TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/IN/JTRP-2003/1		

4. Title and Subtitle	5. Report Date
Evaluation Procedures for Deploying Spread Spectrum Interconnect	November 2003
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
J. V. Krogmeier	FHWA/IN/JTRP-2003/1

9. Performing Organization Name and Address	10. Work Unit No.
Joint Transportation Research Program 1284 Civil Engineering Building Purdue University West Lafayette, IN 47907-1284	
	11. Contract or Grant No. SPR-2395

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Indiana Department of Transportation State Office Building 100 North Senate Avenue Indianapolis, IN 46204	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the Indiana Department of Transportation and Federal Highway Administration.

16. Abstract

This purpose of this project was the design and construction of a testbed network for experimentation with spread spectrum communications in the 900 and 2400 MHz band. The current network consists of five fixed nodes and two portable nodes. Of the fixed nodes, three are located in the Purdue MSEE building, one is located in the Harold L. Michael Traffic Operations Laboratory, and one is located in the experimental traffic signal cabinet at the intersection of Stadium and Northwestern Avenues in West Lafayette. The testbed has been used to evaluate spread spectrum radio technologies from vendors Microwave Data Systems, GINA, and EnCom, as regards radio performance as a function of link distance and with varying levels of interference. The project has produced software that can be used to test any radio presenting a standard RS-232 interface to customer equipment. The testbed can also be interfaced with wireless channel emulator equipment located in the Wireless Communications Research Laboratory in order to test radio performance in multipath and fading environments.

The field tests conducted to date have yielded the following conclusions regarding the design of spread spectrum interconnect for the traffic signal control application. First, as regards the need for line-of-sight between radio antennas, we were able to establish a link even when line of sight was not available although the resulting communication was somewhat error prone. If an additional robust protocol were designed these radios could work even in non-line-of-sight applications. The protocol would need to emphasize an efficient retransmission scheme as opposed to the use of more powerful forward error control codes (this observation comes from the link error statistics gathered from field testing). Second, it was noted that the RSSI (which stands for received signal strength indicator) value was not a completely informative measure of link quality. In other words, the link testing software found many instances of links where RSSI was relatively large yet many errors were incurred. Such situations always involved interference or suspected interference. Third, as regards the choice of antenna (Yagi versus omni) it is concluded that Yagi antennas (with their higher gain) were preferable in long distance rural environments. But in the presence of multipath propagation or interference (as in a town or city scenario) the omni was often preferable. Finally, interference tests showed that the MDS 9810 handled the interference well at the price of the throughput; the EnCom radio had a high throughput but also a high data loss rate in certain situations. It was concluded that if the application only needs between 6 and 14 kbps, the MDS radio is an ideal choice. For higher rate applications the EnCom radio would be preferred.

17. Key Words	18. Distribution Statement
Wireless communications, spread spectrum, traffic signal control.	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	58	

Form DOT F 1700.7 (8-69)

Final Report

FHWA/IN/JTRP-2003/1

Evaluation Procedures for Deploying Spread Spectrum Interconnect

By

James V. Krogmeier
Associate Professor
School of Electrical and Computer Engineering
Purdue University

Joint Transportation Research Program
Project No. C-36-75N
File No. 8-9-14

In Cooperation with the
Indiana Department of Transportation
and the
U.S. Department of Transportation
Federal Highway Administration

Purdue University
West Lafayette, Indiana 47907
November, 2003

Table of Contents

List of Figures

List of Tables

1. Introduction

Traffic management systems under design and deployment by the Indiana Department of Transportation (INDOT) are more and more dependent upon advanced computer and communication technologies. Projects of this sort are found all over the state and include the Advanced Traffic Management System (ATMS) on the Borman Expressway and coordinated traffic signal systems in Indianapolis, Lafayette, and other cities. In the majority of traffic system deployments, communications needs are handled by dedicated cable (copper or fiber optic) buried along the roadside. Such communication systems work very well and can provide enormous bandwidth for data, voice, and video traffic. There are several downsides to dedicated cable, however. First, is the relatively large cost associated with installation. Second, dedicated cable cannot serve in situations requiring mobility or even portability as might occur with temporary communication needs associated with road maintenance, construction, or special traffic impacting events. Finally, utility work regularly severs roadside communication lines resulting in a loss of "interconnect communication." For a coordinated traffic signal system, this may result in severely disrupted traffic flow. For a traffic surveillance system, disrupted communication systems will severely impact traveller information and incident response systems. In some cases it is several days (or weeks) before the problem is identified. By this time, the contractor doing the work is no longer around and it is nearly impossible to assess blame and recover damages. Due to issues with scheduling and cost it may be several months (or longer) before the interconnect line is repaired.

Wireless communication is an attractive alternative to hard interconnect. Advantages include relatively low cost, mobility, and portability. Disadvantages include a limited bandwidth (compared to fiber optic lines), the potential for interference in frequency bands not requiring a license, and the cost and time involved in regulatory issues in frequency bands requiring a license. In fact, licensed frequencies are very difficult to obtain in most urban areas which is a driving factor behind the emergence of unlicensed spread spectrum radio (primarily in the 900 MHz and 2.4 GHz bands). The possibility of interference is an issue although it is partially mitigated by the use of spread spectrum technology which is expressly designed for robustness to interference. In part because of their wide use around the country and the resulting economies of scale, spread spectrum radios are a very cost effective technology; about $2000 per intersection for implementing, for example, a 19.2 kbps traffic signal interconnect. In contrast, copper interconnect costs about $8000 per mile and fiber optic interconnect costs about $20000 per mile.

However, Indiana has a somewhat limited experience in implementing spread spectrum radio systems. A high bandwidth system has been implemented on the Borman Expressway as a backbone communication system and a low bandwidth traffic signal coordination system has been deployed in Crawfordsville. For more information see [1].

Low bandwidth communication needs as in sensor telemetry transmission and control (of e.g., highway advisory radios, roadside signs, and cameras) have been studied in the context of licensed communications using FHWA's dedicated bands at 220 MHz (this is not spread spectrum) [2, 3]. Unfortunately, the 220 MHz bands have not been widely adopted and very little equipment is available that is applicable to transportation. It should be noted, however, that the 220 MHz band is actively used by non-ITS applications, e.g., as paging networks.

2. Problem Statement and Objectives

Wireless communications will occupy an increasingly important place in the portfolio of communication systems that INDOT will design, deploy, and manage in the future. Spread spectrum communications operating in unlicensed bands will probably be the most important technology in the mix of wireless possibilities. This project is intended to enhance INDOT's ability to effectively use the technology. There are a variety of capabilities that will be required including: 1) the ability to do preliminary field inspection, 2) the ability to perform detailed communication system tests, 3) the ability to effectively perform system integrations involving such communication systems, and 4) the ability to effectively work with individual vendors of spread spectrum communication systems. The main objective of this project was to develop a testbed for testing and integration of spread spectrum communication systems and equipment for traffic surveillance telemetry and traffic signal control. The existence of the testbed will also serve in the development of deployment guidelines for spread spectrum interconnect and in training of INDOT personnel in spread spectrum communications.

3. Work Plan

The work plan for this project consisted of six separate tasks, which are briefly described below. In addition, an expansion of scope and time extension was granted to allow the project to test additional spread spectrum radios.

1. Task 1: Literature Review. A literature review concentrating on spread spectrum communication systems was performed, with particular attention paid to the propagation and expected interference environments at 900 MHz and 2.4 GHz (the applicable bands for unlicensed spread spectrum). In addition, some literature on the relative merits of frequency hopping vs. direct sequence spreading was reviewed although most of the available equipment is frequency hopped.
2. Task 2: Develop Radio Specifications. This task was intended to identify the specifications needed for the spread spectrum radios considered for use in the

radio interconnect testbed to be developed in Task 3. All of the usual radio communication specification areas were considered including: 1) unit power consumption in active and standby mode, 2) radio frequency transmit power, 3) line-of-sight range, 4) resistance to interference (i.e., processing gain), 5) supported data rates, 6) number of radios supported in network configuration, 7) flexibility of modes of operation, and 8) ease of integration with INDOT approved traffic surveillance and signal control equipment. In the process of developing the radio specifications, spread spectrum radio products from the following manufacturers were examined.

a. **Encom Radio Services:** a frequency hopped system with software support for several different local area network configurations.

b. **FreeWave Technologies:** a frequency hopped system in the 902-928 MHz band. They claim the radio is capable of an uncompressed data rate of 115.2 kbps over 20 miles or more. Up to 1 W transmitter power. It allows only two modes of operation: point-to-point or point-to-multipoint.

c. **Microwave Data Systems**: a frequency hopped system supporting data rates up to 19.2 kbps. Operation in either the 902-928 MHz or 2.4 GHz bands are supported (with different products). The units have flexible network operation and low power operating modes.

3. Task 3: Design Radio Interconnect Testbed. A major portion of the work of this project was the design and deployment of a radio interconnect testbed. The testbed consists of four nodes. The primary node (i.e., the master in master/slave situations) is located in the Wireless Communications Research Laboratory (WCRLab) in the Materials Science and Electrical Engineering Building on the Purdue Campus. Secondary nodes are located in Prof. Bullock's Transportation Systems Laboratory in the Civil Engineering Building and in a roadside cabinet at the experimental facility at the corner of Northwestern Avenue and State Street in West Lafayette, IN. In addition, two portable nodes were built. The testbed supports:

   a. The evaluation of competing manufacturers radio technologies via their integration into a common (test) network including INDOT approved traffic surveillance and control equipment.

   b. Operation in multiple network modes, e.g., point-to-point, point-to-multipoint (with master/slave operation), and peer-to-peer.

4. Task 4: Conduct Radio Interconnect Tests. The testbed was used in various radio interconnect tests. We examined the satisfaction of various levels of quality of service provided by the testbed network including bit error rates, packet error rates, the performance of retransmission protocols and their effect on network loading. Operation of the network will be observed over long periods of time and in a variety of weather conditions (which are interesting only in so much as they affect the transmission environment).

5. Task 5: Creation of List of Web Links to Radio Tutorial Materials. The project also created a web page consisting of links to useful tutorial material on spread spectrum and related topics.

4. Analysis of Data

This section contains the main results of the project. It is divided into three subsections. The first gives some background information on the two main technologies for implementing spread spectrum radio. The second describes the radio link testing software that was written in order to operate the testbed and extract performance information from the tested radio network. The third and final section contains the results of extensive testing of spread spectrum radios.

4.1 Comparison of Spread Spectrum Technologies and Radios

4.1.1 Comparison of Frequency-Hopping and Direct-Sequence

There are two major techniques used to implement spread spectrum communications, which itself is defined as a communication system using more than the minimum bandwidth required for the specified information rate. Spreading gives the transmission an advantage over interference (both intentional and unintentional) at the cost of bandwidth.

Frequency-hopped spread spectrum (FHSS) uses a conventional narrow band transmission but with a transmission center frequency that changes (or hops) over the course of the transmission. The hopping pattern is the sequence of different center frequencies that the transmission moves over; it is known to the desired receiver and generally unknown to other receivers. The frequency changing can be very fast. Transceivers usually stay less than 400ms[1] at a particular center frequency. That's why it's called frequency "hopping".

Direct sequence spread spectrum (DSSS) takes a narrow band signal and spread it over a much wider band of frequencies by multiplying the original signal by a higher rate sequence called the spreading code.

In either case (FH or DS) the spreading gain is defined to be the ratio (sometimes expressed in decibels) of the transmission bandwidth to the information bandwidth. The advantages of spread spectrum in terms of interference resistance come in proportion to the spreading gain. A typical comparison of a data signal before and after (direct sequence) spreading is shown in Figure 1. The situation for a frequency hopped system is shown in Figure 2.

[1] Per FCC regulation.

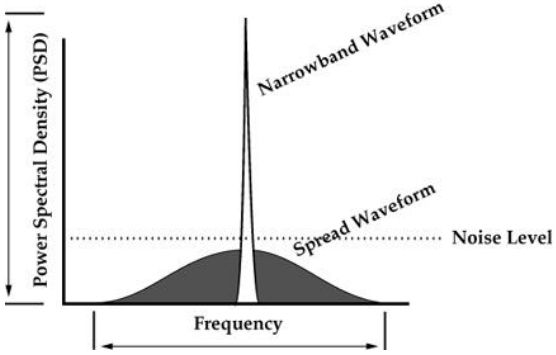

Figure 1: Power density spectrum of direct sequence spread spectrum.

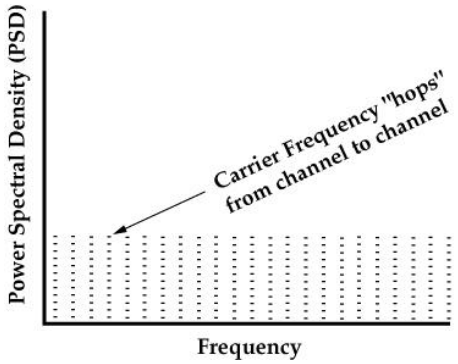

Figure 2: Power density spectrum of frequency hopped spread spectrum.

Theoretically, it is not obvious which technology is best. For our purpose of radio selection, we try to list and explain their differences as below. The comparisons are rather between available practical systems than between two spread spectrum technologies.[2]

- Cost. DSSS is driven by high rate digital multiplication, therefore it has a heavy requirement for large, expensive digital circuitry. Also a more expensive power amplifier is required in DSSS transmitter. This may be the main reason that FHSS is currently the dominant technology used in wireless LAN products.
- Power consumption. It has a direct impact on the battery life for mobile users. FHSS vendors claim their systems are less energy consuming.
- Tolerance to interference. Theoretically, it's hard to say which one is better. Practically, FHSS systems usually hop across the whole spectrum, while DSSS systems usually divide the spectrum into a couple of sub-bands, and then spread the original signal to one of these sub-bands. This means DSSS systems do not spread as wide as FHSS systems, i.e., the latter have higher spreading gains. Therefore,

[2] For example, it is unfair to compare two technologies if the systems we look at have different spreading gains.

practical FHSS systems often show much better interference rejection ability than DSSS systems. In fact, Freewave manufacturer claims that if their product is placed in the same room as DSSS product, the latter will be blocked.

- Throughput. DSSS systems usually claim more capacity per channel. But FHSS systems have more channels (hopping patterns) for a given bandwidth. Combining the two effects together, FHSS systems claim more total capacity.
- Security. FHSS systems are generally more secure, due to pseudo-randomly changing transmission frequency. Using encryption techniques in DSSS systems can minimize this advantage.
- Coverage Area. DSSS systems usually use coherent modulation schemes, while FHSS systems usually employ non-coherent modulation schemes. This difference gives DSSS system 3dB power advantage. For the same transmitted power, DSSS systems claim more coverage.

4.1.2 Comparison of Spread Spectrum Radio Products

In this section, we compare three commercial, off-the-shelf spread spectrum data transceivers, namely, MDS 9810 (MDS), Freewave DGR115 series (Freewave), and GINA 6000N-5 series (GINA). They all operate in the 902-928 MHz ISM band. The comparison is based on the technical documents provided by the product manufactures, including product manuals, and application notes.

