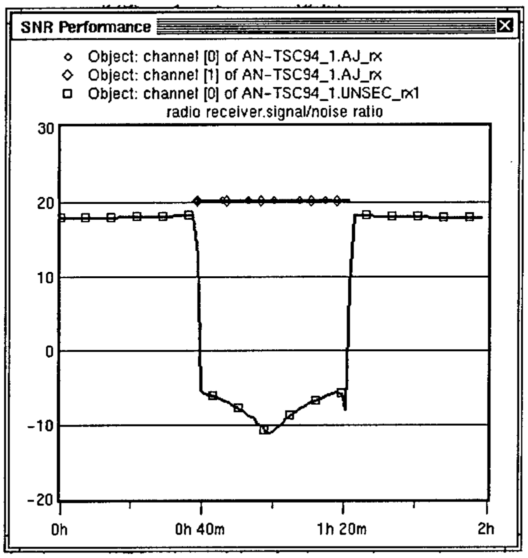


Figure 21. SNR for AJ and Normal channels

5. Conclusion

The DSSS link is able to operate in a jammed environment with improved Bit Error Rate (BER) over a normal OPNET transmission channel. By applying the processing gain to the signal before Direct Sequence Spread Spectrum (DSSS) transmission, the transmitted power is significantly reduced. A reduced transmission power reduces the affect the transmission has on other receivers.

The processing gain is also used to increase the received signal power when the spreading codes and families are matched. Because transmitted and received power cannot be modified in OPNET's Transceiver Pipeline, the gain of the antenna is decreased / increased by a processing gain factor to decrease / increase the transmit / receive power. This accounts for the 'spreading / despreading' features associated with DSSS.

Finally, the processing gain serves to reduce overall noise power which increases the Signal-to-Noise Ratio (SNR) of the signal; received noise power is effectively divided by the processing gain. Reducing the noise power simulates the 'spreading' of the noise beyond DSSS bandwidth limits. By spreading the noise power, noise affects on the system are reduced and the SNR is effectively increased.

Increasing SNR leads to better (lower) BER perfromance, a desired result for this system. By increasing the SNR in the Anti-Jam (AJ) mode, bit errors are significantly reduced and the channel is able to operate within hostile or jammed environments. These changes allow a lower transmission power, increased SNR's, decreased BER, and resistance to jamming attacks.

The cross correlation feature of the link enables codes in the same family to be included in simulations. By increasing the cross correlation, the amount to which the two signals appear similar is increased. Smaller cross correlation values lead to independence of transmissions in the DSSS mode. At these small numbers, there is minimal interference between signals in the same family without the same spreading code.

The AN/TSC-94 uses this link while operating in a Super High Frequency (SHF) environment. The transmission of data in a jamming environment is aided by the DSSS system. The AN/TSC-94 is able to effectively defeat a 1 kW jamming source, by switching to AJ mode.

To effectively incorporate the AN/TSC-94 radio model into NETWARS, simple modifications may be required. To model the traffic patterns more accurately, the "twelve – in – one" data channel used in normal mode communications may need to be modified to obtain more accurate results. Also, antenna patterns may be incorporated into the model; a generic isotropic antenna model was used for this research.

5.1. Future Research

There are many opportunities for future research in this area. NETWARS is a large project and will continue to be developed in years to come. The present radio / link models are simply shells of the operating processes. Follow-on researchers can expect to model individual system components, including encryption devices, multiplexers and demultiplexers, as well as modems used for data transmission.

DSSS is only one method for spread spectrum communication. There are opportunities to model various spread spectrum techniques, including Frequency Hopped

Spread Spectrum. This method of spread spectrum changes the center transmission frequency, providing a difficult lock onto a specific jammable frequency. This technique may be used in multiple communications systems, making it a valuable addition to many simulations.

Finally, the AN/TSC-100 satellite system used in this simulation is simply a 'bent-pipe' representation. Future researchers may model this satellite, to provide true worldwide coverage for the AN/TSC-94.

