UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

FEASIBILITY OF A COGNITIVE EXTENSION TO EXISTING 802.11B WIRELESS DEVICES

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

WILLIAM JUSTIN BARNES

Norman, Oklahoma
2009

FEASIBILITY OF A COGNITIVE EXTENSION TO EXISTING
802.11B WIRELESS DEVICES

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

____
Dr. Hazem Refai, Chair

____
Dr. James J. Sluss, Jr.

____
Dr. John Fagan

____
Dr. Samuel Cheng

____
Dr. William Ray

Acknowledgments

I would like to thank the members of my doctoral committee, Dr. William Ray, Dr. James Sluss, Dr. John Fagan, and Dr. Samuel Cheng for their advice and suggestions in the early stages of this work. I would also like to thank my advisor, Dr. Hazem Refai, for his direction and the help he provided in the fulfillment of this work; it would not have been possible without him. Lastly I would like to thank my family, particularly my wife Allison, for their constant support and understanding throughout this lengthy process.

Table of Contents

List of Tables

List of Figures

Abstract

Cognitive radio presents a means of altering the communication method of a wireless device based on channel conditions and the intended receiving device. However, the design of such a radio is very complicated as it must consider the possibility of multiple forms of modulation, differing transmit frequencies and symbol rates, and the accompany changes to other training procedures such as synchronization. This work proposes that in some cases a simpler, more cost-effective approach can be taken, that builds upon the architecture of existing wireless devices forming a new radio with cognitive capabilities. This approach allows the base device to perform all base-band and MAC-related functions with minimal or no negative effects due to the extension. As changes in modulation type are much more complex, the analysis in this work is restricted to systems wanting to intelligently alter their transmit frequency or power, such as the 802.22 standard. Because of the extensive investment that has already been made in 802.11 technology, 802.11b chipsets and APs are very inexpensive. Therefore a frequency conversion extension was designed and tested as the fixed architecture to enable signal conversion of an 802.11b signal. Cognitive functionalities could be added with little modification to the proposed design in this work.

The overall goal of this work is to achieve throughput and packet loss results comparable to the base design at the converted frequency of approximately 1.7 GHz. The successful conversion with a fixed design proves the concept feasible, as the only additional requirement is to interface a cognitive subsystem with a configurable architecture employing the same design as the fixed architecture. The nodes under test were isolated in an anechoic chamber to prevent interference from nearby networks. A program called *IxChariot* is used to experimentally conduct network performance tests to confirm that the extended device operates nearly identically to a normal 802.11b radio. Tests were performed for one-hop and two-hop scenarios collecting throughput and packet loss statistics. A number of undesirable effects such as increased switching delay time are also examined as well as their impact on the MAC and physical layer of the base device. The results of testing established the feasibility of a cognitive extension with no perceivable throughput/packet loss degradation for reasonable switching delays. Analysis of poor switching delay performance and 802.11g is also presented to illustrate the additional design constraints these challenges present.

1. Introduction

Cognitive radio has been the focus of much research in recent years as it provides a means by which a radio can automatically adjust to meet spectrum and communication method requirements. The 802.22 standard, for example, makes use of cognitive radio to intelligently share unused spectrum in the television frequency bands. There are also situations in which it may be necessary to facilitate communication between two or more wireless devices inherently incapable of communicating, or to modify the transmission characteristics of an established link. These characteristics include cross-platform communication, noise/interference reduction, spectrum utilization, and improved security. An extension to a wireless device can be designed that essentially acts like a receiver/transmitter. It decodes the information from the transmitted signal and encodes this information into a new signal with the desired properties before relaying the signal. These types of converter units have existed for years, but have been strictly limited to their narrow application. Herein, robust converter design is researched in detail to determine the feasibility of a cognitive extension to existing wireless devices. Background research for existing parameter estimation for use in protocol conversion is provided in Chapter 2. However, the central focus of this work is placed on the development and evaluation of a cross-layer conversion architecture, culminating in a cross-layer extension to existing 802.11b devices.

Protocol conversion involves turning one communication protocol into another, such as 802.11a to 802.11b, for example. To improve upon the robustness of the application-specific converters, the platform should be capable of identifying the transmission parameters of the signal output from the wireless device. It can then decipher the data of multiple protocols, which can be relayed using the desired wireless protocol. The process of decoding an unknown signal is often referred to as blind signal demodulation. The primary parameters needed for signal demodulation are the carrier frequency, symbol rate, and modulation. Methods of estimating each of these parameters are covered in Chapter 2. Once the wireless characteristics have been determined, a means of implementing various protocols is needed. Software-defined radio (SDR) provides a flexible platform for reconfiguring the architecture of a transceiver to enable communication under a wide variety of scenarios, including different transmission power, modulation, carrier frequency, bandwidth, channel and error correction coding, equalization, diversity, MAC, and routing protocols.

Cognitive radio came about as a logical extension of SDR. The origins of SDR can be traced back to the Department of Defense with the creation of the SPEAKeasy system[1]. The purpose of the SPEAKeasy system was to allow communication from 2 to 2000 MHz with programmable waveforms and enhanced security. However, waveform generation could be altered by changing the waveform modules deployed in the unit [1]. These drawbacks spawned SPEAKeasy Phase II, in which these problems were addressed by focusing on “off the shelf” components and standards, thereby augmenting the DSP core [2]. The development of the next generation military radio, JRTS [3], is underway. The primary advantage of JRTS is in its adherence to the Software Compliance Architecture (SCA) standard [4], which allows for portability of software components between SDRs. More recently, commercial SDR applications have dominated the market as multi-purpose radios are becoming more and more prevalent. The debate on commercial SDR architecture is still ongoing, with some advocating DSP-based designs [5, 6], and others FPGA-based[7-9] or DSP-FPGA combination architectures [10]. The applications of SDR, however, have been fairly restricted in the RF community. Military motivations are largely limited to jamming/interference avoidance [11, 12] and electronic surveillance [13, 14]. An explosion of renewed commercial interest in SDR platforms was brought about by the advent of cognitive radio, producing a self-configurable SDR. This development has given rise to new research areas outside the scope of military use, most notably dynamic/shared spectrum access [15-18]. Specific building blocks of SDRs are covered in the Chapter 2 to give the reader an understanding of the basic operations and the inherent segregation of physical layer processes.

Cross-layer conversion is defined as the translation of an existing wireless protocol to operate in a new frequency range without exhibiting significant performance degradation. Because the main goal is to translate operation of a wireless device to different operating frequencies, this type of converter will also be referred to as a frequency converter. To avoid the excessive delays and stalled throughput of existing conversion technology, this process should occur without demodulating the original signal. At first this may appear to be solved simply by the addition of a mixer, amplifier, and appropriate filtering. However, since the converter does not have access to the control signal of the wireless device, it must determine autonomously when to enable the transmitting and receiving paths. Because the signal is not digitized at any point in the converter, any noise generated in the converter is also of critical importance, as it is added to the noise generated in the wireless device. For modest noise levels it is unlikely that the noise will significantly impact the transmit path, because any propagated noise is likely to reach the noise floor prior to reception. However, noise generated in the receive path can hinder the performance of the link as the bit-error rate is directly related to the signal-to-noise ratio of the received signal. Poor noise figure of the converter could even result in dropped packets, particularly for weak signals.

This process is referred to as cross-layer conversion because it should enable proper functionality of the wireless device across all layers of the OSI stack. Of particular importance is the performance of the MAC when a frequency converter is employed. While simple MAC protocols like ALOHA or slotted ALOHA should perform reasonably despite the converter design, more popular protocols like CSMA require the spectrum monitoring to be extended to the new frequency range. Delays during switching to the transmit path may result in other units transmitting when they should not, creating a small window of time in which a hidden node problem exists. The conversion process and design considerations are discussed in detail in future chapters.

The research carried out in this work resulted in a number of significant contributions to this field of study including the following:

- Detailed theoretical analysis of the impact a truncated 802.11b preamble has on the MAC operation and performance including changes in the inter-frame spacing and ACK-timeout.
- Evaluated two different extension architecture: one that enables a weak preamble during the switching interval (favoring the MAC) and another that truncates the preamble during switching (favoring the PHY layer).

- ⦿ Electromagnetically isolated testing confirming the viability of the wireless extension and verifying that base-band MAC and PHY layers are not significantly affected by said extension provided that the switching delay is reasonable thereby proving the feasibility of a cognitive wireless extension.
- ⦿ A detailed examination of delays in switching between transmit and receive paths, including the modified packet structure, its impact on network performance characteristics (packet loss, throughput, etc.), and justification for the selection of a truncated preamble approach.
- ⦿ Examination of the additional design constraints and limitations that an 802.11g network would introduce, including aspects such as the shortened preamble and the peak-to-average power ratio problems introduced by orthogonal frequency division multiplexing.

The organization of this dissertation can be broken down as follows. Chapter 2 provides the background on blind signal analysis, including a description of the overall architecture and capabilities of a standard SDR platform. Chapter 3 discusses the background involved with frequency converter design culminating in a preliminary wired design. Chapter 4 describes the 802.11b/g physical and MAC layers in detail. Chapter 5 introduces the wireless frequency converter design and Chapter 6 discusses the corresponding impact on the MAC and physical layer of the base device. Chapter 7 presents network performance tests to validate the design and examine the problems it presents. Lastly, a brief conclusion of this work is provided in Chapter 8.

2. Parameter Estimation

2.1 Software Defined Radio Components

The architecture of the Lyrtech© SDR SFF DP is a good modular representation of SDR as a whole, and is therefore used as an example to describe the various functions performed by software defined radios. A picture and behavioral description of the platform can be found in Figure 1. The SDR is composed of three boards: an RF board, a data conversion board, and a base-band/communication board. The RF board consists of all transmit and receive hardware used to convert the signal to or from IF, including RF amplifiers, mixers, controllable oscillators, and selectable filters. The data conversion board performs the sampling and conversion at the IF stage to either an analog (transmitting) or digital (receiving) signal. For this SDR specifically, the base-band processing/communication board is comprised of an FPGA for base-band hardware realization, and a DSP to facilitate signal generation and communication from the host PC to the board.

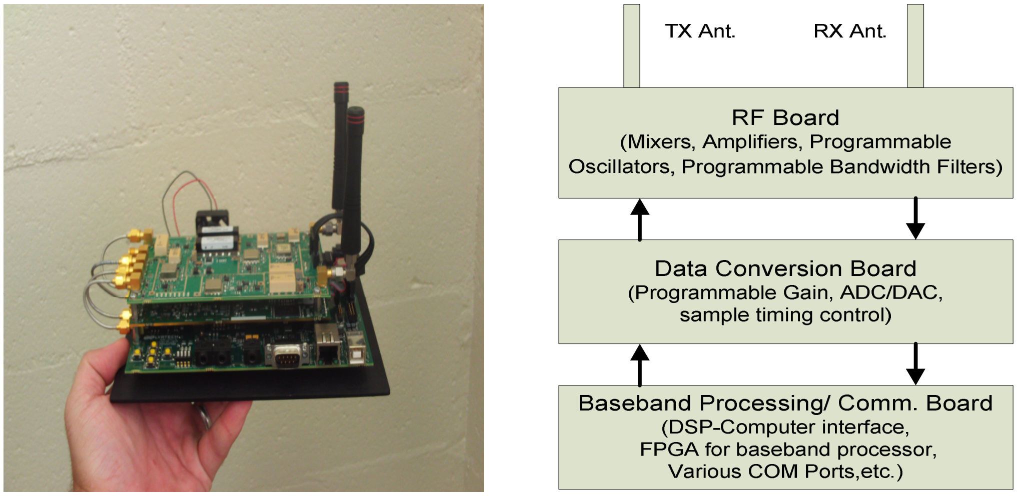

TX Ant.
RX Ant.
RF Board
(Mixers, Amplifiers, Programmable
Oscillators, Programmable Bandwidth Filters)
Data Conversion Board
(Programmable Gain, ADC/DAC,
sample timing control)
Baseband Processing/ Comm. Board
(DSP-Computer interface,
FPGA for baseband processor,
Various COM Ports,etc.)