1. Spread spectrum technology

GINA radios are direct sequence spread spectrum (DSSS) radios. Freewave and MDS radios employ frequency hopping spread spectrum (FHSS) technology.

2. Radio specifications

i. Baud rate

GINA and MDS support baud rates of 1.2 – 38.4 kbps, while Freewave supports baud rates of 1.2 – 115.2 kbps.

ii. Nominal range

The coverage of an established radio link depends on a number of factors, including the antenna gains, the feedline losses, the height of transmitting and receiving antennas, existence of line-of-sight, etc. The nominal range is the expected range if the propagation environment is close to ideal. The number is given by the manufactures and represents somewhat the best case scenario. It

should be stressed that careful site planning is crucial for any successful deployment.

The Freewave manufacturer claims that, with a good antenna system, the radio can cover up to 20 miles. The GINA manufacturer claims 12 miles, while MDS claims 10–15 miles. The advantage of Freewave radios comes from its coherent modulation techniques.

iii. Number of Channels/hopping patterns

GINA divides the 900 MHz ISM band into 21 channels, with neighboring channels overlaping each other. The user can configure the radio to operate within one of these channels. Any other system transmitting in the same frequency band will interfere with the GINA radios.

For FHSS systems, channel should be defined as the product of hopping pattern and time. For example, two systems using exactly the same hopping pattern don't completely block each other as long as they are not synchronized to each other. Therefore, the implication of total number of hopping patterns is not clear. We are listing them below with the note that more patterns are not necessarily better.

Freewave has 15 hopping patterns with the number of frequencies configurable from 50 to 112. MDS has 65000 different hopping patterns. The MDS radio can be configured to skip frequency zones to avoid constant interference.

iv. Power Consumption

GINA is clearly the most power-consuming one with Tx/Rx power consumption of 750/400 mA. MDS is the most power efficient radio with 400/125 mA. It also has an idle mode with a current draw of 30 mA. The MDS is the best choice in power-sensitive applications, such as applications with remotes powered by solar source. Freewave consumes 650/100/50 mA when Tx/Rx/idle.

v. Adjustable RF power

Freewave and MDS radio offers adjustable RF power feature while GINA radio does not.

3. Configuration flexibility

i. Operation modes

All of them support point–point (P/P) and point–multipoint (P/MP) operations. GINA and MDS also support peer–peer (Pr/Pr) operation.

ii. Configuration with repeater

All three radios support the use of repeater in P/P and P/MP modes. However, in P/MP mode, GINA radio requires all traffic to go through the repeater, i.e., every slave has to communicate with the master via the repeater. Freewave and MDS also support the use of single radio repeater with store and retransmit technique (Such configuration will have an impact on the maximum throughput of the system.). Moreover, a single Freewave radio can be used as slave and repeater simultaneously in P/MP mode, therefore saving the total number of required radios.

4. Security

It'll be not wise to believe any one of these three radios is secure and impossible to penetrate. However, they do offer some security features to avoid security attacks and unintended interference by nearby systems made of same brand of radios.

GINA offers has very limited security. A GINA receiver has a two-digit configurable RID (receiver ID). The receiver will listen to any transmitter in the same frequency band with the same number configured as TXP (transmission path). It's very easy to intercept and interfere with GINA's operation.

Every Freewave radio has a unique factory-preset 7-digit serial number. With the call book feature, a Freewave radio only listens to those radios whose serial number is in its call book list. This makes it difficult for the perpetrator to eavesdrop or to interfere with other sets of Freewave radios.

Every MDS radio network has a configurable 4-digit network ID. Every radio in the network has to have the same network ID. If a perpetrator wants to eavesdrop or interfere with the network with his/her own MDS radio, he/she has to somehow set the interfering network ID the same as the interfered network.

5. Performance analysis and management tool

The Freewave and MDS radios manufactures offer additional radio management software for collection of link statistics, network diagnosis, and radio configuration. The GINA radio manufacture does not offer diagnostic/management software.

The comparison among the radios is tabulated in Table 1 on the next page. Radios with clear advantage over the other candidates are described with bold font entries in Table 1.

Table 1 : Comparison of three Spread Spectrum Radio Candidates

Feature		GINA6000N	FREEWAVE DGR-115	MDS9810
Spread Spectrum Technology		DSSS	FHSS	FHSS
Radio Specs	Baud Rate(bps)	1.2-38.4k	**1.2-115.2k**	1.2-19.2k
	Range	12miles	**20miles**	10-15miles
	# of channels/hopping patterns	21	15	65000
	Adjustable RF power	No	**Yes**	**Yes**
	Power consumption(mA)	750/400	650/100/50	**400/125/30**
Flexibility	Operating modes	**Point-point, point-multipoint, peer-peer**	Point-point, Point-multipoint,	**Point-point, Point-multipoint, Peer-peer**
	Configuration with repeater	Not flexible	**Most flexible**	Flexible
	Controlled communication path	N/A	**Yes, via call book feature or subnet ID.**	Yes, but not as flexible as freewave
Performance Analysis	Adjustable RF power	No	**Yes**	**Yes**
	Link statistics	Not available	**Available**	**Available**
	Management software	Not available	**Available**	**Available**
Security		Low	**High**	**High**

4.2 The Radio Link Testing Software

4.2.1 Introduction

I. Program Overview

The Radio Link Testing Software (RLTS) was developed in the Wireless Communications Research Laboratory (WCRLab) of Purdue University as part of the spread spectrum interconnect project sponsored by the Indiana Department of Transportation (INDOT). INDOT is interested in using radio links in place of expensive copper cables to control the timings of traffic signals, and to gather data from traffic equipment, etc.. The purpose of developing the RLTS program is to provide a tool for field testing commercial, off-the-shelf radios. The results of field tests help the radio users to understand the coverage, line-of-sight, sensitivity to noise, and other issues of the wireless network. More importantly, the knowledge gained through field tests provides precious information as how to extend the coverage of the network in order to use these low-cost radio links in situations where they cannot be used right now.

Although it was initially designed to measure the quality of wireless links established by MDS 9810 spread spectrum radios, the RLTS program should work well with any data communication radios with RS-232 interfaces to user equipment provided that proper adjustments are made to some program parameters. More discussion can be found in the later sections.

II. System Diagram and Error Model

In most point-to-point communication systems, there are two data links. One link goes from user A to user B, while the other goes from user B to user A. At any give moment, only one of the two links is active. The quality of these two links can be roughly the same, or very different, depending on whether they use the same frequency band, the same modulation technique, etc. The RLTS program measures link quality one way at a time. One radio is connected to the data source, which is a computer that sends out data. The other radio is connected to the data sink, which is a computer that receives data. Upon completion of testing, the program provides measurements for the link from the source to the sink. These measurements include bit error rate (BER), data loss rate (DLR), distribution of the lengths of lost data segments, etc. A flow chart of this program is provided in Appendix A.

Figure 3 provides a diagram of the system. Data originates at the data source, then travels through the air link before reaching the data sink. The air link consists of the transmitter radio, the receiver radio, and the physical wireless channel between them. The wireless channel is inherently very hostile. The function of the radios is to work with this hostile channel and try to provide an error-free communication link to the user equipment. Unfortunately, the air link can never be truly error free. How good the link is, i.e., how close it is to error-free, is dependent on the propagation environment. For example, the distance between the transceivers, the existence of line-of-sight (LOS), and

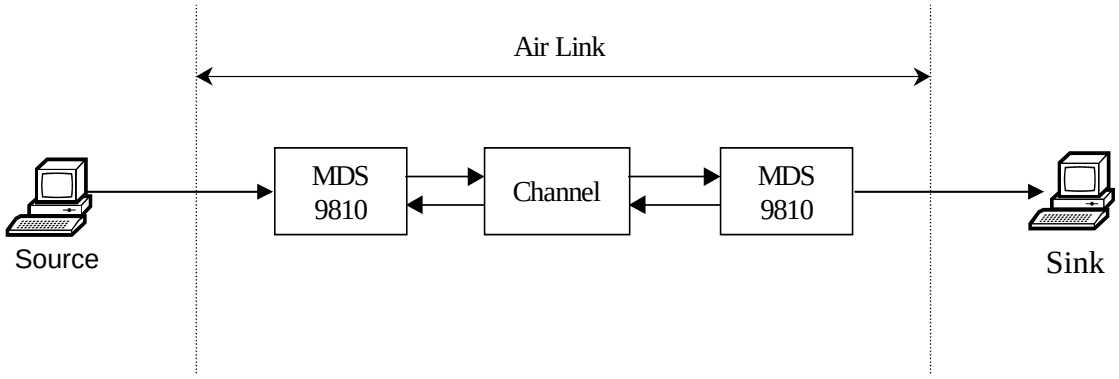

Figure 3: System Diagram

the type of antenna used all have an impact on the quality of the link. The job of the RLTS program is to help us to understand the dependence of air link quality on different environment parameters in order to find possible ways to improve it.

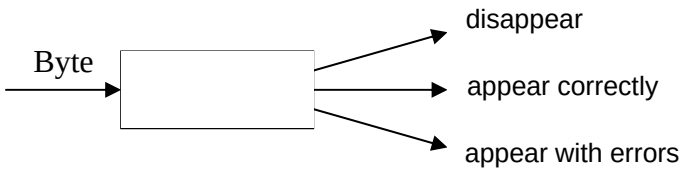

Figure 4: Error Model

Figure 4 depicts our error model. This model is a byte-oriented model. When a byte is sent through the air link one of three possibilities must occur; it either disappears, or appears correctly, or appears with errors. Although the underlying wireless channel is a bit-oriented channel, the RS-232 interface between the radio and user equipment guarantees that the air link seen by the user is byte-oriented. This model can apply to almost all data communication radios with RS-232 interfaces to user equipment. Therefore the algorithm developed from this model should work with all radios with RS232 interfaces.

Most of the commercial data communication radios implement proprietary sets of air interface specifications and protocols. For example, the hopping patterns of FHSS radios are usually unknown to the users, nor do the users know the forward error control (FEC) codes or the retransmission protocols. By including the radios in the air link and

using the simple byte-oriented model, we free the program from the impossible mission of guessing the specifications and protocols used by the radios.

III. Program Initiation

To initiate the program, the user has to make a few inputs. First, he/she must choose which serial port of the computer is to be used and the data rate for the test. Then the user has to specify whether the computer is the data source or the data sink. If the computer is specified as data source, then no other information is needed and the program is properly initiated. If the computer is specified as the data sink, a few more inputs are required. They are the name of the output file and length of the test (either by time or by number of bytes processed). Screen shots of these input prompts can be found in appendix B.

IV. Operation at the Data Source

The operation at the data source is simple and straightforward. The program waits for a special phrase to be received from the data sink as the request for data. Then the program sends a pseudo-random sequence 60,000 bytes at a time through the air link. After that the program waits for another request to start the next session of 60,000 bytes. The buffer size at the data sink is larger than 60,000 bytes, which guarantees the sink buffer never overflows. A new session starts from the exact place in the pseudo-random sequence that the previous session ended. The pseudo-random sequence is the sequence of two-byte words from 0x0000 to 0xFFFF with increment 1. The period of the sequence is 131072 bytes. This period should be greater than the maximum length of a missed data segment. Otherwise, if a segment longer than the period is lost, the program cannot count its length correctly. An extreme example is when the period of the sequence is 1, i.e., the transmitter keeps sending the same byte again and again. If this is the case, the program loses all its ability to recognize a data loss. In our tests, the maximum data rate for the MDS9810 is 38.4kbps. Hence it'll take at least $131072 \times 8 \div 38400 = 27.3$ seconds of communication outage to miss a whole period of sequence. 27.3 seconds are much longer than the channel coherence time. Our conclusion therefore is that the sequence is longer than the maximum length of missed data segment.

4.2.2 Operation at the Data Sink

The data sink is where most of the processing takes place. In the following, we discuss its different functions separately.

I. Flow Control

There are two purposes for flow control in the RLTS program. It makes sure the buffer at the data sink doesn't overflow and it is also responsible to ensure the two radios don't transmit at the same time, which causes collision and data loss.

As mentioned earlier the data source waits for a request from the data sink to begin a session of 60,000 bytes. The data sink draws bytes from its buffer, which is filled in by the radio via the RS-232 interface. If the buffer is empty, the sink keeps monitoring it till either new bytes enter the buffer or 180 ms has elapsed and the buffer stays empty. If the latter is the case, the data sink decides that the current session had ended, and a request is sent to the data source for the next session. If the request is lost or corrupted such that the data source doesn't start a new session, another request will be sent after another 180 ms.

The choice of the time out parameter is a tradeoff. If it is too small, the data sink may decides the current session is over when it really is not. Then the request collides with the oncoming data stream and causes data loss. This data loss would be counted towards link statistics even though it was due to the imperfect protocol. In our tests, the hop length is 80 ms. It would take at least two consecutive lost hops to cause the collision. From our test results, we have found that the frequency of this occurrence is very low. On the other hand if the time out parameter is too large, the data sink waits too long before requesting a new session decreasing the throughput. In our case, a session takes at least 12.5 seconds. Simple analysis shows that the choice of 180 ms time out parameter causes a throughput decrease of less than 1.5%. We recommend choosing the time out parameter no longer that 10% of the duration of a typical session.

The buffer size in the data sink is 80,000 bytes, which is more than the length of a session. Therefore the buffer never overflows unless there exist unrequested sessions. This can be the case if there is interference and somehow it was decoded by the data source to be the request phrase. The data source then starts the next session before the receiver requests it. In addition, the receiver buffer has to have at least 20,000 bytes (80,000 – 60,000) in it when the mistaken new session starts for the buffer to overflow. Basically, it takes a very unusual interference and a very slow data sink for the buffer to over flow. In our tests, the buffer seldom exceeded the 5,000 bytes mark. This is the result of the combination of high-power CPU and low-speed RS-232 interface. If the user wants to feel more secure, two parameters can be changed. One is the request phrase. A longer phrase makes it less likely for the interference to be mistaken as the request. The user can also increase the buffer size by modifying the constant value defined by #define MAX_BUFF.

The Xon/Xoff flow control option of the radio is an alternative for avoiding data collision. With this option, the radios coordinate their transmission such that they don't talk at the same time. However, it requires the radios being set at time division duplex mode, which cuts the throughput in half. On the other hand, programs that rely on this option to avoid data collision won't work with radios without this option.

II. Frame Synchronization

Frame synchronization occurs right after the program is properly initiated at both computers, as well as right after a reset takes place. Reset is basically a restart of communication by the program and will be discussed in a later section. Through frame synchronization the data sink acquires the phase of the pseudo-random sequence. This

phase information is required by the error checking process that starts right after frame synchronization.