BIBLIOGRAPHY

1. Sklar, Bernard. Digital Communications: Fundamentals and Applications. New Jersey: Prentice Hall, 1988.

2. Peterson, Roger L. et. al., Introduction to Spread Spectrum Communications. New Jersey: Prentice Hall, 1995.

3. Pritchard, Wilbur L. et. al. Satellite Communication Systems Engineering. New Jersey: Prentice Hall, 1993.

4. Rappaport, Theodore S. Wireless Communications: Principles and Practice. New Jersey: Prentice Hall, 1996.

5. Training Order Number 31R2-2TSC94-11

6. Atamna, Youcef. "NETWARS: Toward the Definition of a Unified Framework for Modeling and Simulation of Joint Communication Systems." SPIE Conference on Digitization of the Battlespace III, April 1998. SPIE Vol. 3393

VITA

Second Lieutenant David M. Banker was born on 9 November 1976 in Del Rio, Texas. He graduated from Juanita High School in Kirkland, Washington in June 1994. He entered undergraduate studies at the United States Air Force Academy in Colorado Springs, Colorado where he graduated with a Bachelor of Science degree in Electrical Engineering in May 1998. He was also commissioned through the United States Air Force Academy. His first assignment was as a Masters Student, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, in August, 1998. His areas of study included communications and digital communications networks. Upon Graduation, he will be assigned to the Sensors branch of the Air Force Research Laboratory.

PERMANENT ADDRESS: 14329 117th Ave NE, Kirkland, WA 98034

APPENDIX A

CODE CHANGES TO OPNET PIPELINE STAGES

Channel Match Stage

```
/* dra_chanmatch_DSSS.ps.c */
/* Default channel match model for radio link Transceiver Pipeline */

/****************************************/
/*              Copyright (c) 1993              */
/*                     by MIL 3, Inc.                          */
/*            (A Delaware Corporation)          */
/*     3400 International Drive,  N.W.          */
/*            Washington, D.C., U.S.A.          */
/*                     All Rights Reserved.     */
/*****************************************/
/* This model includes modifications for */
/* using the Transceiver Pipeline as a   */
/* Direct Sequence Spread Spectrum comm  */
/* link.                                 */
/*****************************************/


#include <opnet.h>


void
dra_chanmatch_DSSS (pkptr)
        Packet*         pkptr;
        {
        double          tx_freq, tx_bw, tx_drate, tx_code;
        double          rx_freq, rx_bw, rx_drate, rx_code;
        double          inoise, bnoise;
        int             tx_fam, rx_fam, objid_tx, objid_rx;
        Vartype         tx_mod;
        Vartype         rx_mod;

        /** Determine the compatibility between transmitter and receiver channels. **/
        FIN (dra_chanmatch_DSSS (pkptr));

        /* Obtain the object id for transmitter and receiver. */
        objid_tx = op_td_get_int (pkptr, OPC_TDA_RA_TX_OBJID);
        objid_rx = op_td_get_int (pkptr, OPC_TDA_RA_RX_OBJID);

        /* Obtain transmitting channel attributes. */
        tx_freq         = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_FREQ);
        tx_bw           = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_BW);
        tx_drate        = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_DRATE);
        tx_code         = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_CODE);
        tx_mod          = op_td_get_ptr (pkptr, OPC_TDA_RA_TX_MOD);
        if (op_ima_obj_attr_exists(objid_tx, "Family") == OPC_TRUE)
                op_ima_obj_attr_get (objid_tx, "Family", &tx_fam);
        else tx_fam = -2;

        /* Obtain receiving channel attributes. */
        rx_freq         = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_FREQ);
        rx_bw           = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_BW);
        rx_drate        = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_DRATE);
        rx_code         = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_CODE);
        rx_mod          = op_td_get_ptr (pkptr, OPC_TDA_RA_RX_MOD);
        if (op_ima_obj_attr_exists(objid_rx, "Family") == OPC_TRUE)
                op_ima_obj_attr_get (objid_rx, "Family", &rx_fam);
        else rx_fam = -2;

        /* For non-overlapping bands, the packet has no      */
        /* effect; such packets are ignored entirely.        */
        if ((tx_freq > rx_freq + rx_bw) || (tx_freq + tx_bw < rx_freq))
```

```
                {
                op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS, OPC_TDA_RA_MATCH_IGNORE);
                FOUT;
                }

        /* Otherwise check for channel attribute mismatches which would     */
        /* cause the in-band packet to be considered as noise.              */
        /* The family for the spreading code is checked to see if the       */
        /* two signals can communicate.  If so the code will be checked.*/
        if ((tx_freq != rx_freq) || (tx_bw != rx_bw) ||
                (tx_drate != rx_drate) || (tx_fam != rx_fam) || (tx_mod != rx_mod))
                {
                op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS, OPC_TDA_RA_MATCH_NOISE);
                FOUT;
                }

        /* Otherwise the packet is considered a valid transmission which    */
        /* could eventually be accepted at the error correction stage.              */
        op_td_set_int (pkptr, OPC_TDA_RA_MATCH_STATUS, OPC_TDA_RA_MATCH_VALID);

        FOUT;
        }
```