Figure 1. Lyrtech SDR SFF DP (left) and behavioral description (right).

2.1.1 The RF Board

The platform achieves full duplex by maintaining completely independent transmit and receive paths. A frequency range of 200-960MHz transmitting and 30-960 MHz receiving is supported by the SDR, however the antennas are optimized for 450-490 MHz. An RF filter bandwidth of 5 MHz or 20 MHz can be selected through software. Figure 2 is a description of the RF board hardware. The received signal is sent to a filtered and up-converted to a fixed center frequency of 1575 MHz. Local oscillators are controlled through software to ensure proper translation to this frequency. The signal is then down-converted to a center frequency of 300 MHz. One of the two possible bandwidths is next used to filter the signal appropriately before the signal is down-converted to IF by a fixed oscillator of 330 MHz. The transmitter section of the RF board is composed of a quadrature mixer driven by a local oscillator (523-876 MHz) that has a divide-by-2 prescaler to achieve the lower frequencies (262-438 MHz). Although it is not explicitly stated in the manual, it is assumed that the lower image frequency covers

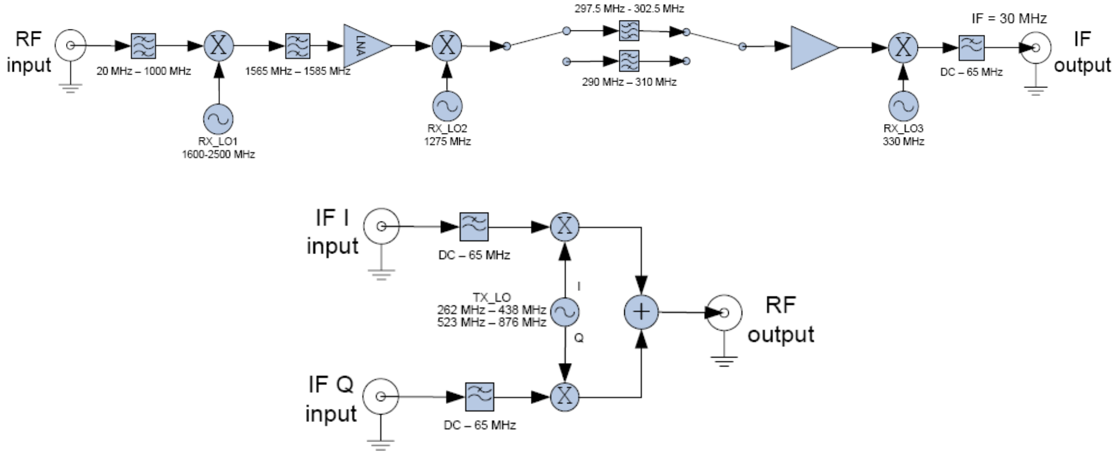


Figure 2. RF receive (top) and transmit (bottom) architecture [19].

the gap in frequencies when 30 MHz is mixed with 523 MHz, and the upper image frequency when the local oscillator is at 438 MHz.

2.1.2 The Data Conversion Board

The data conversion module is responsible for maintaining synchronous behavior between the ADC, DAC, and RF board using a series of PLLs for clock management. After a received signal has been down-converted to IF, it is adjusted by a programmable gain/attenuation prior to digitization. A 14-bit ADC sampling at 125 MSPS sends the received digital signal to an FPGA that handles communication between the data conversion and base-band processing board. Conversely, a transmitted signal is fed to the DAC by the aforementioned FPGA. The DAC itself is 16-bit and outputs an analog sample at a rate of 500 MSPS. The DAC has built-in programmable gain amplifiers that can be configured through software. Figure 3 shows a functional block diagram for the data conversion board.

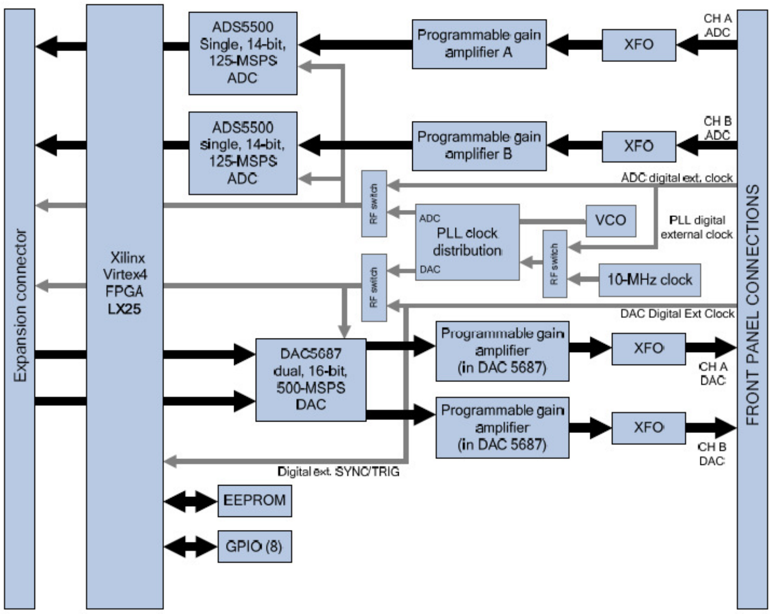

Expansion connector
Xilinx Virtex4 FPGA LX25
ADS5500 Single, 14-bit, 125-MSPS ADC
ADS5500 single, 14-bit, 125-MSPS ADC
Programmable gain amplifier A
Programmable gain amplifier B
XFO
XFO
CH A ADC
CH B ADC
ADC digital ext. clock
RF switch
RF switch
RF switch
ADC
PLL clock distribution
DAC
VCO
PLL digital external clock
10-MHz clock
DAC Digital Ext Clock
DAC5687 dual, 16-bit, 500-MSPS DAC
Programmable gain amplifier (in DAC 5687)
Programmable gain amplifier (in DAC 5687)
XFO
XFO
CH A DAC
CH B DAC
Digital ext. SYNC/TRIG
EEPROM
GPIO (8)
FRONT PANEL CONNECTIONS

Figure 3. Data conversion module architecture for Lyrtech SDR platform [19].

2.1.3 Base-band Processing Software Interface

The architecture of the base-band processing board is quite complex and not the primary focus of this dissertation. Consequently, this section emphasizes the interaction between the software and the DSP/FPGA within the base-band processing board. Most hardware configuration goals can be achieved in the Simulink environment. The SDR SFF DP installation software provides a Lyrtech blockset compatible with Simulink. This blockset can be used for visual programming of the transmit/receive settings of the RF and data conversion hardware as well as specifying the outputs to and from base-band. The actual base-band processing occurs within a Xilinx FPGA. Base-band operations are controlled by a Xilinx blockset within the Simulink environment. The overall structure of a program to be implemented on the SDR can be found in Figure 4, and an example of FPGA programming can be found in Figure 5.

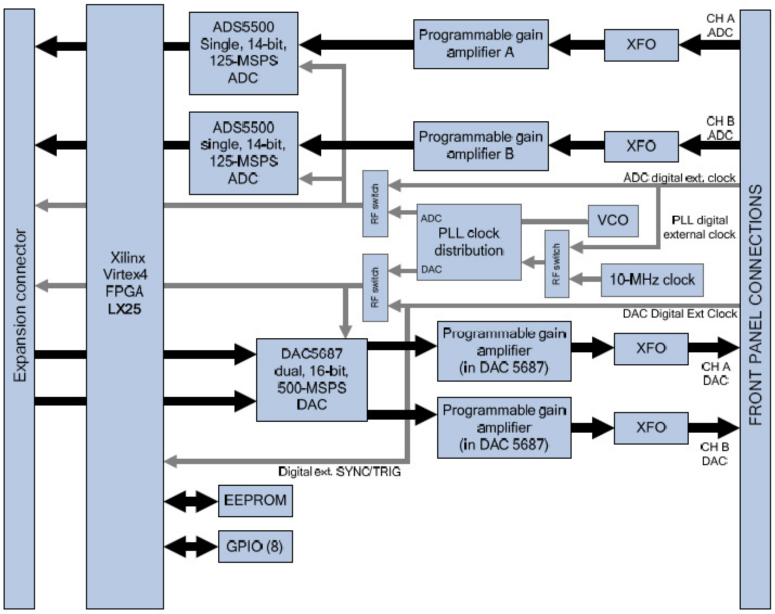

Expansion connector
Xilinx Virtex4 FPGA LX25
ADS5500 Single, 14-bit, 125-MSPS ADC
ADS5500 single, 14-bit, 125-MSPS ADC
Programmable gain amplifier A
Programmable gain amplifier B
XFO
XFO
CH A ADC
CH B ADC
ADC digital ext. clock
RF switch
RF switch
RF switch
ADC
PLL clock distribution
DAC
VCO
PLL digital external clock
10-MHz clock
DAC Digital Ext Clock
DAC5687 dual, 16-bit, 500-MSPS DAC
Programmable gain amplifier (in DAC 5687)
Programmable gain amplifier (in DAC 5687)
XFO
XFO
CH A DAC
CH B DAC
Digital ext. SYNC/TRIG
EEPROM
GPIO (8)
FRONT PANEL CONNECTIONS

Figure 4. Simulink-SDR programming structure. The red areas involve communication between the SDR and the host computer (i.e. signal generator/ signal scope), the green area enclosed by gateways is the FPGA base-band processing, and the gray area is the system configuration where RF and sampling parameters can be set.

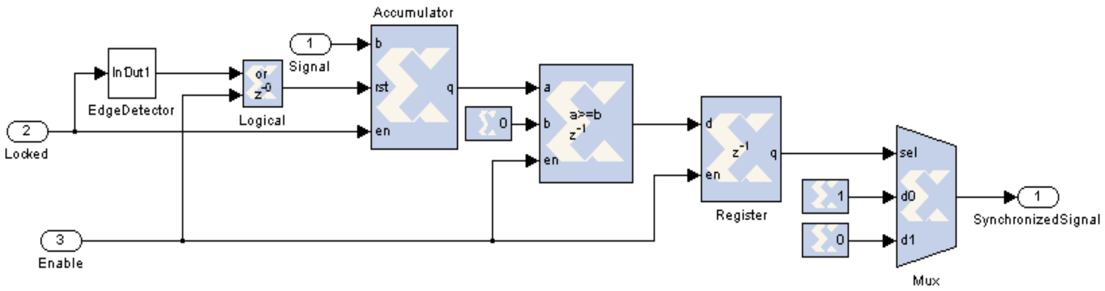


Figure 5. Example of FPGA programming using Simulink-based Xilinx© block-set.