During frame synchronization, the data sink checks the consistence of the received sequence. A sequence is consistent if every word (except the first one) is equal to the previous word plus 1. Frame synchronization is accomplished when the data sink locates a consistent segment of certain length. The phase of this segment is considered as the current phase of the pseudo-random sequence. In our tests, the length is set to be 8 words (16 bytes). Frame synchronization is erroneous if the acquired phase is not the phase of the last word of the segment. Two kinds of received sequence error can cause frame synchronization errors. They are pure bit errors, and combination of bit errors and data loss. For obvious reason, pure data loss doesn't cause frame synchronization failure. For pure bit errors, there must be at least one bit error in each word to cause the failure. The probability of this happens is P_{eW}^{N}, where P_{eW} is the bit error probability, N is the number of words. In our tests, the toughest environment gives a bit error probability of 5.3e-3, which corresponds to $P_{eW} = 0.081$. For this case, the probability of pure bit errors causing frame synchronization failure is less than 1.9×10^{-9}, which is an extremely small number. The actual probability is likely to be much smaller, since it takes a weird error pattern for the resulting sequence to be consistent. We recommend that the required consistent segment length not to exceed 8 words generally. User can change this setting by modifying the value of Reset_Length, which is in bytes. This number should be even because every word consists of two bytes. For the case of combination of bit errors and data loss causing frame synchronization error, let's visualize that a data segment just before the last word is lost. If this happens, as few as one bit error in the last word can make the corrupted sequence look consistent. The probability of synchronization error caused by the combination of bit error and data loss is difficult to compute and may not be negligible. However, this kind of frame synchronization error is benign. If the acquired phase is equal to the phase of the sequence before the data loss, the data loss in the frame synchronization segment will be detected in the following error checking process and included in the link statistics. If the acquired phase is equal to the phase after the data loss, the frame synchronization is successful. If neither are the case, in the following error checking process the reset mechanism will restart the frame synchronization.

III. Error Checking

The error checking process is the core of the RLTS program. It starts right after the frame synchronization. The process can be divided into three steps. When inconsistency is located in the received sequence, the data link first creates a list of hypotheses (hypothesis formation). Then it decides which hypothesis is the most probable one according to certain criteria (hypothesis testing). Finally, the bit errors and data losses are counted and link statistics are updated (statistics update).

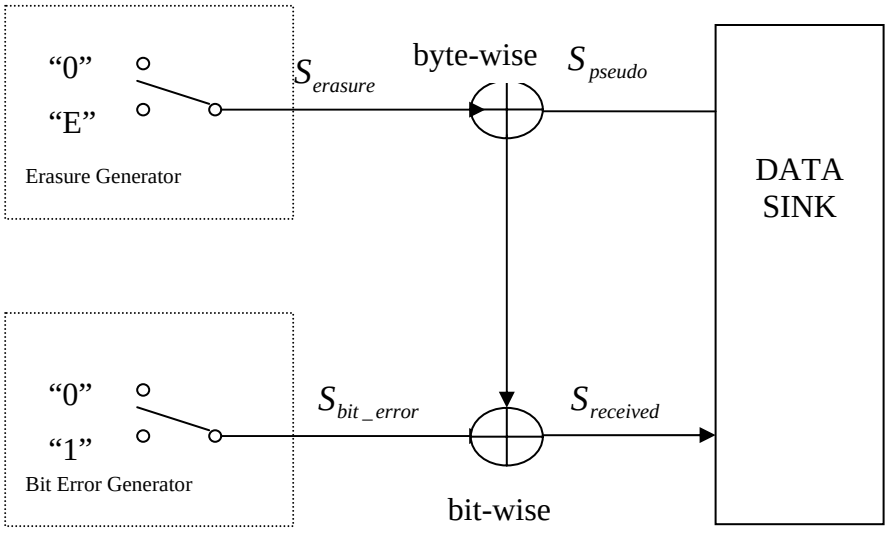

Figure 5: Error Checking Problem

To understand the process, we abstract the error checking problem in Figure 5. The air link errors are represented by two error sources. One of them is the erasure source, which models the data loss in the air link. It's a random sequence generator that generates a sequence with alphabet '0' and 'E'. Each element in the sequence corresponds to a byte. When a byte is added with '0', it is unchanged. When it is added with 'E', it becomes the null byte, which means it is lost in the link and disappears from the received sequence. The other error source is the bit error generator, which produces a bit error sequence out of alphabet '0' and '1'. The entries of this sequence correspond to bits. When a '0' is added to a bit, it remains unchanged, while a '1' flips the bit. In Figure 3, the data stream originates from the data sink because the pseudo-random sequence is known. Then it is added byte-wise with the erasure sequence then bit-wise with the bit error sequence to produce the received sequence. In an equation, that is, $S_{received} = S_{pseudo} + S_{erasure} + S_{bit\_error}$.

The function of the error checking is to detect the erasure sequence $S_{erasure}$ and the bit error sequence S_{bit_error} based on the received sequence $S_{received}$ and the pseudo-random sequence S_{pseudo}. Due to their relationship described in the above equation, the data sink only need to detect sequence $S_{erasure}$, then S_{bit_error} is uniquely determined by that relationship. The error checking problem can be formulated as to detect the sequence $S_{erasure}$ based on sequences S_{pseudo} and $S_{received}$.

Based on classical detection theory, we know that the optimal error checking procedure starts after the whole sequence is received. The number of minutes or number of bytes specified by the user during program initiation determines the length of this whole sequence. Hypotheses are formed to reflect each and every possible instance of $S_{erasure}$. Then the maximum a posterior (MAP) criteria is used to identify the most

probable hypothesis. This winning hypothesis is then used to calculate S_{bit_error}, followed by link statistics update.

To understand the complexity of this optimal approach, let's first imagine an air link that only produces bit errors. For this air link, the all '0' $S_{erasure}$ sequence is the only hypothesis. There is no need for hypothesis testing. The program counts the number of bit differences (which is also called the Hamming distance) between S_{pseudo} and $S_{received}$ to calculate bit error rate. The complexity of the program comes solely from the statistics update, and it grows linearly with the length of the received sequence.

However, if the air link also produces data loss, the complexity of this algorithm increases at least exponentially with the length of the sequence. To see this, let's count the number of hypotheses. We'll assume that data loss is possible between any two neighboring received bytes, and the occurrences of data loss are independent of each other. For a received sequence of length M, there are M places where data loss may have happened. These M slots are, right before the first byte, between the first and the second byte, …… , between the M-1 th and M th byte. For each on of these M slots, there are $N_{\max\_Loss}+1$ possibilities, where $N_{\max\_Loss}$ is the maximum length of the lost data segment. These $N_{\max\_Loss}+1$ possibilities correspond to 0, 1, 2, …… , $N_{\max\_Loss}$ bytes are lost in this slot, respectively. Hence the total number of hypotheses is $(N_{\max\_Loss}+1)^M$. This means the complexity of hypothesis formation (and hence the program) grows at least exponentially with the length of the sequence.

Therefore, the optimal hypothesis testing is not practical at all. Any practical algorithm has to employ some sort of finite window length. Not surprisingly, this is the case with the RLTS program.

Table 2: Window structure of the RLTS program.

1	2	A	B	C	D	E	F	G	H	11	12	13	14
Reference		Hypothesis basis						Add. basis		Data word			

Table 1 shows the window structure of the RLTS program. The length of the window is 14 words (28 bytes). In this table, each entry corresponds to a word. The first two words (word 1 and 2) are called reference words. They carry the current phase of the pseudo-random sequence. At the very beginning of error checking, the last two words of frame synchronization segment are used as the reference words of the first window and the error checking process starts from the third word (word A). As a matter of fact, we make sure every time error checking starts from the third word by not shifting the winning hypothesis completely out of the window. We assume the reference words are

free of errors since they are part of the previous winning hypothesis (or frame synchronization). This is why they are called reference words. Essentially, the phase information can be restored elsewhere in the program and the reference words don't have to be in the window at all. Their existence mainly serves the purpose of easy debugging for the programmer. If however, a wrong hypothesis won in the previous window, the program does not go back and correct it. Instead, we deal with it with a set of techniques that will be discussed later.

The next 6 words, denoted as word A through F, are used as basis for hypothesis formation, which will be discussed in depth in the next section. The next 2 words, word G and H, are used as additional basis if the program finds it's necessary to get additional hypothesis. The last four words, numbered from 11 to 14, are not used as hypothesis basis. They help the program to decide how likely each hypothesis is. After the current window is processed, the reference words are generated using the winning hypothesis. Then word 11 is shifted to the third position in the window (word A) and the error checking process continues from it.

In the next three sections, we discuss the three steps of error checking, i.e., hypothesis formation, hypothesis testing, and statistics update in chronological order.

IV. Hypothesis Formation

When the frame synchronization is accomplished, the detection window is placed in the received sequence such that the last two words of the frame synchronization segment are in the reference words position. Then the data sink checks the consistency between word 2 and word A. If the word A is equal to word 2 plus one, they are consistent. The data sink then shifts the sequence out of the window (to the left) by one word and checks the consistency between the new word 2 and word A again. If the two words are not consistent, the data sink tries to decide what happened through the error checking process. There are a number of possibilities. The inconsistency in the window can be caused by bit errors in the word A or by missed bytes between word 2 and word A. It can also be cause by a combination of bit errors and data loss, in which case the data loss could have occurred anywhere in the window after word A (with bit errors in word A). The first step of error checking is the formation of hypothesis list, each entry of which corresponds to a possible $S_{erasure}$ as described in the last section.

To detect $S_{erasure}$ is equivalent to detect the locations and lengths of blocks of consecutive 'E's in $S_{erasure}$. The locations of these blocks are the location of loss data segments, and the lengths of the blocks are the lengths of loss data segments. As stated in the last section, for a window of size M, there are up to M of these blocks, each with length up to $N_{\max\_Loss}$. The total number of hypotheses is $(N_{\max\_Loss}+1)^M$.

The RLTS program simplifies the above procedure by assuming there is at most one occurrence of data loss in each detection window. Equivalently, this is to assume certain dependence among occurrences of data loss, i.e., the conditional probability of other data losses occur in the window given one occurrence is zero. There are a couple of

reasons why this assumption is made. The first reason is the complexity issue. As argued before, if multiple data losses are independent of each other, the total number of all hypotheses grows exponentially with the length of the window. The second reason has something to do with our program's hypothesis testing criteria and will be elaborated in the next section. During the making of the RLTS, we experimented with including two data loss occurrences in the hypothesis list. This turned out did more harms to the program than good. The reason will be discussed in the next section.

Now let's consider what if there are more than one data loss occurrences in the window, in which above assumption is not satisfied. It might be necessary to read the next two sections before one can fully understand some of the arguments here. Each one of data loss occurrences shifts the phase of the pseudo-random sequence forward by some integer $P_i, i = 1,\ldots,L$, where P_i is equal to the number of lost bytes associated with the *i*-th occurrence and L is the total number of occurrences. After the *i*-th occurrence, the phase of the sequence is shifted by $T_i = \sum_{l=1}^{i} P_l, i = 1,\ldots,L$. The job of the hypothesis testing is to acquire T_L then update the link statistics properly. Because of the simplification made by the RLTS program, the corresponding hypothesis is that ONE segment of T_L bytes are lost somewhere in the window. This hypothesis has a good chance of winning since the data after the *L*-th occurrence support this hypothesis. If however, this hypothesis doesn't win, it is highly likely that one of the T_i's is acquired instead, because part of the window supports the corresponding hypothesis. Or, some number less than T_L is acquired. In either case, the correct number of lost bytes (T_L) is not registered in the current window. However, when new data are read into the window, either the uncounted offset will be counted based on the new data (since the data sink is out of phase by that offset), or the error checking process resets. Reset is the mechanism the program gets back to frame synchronization when it's out of synch. Details of the reset mechanism will be discussed in later sections. The other possibility is a number greater than T_L is acquired. If this is the case, the program will reset in the next window with all likelihood unless the data loss in the next window offset the difference (in which case the number of lost bytes will be correctly counted). In short, either the program acquires the correct phase information (thus the number of lost bytes), possibly in later windows, or the reset mechanism kicks in and the communication is restarted. Generally, multiple data loss occurrence doesn't have much impact in the data loss rate statistics.

However, for each occurrence of multiple data loss, some bit errors are to be wrongfully counted by the RLTS. In the next section, we argue that incorporating multiple losses in the hypothesis impacts more on the accuracy of the bit error rate statistics.

The RLTS program further simplifies the hypothesis formation by creating only one (instead of $N_{\max\_Loss}$) hypothesis for each one of M slots. Let's take as an example the hypothesis for which the data loss occurred between word B and word C. As discussed in the last section, there are $N_{\max\_Loss}+1$ possibilities, each corresponding to the length of

the segment is 0, 1, 2, …… , $N_{\max\_Loss}$. The RLTS program generates only one hypothesis using word C as the basis. The hypothesis is generated by assuming (a) there are no bit errors in word C, (b) the location of the lost segment can be any where between the end of reference words and word C, (c) the length of the segment is an even number. By assumption (a), the length of the lost data is calculated by the phase difference of the word C and the reference words. If there are bit errors in word C, such calculated length is wrong. However, if one of the words after word C is free of bit errors, lets say word D, the hypotheses based on word D will produce true data segment length, although the data loss might not have happened right before word D. This is the exact reason for assumption (b), which doesn't assume the location of data loss is right before the current hypothesis basis. Instead, the program chooses the location of data loss to minimize the Hamming distance between the hypothesis vector and the detection window. Example 1 illustrates this process.

EX 1. Formation of hypothesis based on word C

The received window is as follows. There is a segment of data missing between word B and word C. In the table below, word C is underlined to show that it's the basis for the hypothesis.

Position	1	2	A	B	C	D	E	F	G	H	11	12	13	14
Recv.	0004	0005	0006	0007	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013

Step 1. The length of lost data segment is calculated.

Length = 000A – 0004 –4 = 2 words. (4 is the position difference between word 1 and word C).

Step 2. The location of the data loss is chosen such that the Hamming distance between the resulting hypothesis vector and the received sequence is minimized.

Position	1	2	A	B	C	D	E	F	G	H	11	12	13	14
Recv.	0004	0005	0006	0007	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #1	0004	0005	|0008	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #2	0004	0005	00|08	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #3	0004	0005	0006|	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #4	0004	0005	0006	00|09	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #5	0004	0005	0006	0007|	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013

The candidates are numbered from 1 to 5. The '|' in the sequence marks the corresponding location of data loss. The Hamming distances between candidates 1 to 5

and the received sequence are, respectively, 6, 6, 3, 3, 0. Hence candidate #5 is chosen as the hypothesis based on word C.

The next example illustrates how the correct hypothesis is included based on word D if there is a bit error in word C.

EX 2. Formation of hypothesis based on word D with existence of bit errors in word C

The received window is the same as the last example except that there is a bit error in word C (000A becomes 000E). Again, the word D is underlined since it's the hypothesis basis.

Position	1	2	A	B	C	D	E	F	G	H	11	12	13	14
Recv.	0004	0005	0006	0007	000E	000B	000C	000D	000E	000F	0010	0011	0012	0013

Step 1. The length of lost data segment is calculated.