Transmitter Gain Stage

```
/* dra_tagain_DSSS.ps.c */


/****************************************/
/*              Copyright (c) 1993              */
/*                      by MIL 3, Inc.                          */
/*              (A Delaware Corporation)        */
/*      3400 International Drive,  N.W.         */
/*              Washington, D.C., U.S.A.        */
/*                      All Rights Reserved.    */
/*****************************************/
/* This model includes modifications for */
/* using the Transceiver Pipeline as a   */
/* Direct Sequence Spread Spectrum comm  */
/* link.                                 */
/*****************************************/


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

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

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


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

void
dra_tagain_DSSS (pkptr)
        Packet*         pkptr;
        {
        double          tx_x, tx_y, tx_z;
        double          rx_x, rx_y, rx_z;
        double          dif_x, dif_y, dif_z, dist_xy;
        double          rot1_x, rot1_y, rot1_z;
        double          rot2_x, rot2_y, rot2_z;
        double          rot3_x, rot3_y, rot3_z;
        double          cos_pt_th, sin_pt_th;
        double          cos_sw_th, sin_sw_th, cos_sw_ph, sin_sw_ph;
        double          rx_phi, rx_theta, point_phi, point_theta;
        double          bore_phi, bore_theta, lookup_phi, lookup_theta, gain;
        Vartype         pattern_table;
```

```
double          sweep_phi, sweep_theta;
double          proc_gain, rx_code, tx_code;
int             objid;

/** Compute the gain associated with the transmitter's antenna. **/
FIN (dra_tagain_DSSS (pkptr));

/* Obtain handle on receiving antenna's gain. */
pattern_table = op_td_get_ptr (pkptr, OPC_TDA_RA_TX_PATTERN);

/* Special case: By convention a nil table address indicates an isotropic  */
/* antenna pattern. Thus no calculations are necessary.
        */
if (pattern_table == OPC_NIL)
        {
        /* Assign zero dB gain regardless of transmission direction. */
        op_td_set_dbl (pkptr, OPC_TDA_RA_TX_GAIN, 0.0);
        FOUT;
        }

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

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

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

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


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

/* Setup a new coord. system where x axis is in same theta plane    */
/* as pointing direction. This allows simple computation of effect */
/* of phi rotation on the transmission vector.
*/
cos_pt_th = cos (DEG_TO_RAD * point_theta);
sin_pt_th = sin (DEG_TO_RAD * point_theta);
rot1_x = dif_x * cos_pt_th - dif_y * sin_pt_th;
rot1_y = dif_x * sin_pt_th + dif_y * cos_pt_th;
rot1_z = dif_z;

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

```
        rot2_x = (rot1_x * cos_sw_ph - rot1_z * sin_sw_ph) * cos_sw_th + rot1_y *
sin_sw_th;
        rot2_y = rot1_y * cos_sw_th - (rot1_x * cos_sw_ph - rot1_z * sin_sw_ph) *
sin_sw_th;
        rot2_z = rot1_x * sin_sw_ph + rot1_z * cos_sw_ph;