Any real-time communications between the host computer and the SDR are made through the DSP. Direct programming of the TI DSP present on the board can be made using normal Simulink blocks. As a consequence, signal generation and live feedback to the host computer is managed by the DSP. It should be noted that making live updates of transmission characteristics (carrier frequency, gain, etc.) cannot be achieved directly through the Simulink interface and require manipulation of the built-in API functions of the radio.

2.2 Modulation Identification

A detailed survey of various methods of modulation classification was conducted by Dobre *et al.* [20]. This section discusses the basics of how the various types of modulation analysis are performed, and the disadvantages associated with each technique. A number of unique methods for modulation identification exist [21, 22]; however, the vast majority are comprised of likelihood methods, wavelet transform-based methods, and cumulant-based methods. The discussion begins with a brief overview of likelihood techniques. Detailed examination of feature-based techniques including the use of cumulants and the wavelet transform are discussed more thoroughly, as they represent the most recent trends in modulation identification.

2.2.1 Maximum Likelihood-Based Modulation Identification

Maximum likelihood based modulation identification [23-26] comprised the majority of modulation classifiers throughout the 1990s and early 2000s. As these techniques had already found numerous applications in classification problems, it seemed a logical choice place to begin. The basic principle behind maximum likelihood (ML) classification is derived from the relationship between observed measurements and their probability distribution functions (pdfs). Assuming a static pdf and *N* observations, the likelihood function can be expressed [27] as:

$$L(\theta) = \prod_{i=1}^{N} f(x_i, \theta) \quad , \qquad (1)$$

where x_i is the i^{th} observed measurement and θ is the parameter of interest. The likelihood function can be maximized in the standard way by taking the derivative with respect to θ, setting it equal to zero, and solving for θ. The form in (1) is useful in determining estimates for a numerical parameter such as the mean or variance of the pdf, but it can also be applied to modulation analysis.

In most cases the likelihood function is a piecewise function of the modulation type, with each modulation type represented by a separate piece. As an example, consider an MPSK signal where the objective is to determine the number of phases used (i.e. the modulation type). It has been shown [24] that the joint pdf of the real and imaginary components of a PSK signal can be expressed by:

$$f_M(x, y) = \frac{e^{\frac{-x^2-y^2+2\rho}{2}}}{2M\pi} \sum_{i=1}^{M-1} e^{\sqrt{2\rho}\left(x\cos\theta_i + y\sin\theta_i\right)}, \qquad (2)$$

where ρ is the SNR, θ_i is the phase of the i^{th} symbol, and *(x, y)* are the real and imaginary components respectively. By obtaining the likelihood function from this equation and taking the natural logarithm we obtain a log-likelihood function. Solving for the maximum log-likelihood function with respect to *M* yields likelihood values that are dependent on the SNR as well as *L* real and imaginary received components [24].

$$L_{\log}(M) = \sum_{i=1}^{L-1} \lambda_M(x_i, y_i),$$

$$\lambda_i = \begin{cases} \sqrt{2\rho}x_i & M = 2 \\ \ln[\cosh(\sqrt{\rho}x_i)\cosh(\sqrt{\rho}y_i)] & M = 4 \\ \ln\left[\frac{1}{2}\cosh(\alpha\sqrt{2\rho}x_i)\cosh(\beta\sqrt{2\rho}y_i) + \frac{1}{2}\cosh(\alpha\sqrt{2\rho}x_i)\cosh(\beta\sqrt{2\rho}y_i)\right] & M = 8 \end{cases},$$

where $\alpha = \sqrt{\frac{\sqrt{2}+1}{2\sqrt{2}}}, \qquad \beta = \sqrt{\frac{\sqrt{2}-1}{2\sqrt{2}}}.$ (3)

While values for the SNR must be estimated, under certain conditions (such as high or low SNR) simplified equations can be used in which the SNR is negligible. Thus the likelihood function can be computed from a set of observed real and imaginary components, and the value of *M* for which it is maximized can be computed. In most cases, variations in the estimators used in different algorithms can be traced back to different signal models/assumptions based on things such as synchronous/asynchronous, coherent/non-coherent detection [25, 26].

Several more robust methods for deriving maximum likelihood classifiers were discussed by Boiteau and Martret [23]. The framework for this method is based on the widely popular average likelihood classifier (ALC):

$$L_{avg}\left(\overline{\theta}_C\right) = E_{\theta_C}\left\{ e^{\frac{1}{N_0}\int_0^N \mathrm{Re}\,al\{r(t)s^*(t,\overline{\theta}_C)\}} \right\}, \tag{4}$$

where *N₀* is the noise floor, θ_C is the set of parameters defining the modulation class *C*, *r(t)* is the received signal, and *s(t,θ_C)* is the uncorrupted received signal. Expressing the exponential with its power series approximation gives way to the general maximum likelihood classifier (GMLC) [23]:

$$L_{avg}\left(\overline{\theta}_C\right) = 1 + \sum_{n=1}^{\infty} \frac{\lambda_{nC}}{n!2N_0}, \tag{5}$$

$$\lambda_{nC} = E_{\theta_C}\left[\int_0^N \left\{ r(t)s^*(t) + r^*(t)s(t) \right\} dt \right]^n. \tag{6}$$

Subspace-based methods can be used to further simplify this algorithm by eliminating the exponent. It should be noted that detailed derivations of the AMLC and GMLC classifiers is beyond the scope of this dissertation, but is described in detail by both Dobre *et al.* and Boiteau and Martret [20, 23]. The GMLC has the advantage of not having base-band constraints. While the maximization of the ALC and GLC results in good performance with respect to SNR, it is plagued by the computational burden and design complexity involved with most ML estimators. Thus more simplified methods have been created that are based on extracting features of a signal to determine the modulation employed.

2.2.2 Direct Estimation of the Number of Amplitudes, Phases, and Frequencies

Perhaps the simplest method of performing narrowband modulation identification is the estimation of the frequencies, amplitudes, and phases present in the received signal over time. Amplitude changes restrict possible modulation schemes to ASK or QAM, while frequency changes are an indication of some form of FSK. Similarly phase changes exclude FSK as a potential candidate and limit modulation schemes to QAM or some derivative of PSK. Simple frequency estimators can be used to establish the presence of an FSK signal [28]. Knowledge of these parameters is sufficient information for modulation classification for PSK, QAM, and FSK signals.

A basic estimation of frequency, amplitude, and phase was performed by Rashid *et al*. [29]. Samples of the received signal over a length of *N* symbols are saved in memory. Amplitude estimates are taken of the symbols. Although the estimation algorithm itself is not discussed, a simple average of a large number of samples within a symbol can provide a decent estimate of the amplitude. Next the saved samples are normalized with respect to the estimated amplitude and sent to phase and frequency estimators. Frequency estimation is covered thoroughly in Section 2.3. It should be noted that the phase is not estimated, but rather the number of phases present is estimated from a histogram of phase changes. As the modulation schemes of interest can be fully discriminated from these estimates, the discussion of the classification process is now concluded.

One very distinct advantage of this type of identification is that it is directly applicable to real-valued signals. A number of the classifiers developed, including ML and wavelet-based identifiers, rely on complex signal representation. They are able to test their methods because most signal generators provide I and Q channel outputs. In reality, if the frequency is unknown at the receiver, such a representation is difficult to obtain. Despite this advantage, the method discussed in this section is very rudimentary and rarely used in practice. There is significant redundancy in the hardware of the modulation classifier, and a basic receiver decreases the overall design efficiency. Additionally, the phase, frequency and amplitudes are all examined independently using different algorithms for each parameter. Such an architecture may increase the time to identify (TTI) the modulation, as the identification process is broken into stages. The experimental SNR performance of direct parameter estimation modulation identifiers is significantly below that of wavelet transform and cumulant-based techniques discussed in the next sections. It should be noted, however, that amplitude estimation is often used, as it is an essential step in distinguishing QAM from PSK signals. This is frequently employed in hierarchical-based modulation classifiers, in which potential modulations are eliminated over the course of several stages [30-32].

2.2.3 Wavelet Transform-Based Techniques

A number of modulation classifiers make use of the wavelet transform [33-38]. This section is a description of the wavelet transform approach to modulation identification developed by Ho *et al.* [35]. Wavelets are a powerful tool used to perform time/frequency analysis. Scales are used as a means of controlling the length of the wavelet while maintaining its other properties. Choosing a small scale produces a narrow wavelet with good time localization, but sacrifies frequency localization. Since the primary focus is to determine abrupt changes in the time domain signal to obtain modulation information, a small scale value of a = 2 was chosen. Discrete representation of the wavelet transform is given by:

$$WT(a,n) = \frac{1}{\sqrt{a}} \sum_{k} s(k) \psi\left(\frac{k-a}{n}\right) , \tag{7}$$

$$\frac{1}{\sqrt{a}} \psi\left(\frac{k}{a}\right) = \begin{cases} 1/\sqrt{a} & k = -a/2 \ldots -1 \\ -1/\sqrt{a} & k = 0,1 \ldots\ldots. a/2 \\ 0 & elsewhere \end{cases} , \tag{8}$$

where a Haar wavelet was chosen for simplicity. The signal model described by Ho *et al.* [34] was assumed to have the complex form:

$$s(k) = \sqrt{S} e^{j(w_c k + \varphi)} , \tag{9}$$

where S is the signal power, w_c is the received signal power, and φ is the received signal phase. The magnitude of the wavelet transform for PSK, QAM, and FSK modulated signals can be expressed [34] as:

$$\left| WT(a,n) \right| = \frac{4\sqrt{S}}{w_c \sqrt{a}} \sin\left(\frac{w_c a}{4}\right) \sin\left(\frac{w_c a}{4} + \frac{2\pi k}{M}\right) \qquad k = 1,2 \ldots M-1 \qquad \text{(PSK/QAM)} \tag{10}$$

$$\left| WT(a,n) \right| = \frac{4\sqrt{S}}{(w_c + w_i)\sqrt{a}} \sin^2\left(\frac{(w_c + w_i)a}{4}\right) . \qquad \text{(FSK)} \tag{11}$$

The magnitude of the wavelet transform has distinct behavior for PSK and FSK signals. When a complex representation of the signal is used, the behavior is constant between symbol changes for PSK. In the case of FSK, it is a constant related to the transmit frequency. For example, if MFSK is employed there exist M possible constant levels between symbol changes. Ho realized the variance of the wavelet transform magnitude was therefore sufficient to distinguish between FSK and PSK signals. However, inter-modulation classification is also possible. When the wavelet is passed over a region of transition, a spike occurs in the wavelet transform magnitude. This spike is a different level depending on the phase jump that occurs. Thus by grouping phase jumps of nearly identical value through histograms or other means, the number of groups is related to *M*. One spike level is possible for PSK (± 180 degrees), two are possible for QPSK (± 90, ± 180), four are possible for 8PSK (±45, ±135, ± 90, ± 180), and so forth. As previously mentioned, the level of FSK employed is determined by the number of levels seen between symbol changes.