Length = 000B – 0004 –5 = 2 words = 4 bytes. (5 is the position difference between word 1 and word C).

Step 2. The location of the data loss is chosen such that the Hamming distance between the resulting hypothesis vector and the received sequence is minimized.

Position	1	2	A	B	C	D	E	F	G	H	11	12	13	14
Recv.	0004	0005	0006	0007	000E	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #1	0004	0005	|0008	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #2	0004	0005	00|08	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #3	0004	0005	0006|	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #4	0004	0005	0006	00|09	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #5	0004	0005	0006	0007|	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #6	0004	0005	0006	0007	00|0A	000B	000C	000D	000E	000F	0010	0011	0012	0013
Can. #7	0004	0005	0006	0007	0008|	000B	000C	000D	000E	000F	0010	0011	0012	0013

The candidates are numbered from 1 to 7. Candidates 1 through 5 are identical to those of last example. The Hamming distances between candidates 1 to 7 and the received sequence are, respectively, 7, 7, 4, 4, 1, 1, 2. Again, candidate #5 is chosen as the hypothesis based on word D.

Through this example, we see that although there is an error in word C, a hypothesis with correct number of missed bytes enters the hypothesis list because one of the words after word C is error-free.

If the number of missed bytes is an odd number, in which case assumption (c) is not satisfied, all words in the window after the data loss consist of two bytes from two different words of the original sequence. The RLTS program uses the second byte of one basis word and the first byte of the next word as hypothesis basis for such situations. See the following example.

EX 3. Formation of hypothesis based on word C and D

The received window is as follows. There are an odd number of bytes missing between word B and word C. The second byte of word C and the first byte of word D is underlined to show that they are the basis for the hypothesis.

Position	1	2	A	B	C	D	E	F	G	H	11	12	13	14
Recv.	0004	0005	0006	0007	09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200

Step 1. The length of lost data segment is calculated.

Length = (000A – 0004 –4) words – 1 byte = 3 bytes (4 words is the position difference between word 1 and word C, 1 byte is the offset of the basis from the beginning of word C).

Step 2. The location of the data loss is chosen such that the Hamming distance between the resulting hypothesis vector and the received sequence is minimized.

Position	1	2	A	B	C	D	E	F	G	H	11	12	13	14
Recv.	0004	0005	0006	0007	09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200
Can. #1	0004	0005	|0600	0800	09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200
Can. #2	0004	0005	00|00	0800	09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200
Can. #3	0004	0005	0006	|0800	09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200
Can. #4	0004	0005	0006	00|00	09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200
Can. #5	0004	0005	0006	0007	|09<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200
Can. #6	0004	0005	0006	0007	00|<u>00</u>	<u>0A</u>00	0B00	0C00	0D00	0E00	0F00	1000	1100	1200

There are 6 candidates. The Hamming distances between these candidates and the received sequence are, respectively, 8, 6, 4, 3, 0, 2. Hence candidate #5 is chosen as the hypothesis.

In the same way, the RLTS program forms hypothesis based on word A through F. The total number of hypotheses is 13 (or possibly 17 if word G and H are used as additional basis). Words 11 through 14 are not used as hypothesis bases. The reason is that there are too few data bytes supporting hypotheses corresponding to data loss in these words. To see this, let's look at Ex. 2 again. If there is a bit error in word C, the hypothesis based on it has a large Hamming distance with the received sequence, basically because the calculated phase is wrong and there are 9 words after word C that contradict the wrong hypothesis. Instead, the right hypothesis enters the list based on word D. The words after word C supports the following hypothesis testing by (statistically) favoring the right hypothesis. Now let's imagine there is a bit error in word 14, and it's used as the hypothesis basis. There are not bytes after the word (in our window) to work against the resulting wrong hypothesis in the hypothesis test. For the same reason, there are more data bytes after word A to check the corresponding hypothesis then there are for word F. Therefore, each one of the hypothesis basis is not the same 'reliable'. In the program we choose to ignore the difference between word A and F and not to ignore the difference between word A and word 14.

Not using the last 6 words as hypothesis bases can cause problems when the data loss actually happened among these words. If the winning hypothesis of hypothesis testing has a Hamming distance from the received sequence larger than some preset threshold, the program uses word G and H to create 4 additional hypothesis. Basically, the program takes the large distance as an indication that the data loss didn't happen between word 1 and word F. In the mean time, bit errors in the last 6 words (4 words if one of the hypotheses generated by additional basis wins) are not included in the link statistics in the statistics update stage. Nor are these words shifted out of the window when the current error checking is done. Instead, word G (or 11) is shifted to the position of word A to form the new detection window. In some sense, they are in the previous detection window only as support data for the hypotheses. This is whey they are called data words.

V. Hypothesis Testing

Hypothesis testing is the procedure through which the winning hypothesis is picked according to certain criteria. After hypothesis testing, the link statistics are duly updated in the statistics update step. Naturally, we'd like to pick the most probable hypothesis, which requires the use of the maximum a posterior (MAP) criterion. However, this is not a practical approach. Generally, the evaluation of a posterior probability requires the knowledge of the a priori probability associated with each hypothesis. For example, we need to know the probability of no data loss occurred in the window, as well as the probability of a segment of 50 bytes long data segment is lost between word C and D. These probabilities are complicated function of the underlying physical link as well as the proprietary radio protocols, which are virtually impossible to acquire. Generally, it's not practical to use the MAP criterion.

In light of this practical constraint, the RLTS uses the minimum Hamming distance criterion with ranking. It is associated with the maximum likelihood criterion in detection theory. Assuming bit errors are independent and identically distributed (i.i.d.),

the likelihood of each hypothesis is a decreasing function of the Hamming distance between the observation and the hypothesis vector. The program picks the hypothesis with least Hamming distance to the observation as the winning hypothesis. All but one (the one corresponds to pure bit error hypothesis) such distance is already calculated in the hypothesis formation stage. In case of a tie, the winning hypothesis is picked based on preset ranking. In our program, the bit error only hypothesis has the highest ranking. Hypothesis with basis that's in the beginning of the window has higher ranking than that with basis in the end of the window. For example, hypothesis based on word A has the second highest ranking (just after the bit error only hypothesis), while hypothesis based on word F (or G, if it's used as bases) has the lowest ranking. This ranking scheme is used simply because there are more data bytes supporting hypothesis with basis in the front.

Now let's get back to the discussion in Section II about including multiple data loss occurrence in the hypothesis lost. Let's consider that data loss can occur anywhere in the window after the reference words regardless of other occurrences, and there is no inconsistency among data words (word G through 14). Then we can always create a hypothesis that has zero Hamming distance with the received sequence by attributing every inconsistency in the received sequence to data loss. With our hypothesis testing, this hypothesis always wins. The algorithm then degenerates into an inconsistency counter, which simply registers each inconsistency in the received sequence and practically kills the bit error nature of the air link.

Above arguments are crude in the sense that the reset mechanism is not considered, nor is the inconsistency in data words (although this can be shown not to be a factor). However, it reveals the dilemma faced by the multiple data loss hypothesis. On one hand, hypothesis allowing multiple data loss 'fits' the data better. This advantage is demonstrated in an extreme fashion in arguments of the last paragraph. On the other hand, the a priori probability of multiple occurrences is much lower than that of single occurrence while both of them are hard to evaluate. If we don't include multiple loss hypotheses, the link statistics will be penalized because occasionally (i.e. when multiple losses do happen in one window) the right hypothesis is not picked (even included in the list). If we do include the multiple loss hypotheses and don't have a way of obtaining the appropriate a priori probabilities, we run a greater risk of picking the wrong hypothesis in hypothesis testing. This is why including hypotheses corresponding to two occurrences of data losses seemed to do less good to the program than harm. In short, multiple data losses hold a much smaller probability and it's hard to assess this probability in order to properly offsetting its impact in hypothesis testing. This is the other one of the two reasons why they are not included in the hypothesis list.

VI. Statistics update

The step after hypothesis testing is statistics update. Number of bit errors, lost bytes and counted and registered. As mentioned in the last section. Only bit errors in the hypothesis basis are included in the statistics. Therefore this number can be different from the Hamming distance in hypothesis testing.

Most of the processing in this stage is straightforward bookkeeping. After the errors are registered, the hypothesis basis words are shifted out (to the left) of the window. If the additional basis words are used to generate the winning hypothesis, they are shifted out too. In the end, the data word section is going to be the position for the new error checking process.

VII. Reset Mechanism

Reset mechanism is a very important part of the algorithm. When the data sink believes it has lost the correct phase information of the received signal, it resets the error checking process by restarting frame synchronization.

The reset mechanism kicks in if the winning hypothesis corresponds to a lost data segment longer than $N_{\max\_Loss}$. In the implementation, a threshold is preset as $N_{\max\_Loss}$. This threshold can be either estimated as the product of throughput and channel coherence time (with margin for error), or set by trials. The program resets in two situations. The frame synchronization for the current window can be good, while the data are too corrupted for any reasonable hypothesis to win. This can be a result of channel variation or interference. After the reset, the frame synchronization procedure makes sure the communication restarts only after a segment of data with certain length are received correctly, as described in Section I. It's also possible that the data sink is out of synchronization with the received sequence. This happens if a wrong hypothesis has won in some previous hypothesis testing, and went undetected by the reset mechanism in that testing and every testing since. Most likely it happened in the window immediately before the current window since the probability that false frame synchronization went undetected is very low. Assuming the current hypothesis testing recovers frame synchronization, the sum of the lost data lengths of the current and the previous winning hypothesis will be near to a period of the pseudo-random sequence, which is 131072. To understand this intuitively, imagine a whole period of the sequence is missing, the resulting received sequence shall have exactly the same phase as the original sequence. The program loses and then regains the frame synchronization by advancing the phase for a whole period in two consecutive hypothesis testing.

Therefore, when the winning hypothesis corresponds to a lost segment longer than the threshold, the program resets the communications, more over, it looks at the length of lost data segment of the previous hypothesis testing. If the sum of it and current winning hypothesis's lost segment length is equal to the period of the pseudo-random sequence, the program goes ahead and deletes the previous entry from link statistics.

Extremely rarely, the wrong hypothesis won before the previous window. This hypothesis first escapes the reset mechanism of its own window, and the resulting false frame synchronization escapes again in every following windows, until it was finally captured. When this happens the RLTS program will not be able to go back and remove the impact of the first wrong hypothesis from the link statistics. This causes some error in the produced statistics. Throughout our testing, this occurs extremely rarely. Each occurrence will bring less than the preset nominal $N_{\max\_Loss}$ value (otherwise the program

should have reset) to the total number of lost bytes. Because of its rare occurrence, its impact on the final statistics is minimal. Also, this impact can be adjusted by changing the nominal $N_{\max\_Loss}$ value in the reset mechanism.

VIII. Program Output

When the RLTS program is running, there is always a line of output at the bottom of data sink's window. An example of this output line can be seen in figure B.5 in Appendix B.

The line contains the following information:

- The word that is being read into the window.
- The total number of bit errors counted.
- The total number of bytes read and tested.
- The number of bytes currently in the buffer.

Meanwhile, the information on current error checking process scrolls on the screen. It's in the same format as the first type of output file as described below.

When the program terminates, it produces four types of text output files. The first type of file has detailed information about each error. Depending on the length of the test, the file can be too big to be opened by Windows Notepad program. To avoid this, the program creates a new file every time when the current file size reaches some preset limit. These files are assigned serial numbers in chronological order. The filename of this type of file is FILENAME_MMDDYYHHMMSS_XXX.txt, where FILENAME is the filename entered at the start of the program, MMDDYY is the date that the program was started, HHMMSS is the time the program was started, and XXX is the serial number of the file starting at 000.

The second type of file contains the final statistics such as the number of bit errors, the number of missed bytes, the number of packet losses, the number of resets, the bit error rate, the missed byte rate, etc. The filename for this type of file is FILENAME_MMDDYYHHMMSS_final.txt.

The third type of file contains a list of each occurrence of data loss and includes the date, time, and the number of bytes missed for each packet loss. The filename for this type of file is FILENAME_MMDDYYHHMMSS_ploss.txt.

The last type of file is similar to the packet loss list. It contains a list of each reset and information pertinent to the reset, such as the time and date stamp as well as how many bytes passed during the reset, and how many bytes were read in during the reset. The filename for this type of file is FILENAME_MMDDYYHHMMSS_reset.txt.

The data loss and reset files can be easily imported into excel for further analyzing and graphing if it is desired. Examples for all four types of files can be found in Appendix C.

4.3 Field Tests

Field tests were conducted at several locations around Purdue's West Lafayette campus and in Indianapolis near a stretch of US 40. The main goal of the field tests was to measure the link quality of MDS 9810 spread spectrum radios to understand some critical issues in the deployment of spread spectrum radio interconnect. These issues include:

- coverage area of the air link.
- sensitivity of the link to loss of line-of-sight.
- impact of choice of antenna (omni vs. Yagi) on the link quality.
- ways to extend the coverage of the air link.
- performance of MDS 9810 in the presense of interference.

Seven field test sites were chosen to study link quality in the following 6 scenarios:

1. behind small hill without line-of-sight (LOS)
2. long distance without line-of-sight
3. moderate distance without line-of-sight
4. around the turn without line-of-sight
1. on extension of the valley without line-of-sight
2. behind city blocks without line-of-sight, with interference

Another set of experiments was conducted to compare the interference tolerance capability of FHSS and DSSS radios. Radios that were used in this test were the MDS 9810 (FHSS), the ENCOM 5200 (FHSS), and the GINA 6000N (DSSS).

In the conducted tests, one radio was set up as the master, while the other radio was set up as the slave. Both radios were hooked up with computers running testing programs. The master radio transmits mock data that is known by the slave. The slave compares the received data against a correct copy to determine whether and what kinds of errors have occurred. There are three kinds of errors. They are bit error, data loss, and the combination of these two. Based on the comparison, the slave comes up with link quality statistics such as bit error rate, data loss rate, and the distribution of the lost packet length. For details of the testing program, refer to the program description. In all tests, the data rate of the radios has been set to 38.4kbps, the highest rate supported by the MDS 9810 radio.

4.3.1 Field Tests without Line-of-Sight

Figure 6. shows map of the sites of tests without line-of-sight. Yellow flags mark the location of master stations, while Blue flags marks the location of portable (slave)

stations. In the following sections, we present each test and the corresponding results in detail.