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

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

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

                if (rot3_y > 0.0)
                        rx_theta = -RAD_TO_DEG * acos (rot3_x / dist_xy);
                else
                        rx_theta = RAD_TO_DEG * acos (rot3_x / dist_xy);
                }


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

        /* Obtain gain of antenna pattern at given angles. */
        gain = op_tbl_pat_gain (pattern_table, lookup_phi, lookup_theta);

        /* Account for spread spectrum effects by 'spreading' power.  */
        objid = op_td_get_int (pkptr, OPC_TDA_RA_RX_OBJID);
        rx_code = op_td_get_dbl(pkptr, OPC_TDA_RA_RX_CODE);
        tx_code = op_td_get_dbl(pkptr, OPC_TDA_RA_TX_CODE);
        if ((tx_code > 0))
                {
                if (op_ima_obj_attr_exists(objid, "Processing Gain") == OPC_TRUE)
                        op_ima_obj_attr_get (objid, "Processing Gain", &proc_gain);
                else
                        proc_gain = 1.0;
                proc_gain = 10*log10(proc_gain);
                gain = gain - proc_gain;
                }

        /* Set the tx antenna gain in the packet's transmission data attribute. */
        op_td_set_dbl (pkptr, OPC_TDA_RA_TX_GAIN, gain);

        FOUT;
        }
```

Receiver Gain Stage

```
/* dra_ragain_DSSS.ps.c */


/****************************************/
/*              Copyright (c) 1994              */
/*                    by MIL 3, Inc.                        */
/*             (A Delaware Corporation)         */
/*     3400 International Drive,  N.W.          */
/*             Washington, D.C., U.S.A.         */
/*                    All Rights Reserved.       */
/****************************************/

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


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

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


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

void
dra_ragain_DSSS (pkptr)
        Packet*         pkptr;
        {
        double          tx_x, tx_y, tx_z;
        double          rx_x, rx_y, rx_z;
        double          dif_x, dif_y, dif_z, dist_xy;
        double          rot1_x, rot1_y, rot1_z;
        double          rot2_x, rot2_y, rot2_z;
        double          rot3_x, rot3_y, rot3_z;
        double          cos_pt_th, sin_pt_th;
        double          cos_sw_th, sin_sw_th, cos_sw_ph, sin_sw_ph;
        double          tx_phi, tx_theta, point_phi, point_theta;
        double          bore_phi, bore_theta, lookup_phi, lookup_theta, gain;
        Vartype         pattern_table;
        double          sweep_phi, sweep_theta;
        double          proc_gain, tx_code, rx_code;
        int             objid_rx;

        /** Compute the gain associated with the receiver's antenna. **/
        FIN (dra_ragain_DSSS (pkptr));

        /* Obtain handle on receiving antenna's gain. */
        pattern_table = op_td_get_ptr (pkptr, OPC_TDA_RA_RX_PATTERN);

        /* Special case: by convention a nil table address indicates an isotropic  */
        /* antenna pattern. Thus no calculations are necessary.
                */
        if (pattern_table == OPC_NIL)
                {
                /* Assign zero dB gain regardless of transmission direction. */
                op_td_set_dbl (pkptr, OPC_TDA_RA_RX_GAIN, 0.0);
                FOUT;
                }


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

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

```
        rx_z = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_GEO_Z);

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


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


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

        /* Setup a new coord system where x axis is in same theta plane      */
        /* as pointing direction. This allows simple computation of          */
        /* effect of phi rotation on the transmission vector.                        */
        cos_pt_th = cos (DEG_TO_RAD * point_theta);
        sin_pt_th = sin (DEG_TO_RAD * point_theta);
        rot1_x = dif_x * cos_pt_th - dif_y * sin_pt_th;
        rot1_y = dif_x * sin_pt_th + dif_y * cos_pt_th;
        rot1_z = dif_z;

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

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

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

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

```
                tx_phi = RAD_TO_DEG * atan (rot3_z / dist_xy);

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

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

        /* Obtain gain of antenna pattern at given angles. */
        gain = op_tbl_pat_gain (pattern_table, lookup_phi, lookup_theta);