One advantage of wavelet-based techniques is that they are capable of classifying signals that are traditionally difficult, such as CDMA [36, 37] and GMSK [33, 36, 37]. Wavelet-based techniques are also very easily implemented when simple wavelets (i.e. Haar) are used. However, there are a number of significant drawbacks to the wavelet transform method. As previously stated, complex representation is very difficult to achieve in real world applications without knowledge of the carrier frequency. One of the major purported advantages of this method is its claim to being “frequency blind”. While most other algorithms are completely invalidated when this assumption is violated, the wavelet transform methods can still operate at significantly decreased accuracy. Revising the signal model to be representative of a real-valued signal yields:

$$s(k) = \alpha_i \cos(w_c k + \varphi) + \beta_i \sin(w_c k + \varphi) \, . \tag{12}$$

Limiting the scale to a = 2 in (7) causes the wavelet transform (WT) to be a scaled approximation of the product of the derivative of *s(k)* and the sampling time. Since the signal is assumed to be over-sampled, $T_s w_c$ <<1. The resulting WT magnitude ($|WT|$) between symbol changes is given by a heavily attenuated sinusoid:

$$WT = k_1 \cos(w_c k + \varphi) + k_2 \sin(w_c k + \varphi) \qquad k_1, k_2 << \alpha_i, \beta_i \, . \tag{13}$$

The different signal model results in an attenuated sinusoidal behavior of the WT between symbols rather than a constant. For PSK (Figure 6) this results in spikes separating attenuated sinusoids, and for FSK this results in sinusoids of differing amplitudes for each frequency present. Over-sampling is not always realizable. Consequently it is important to examine the impact on the magnitude of the wavelet transform as the sampling rate is reduced. Conceptually, it is clear that down-sampling will result in an increase in the distance between adjacent sample points. As a consequence |*WT*| will be given by the magnitude of a sinusoid with increased amplitude.

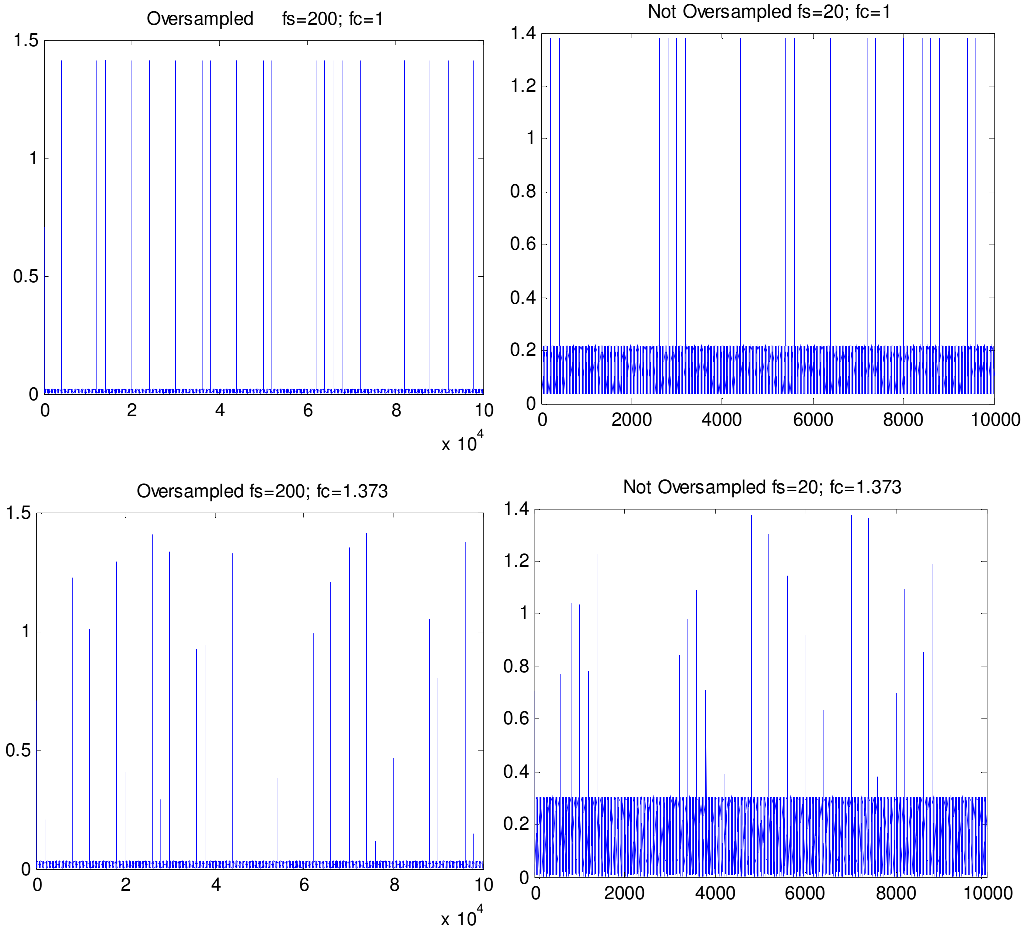


Figure 6. Four scenarios that occur using the WT approach on a BPSK signal.

As this amplitude grows with decreasing sampling rates, the sinusoidal floor causes symbol changes to become increasingly difficult to detect in a PSK signal. FSK symbol changes will become easier to detect as the amplitude difference between adjacent symbols increases. However, the algorithm proposed in Chapter 3 uses a frequency estimator for determining FSK, and consequently we limit our discussion to PSK behavior.

Another significant problem arises when the sampling frequency is not divisible by the carrier frequency. Phase changes are not captured at the moment of transition, and this effect propagates, causing multiple values for |WT| despite only a few different phase changes. It has been suggested that this problem can be alleviated by interpolating |WT| at the time of transition, but the authors herein fail to see how this is achievable. At a minimum, this problem will complicate the simple, elegant architecture that was developed. Simulation for an uncorrupted BPSK signal under four possible test scenarios is given in Figure 6. The first scenario illustrates how the wavelet transform is meant to look under ideal conditions, where the signal has been over-sampled and f_s/f_c is given by an integer. The rise in the floor of the wavelet transform magnitude is apparent when the sampling rate is reduced. When the sampling rate is not evenly divisible by the transmit frequency, different magnitudes are present despite having only one potential phase change. Decreasing the sampling rate further clearly complicates the modulation identification process.

2.2.4 Cumulant-Based Modulation Classification

One very popular technique of distinguishing modulation types is estimation of the cumulants of the base-band signal. Cumulants are powerful diagnostically as they differ significantly depending on the modulation employed and the cumulant selected. Consequently, a frequently-used modulation identification technique is to derive an estimate for a series of cumulants, and then use the estimates to distinguish modulation schemes. The n^{th} cumulants of a random variable X are the values κ_n satisfying the equation:

$$c(t) = \log E\{e^{Xt}\} = \sum_{n=1}^{\infty} \kappa_n \frac{t^n}{n!}\,. \tag{14}$$

However, joint cumulants are more often used in modulation identification:

$$cum\{X_1, X_2, ... X_n\} = \sum_{k} (|k|-1)(-1)^{|k|-1} \prod_{k \in G} E\left\{\prod_{i \in G} X_i\right\}, \tag{15}$$

where k spans $\{1,2 \ldots.\ n\}$, and G are the groups of length k within the span $\{1,\ldots n\}$. A simpler interpretation of this equation is that the term in front of the summation dictates the coefficient, and the product terms multiply all possible expected value products of length k by the remaining expected value terms in the partition. Thus the first, second, and third joint cumulants can be expressed respectively as:

$$cum\{X_1\} = E\{X_1\} \tag{16}$$

$$cum\{X_1, X_2\} = E\{X_1 X_2\} - E\{X_1\}E\{X_2\} \tag{17}$$

$$\begin{aligned} cum\{X_1, X_2, X_3\} = E\{X_1 X_2 X_3\} - E\{X_1\}E\{X_2 X_3\} - E\{X_2\}E\{X_1 X_3\} - \\ E\{X_3\}E\{X_1 X_2\} + 2E\{X_1\}E\{X_2\}E\{X_3\} \end{aligned}\,. \tag{18}$$

The random variables used for modulation classification are the base-band signal and its conjugate. It should be noted that the signal model at base-band is zero mean, which combined with the relationship between central moments and the cumulant, dramatically reduces the complexity in developing cumulant estimators. Typical notation used for modulation analysis revolves around joint cumulants. If *r(t)* is the received signal and *s(t)* is the signal without noise corruption, a joint cumulant estimate of *s(t)* can be derived from the cumulant estimate of *r(t)* yielding*:*

$$C_{n,m} = cum(r_1, r_2, r_3 ... r_{n-m}, r_1^*, r_2^*, r_3^*, .. r_m^*),$$

$$\text{where } r_n = r_m \qquad for\ all\ n,m\ . \tag{19}$$

This is the convention used in numerous papers to designate the joint cumulants of the received signal and its conjugate. The survey conducted by Dobre *et al.* [20] shows a number of 2nd, 4th, 6th, and 8th order cumulant estimates for PSK and QAM modulation. Based on these values, $C_{8,0}$ is a reasonable discriminator between BPSK, QPSK, 8PSK, 16QAM, and 64QAM. In general a higher order modulation requires a higher order cumulant to classify modulation schemes. This gives way to a hierarchical structure in which multiple cumulants are calculated and used to eliminate potential modulation schemes. Such is the case in the work by Swami and Sadler [39], where statistics of the C_{20}, C_{40}, and C_{42} are used to identify QAM, PAM, and PSK modulations. This method of analysis can also be used to distinguish QAM constellation shapes using C_{80}, C_{40}, and C_{42} [40]. More recently it has been shown that multi-path effects can even be accounted for in the estimation of cumulants [41].

Recent trends have applied cumulants to cyclical analysis. A joint cyclic cumulant is formed by taking the joint cumulant of random variables at over time:

$$C_{n,m}(\overline{\tau}) = cum\Big(r_1(t), r_2(t-\tau_2), ... r_{n-m}(t-\tau_{n-m}), r_1(t-\tau_{n-m+1})^*, r_2(t-\tau_{n-m+2})^*, ... r_m^*(t-\tau_n)\Big), \tag{20}$$

where τ is a vector of time lags. This is somewhat analogous to cyclical autocorrelation as it pertains to symbol rate estimation, which is discussed in Section 2.4.1. Despite the different values obtained by cyclical joint cumulants, they can be used in a similar fashion to classify signals. Using the 2nd and 4th order cyclical cumulants enhances modulation discrimination, but this is further improved for QAM and PSK when 6th order cyclical cumulants are included [42]. Higher order cyclic cumulants are effective at classifying higher order modulations [43]. Cumulant-based techniques produce straightforward, highly accurate modulation classifiers. However, these methods require complex signal representation. As previously mentioned this is very difficult to achieve without first obtaining a carrier frequency estimate.