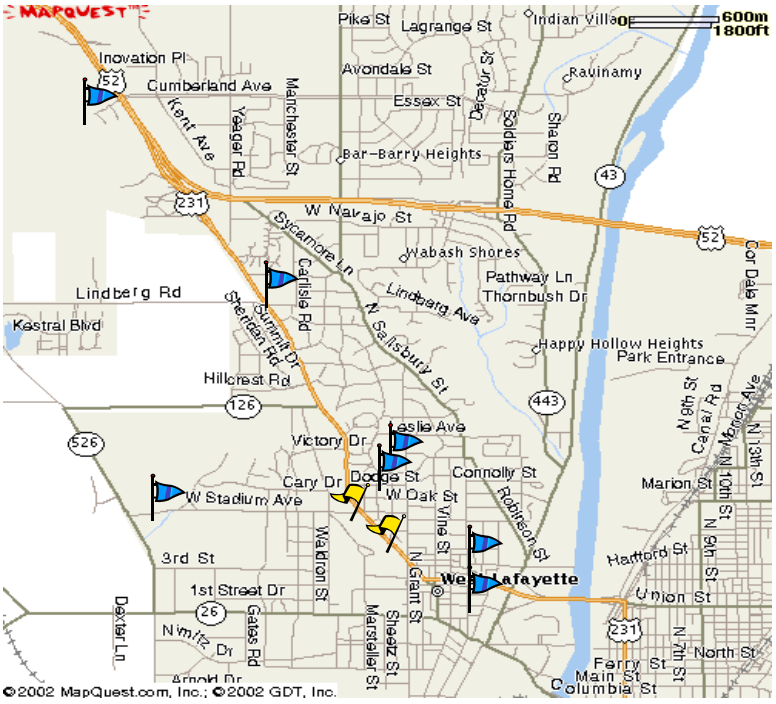

Figure 6: Field test map.

Scenario #1: behind small hill without LOS

The master antenna is located on top of the Civil Engineering building. The slave station is set up on Stadium Ave. near Airport Road behind a 30 feet high small hill. The distance between two radios is about 0.8 mile. Figure 7. Shows the map of this test. There is no line-of-sight between portable station and base station. For comparison, one set of test is done on top of Picket hill with line-of-sight. Omni antennas are used in both locations. Test results are listed in the table below.

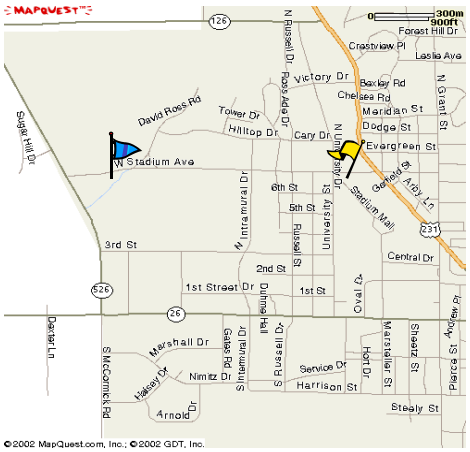

Figure 7: Map for scenario #1.

Location	Test Length	Throughput (kbps)	Bit Error Rate	Data Loss Rate
Top of the hill	20 mins	15.9	0	0
Behind the hill	20 mins	15.9	7e-6	7e-4

Table 3: Test Results for Scenario #1.

The hill clearly introduced degradation in link quality. It seems for this scenario bit error is much less a problem than data loss although the impact of the errors on the user application depends heavily on the way the user application handles the errors.

Scenario #2: Long distance without LOS

This is the toughest propagation environment we have tested so far. The master antenna is located at the top of the Civil Engineering building. The slave radio is located at the West Lafayette Wal-Mart parking lot, near the intersection of US 231 and Cumberland Ave.. The distance between two radios is about 2.5 miles. There are two ridges lying between the Master station and the slave station as highlighted in the elevation map on the right. They are 700 feet above the sea level. Antennas of both radios are about 690 feet about the sea level. There is no line-of-sight between them. With the omnidirectional antenna, the portable radio failed to establish communication with the master.

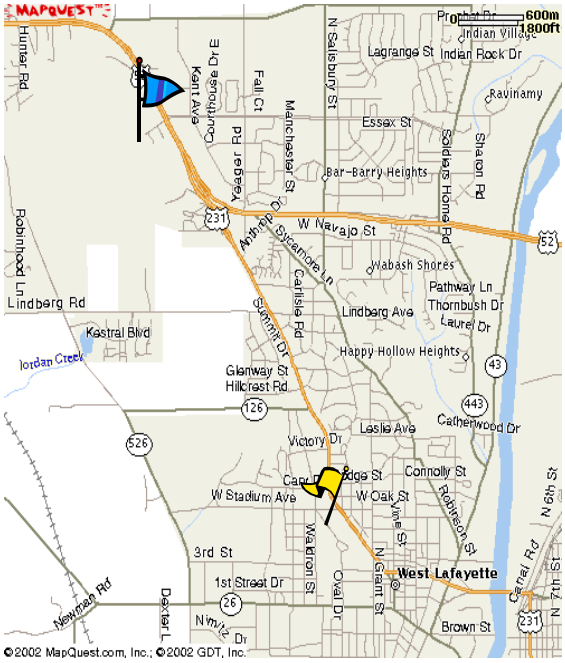

Figure 8: Map for scenario #2.

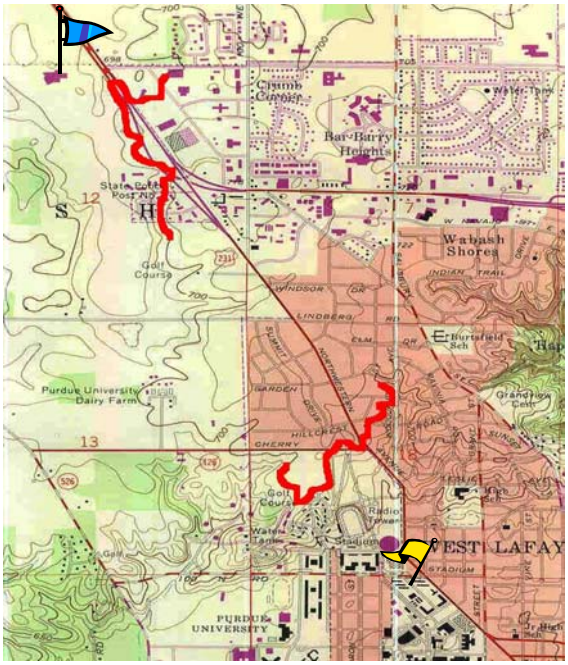

Figure 9: Elevation map for scenario #2.

Results shown below are obtained with the Yagi. Although there is only 3 dB antenna gain advantage for the Yagi over the omni antenna, it did make a difference. About a quarter of transmitted data is lost. Data loss is still a more serious problem than bit errors. Throughput decreased by about one third compared to scenario #1.

Location	Test Length	Throughput (kbps)	Bit Error Rate	Data Loss Rate	RSSI (dBm)
Wal-Mart	35 mins	9.67	5.3e-3	22.5 %	-111 to -113

Table 4: Test results for scenario #2.

Scenario #3: Moderate distance without LOS

We then moved the portable station closer to the master to the intersection of Elm street and Northwestern Avenue. Two sets of testing are done with two different locations of master antenna. One location is on the top of Civil Engineering building. The other one is the top of MSEE building. The distance between slave and master radios is about 1 mile. There is one ridge lying between two radios, as highlighted in the elevation map on the left. There is no line-of-sight. Communication can be established with both omni and Yagi antennas between the portable radio and the Civil Engineering master. Only with Yagi antenna can the portable radio talk to the #2 radio of MSEE building, which is the closest one of three radios in MSEE building to the portable radio. Communications can not be established with the other two radios of MSEE building. Test results are collected with Yagi antenna only.

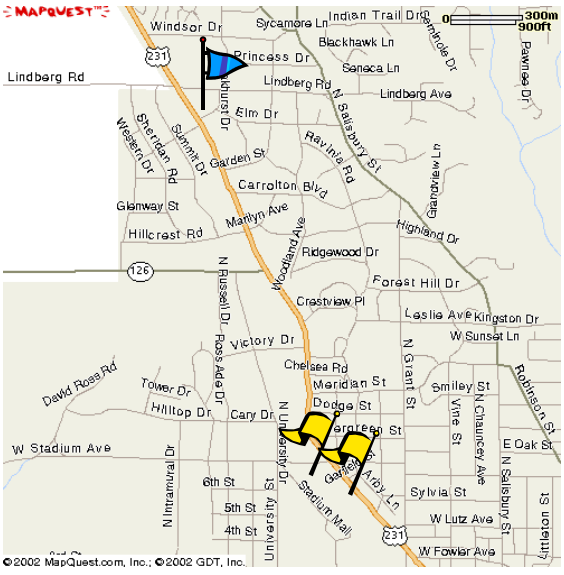

Figure 10: Map for scenario #3.

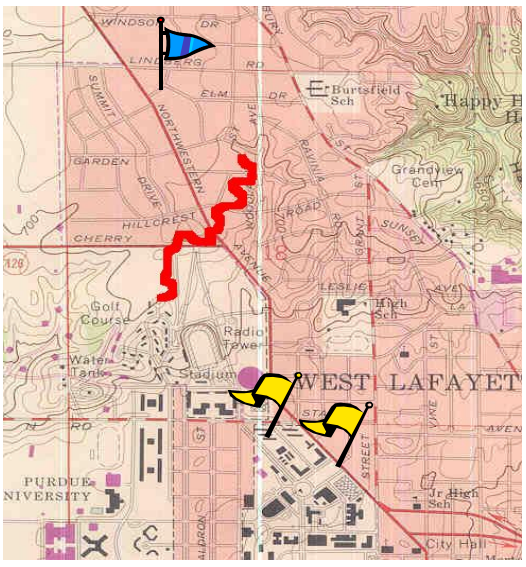

Figure 11: Elevation map - scenario #3.

Master Station	Test Length	Throughput (kbps)	Bit Error Rate	Data Loss Rate	RSSI (dBm)
Civil Engr.	20 mins	14	7e-4	9.7%	-107 to -109
MSEE #2	20 mins	15	3e-4	4.5%	-112 to -114

Table 5: Test results for scenario #3.

For both tests, the bit error rate and data loss rate are better than those of scenario #2. The throughput is higher than that of scenario #2 and close to that of scenario #1. MSEE link works better than the Civil link since it has less bit errors/lost data and higher throughput. However, the RSSI measure for MSEE master is lower than that of Civil master. This indicates that RSSI reading is not a good measure of link quality.

Scenario #4: Around the turn

The master antenna is located at the sidewalk in front of the MSEE building, 15 feet above the street level. The slave radio is located at a parking lot along US 231 between Chauncey Avenue and Salisbury Street. Tall Purdue buildings along US 231 form a `valley'. The master radio is located right in the valley. US 231 took a turn of about 60 degrees. The slave radio is located around the corner. The location of portable radio is so chosen such that the two radios are not blocked by the huge Northwest Avenue parking garage. The distance between two radios is about half a mile. There is no line-of-sight. Houses, apartment buildings, and trees are in the way.

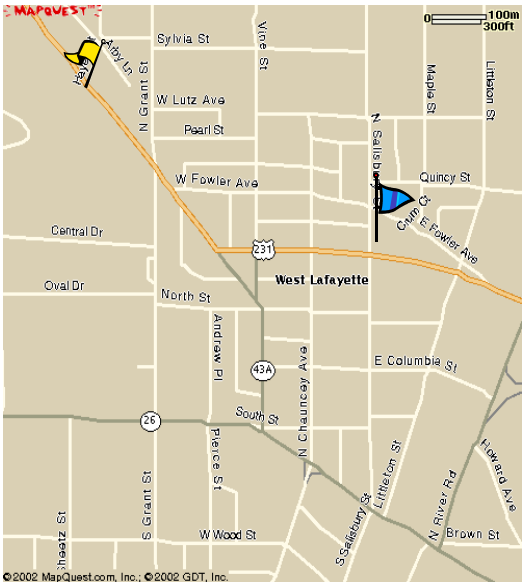

Figure 12: Map for scenario #4.

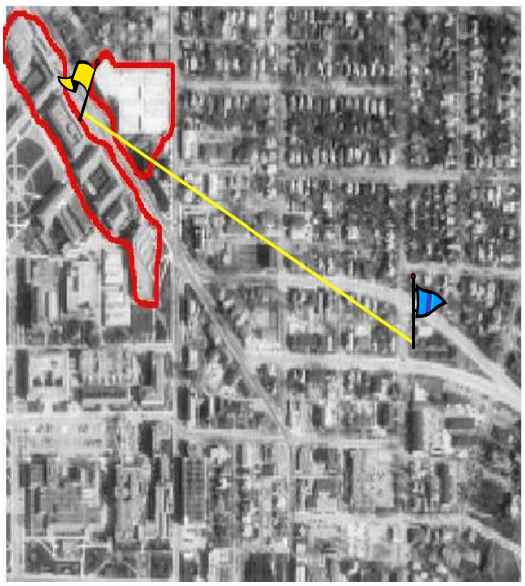

Figure 13: Ariel photo for scenario #5.

Two sets of tests are conducted with Yagi and omni antennas respectively. Although Yagi antenna did give about 4 dB difference in RSSI, link statistics from both tests are very close. The EM wave was scattered by buildings in surrounding city blocks. As we changed the direction of Yagi antennas around, RSSI readings remained in the same 4dB region. We conclude that Yagi antenna doesn't offer much advantage in scattering environment.

Antenna	Test Length	Throughput (kpbs)	Bit Error Rate	Data Loss Rate	RSSI (dBm)
Omni	20 mins	14	0.1 %	9.1 %	-106 to -109
Yagi	20 mins	13.8	0.1 %	10 %	-102 to -105

Table 6: Test results for scenario #4.

Scenario #5: On extension of the valley

We moved the portable station 3 blocks south. It is now on the extension of the valley. There is no line-of-sight. Houses and apartment buildings are in the way of two radios. Omni antennas are used in the test. The link quality is much better than that of scenario #4, although it is probably still not good enough for traffic applications without additional protocol.

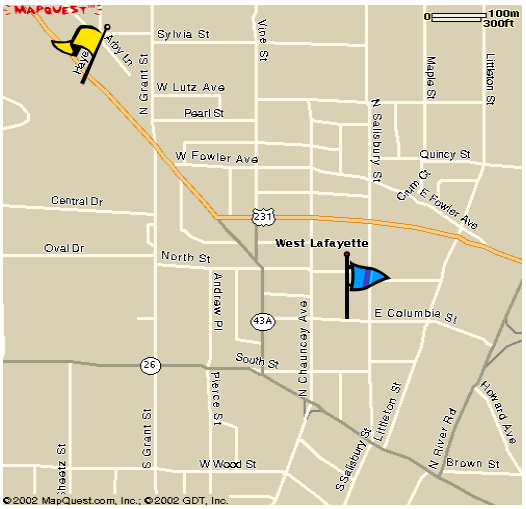

Figure 14: Map for scenario #5.

Figure 15: Ariel photo for scenario #5.

Test Length	Throughput (kbps)	Bit Error Rate	Data Loss Rate	RSSI (dBm)
30 mins	15.5	5.3e-5	0.37 %	-94

Table 7: Test results for scenario #5

Scenario #6: Behind city blocks with interference

The master station is on the sidewalk in front of MSEE building. The portable radio is placed at two locations behind McDonald's restaurant. There is no line-of-sight. Houses, a gas station, and other buildings are in the way. The distances between master radio and two portable radios are about 300 yds and 500 yds respectively. Link quality degrades rapidly as portable radio moves away from the master. At Hayes & Meridian, although Yagi antenna provides a 4 dB gain over Omni antenna, link performance is actually poorer. We believe this is because the Yagi picks up more interference since it points to McDonald's restaurant.