        /* Account for spread spectrum effects by 'spreading' power.  */
        objid_rx = op_td_get_int (pkptr, OPC_TDA_RA_RX_OBJID);
        tx_code = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_CODE);
        rx_code = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_CODE);
        if (tx_code == rx_code)
                {
                op_ima_obj_attr_get (objid_rx, "Processing Gain", &proc_gain);
                proc_gain = 10 * log10 (proc_gain);
                gain = gain+proc_gain;
                }

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

        FOUT;
        }
```

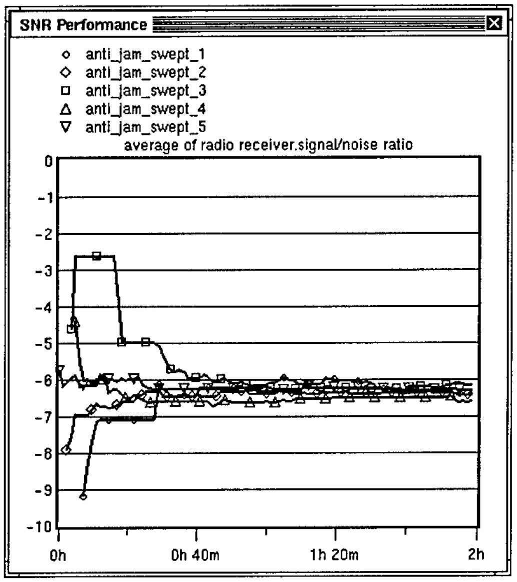
SNR Stage

```
/* dra_snr_DSSS.ps.c */
/* Default Signal-to-Noise-Ratio (SNR) model for radio link Transceiver Pipeline */

/****************************************/
/*              Copyright (c) 1993              */
/*                   by MIL 3, Inc.                     */
/*             (A Delaware Corporation)         */
/*      3400 International Drive,  N.W.         */
/*             Washington, D.C., U.S.A.         */
/*                   All Rights Reserved.       */
/****************************************/

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


void
dra_snr_DSSS (pkptr)
        Packet*         pkptr;
        {
        double          bkg_noise, accum_noise, rcvd_power;
        double          proc_gain, cross_corr;
        double          tx_code, rx_code;
        int             objid_tx, objid_rx, tx_fam, rx_fam;

        /** Compute the signal-to-noise ratio for the given packet. **/
        FIN (dra_snr_DSSS (pkptr));

        /* Get the packet's received power level. */
```

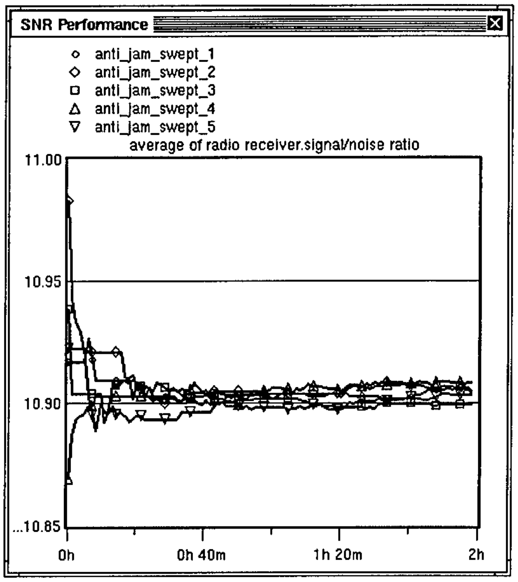
```
rcvd_power = op_td_get_dbl (pkptr, OPC_TDA_RA_RCVD_POWER);

/* Account for cross correlation, if necessary. */
objid_tx = op_td_get_int(pkptr, OPC_TDA_RA_TX_OBJID);
objid_rx = op_td_get_int(pkptr, OPC_TDA_RA_RX_OBJID);
op_ima_obj_attr_get (objid_rx, "Family", &rx_fam);
op_ima_obj_attr_get (objid_tx, "Family", &tx_fam);
rx_code = op_td_get_dbl (pkptr, OPC_TDA_RA_RX_CODE);
tx_code = op_td_get_dbl (pkptr, OPC_TDA_RA_TX_CODE);