2.3 Frequency Estimation Techniques

Frequency estimation is one of the oldest, most well researched signal processing topic. Despite the substantial amount of time that has already been devoted to this subject, it remains one of the most common signal processing problems. This is also true to a lesser extent for modulated signal carrier estimation. The demodulation of the vast majority of signals requires knowledge of the frequencies employed (FSK) or translation of the signal to base-band (PSK, QAM, etc.). Thus any system wishing to perform blind demodulation will almost certainly require an accurate estimate of the transmit frequencies employed. It should be noted that the carrier frequency must be estimated directly. This is not analogous to frequency-offset estimation, a method by which corrections to a predetermined frequency are made due to changes in the signal caused by the wireless channel. At times, however, the frequency estimation problem can become one of offset estimation if the frequency is localized to within the symbol rate. In the following sections, popular techniques for carrier estimation are presented, as well as the challenges associated with each technique as it relates to signal demodulation.

2.3.1 FFT-Based Techniques

The most well known tool for frequency-based analysis is the fast Fourier transform (FFT). As previously discussed, the FFT can be used to determine the frequency content of a signal:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{j2nk\pi}{N}} \; . \tag{21}$$

Nyquist proved that the maximum retrievable frequency is given by $f_s/2$. The various forms of the Fourier transform divide the spectrum from 0 to $f_s/2$ into N evenly spaced segments, and approximate the magnitude of the frequency content at these points. If the frequency of interest is less than $f_s/2$, the FFT resolution can be improved by down-sampling (i.e. decreasing the amount of spectrum monitored). Similarly, taking a longer-length FFT by using more signal samples increases the number of points into which the spectrum is divided, thereby also increasing the resolution. Implementing FFT techniques for carrier estimation in a digitally modulated signal becomes much more complex as the signal experiences periodic changes in its transient behavior.

The limits on the resolution of the FFT make its applications limited to broadly monitoring the spectrum. Obtaining a precise estimate for the frequency of a sinusoid is not possible in most practical implementations. Attempts to overcome this challenge and enhance the flexibility of the FFT were resulted in the chirp z-transform:

$$X(k) = \sum_{n=0}^{N-1} x(n) A^{-n} W^{nk} , \quad A = A_0 e^{j2\pi\theta_0} , \quad W = W_0 e^{j2\pi\varphi_0} \ , \ k = 0,1,...M-1 , \tag{22}$$

where *A* and *W* are both complex constants. The chirp z-transform can be used to represent points inside and outside of the unit circle. *A* controls the starting point, and *W* controls the rate at which the contour spirals inside or out of the unit circle as well as the angular span covered by the transform. Consequently, the chirp z-transform can be used to analyze narrow-band signal over a small frequency range by setting a small angular span over the region of interest. The chirp z-transform can thus be used as a means of localizing a scaled Fourier transform over a small frequency range, which increases the resolution, as illustrated in Figure 7.

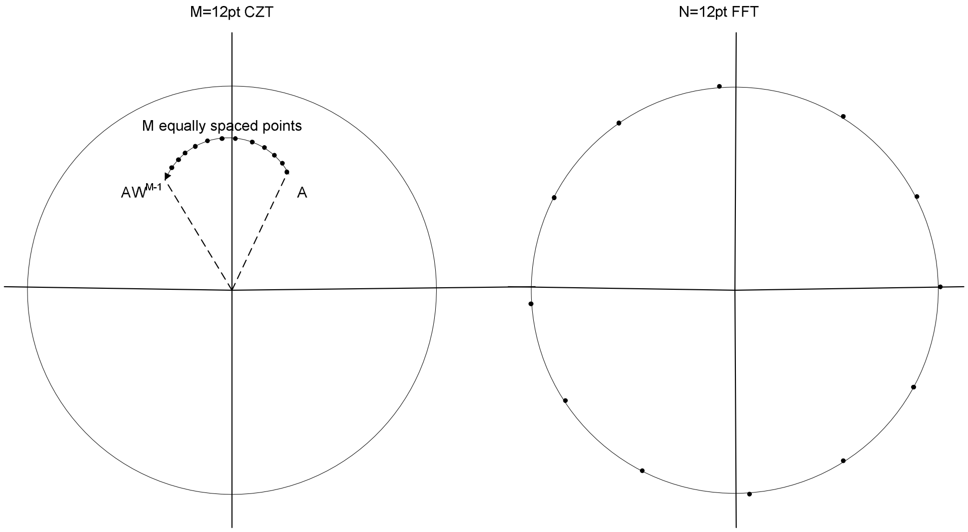


Figure 7. Example of a 12-point chirp z-transform and FFT representation on z-plane.

The chirp z-transform and several modified algorithms derived from it [44, 45] can be used to greatly enhance resolution in narrowband analysis. However, for digitally modulated signals such algorithms are very difficult to implement. The peak power of a modulated signal is frequently not the carrier frequency. Frequency and phase jumps contaminate the estimates of FFT-based methods, resulting in erroneous peak frequency positions. Some carrier estimation algorithms [46, 47] attempt to isolate regions of pure sinusoidal behavior. Such isolation is necessary for an FFT-based algorithm as well. There exist a number of implementation issues, such as the amount of memory needed and the complexity of the algorithm that make this approach less optimal. FFT-based methods, along with periodogram-based power spectral density techniques, are usually relegated to spectrum analyzers rather than blind carrier estimation.

2.3.2 Autocorrelation and Maximum Likelihood Estimators

A number of frequency estimation techniques make use of the autocorrelation of the received signal either directly, or as a means of developing a maximum likelihood estimator. The autocorrelation of a sequence *x[n]* is given by:

$$r(m) = E\{x(k)x(k-m)\} = \frac{1}{N-1}\sum_{n=0}^{N} x(k)x(k-m)\,. \tag{23}$$

The autocorrelation is a very useful mathematical tool that is often related to the fundamental properties of a signal. The autocorrelation of a modulated signal is very dependent on the modulation scheme employed. For this reason, estimators employing the autocorrelation of a signal are usually developed for specific types of modulation and/or modulating conditions, such as a linearly modulated signal.

The phase of the autocorrelation for complex signals is the basis for a number of popular frequency estimators in phase shift-keyed signals. If the frequency of a signal has been identified to within a moderate percentage (20%-50%) of the symbol rate, using techniques like the FFT and CZT, then it is possible to estimate the frequency using a robust frequency offset technique. Most frequency-offset algorithms operate on the condition that the offset is small. If the complex model for the received signal in (9) is assumed, then after down-conversion the signal is given by:

$$s(k) = \alpha e^{j2\pi f_d kT + \varphi_k} + n(k), \qquad k = 0,1 \ldots M-1, \tag{24}$$

where f_d is the frequency deviation from the true frequency, and T is the sampling period. Letting *x[k]* represent the non-noise term yields a non-zero lag correlation expression:

$$r(m) = E\{s(k)s(k-m)\} = \frac{1}{N-1}\sum_{k=1}^{N}[x(k)+n(k)][x(k-m)+n(k-m)]. \tag{25}$$

From a probabilistic viewpoint this can be expresssed as:

$$E\{[x(k)+n(k)][x(k-m)+n(k-m)]\} = E\{x(k)x(k-m)\} + E\{n(k)n(k-m)\}, \tag{26}$$

where the cross-correlating terms are zero because *n(k)* is assumed to be zero mean white noise. Having left out the zero lag term, the second term is ideally zero (but more realistically very small). The uncorrupted signal has the correlation expression:

$$r_x(m) = \frac{1}{N}\sum_{m=1}^{N}(e^{j2\pi f_d kT + \varphi_k})(e^{-j2\pi f_d [k-m]T + \varphi_k}) = e^{j2\pi f_d mT}. \tag{27}$$

Therefore, the argument of *r(m)* is a combination of (27) and the small term, λ_k, which is caused by deviations from the uncorrelated behavior of the noise term, yielding:

$$\arg\{r(k)\} = [2\pi k f_d T + \lambda(k)]. \tag{28}$$

Neglecting the noise term yields a very basic estimator for *k*:

$$f_d = \frac{\arg\{r(k)\}}{2\pi T}. \tag{29}$$

If instead this value is averaged over *N* different lags, the noise term vanishes leaving:

$$\frac{1}{N}\sum_{k=1}^{N}\arg\{r(k)\} = \frac{2\pi f_d T}{N}\sum_{k=1}^{N}k = \frac{2\pi f_d T(N+1)(N/2)}{N}, \qquad \text{for N even.} \tag{30}$$

This bears a striking resemblance to the estimator suggested by Fitz [48]:

$$f_d = \frac{2}{\pi T N(N+1)} \sum_{k=1}^{N} \arg\{r(k)\}\,. \tag{31}$$

A periodogram is essentially a time-windowed Fourier transform. By isolating the portion of the signal for which the frequency of a modulated signal is constant, it is possible to extract a maximum likelihood estimator from the magnitude of the periodogram squared:

$$L(f_d) = \arg\left\{ \max \left| \sum_{n=0}^{N-1} z(n) e^{j2\pi f_d T} \right|^2 \right\}. \tag{32}$$

The well-known L&R technique is derived from this expression. Maximization of this equation with respect to f_d results in:

$$\mathrm{Im}\left\{ \sum_{k=1}^{N-1} k(N-k) r(k) e^{j2\pi f_d Tk} \right\} = 0\,. \tag{33}$$

Luise and Reggianni [49] recognized that the primary purpose of the weighting function, *k(N-k)*, was to weight samples close to *N-1* much less than for samples far from *N-1*. Consequently they set forth a suboptimal approach where they assume a weight of one for *M<N-1* iterations. They then replace the exponential with the linear approximation given by the first term of its Taylor series:

$$\mathrm{Im}\left\{ \sum_{k=1}^{M} r(k)[1 - j2\pi f_d Tk] \right\} = \sum_{k=1}^{M} \mathrm{Im}\{r(k)\} - 2\pi f_d T \sum_{k=1}^{M} k\,\mathrm{Re}\{r(k)\} = 0\,. \tag{34}$$

Simplification of this equation leads to the L&R estimator:

$$f_d = \frac{1}{\pi T(M+1)} \arg\left\{ \sum_{k=1}^{N} r(k) \right\}\,. \tag{35}$$

A number of other methods derive their estimators using (13) One problem associated with the computation of arg{*r[k]*}, is that the phase must be unwrapped to a fixed interval [0, 2π] in order to maintain the linearity assumed. However, this problem can be avoided by considering the product $r[k]r^{*}[k]$ instead, and replacing the original weights *k(N-k)* with those obtained by applying conversion equations set forth by Tretter and Kay [50]. This produced a frequency estimate given by [50]:

$$f_d = \frac{1}{2\pi T}\sum_{k=1}^{N} w(k)\arg\{r(k)r^{*}(k-1)\}, \tag{36}$$

$$where\ \ w(k) = \frac{2N(N^2-1)-k(k-1)(3N-2k+1)}{N^2(N^2-1)}. \tag{37}$$

By using the difference between subsequent correlations, it is no longer necessary to unwrap the phase to maintain the linearity of the estimator. However, the multiplication of *r(k)* and *r(k-1)* adds additional complexity to the estimator. This problem was resolved by Wu *et al*. [51] through estimation of the frequency, $f_d(k)$, at each iteration. Estimates are then adjusted by the multiple of 1*/kt,* such that the difference between subsequent iterations is minimized:

$$\hat{f}_I(k) = \hat{f}(k) + C\frac{1}{kt}. \tag{38}$$

Making use of (16) and (17), the frequency estimate at each iteration, f_I, is weighted similarly:

$$\hat{f}_d = \sum_{k=1}^{N} \frac{12k(N+1-k)k}{(N+1)^2[(N+1)^2-1]}\,\mathrm{f_I(k)}. \tag{39}$$