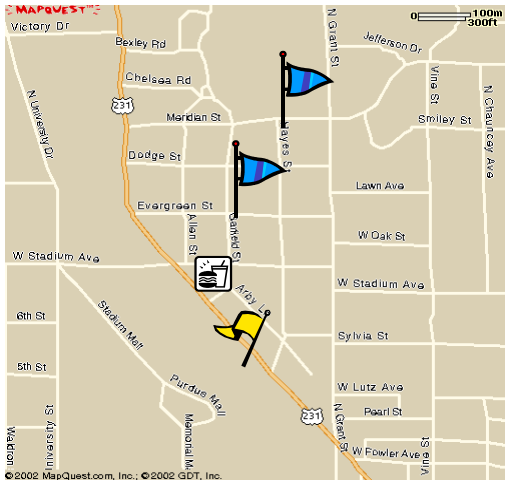

Figure 16: Map for scenario #6.

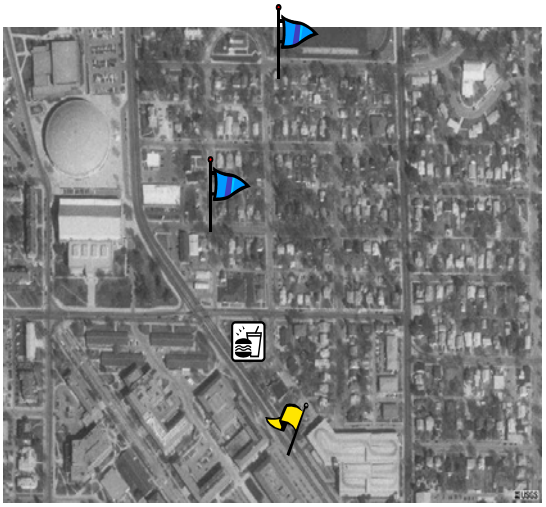

Figure 17: Ariel photo for scenario #6.

Location	Antenna	Test Length	Throughput	BER	DLR	RSSI (dBm)
Evergreen & Garfield	Omni	40 mins	15.6 kbps	1.3e-4	0.36 %	-84 to -87
Hayes & Meridian	Yagi	20 mins	14.8 kbps	7.3e-4	5 %	-102 to -104
Hayes & Meridian	Omni	40 mins	14.8 kbps	6.4e-4	4.8 %	-106 to -108

Table 8: Test results for scenario #6

4.3.2 Field Tests with Interference

Field tests in Indianapolis

Field tests are carried out on a stretch of US 40 near the Indianapolis Int'l airport. There are a number of warehouses and shipping centers that are presumably interference sources to the ISM band spread spectrum radios. The purpose of the test was to measure the performance of the investigated radio link in such an interference environment.

The master station is set up to the south side of US 40 near Dollar Inn. Two tests were carried out with slave stations set up in front of the K-Mart and at the intersection of Bailey Drive and US 40. Both slave stations are to the north side of the street. All antennas are 15 ft above the ground. Line-of-sight exist for both tests. The distances between the master station and the slave station of the two tests are about 0.6 and 1 mile respectively. The results for the tests are shown in the table below. It seems additional user protocols are necessary for the application to function properly if the distance between radios is equal or longer than 1 mile.

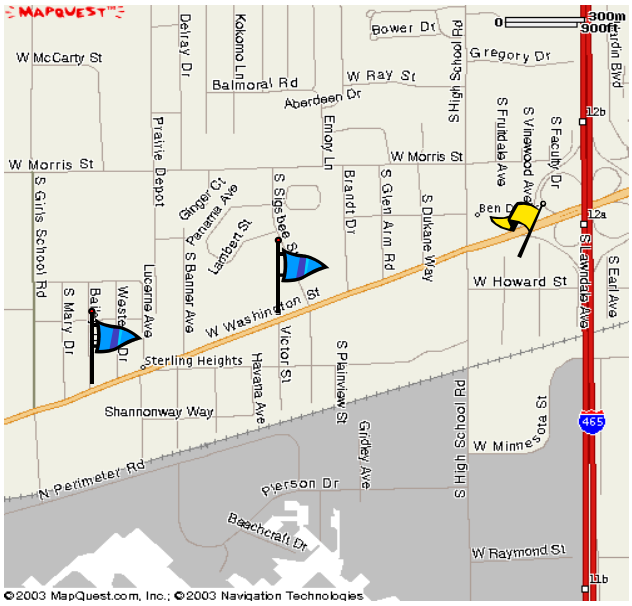

Figure 18: Map for Indy test

Location	Antenna	Test Length	Throughput	BER	DLR	RSSI (dBm)
K-Mart	Omni	20 mins	14.5	1.4e-4	0.5 %	-72
K-Mart	Yagi	20 mins	14.5	0	1e-6	-64
Bailey	Omni	20 mins	13.7	1.8e-3	6.1 %	-96
Bailey	Yagi	20 mins	14.4	4.9e-4	1.5 %	-91

Table 9: Test results for Indy test

Test with interfering network

Interference tests were carried out to compare the sensitivity of MDS 9810, ENCOM 5200, and GINA 6000N radios to interfering network. Among these three radios, MDS and ENCOM are FHSS radios while GINA is a DSSS radio.

In the test, radio #1 and radio #3 of the MSEE node, i.e. two radios to the right hand side in the aerial photograph, are configured as the tested network. Radio #2 and Civil radio are configured as the interfering network. Four copies of the RLTS programs were running to generate traffic in the two networks. Data flows from radio #1 to radio #3 and from radio #2 to Civil radio, respectively. All tests use omni antennas with test length equal to 60 minutes.

The transmitting power of the tested network is fixed at 30 dBm (1Watt) while the power of the interfering network varies from 20 dBm to 30 dBm. Since the output power of GINA is fixed at 30dBm and can't be adjusted easily, a 10 dB attenuator was used in order to obtain the desired 20 dBm interfering power. For MDS and ENCOM radios the output power can be adjusted by the driver shipped with the product. In the sequel we define the near far radio (NFR) to be

$$\mathrm{NFR\ (dB)} = 10\log\left(\frac{\text{power of interfering radio}}{\text{power of tested radio}}\right).$$

Figure 19: Ariel photo for the test with interfering network.

In the following three sections each radio is implemented as the tested radio while the remaining two play the role of interfering networks. The statistics are collected at the end of each test. Three metrics, namely throughput, bit error rate, and data loss rate, are used to compare the performance of these three radios. A comparison test was also carried out with the interfering network turned off.

I. MDS as the tested network

Other than ENC and GINA, here we let MDS be the interfering network as well, and the two (tested and interfering) MDS networks are configured with different network addresses but with the same hopping pattern.

If the interference is coming from ENC or GINA, MDS sacrifices throughput in order to keep a low bit error rate (BER) and data loss rate (DLR). In fact, it is observed that the BER and DLR are even lower than without interference. This phenomenon may be an indication that MDS changes its channel coding or retransmission mechanism to combat interference.

On the other hand, if the interference is coming from another MDS network with the *same* hopping pattern, i.e., occupies the same frequency bands, then the throughput will drop and bit error rate increases. Therefore, to eliminate possible interference from the same brand of radios, the hopping pattern of each network must be set different.

Interference	NFR (dB)	Throughput (bps)	Bit error rate	Data loss rate
None	N/A	14933	5.0e-7	1.5e-5
MDS	-10	12771	1.9e-5	2.4e-3
	0	9510	2.0e-5	2.9e-3
ENC	-10	5190	4.0e-8	1.0e-6
	0	4681	4.9e-8	1.0e-6
GINA	-10	7790	1.8e-8	8.0e-7
	0	6671	2.0e-8	1.0e-6

II. ENC as the tested network

The data loss rate of ENC is very high (38%) when the interference is coming from MDS. Other then that, ENC has a descent throughput and bit error rate.

Interference	NFR (dB)	Throughput (bps)	Bit error rate	Data loss rate
None	N/A	24004	1.0e-7	5.4e-2
MDS	-10	8796	1.4e-6	3.7e-1
	-5	7852	1.5e-6	3.8e-1
	0	7013	2.0e-6	3.8e-1

GINA	-10	10055	9.2e-8	2.9e-3
	0	9030	1.2e-7	6.0e-3

III. GINA as the tested network

The data loss rate of GINA is high (13% to18%) when either MDS or ENC acts as an interferer. Also, the throughput drops quickly when interferers are present.

Interference	NFR (dB)	Throughput (bps)	Bit error rate	Data loss rate
None	N/A	18032	2.8e-5	9.5e-2
MDS	-10	5017	5.0e-6	1.7e-2
	-5	4786	9.8e-5	8.2e-2
	0	4676	1.9e-4	1.3e-1
ENC	-10	5151	4.0e-6	1.1e-2
	0	4916	4.7e-4	1.8e-1

The following three figures compare the performance of these radios in terms of throughput, bit error rate, and data loss rate, respectively.

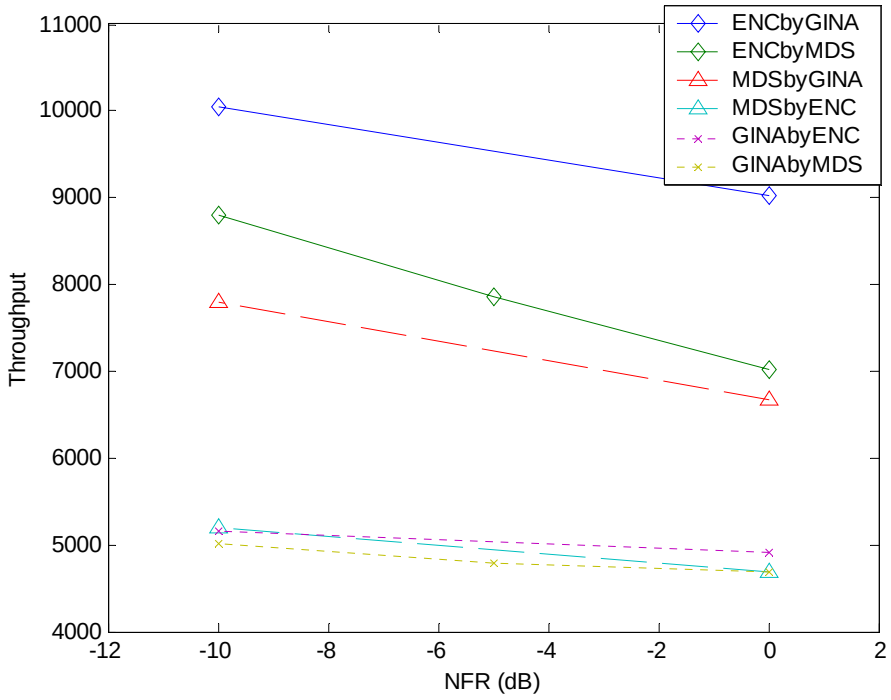

Figure 20: Throughput comparison.

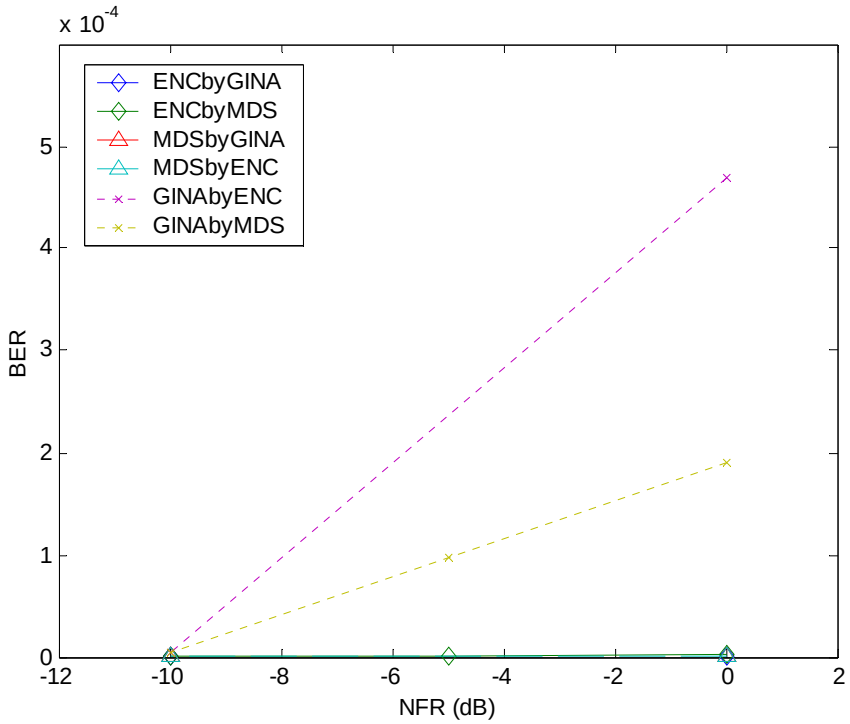

Figure 21: Bit error rate comparison.

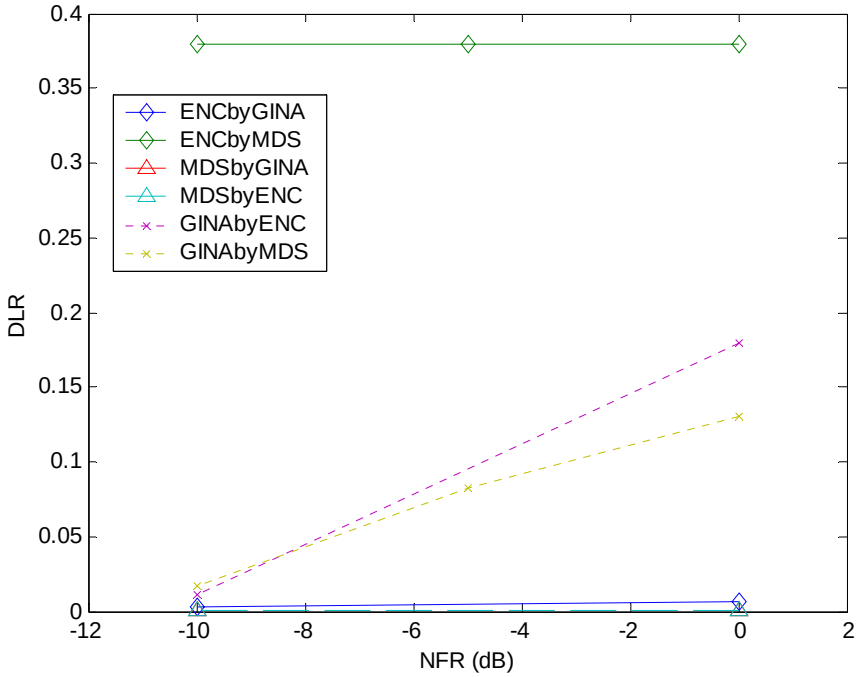

Figure 22: Data loss rate comparison.