/* Get the packet's accumulated noise levels calculated by the */
/* interference and background noise stages. */
accum_noise = op_td_get_dbl (pkptr, OPC_TDA_RA_NOISE_ACCUM);
bkg_noise = op_td_get_dbl (pkptr, OPC_TDA_RA_BKGNOISE);
if ((tx_fam == rx_fam) && (tx_code != rx_code))
        {
        op_ima_obj_attr_get(objid_rx, "Cross Correlation", &cross_corr);
        rcvd_power = rcvd_power * cross_corr;
        }

/* Obtain the processing gain from the model attributes. */
op_ima_obj_attr_get (objid_rx, "Processing Gain", &proc_gain);

/* Account for Spread Spectrum through Processing Gain. */
/* Assign the SNR in dB. */
op_td_set_dbl (pkptr, OPC_TDA_RA_SNR,
        10.0 * log10 (rcvd_power * proc_gain / ((accum_noise + bkg_noise))));

/* Set field indicating the time at which SNR was calculated. */
op_td_set_dbl (pkptr, OPC_TDA_RA_SNR_CALC_TIME, op_sim_time ());

FOUT;
}
```

APPENDIX B

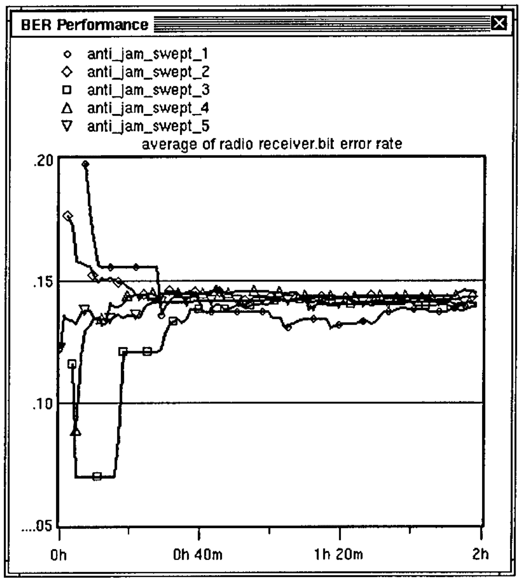
SNR AND BER PLOTS FOR SWEPT BAND JAMMER CASE



SNR Normal Receiver



SNR DSSS Receiver



BER Normal Receiver

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* xx-03-2000	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From - To)* Jun 1999 - Mar 2000; Mar 2000

4. TITLE AND SUBTITLE

Modeling and Simulation of Communication Systems in OPNET

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

David M. Banker, 2nd Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Wright - Patterson AFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/00M-03

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

Kenneth Fore
HQ AFCA/ITAT
203 West Losey Street
Scott AFB, IL 62225-5222

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION LIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research aims to present accurate computer models of a communication link and a Super High Frequency (SHF) radio communication system. Network Warfare Simulation (NETWARS) is a J-6 initiative aimed at modeling all communication traffic in the Department of Defense (DoD) for testing and analysis of specific real world scenarios. The AN/TSC-94 is a SHF radio system with satellite communication capabilities. The AN/TSC-94 incorporates a Direct Sequence Spread Spectrum (DSSS) radio link for certain Anti-Jam (AJ) features. A DSSS 'spreads' signal power over a large bandwidth, reducing power previously concentrated within the original system bandwidth. The simulations were performed using OPNET. Simulation results show DSSS lowered Bit Error Rate (BER) over links not using spread spectrum. Results show that in the presence of multiple jamming forms, the DSSS link performed without bit errors while the normal (non-DSSS) link was disrupted by the jammer, experiencing BER's of up to 0.43. The AN/TSC-94 was able to defeat the jammer using the DSSS link. By performing in normal mode during unjammed scenarios, and switching to AJ mode in the presence of a hostile transmitter, the AN/TSC-94 demonstrated its ability to successfully communicate in multiple access and hostile environments.

15. SUBJECT TERMS

Direct Sequence Spread Spectrum, NETWARS, Super High Frequency Radio

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Michael A. Temple
Unclassified	Unclassified	Unclassified	UL	64	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255 - 3636 x4703