This method was shown to have better performance than earlier methods [50] for a greater range of frequency deviation. An algorithm developed by Mengali and Morelli [52] develops a new model for the scaled autocorrelation up to $L_0/2 < N$ lags:

$$R(m) = e^{j2\pi f_d T}[1 + \lambda(m)] \ , \tag{40}$$

$$where \ \ \lambda(m) = \frac{1}{L_0 - m}\sum_{k=m}^{L_0-1} c_k^{\ *} n(k) e^{j2\pi f_d Tk} + c_k^{\ *} n(k-m) e^{j2\pi f_d T(k-m)} \, . \tag{41}$$

Again, the difference between subsequent complex arguments of *r(m)* is used to avoid the phase unwrapping problem. Similar analysis as seen above can be performed to yield a new windowing function, resulting in good performance for very low SNR while maintaining wide error range capabilities:

$$w(m) = \frac{3[(L_0 - m)(L_0 - m + 1) - N(L_0 - N)]}{N(4N^2 - 6NL_0 + 2L_0^{\ 2} - 1)} \tag{42}$$

There are a number of reasons why autocorrelation-based ML estimators may not be employed. First among these is the necessity to obtain some basic approximation of the frequency within at least fifty percent of the symbol rate. Failing to do so makes the techniques listed above useless. The signal model itself places additional restriction on the design from the hardware perspective, as down-conversion to near base-band is needed prior to frequency analysis. A recent trend in carrier and timing analysis is to analyze the behavior of the cyclic autocorrelation:

$$r(\alpha, \tau) = \frac{1}{N}\sum_{k=0}^{N-1} x(k) x(k+\tau) e^{-j2\pi\alpha k} \, . \tag{43}$$

Derivation of the cyclic autocorrelation for a PSK signal was performed by Jin and Ji [53], and it was found to relate to the modulating signal *a(t)* in *a(t)cos(2πf_cτ+φ)* by:

$$R_x(\alpha, \tau) = \frac{1}{2} R_a(\alpha, \tau)(\cos(2\pi f_c \tau) + \frac{1}{4} R_a(\alpha + 2f_c, \tau) e^{j2\varphi} + \frac{1}{4} R_a(\alpha - 2f_c, \tau) e^{-j2\varphi} \, , \tag{44}$$

where $R_a(\alpha,\tau)$ is the cyclic autocorrelation of the a(t). Further analysis showed that $R_a(\alpha,0)$ only has nonzero values for α = 0, ± $2f_{c.}$ Thus analysis of the cyclic autocorrelation for zero lag should reveal the carrier frequency. Note that unlike the previously discussed methods, this has been accomplished without down-conversion or pseudo-knowledge of the transmit frequency. However, this algorithm is limited by its assumption that the symbol period is a multiple of the sampling period, which in practice may not be true.

2.3.3 Subspace-Based Methods: Pisarenko and MUSIC

One of the most well known methods for frequency estimation was introduced in the early 1970s when Pisarenko began studying the problem of determining the frequency of a sinusoid in the presence of white noise. Pisarenko was able to establish a relationship between the roots of the eigenvectors and the frequency of the sinusoid. If there are assumed to be *p* sinusoids in white noise, then we can derive an autocorrelation:

$$r(k) = \sum_{i=0}^{p-1} C_i e^{j2\pi k f_i} + \sigma_n{}^2 \delta(k) \ . \tag{45}$$

The first term is obtained by assuming spikes in the power spectrum at the sinusoids and taking the inverse Fourier transform. The second term is derived from the fact that the noise is uncorrelated except at zero lag, where it has a value equal to the noise power/variance σ_n^2.

Pisarenko realized that the eigenvectors of the autocorrelation matrix span the signal space of the sinusoid term except for λ = 0. Therefore, assuming an autocorrelation matrix of rank *p + 1*, the eigenfilter:

$$V_{P+1}(e^{jw}) = \sum_{k=0}^{p} v_{P+1}(k) e^{-jwn} , \tag{46}$$

has zeros corresponding to the frequencies of the sinusoid. Thus an equation representing a spectrum estimate for identifying peaks was suggested to be:

$$\hat{S} = \frac{1}{|V_{P+1}(e^{jw})|^2} . \tag{47}$$

This method does not perform well in practice because of the relatively few measurements taken to obtain the estimate. This is particularly true for the single frequency estimate that is of interest in signal down-conversion. If instead the eigenfilter is considered for additional eigenvectors (*p+1…N*), then zeros will exist for each of these eigenvectors and their summation yields an improved estimator:

$$\hat{S} = \frac{1}{\sum_{p+1}^{N} \| V_{P+1}(e^{jw})|^2} , \tag{48}$$

which is known as the MUSIC algorithm. The minimum of each eigenvector occurs at the frequencies of the sinusoids, and their summation provides for a more accurate minimum to be determined. Other methods of improved performance [54] have been suggested, but suffer from the same disadvantages that all these type of estimators share.

Subspace-based methods do not require localization of the frequency prior to their application, as was seen for the methods in the previous section. However, they have a few very significant disadvantages. These methods only work for moderate SNR values, reducing the robustness of their applications. However, more significant is the computational burden placed on the calculation of the eigenvectors. It is this limitation that restricts the application of these methods primarily to post-processing analysis after data has been collected.

2.3.4 Direct Frequency Estimation

Direct frequency estimators (DFEs) estimate the frequency of a sinusoid without knowledge of the properties of the signal (autocorrelation/eigenvectors). One way to accomplish this is through application of an adaptive notch filter [55-57]. The output of adaptive notch filters is monitored to determine when the sinusoid has been filtered (i.e. the power output is used as a form of error signal). However, the signals of interest are being modulated, spreading the bandwidth of the signal. Additionally, unlike the previously discussed methods, we cannot simply apply the notch filter to a time-isolated segment of the received signal. Consequently these methods are not applicable for the intended purpose.

However, there are a handful of direct frequency estimators that do not apply adaptive notch filters. The method proposed by Ho and Ching [58] is particularly attractive for frequency estimation of modulated signals. Their algorithm utilizes the recursive relationship of a digitized sinusoid $x(k) = cos(2\pi f_c kT)$, which is expressed as:

$$x(k) = \cos(2\pi f_c)x(k-1) - x(k-2)\,. \tag{49}$$

An expression for the error *e(k)* is derived from the difference between the estimate and the actual value. The expected value for the error cannot be minimized without knowledge of the noise variance. Consequently, their work suggests a constrained optimization problem that results in an expression of the form:

$$\eta = F(\hat{w}, \sigma_s) + \sigma_n{}^2 \ , \tag{50}$$

which, when minimized using gradient methods, has no dependency on the noise power $\sigma_n{}^2$. The resulting frequency estimator, employing a step size μ, is expressed:

$$w(k+1) = w(k) - \mu e(k)[(x(k) - x(k-2)\cos[w(k)] + x(k-1)]\,. \tag{51}$$

This method of frequency estimation is computationally inexpensive, has very straightforward implementation, flexible convergence properties, operates at IF, and does not require isolation of pure sinusoidal segments. Selection of the step size can be made based on the sampling rate and the maximum expected symbol rate, which ensures reasonable convergence within a symbol period. However, this method requires a substantial number of samples/symbols which, depending on the system, may not be possible.

2.4 Symbol Rate Estimation

Symbol synchronizers are an important part of a digital receiver. Failure to properly synchronize an incoming signal can result in the smearing of information from adjacent symbols as well as improper signal demodulation. In order to correctly demodulate a signal it is necessary to know the time duration of a single symbol. In the case of a non-cognitive radio, the symbol duration is known prior to signal reception. For narrowband signals, synchronization is usually achieved by monitoring the output of a matched filter *h[n]*. Assuming the PSK/QAM signal has been translated to base-band, the in-phase and quadrature phase components should be of the form:

$$r_I(t) = C_k \;\;,\; r_Q(t) = C_l \qquad k, l = 1,2,...N \;, \tag{52}$$

where *k* and *l* refer to the symbol index. Consequently, the appropriate matched filter has a constant magnitude and a length given by the duration of a symbol:

$$h[n] = u[n] - u[n-N] \tag{53}$$

It is clear that the output of this filter is dependent on the behavior of the input.

The beginning symbols in a received packet often contain known training symbols to be used by the synchronizer. The most common training sequence for BPSK is the antipodal alternating sequence $C_k = [C,-C,C,-C....]$. The matched filter output would result in a triangular wave with maxima/minima separated by the symbol duration T_s as Figure 8 illustrates. Because the matched filter simultaneously smoothes the noise contribution, the local maxima/minima can be determined using rudimentary peak-finding algorithms. Notice that the only theoretical change required for differing symbol rates is the extension of the length of the matched filter. The problem of designing a configurable synchronizer for phase-shift keyed signals was undertaken by Tachwali and Barnes [59], and a structure encompassing multiple symbol rates was developed.

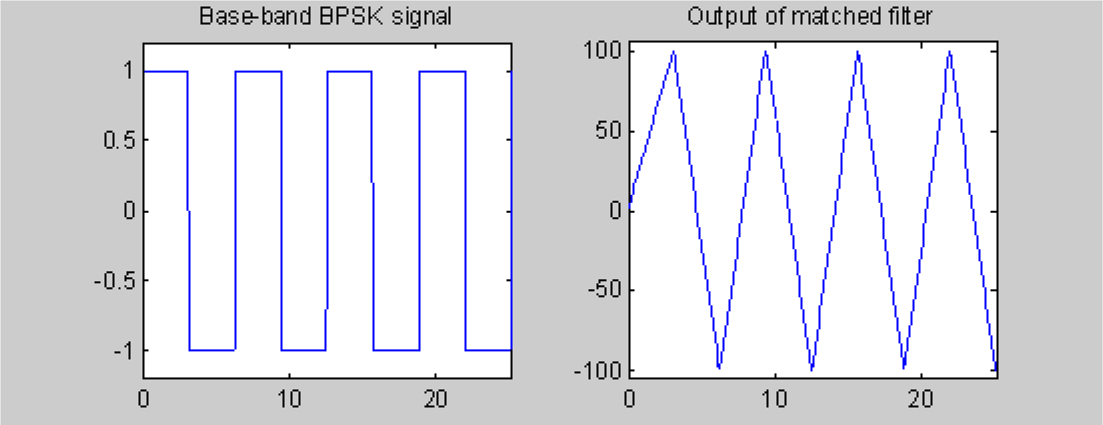


Figure 8. Impact of matched filter on base-band pulse for BPSK.