From the figures above we have the following observations:

I. In terms of throughput

- Without interference, ENC has the highest throughput (24kbps) while MDS has the lowest (14.5kbps).
- Both FHSS radios are less affected by DSSS type of radios than by FHSS radios.
- ENC still has the highest throughput (7kbps) when interference is present.

II. In terms of bit error rate

- Without interference, both ENC and MDS have low BER (10^{-7}) while GINA has BER around 10^{-5}.
- When the interference is present, both MDS and ENC have very small BER ($<10e^{-6}$), which is even better than when no interference is present. GINA has its BER ranging from 10^{-6} to 10^{-4}, depending on the power of the interferer.

III. In terms of data loss rate

- Without interference, MDS has the lowest DLR (10^{-5}) while both ENC and GINA have DLR around 10^{-2}.
- With interference, again MDS (10^{-6}) performs even better when no interference is present.
- ENC has high DLR (38%) when MDS is the interferer.
- GINA has high DLR when either MDS or ENC is the interferer.

In summary, ENC has the highest throughput whether interference is present or not. However the high data loss rate of ENC (38%) caused by MDS is generally unacceptable. MDS has throughput in the range of 6 kbps to 14 kbps, although not as high as ENC (7kbps to 24kbps), but should be enough for traffic signal applications. The quality that distinguishes MDS from the other radios is the low BER and DLR in every situation we tested. Finally, GINA is not as competitive as the other two FHSS radios in general.

To find out which radio to use, we consider the following two scenarios:

(a) If there is very little interference, or a data link layer protocol is present to check for lost packets, ENC is the best radio to achieve high data rate.

(b) If there is a concern about interference from other spread spectrum networks, or there is no data link layer to check for bit errors or missed bytes, MDS is a conservative choice because of its good BER/DLR performance.

5. Conclusions

A spread spectrum radio testbed network has been built on the Purdue University campus. The testbed has been used to compare radios from three different manufacturers and can be used for further testing as INDOT needs arise. The results of the particular field tests conducted to date also yield the following more specific conclusions.

1. An additional robust protocol is essential to the successful use of MDS 9810 radios in non line-of-sight situations. This protocol is responsible for removal of residue bit errors and retransmission of lost data. It works as a filter, preventing any errors from reaching user application.

2. With a robust protocol, the coverage of wireless links can be improved significantly. Otherwise, the link quality doesn't seem good enough to support important applications without line-of-sight.

3. The design of such a protocol should emphasize an efficient retransmission scheme instead of powerful error control coding. Distribution of loss packet length produced by our testing program provides important information for the design of retransmission protocol.

4. RSSI is not a very good measurement of link quality. Testing programs such as the one we wrote should be used whenever possible.

5. Omni-directional vs. Yagi antenna:

   (a) High-gain Yagi antenna should be used for long distance, rural environment.

   (b) In scattering city environment, Omni with a couple of dB lower gain performs just as well as Yagi antenna.

   (c) If there is an interference source in the middle of the radio link, Yagi should be avoided since it picks up more of the interference signal.

6. Interference tests show that MDS 9810 handles the interference well at the price of the throughput; ENC has a high throughput but also a high DLR in certain situations. Therefore, if 6 kbps to 14 kbps data rate is enough for the application, MDS would be an ideal choice. Also, it is important to set the hopping patterns to different values, whenever possible, when different MDS networks are operating in the same area.

6. Recommendations and Implementation Suggestions

This project has funded the construction of a testbed network for experimentation with spread spectrum communications in the 900 and 2400 MHz band. The current network consists of five fixed nodes and two portable nodes. Of the fixed nodes, three are located in the MSEE building, one is located in the Harold L. Michael Traffic Operations Laboratory, and one is located in the experimental traffic signal cabinet at the intersection of Stadium and Northwestern Avenues in West Lafayette. The testbed has been used to evaluate spread spectrum radio technologies from vendors Microwave Data Systems, GINA, and EnCom, as regards radio performance as a function of link distance and with varying levels of interference. The project has produced software that can be used to test any radio presenting a standard RS-232 interface to customer equipment. The testbed can also be interfaced with wireless channel emulator equipment located in the Wireless Communications Research Laboratory in order to test radio performance in multipath and fading environments.

With the completion of the testbed and the verification of its operation the main item for implementation in this project is complete. However, in order to maximize the testbed's benefit to INDOT it is recommended that testbed operation be continued to allow further radio testing and experimentation with radio and traffic control and sensor integration. In this task, it is proposed to continue the operation of the testbed to accomplish the following:

1. Testing of improved vendor radios as they become available.
2. Experimentation with the setting of various radio network parameters (e.g., error control coding, data rate, retransmission protocols) and how they influence the performance of application specific software, such as traffic signal control.
3. Experimental verification of analytical models for network performance evaluation (e.g., throughput, etc.).

7. References

1) J. V. Krogmeier, K. C. Sinha, M. P. Fitz, S. Peeta, and S. Y. Nof. Borman expressway ATMs equipment evaluation. Technical Report FHWA/IN/JHRP-96/15, School of Civil Engineering, Purdue University, West Lafayette, IN, 1996.
2) J. V. Krogmeier and M. P. Fitz. Borman expressway point-to-point wireless modem. Technical Report SPR-2036 FHWA/IN/JTRP-98/3, Joint Transportation Research Project, West Lafayette, IN, February 1998.
3) J. V. Krogmeier and N. B. Shroff. Wireless local area network for its communications using the 220 MHz ITS spectral allocation. Technical Report SPR-2193 FHWA/IN/JTRP-99/12, Joint Transportation Research Project, West Lafayette, IN, November 1999.
4) R. L. Pickholtz, D. L. Schilling and L. B. Milstein. Theory of spread spectrum communications – a tutorial. IEEE Trans. Commun., COM-30:855-884, May 1982
5) G. L. Turin. Introduction to spread-spectrum antimultipath techniques and their applications to urban digital radio. Proceedings of the IEEE, 68:328-353, March 1980.
6) Microwave Data System, MDS 9810/24810 Installation and Operation Guide, MDS 05-3301A01, Rev. B, April 2000.
7) GRE America, GINA User's Manual, Model: 6000N-5/6000NV-5/8000N-5/8000NV-5. 1999.
8) ControlPak Programming and Diagnostic Software for ENCOM 5000 Series Serial Interconnect Radios User Manual, Rev02, 2002.
9) Freewave Technologies, Inc., FreeWave Spread Spectrum Wireless Data Transceiver User Manual, V4.2f, 2000.

Appendix A – RLTS Program Flow Chart

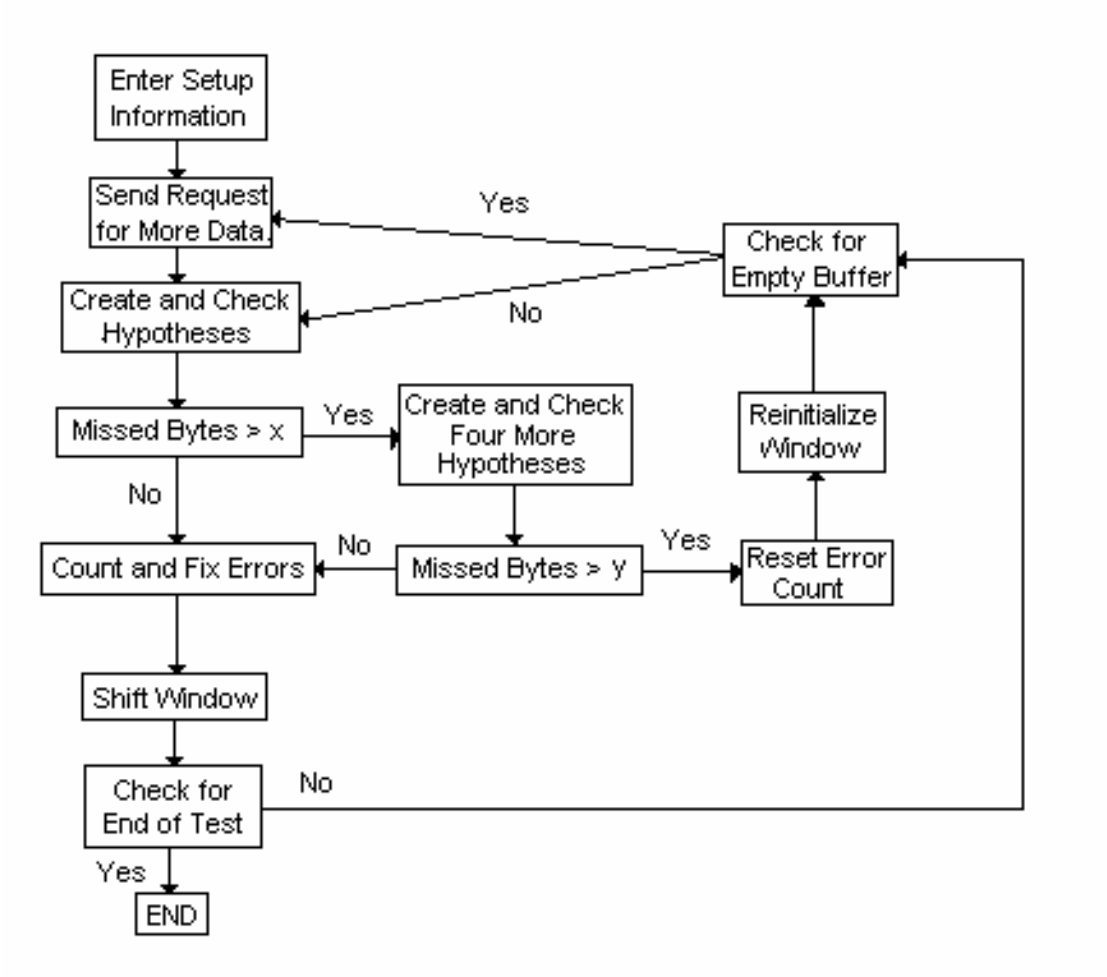

Figure A.1 – Flow chart of the RLTS progam.

Appendix B – RLTS Program Screen Shots

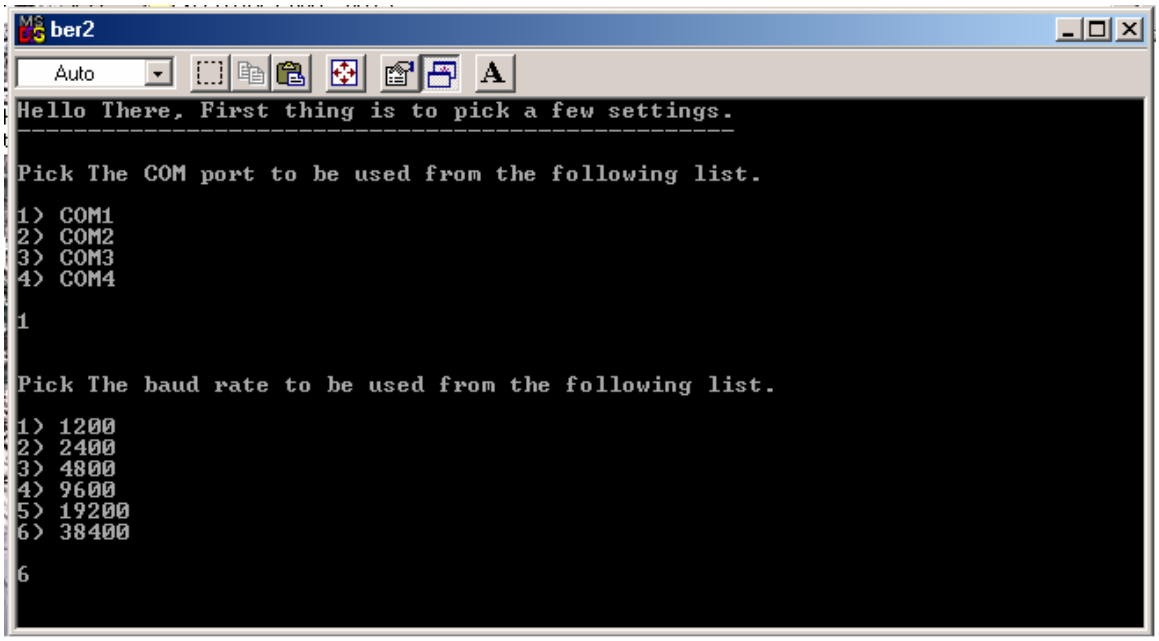

Figure B.1. A screen shot of the program initiation. User is prompted to choose the RS-232 port and its baud rate.

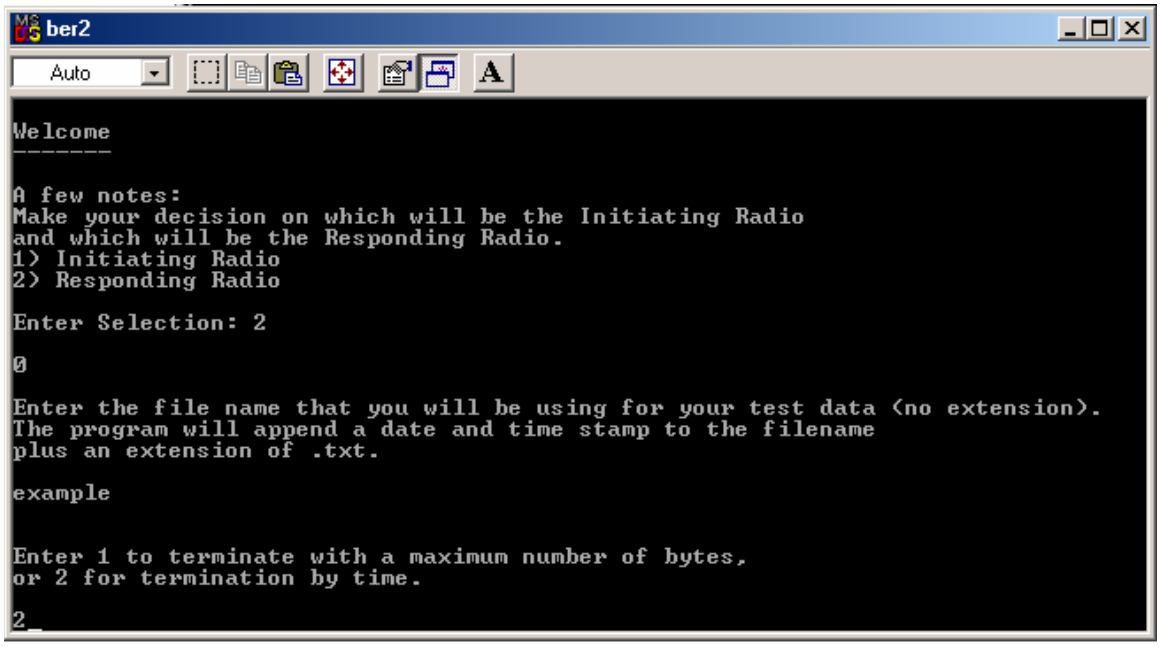

Figure B.2. Another screen shot during program initiation.