Another method for symbol synchronization applies correlators to the received signal and the expected sequence. The output of the correlator is maximized when the expected sequence has the most overlap with the received sequence. As a result the correlation between the expected sequence and the actual sequence can be computed and, if a threshold is not surpassed, the expected sequence can be delayed. Once the threshold has been met, the sequence is synchronized to within the delay length. DSSS correlators employ this technique to achieve coarse lock to within a half chip period. A refined process is needed to enhance the timing estimation of the correlator beyond this resolution. This is achieved by the application of a delay-locked loop (DLL). The DLL takes a single sequence sampled “early” and “late”, and it determines the correlation from each. The difference in the correlation is used as an error signal to drive the frequency of the sequence generator until eventually the error signal is driven to zero, at which point synchronization is achieved. A configurable synchronizer of this form [60] is summarized in Figure 9.

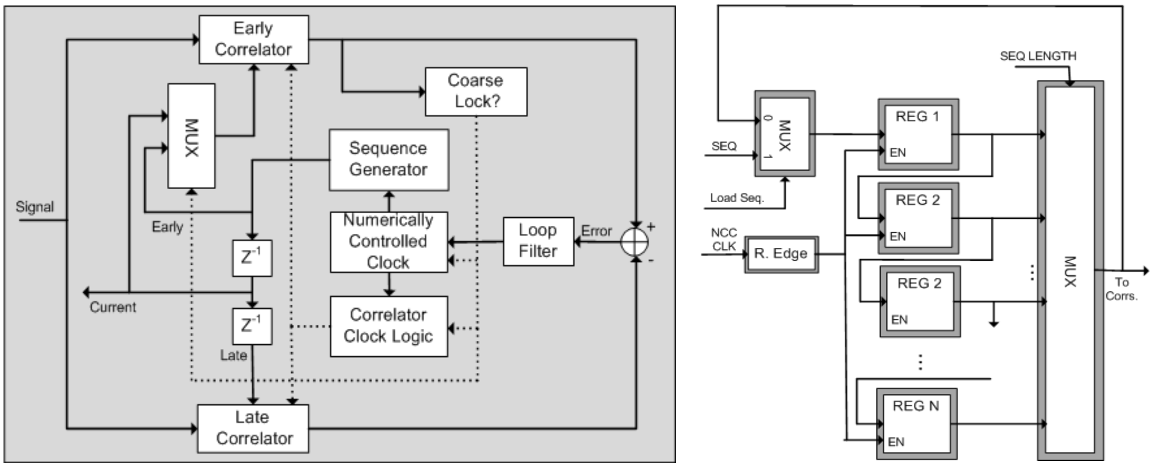


Figure 9. Configurable synchronizer (left) and sequence generator (right) architectures.

The rightmost architecture in Figure 9 depicts the sequence generator. The sequence is serially loaded into a maximum of *N* registers such that each register contains a value in the expected sequence. The sequence is then output at a rate dictated by the numerically controlled clock (NCC), which by default, inserts a half chip delay each time the sequence is output. Upon achieving coarse-lock, the system utilizes a DLL to control the NCC until fine synchronization has been reached. Analysis of the discussed architectures shows that three parameters are sufficient for successful timing recovery: symbol duration, the synchronizing sequence, and the matched filter response.

2.4.1 Cyclic Autocorrelation-Based Techniques

As with the carrier frequency, the cyclic autocorrelation has been found to have a distinct relationship with symbol timing for PSK signals. Recalling the cyclic autocorrelation equation (43), calculation gives an indication of whether the properties of a signal change cyclically over time. It has been established that the relationship between the autocorrelation and cyclic autocorrelation can be exploited to determine the symbol rate in MPSK modulation:

$$r(n) = r(0) + r(\alpha_0, n)e^{j2\pi\alpha_0} + r(-\alpha_0, n)e^{-j2\pi\alpha_0}\,. \qquad (54)$$

It was found [61] that this expression could be exploited to determine a weighted cyclic autocorrelation statistic that was maximized at α_0. However, the method required very large SNR and was computationally burdensome, making it worthless in real time situations.

Recall the the relationship between the cyclic autocorrelation time-varying amplitude and the cyclic autocorrelation of the overall signal established in (24). The cyclic autocorrelation, $R_a(\alpha,\tau)$, of the modulating function $a(t)$ has non-zero values for $\alpha = kf_{sym}$ shows that the cyclic autocorrelation of the received signal has non-zero values [53]:

$$\alpha = \{kf_{sym}, \pm 2f_c + kf_{sym}\} \quad k = 1,2.....M\ , \qquad (55)$$

where *M* is f_{samp}/f_{sym}. Thus the technique presented in Section 2.3.2 can be used to estimate the carrier frequency. Once the frequency is known, the time difference between the spike associated with the carrier frequency and the nearest neighboring spike gives an estimate of the symbol rate. This joint estimation technique is quite powerful as it reduces the necessary hardware, while providing good estimates at low SNR for both timing and frequency. It does, however, suffer from the restriction that f_{samp}/f_{sym} must be an integer, which is often not the case, particularly if the characteristics of the signal are not known prior to reception.

Alternatively, the assumption could be made that the received symbols for PSK have an expected value of zero as well as an autocorrelation of zero for non-zero lags because each symbol is equally likely. Under these assumptions, the magnitude of the cyclic autocorrelation for the modulation waveform (assuming square wave), and the magnitude of the overall signal, are equivalent and given by [62]:

$$|R_s(\alpha,\tau)| = |R_a(\alpha,\tau)| = \frac{1}{\pi}\cos\left(\pi\alpha\left[|\tau| - T/2\right]\right). \tag{56}$$

Setting α = *(+/-)1/T*, the above function is maximized when τ = *(+/-)T/2*. Again, localization of the maximum of the function yields the symbol rate. Major drawbacks of this method include poor performance at low SNR and the need for a large number of symbols to satisfy the zero mean condition. More generally, cyclic autocorrelation-based symbol rate estimators require a large number of samples due to the need to span multiple symbols. This is particularly true when signals are oversampled.

2.4.2 Wavelet-Based Techniques

Wavelet-based techniques use short-time analysis to monitor significant changes in the behavior of a received signal. For real-value signals, the magnitude of the wavelet transform using a Haar wavelet is given by a sinusoid of very small amplitude in the region where no symbol changes have occurred. To better understand this behavior, consider a simple filter with impulse response given by:

$$h[k] = \partial[k-1] - \partial[k], \qquad (57)$$

which is a good approximation of the behavior of the wavelet transform for a small scale value. Neglecting noise makes the amplitude at the output of this filter nearly constant for a pure sinusoidal input. Now consider the behavior immediately after a symbol change has taken place. In the case of PSK, a spike in the filter output is expected due to the phase jump discontinuity. Conversely, CPFSK would exhibit a small DC change as the difference between subsequent samples has increased/decreased due to the frequency change.

The basic concept discussed above was expanded to wavelets [63], and it was found that the wavelet transform magnitude of a complex sinusoidal signal is given by [34]:

$$\left|WT(a,n)\right| = \frac{4\sqrt{S}}{w_c\sqrt{a}}\sin^2\left(\frac{w_c a}{4}\right), \qquad \text{(pure sinusoid)} \quad (58)$$

$$\left|WT(a,n)\right| = \frac{4\sqrt{S}}{w_c\sqrt{a}}\sin\left(\frac{w_c a}{4}\right)\sin\left(\frac{w_c a}{4}+\frac{2\pi k}{M}\right) \quad k = 1,2...M-1 \ , \qquad \text{(PSK/QAM)} \quad (59)$$

$$\left|WT(a,n)\right| = \frac{4\sqrt{S}}{(w_c+w_i)\sqrt{a}}\sin^2\left(\frac{(w_c+w_i)a}{4}\right) \ , \qquad \text{(FSK)} \quad (60)$$

where w_c is the normalized carrier frequency, a is the scale, S is the signal power, M is the constellation size, and w_i is the frequency deviation. The perturbations in the wavelet transform magnitude are easily detectable in the presence of moderate noise, and the symbol rate is determined by finding the minimum difference between subsequent peaks. It was shown [64] that the performance could be improved even more if the signal is down-converted prior to the formation of the wavelet transform. These algorithms operate on the assumption that the sampling frequency is a multiple of the carrier frequency. If this condition is not met, there will exist a mismatch that propagates over time resulting in different wavelet transform magnitudes. These magnitudes may be attenuated to within the range of the noise variance causing degraded estimator performance.

2.4.3 Additional Methods

The majority of narrowband systems use symbol rate estimators employing either the wavelet transform or cyclic autocorrelation, but a few other techniques can be found in literature. A number of techniques are computationally complex and involve extensive knowledge of Monte Carlo (importance) sampling methods [65, 66] that are beyond the scope of this dissertation. A more straightforward approach taken by Flohberger *et al.* [67] determines the over-sampling factor through computation of the power spectrum and application of the inverse Fourier transform in cases where raised-cosine filters have been applied. The received down-converted signal can be thought of as the sum of time-shifted, scaled, raised-cosine pulses. By calculating the power spectrum using periodogram-based algorithms, then taking the inverse Fourier transform, it is possible to retrieve scaled versions of the original shaping filter:

$$h[k] = C\left(\frac{\sin(kT_s / T_{sym})}{T_s / T_{sym}}\right)\left(\frac{\cos(\pi\beta kT_s / T_{sym})}{1-(4\beta^2[kTs]^2)/T_{sym}}\right), \tag{61}$$

where T_s is the sampling period, T_{sym} is the symbol rate, and β is the roll-off factor. The raised-cosine filter has zeros at multiples of the symbol duration. Thus if we take the magnitude of *h[k]* and search for local minima, they should occur at multiples of the data rate as well.

This method works well if the sampling rate is a multiple of the symbol rate. It even outperforms more traditional cyclic autocorrelation-based methods for very low SNRs under this condition [68]. If the sampling rate is not a multiple of the symbol rate, performance degrades rapidly, and tracking algorithms are necessary to estimate the exact symbol rate. This limitation combined with the computational complexity of computing the power spectrum and its inverse Fourier transform make this algorithm less practical than those discussed in the previous sections.

3. Frequency Conversion Extension

3.1 Frequency Converter Advantages

The FCC is constantly evaluating spectrum utilization and consumer needs, opening new unlicensed bands, and selling rights to use portions of the spectrum for limited purposes. These new frequency bands may exhibit beneficial properties such as more desirable signal propagation, decreased interference, enhanced security, or an increase in the available bandwidth. A large number of devices implementing popular communication protocols currently exist in the marketplace, and all must adhere to current FCC regulations. An extension to these systems that makes use of the duplicate base-band hardware and MAC implementations could be an invaluable tool in the rapid deployment of devices operating at non-traditional frequencies.

Any frequency converter extension (FCE) must maintain the basic properties of the signal while altering the carrier frequency. The FCE must intelligently determine when to enable transmission or reception without losing a portion of a packet or distorting the signal. All of this must be accomplished without significantly altering the performance of the MAC. The following sections describe a generic procedure for developing an intelligent FCE as well as some of the design challenges such a system presents.