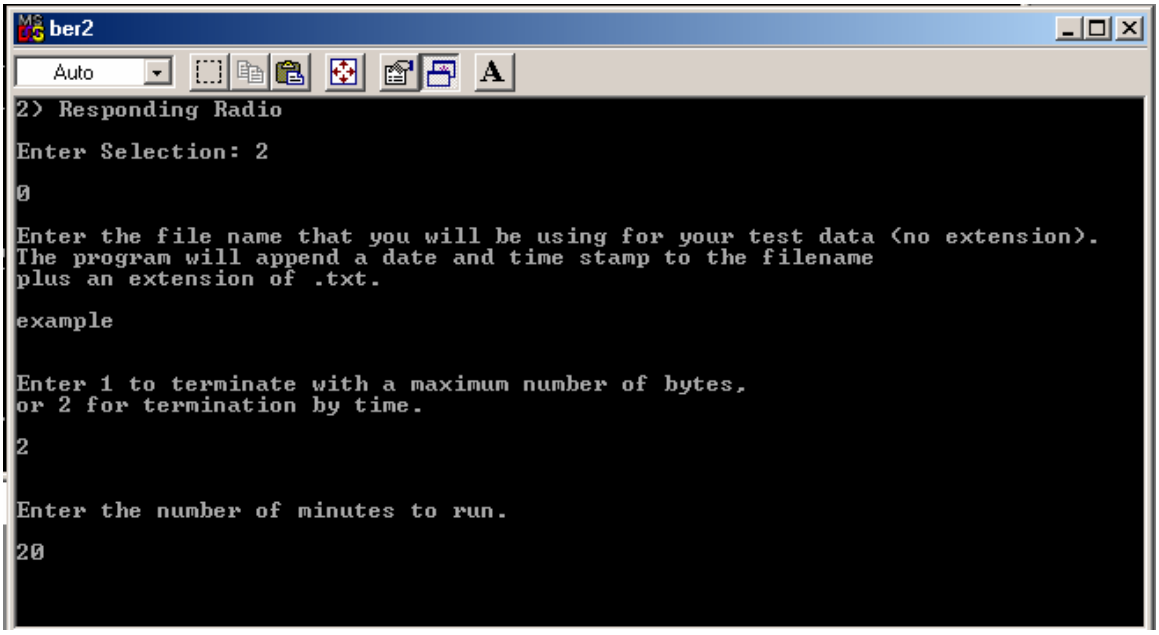

Figure B.3. Another screen shot during program initiation.

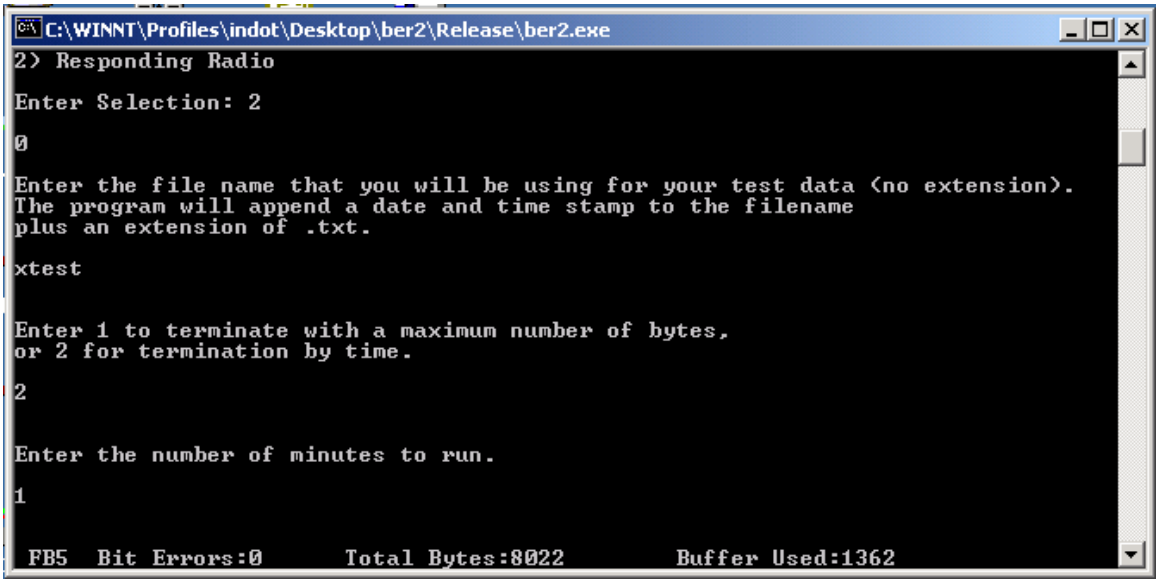

Figure B.4. A screen shot of the data sink window while the program is running. The bottom line contains information of the program as described in Section 2.VIII.

Appendix C – Example Output Files

C.1 - Detailed Error Listing Output File

This is a section of ‘SatRun1GarfieldEvergreen_080302105416_000.txt’

```
======================================================================
08/03/02 10:54:17
Hypothesis Outcomes: 2 25 73 2 51 2 48 2 44 2 43 2 37 200 200 200 200
Winning Hypothesis Contents:  275  276  277  278  279  27A  27B  27C  27D  27E  27F  280  281  282
            Received Vector:  275  276  247  278  279  27A  27B  27C  27D  27E  27F  280  281  282


Bit Errors: 2
Missed Bytes: 0


Current Totals
--------------
Bit Errors: 14
Missed Bytes: 0
Bytes Received with Bit Errors: 10
Bytes Received Correctly: 1242
Packet Losses: 0
Reset Count: 0
======================================================================
08/03/02 10:54:17
Hypothesis Outcomes: 2 24 54 11 51 2 39 2 35 2 34 2 28 200 200 200 200
Winning Hypothesis Contents:  285  286  287  288  289  28A  28B  28C  28D  28E  28F  290  291  292
            Received Vector:  285  286  286 8288  289  28A  28B  28C  28D  28E  28F  290  291  292


Bit Errors: 2
Missed Bytes: 0


Current Totals
--------------
Bit Errors: 16
Missed Bytes: 0
Bytes Received with Bit Errors: 12
Bytes Received Correctly: 1272
Packet Losses: 0
Reset Count: 0
======================================================================
08/03/02 10:54:37
Hypothesis Outcomes: 93 55 0 55 8 61 16 65 25 74 35 64 44 200 200 200 200
Winning Hypothesis Contents: 511A 511B 6B51 6C51 6D51 6E51 6F51 7051 7151 7251 7351 7451 7551 7651
            Received Vector: 511A 511B 6B51 6C51 6D51 6E51 6F51 7051 7151 7251 7351 7451 7551 7651


Bit Errors: 0
Missed Bytes: 159


Current Totals
--------------
Bit Errors: 16
Missed Bytes: 159
Bytes Received with Bit Errors: 12
Bytes Received Correctly: 41497
Packet Losses: 1
Reset Count: 0
======================================================================
08/03/02 10:55:00
Hypothesis Outcomes: 48 56 81 64 64 74 68 69 73 35 58 56 62 200 200 200 200
Winning Hypothesis Contents: A95C A95D A95E A95F A960 A961 A965 A966 A967 A968 A969 A96A A96B A96C
            Received Vector: A95C A95D A952 2979 7179 7264 A965 A140 A967 A968 A969 A97A A971 6D29


Bit Errors: 19
Missed Bytes: 6


Current Totals
```

```
--------------
Bit Errors: 35
Missed Bytes: 165
Bytes Received with Bit Errors: 18
Bytes Received Correctly: 86527
Packet Losses: 2
Reset Count: 0
======================================================================
08/03/02 10:55:00
Hypothesis Outcomes: 60 78 82 60 76 60 53 60 58 54 64 61 55 44 32 45 37
Winning Hypothesis Contents: A964 A965 A966 A967 A968 A969 A96A A96B 6D29 6E29 6F29 7029 7129 7229
            Received Vector: A964 A965 A140 A967 A968 A969 A97A A971 6D29 7EA5 4CB9 7054 C1A9 22A0


Bit Errors: 8
Missed Bytes: 65539


Current Totals
--------------
Bit Errors: 43
Missed Bytes: 65704
Bytes Received with Bit Errors: 22
Bytes Received Correctly: 86538
Packet Losses: 3
Reset Count: 0
======================================================================
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  RESET  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
08/03/02 10:55:00
Current Totals
--------------
Bit Errors: 16
Missed Bytes: 159
Bytes Received with Bit Errors: 12
Bytes Received Correctly: 86523
Packet Losses: 1
Reset Count: 1
Reset Byte Count: 53
Bytes Passed During Reset: 172
Elapsed Time for Reset: 0.180
======================================================================
08/03/02 10:55:20
Hypothesis Outcomes: 6 24 70 50 62 6 44 6 41 6 32 6 28 200 200 200 200
Winning Hypothesis Contents: F232 F233 F234 F235 F236 F237 F238 F239 F23A F23B F23C F23D F23E F23F
            Received Vector: F232 F233 7214 D014 F236 F237 F238 F239 F23A F23B F23C F23D F23E F23F


Bit Errors: 6
Missed Bytes: 0


Current Totals
--------------
Bit Errors: 22
Missed Bytes: 159
Bytes Received with Bit Errors: 16
Bytes Received Correctly: 123649
Packet Losses: 1
Reset Count: 1
======================================================================
```

C.2 - Final Listing Output File

This is ‘SatRun1GarfieldEvergreen_080302105416_final.txt’
The information at the end of the file was typed in by hand for note taking purposes.

```
**********************************************************************
*                            Final Values                            *
*--------------------------------------------------------------------*
* Bit Errors: 2292
* Missed Bytes: 7612
* Bytes Received with Bit Errors: 1127
* Bytes Received Correctly: 2332357
* Packet Losses: 86
* Reset Count: 17
*
* Start Date: 08/03/02    Start Time: 10:54:16
*
* End Date: 08/03/02      End Time: 11:14:16
```

```
* Total Bytes: 2333484
* Timer: 1200
* Throughput (bps): 15556.560000


* Bit Error Rate: 0.000123
* Byte Missed Rate: 0.003251
* Average Missed Bytes Per Packet Loss: 88.511628
**********************************************************************


location


N40.43230
W 086.91248
elevation: 652ft
accuracy 14.7
signal strength: -84-87dbm
```

C.3 – Packet Loss List Output File

This file is ‘SatRun1GarfieldEvergreen_080302105416_ploss.txt’

```
Distribution of Missed Bytes
----------------------------


08/03/02 10:54:37 159
08/03/02 10:55:25 159
08/03/02 10:55:45 27
08/03/02 10:55:51 3
08/03/02 10:55:52 159
08/03/02 10:55:59 159
08/03/02 10:56:30 159
08/03/02 10:56:31 159
08/03/02 10:57:09 147
08/03/02 10:58:10 159
08/03/02 10:58:16 24
08/03/02 10:58:17 72
08/03/02 10:58:19 3
08/03/02 10:58:29 159
08/03/02 10:58:34 159
08/03/02 10:58:37 12
08/03/02 10:58:40 111
08/03/02 10:58:59 141
08/03/02 10:59:01 159
08/03/02 10:59:15 3
08/03/02 10:59:15 132
08/03/02 10:59:20 6
08/03/02 10:59:25 75
08/03/02 10:59:31 3
08/03/02 10:59:56 159
08/03/02 11:00:12 39
08/03/02 11:00:13 159
08/03/02 11:00:45 96
08/03/02 11:00:48 21
08/03/02 11:00:52 159
08/03/02 11:00:53 159
08/03/02 11:01:15 159
08/03/02 11:01:27 159
08/03/02 11:01:32 90
08/03/02 11:01:40 3
08/03/02 11:01:58 3
08/03/02 11:01:59 159
08/03/02 11:02:04 147
08/03/02 11:02:38 159
08/03/02 11:02:42 42
08/03/02 11:02:51 159
08/03/02 11:03:14 63
08/03/02 11:03:32 3
08/03/02 11:04:13 3
08/03/02 11:04:21 3
08/03/02 11:04:39 159
08/03/02 11:04:43 24
08/03/02 11:04:46 36
08/03/02 11:05:27 159
08/03/02 11:05:44 3
08/03/02 11:05:46 159
```

```
08/03/02 11:06:13 9
08/03/02 11:06:25 30
08/03/02 11:06:44 117
08/03/02 11:07:03 159
08/03/02 11:07:13 159
08/03/02 11:07:27 159
08/03/02 11:07:36 3
08/03/02 11:07:36 132
08/03/02 11:08:01 159
08/03/02 11:08:38 90
08/03/02 11:08:56 30
08/03/02 11:09:24 186
08/03/02 11:10:00 39
08/03/02 11:10:34 108
08/03/02 11:10:35 159
08/03/02 11:10:55 3
08/03/02 11:11:04 6
08/03/02 11:11:20 159
08/03/02 11:11:26 3
08/03/02 11:11:36 12
08/03/02 11:11:38 67
08/03/02 11:11:49 159
08/03/02 11:11:49 75
08/03/02 11:12:21 3
08/03/02 11:12:56 72
08/03/02 11:13:00 6
08/03/02 11:13:00 3
08/03/02 11:13:01 357
08/03/02 11:13:02 3
08/03/02 11:13:35 93
08/03/02 11:13:40 21
08/03/02 11:13:55 102
08/03/02 11:14:04 24
08/03/02 11:14:10 48
08/03/02 11:14:10 24
```

C.4 – Reset List Output File

This is ‘SatRun1GarfieldEvergreen_080302105416_reset.txt’

```
Distribution of Resets
---------------------------


08/03/02 10:55:00 172      53       0.180
08/03/02 10:55:20 28       16       0.000
08/03/02 10:56:41 36       32       0.000
08/03/02 10:58:46 58       22       0.000
08/03/02 11:00:18 130      35       0.000
08/03/02 11:00:46 150      98       0.000
08/03/02 11:01:19 46       40       0.000
08/03/02 11:05:00 28       22       0.000
08/03/02 11:06:12 84       61       0.000
08/03/02 11:07:50 68       57       0.000
08/03/02 11:08:47 38       16       0.000
08/03/02 11:09:09 24       16       0.000
08/03/02 11:10:37 126      18       0.000
08/03/02 11:13:07 32       16       0.000
08/03/02 11:13:19 148      26       0.000
08/03/02 11:13:40 20       16       0.000
08/03/02 11:14:06 40       29       0.000
```

Appendix D – Testbed Photographs

Figure D-1: (Top) View of three MDS 9810 spread spectrum radios in the WCRLab in the MSEE building. These radios are connected to three antennas mounted on masts on the roof of the building. (Bottom) View of computers used for radio control and RLTS software.

Figure D-2: (Top) View of MSEE building showing three testbed antennas over the roof-line. (Bottom) View of testbed antennas from on the MSEE roof.

Figure D-3: (Left) Group members setting up the portable master station outside of the MSEE building for a JTRP radio field test. (Right) Portable slave station at the corner of Hayes and Meridian in West Lafayette.

Figure D-4: Portable slave station with Yagi at two West Lafayette locations.