3.2 Existing Frequency Converter Technology

Little research has been performed on the design of an extension of existing communication systems to new frequency bands. However, a number of products have been developed along these lines. For example, TI provides a power extension to low-power 802.11b radios; however, the chip directly relies on control signals from the radio itself [69]. The only frequency converter extension currently on the market of which the author is aware is the 915UDX series developed by RFLINX© [70]. This device converts a specified fixed 802.11b channel to 915 MHz and increases the output power as dictated by the consumer. The primary focus of this device is to increase the effective range of an 802.11b/g network. Another application would be to set up two wireless routers and connect them to two 915UDX units spaced a significant distance apart from one another. With one router acting as a bridge, frequency converters form a link between two or more distant local wireless networks. This structure avoids the burden of each user needing a converter, but maintains the presence of the original 802.11b networks locally. In some situations it may be advantageous to maintain a local network with the same range as the 802.11b network, but utilizing a new carrier frequency, thus eliminating the presence of the original carrier within the network. This may be the case if decreased interference or enhanced security is desired.

3.3 Basic Architecture

Any frequency converter extension will consist of a number of basic design components to enable duplexing and proper signal translation. First the output signal from the original wireless device must be monitored to ensure the transmit and receive paths are enabled at the proper times. Signal monitoring is more well-suited at the output of the wireless device, because that is where the signal should be considerably strong when transmission is occurring. This suggests that the FCE should default to the receiving path, and switch when a new transmission begins. Conversely, if the monitoring was done at the antenna, strong noise within the band of interest or a weaker signal could hinder ideal signal detection, causing the wrong path to be enabled. In terms of the signal monitoring itself, the signal can be detected by examining characteristics of the signal, most notably its power.

While the output of the wireless device is being monitored, the signal itself is converted to the desired frequency using a mixer and appropriate filtering. This procedure should occur outside of the transmit/receive signal paths to avoid hardware duplication. The two switches essentially form a double pull double throw (DPDT) switch with the transmit/receive hardware placed in between the output of the switches. The transmit hardware consists mainly of a power amplifier to recover any loss in signal strength due to attenuation at the mixer output. The receiving hardware consists of a band-pass filter and a low-noise amplifier with static gain. Any received signal sent to the wireless device should have a signal strength comparable to a typical received signal for that particular wireless device. The basic structure suggested is shown in Figure 10.

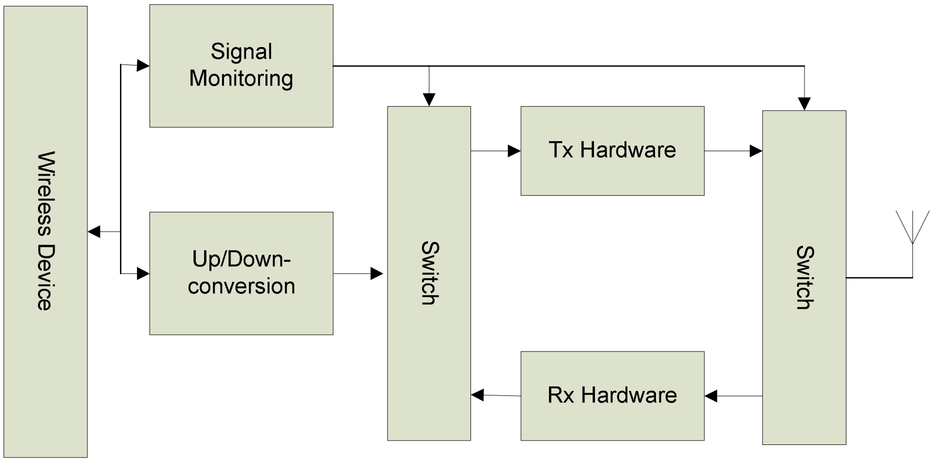


Figure 10. Basic structure of a frequency conversion extension.

3.4 Current Design Limitations

Extension of the original wireless device introduces additional noise and/or distortion to the overall system. In addition, the time delay in switching signal paths will cause the loss of a portion of the transmitted packet. The high power involved in the transmit path also leads to a number of problems with regards to leakage and electrical isolation that must be resolved to enable proper functionality.

3.4.1 Power Loss and Noise Considerations

As the transmit/receive signal passes through portions of the FCE, it will lose some of its power. This is particularly true at any mixers or filters that may exist. In the case of a filter, the loss is usually minimal but may differ by a few dB depending on the frequency of the signal. In the case of a mixer, the conversion loss is both frequency-specific and dependent on the power of the local oscillator. This relationship can be exploited to eliminate the need for an attenuator prior to reception at the wireless device. Any signal power loss experienced in the FCE can be recovered by passing the signal through an amplifier, however, doing so will raise the noise floor in the signal bandwidth. While this is not a significant concern when transmitting, because noise at or near the noise floor is not likely to propagate very far, it can be problematic along the receive signal path and should be considered when evaluating an FCE.

Maintaining the purity of the original signal is also critical to ensure similar wireless performance of the extended system. The bit error rate (BER) is always directly related to the signal-to-noise ratio at the receiver, and any added noise will affect the minimum signal level at which packets are not dropped. To this end, the overall noise figure of the FCE is an important metric when evaluating its overall design. The noise figure of a component is defined as the ratio of the SNR at the input to the SNR at the output of the component:

$$NF = \frac{SNR_i}{SNR_o}. \tag{62}$$

For a system of cascaded components, the noise figure can be expressed as:

$$NF = NF_1 + \frac{NF_2 - 1}{G_1} + \frac{NF_3 - 1}{G_1 G_2} + \ldots \frac{NF_n - 1}{G_1 G_2 \ldots G_n}. \tag{63}$$

Consequently, knowledge of the noise figure and gain of the individual components is sufficient to characterize the overall noise figure of the system. From a noise standpoint, concern should only arise along one transmit/receive path at a time. The noise generated in the control signaling is not relevant except with regards to meeting EMC testing standards. The noise figure is particularly important for amplifiers, as any thermal noise seen at the input will be amplified. Consequently, design of amplifiers exhibiting low noise figure has been thoroughly researched [71-73].

There are three classical methods of measuring noise figures [74]. Of these methods the most popular is the Y-factor method, which is suitable for a wide range of noise figures. This method involves connecting the DUT as it would ordinarily be setup, except producing the input signal with a DC-powered ENR noise head. With the output of the DUT connected to a spectrum analyzer, the difference in the noise spectral density is recorded as *Y*. The noise figure of the device is then given by:

$$NF = 10\log_{10}\frac{10^{(ENR/10)}}{10^{(Y/10)}}, \tag{64}$$

Where γ is a frequency-specific value specified by the manufacturer of the bulkhead. For lower noise figure values this may result in slightly skewed results, which Is unfortunate as low noise figure data is usually provided by the component manufacturer as a strong selling point. It should be mentioned that new methods for interpreting the noise figure of non-linear devices have been suggested [75], but are not necessary for many applications, as they occur over a small range of frequencies in which a non-linear device exhibits locally linear behavior.

3.4.2 RF Leakage and Isolation

Because the FCE is dealing with a fairly strong output signal from the wireless device, RF leakage from one port of a component to another may occur. This makes port isolation of the utmost importance in the system design. Again, the primary concern is with the data path, not the control signal. Some components, such as amplifiers, may have naturally high isolation (in the reverse direction). However, isolation is particularly important with regards to mixers and switches. Poorly isolated ports of a switch may result in a signal being coupled onto a path that should have no signal. Consequently, high isolation is the primary goal of the design of numerous switches [76, 77]. The situation is more complicated for mixers, as they are often followed directly by an amplifier or second mixing stage. Obviously an amplifier would increase the power of a leaked signal, but additional mixers could also result in new mixed frequency products that are within the bandwidth of the desired signal. These problems can be avoided by employing components with high port isolation. However, if such components are not readily available, appropriate filters can be used to increase the overall isolation of the component-filter subsystem, provided that the leaked frequencies are known.

3.4.3 MAC and Timing Considerations

Perhaps the most challenging problem with regard to extending a wireless device to operate in a new frequency range is the need to preserve the operation of the existing MAC implementation and enable packet transmission quickly. If the MAC of the base device is CMSA or some derivative thereof, the FCE must ensure the listening behavior is translated to the new frequency range. In a typical wireless radio, control signals are used to directly enable transmission very quickly. The FCE on the other hand, must first monitor the output from the wireless device to determine if the transmit or receive path should be enabled. The time it takes to switch between signal paths may have a very small impact on the MAC, and as a consequence the throughput of the network. As the switching delay is increased, it is expected to have an increasingly adverse effect on the MAC performance. It should be noted that this impact is expected to be fairly insignificant for reasonable delays.

Of much greater importance is the impact that any delay in enabling the transmit path has on transmitted packets. Two basic approaches can be taken to address this problem. First, the transmit signal could be delayed to ensure the correct path was enabled prior to the signal reaching the switch. This approach is feasible for small delays (less than a few hundred nanoseconds), but becomes significantly harder for delays exceeding hundreds of nanoseconds. The other approach is to limit the delay of the control signals to a relatively insignificant amount for a given packet. For example, if the general packet structure for an 802.11b packet is assumed, Figure 11 shows that the early bits contain synchronization information. In many circumstances, it may be acceptable

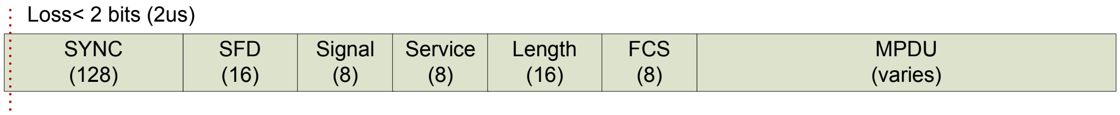


Figure 11. Illustration of partial packet loss due to timing for an 802.11b signal.

In many circumstances, it may be acceptable to lose a few of these bits at the beginning, as the system would recover with the remaining bits. A more acceptable scenario would consist of the loss of part of one or a few chips in the chipping sequence. Since the chipping rate is 11 Mchips/s, the chip duration is approximately 91 ns and the symbol duration is 1 µs. Testing is needed to determine an acceptable synchronization chip/symbol loss.

3.5 Design of a Wired 802.11b Frequency Converter Extension

The main goal of this research is to produce a viable framework-architecture that provides the necessary physical layer changes to an existing wireless system necessary to enable cognitive capabilities. As a means of testing the architecture, an extension to 802.11b devices was proposed. A wired prototype was developed that is capable of translating an 802.11b signal from 2.417 GHz (channel 2) to 1.717 GHz. Design emphasis will be placed on minimizing alterations to the modulating sequence, mainly caused by timing issues, thus ensuring reasonable noise contamination introduced by the FCE. The design will also focus on maintaining similar communication performance for low data rates (1 and 2 Mbps). The FCE output should operate within the limits dictated by the FCC at 1.717 GHz. Overall evaluation of the design is determined by the transmission of a predetermined bit-sequence or file from one radio to another through the prototype FCE. Emphasis is placed on the step-by-step analysis of the architecture, as significant leakage may exist in the wired design and subsequently skew the actual file transfer results.

3.5.1 Wired System Architecture

The FCE architecture is in-line with the basic structure set forth in Figure 10. The wireless device is connected to a directional coupler, which links the forward power from the wireless device to a power detector. The power detector then produces a voltage proportional to the power, which is evaluated by the switch decision circuit to determine whether a signal is currently being transmitted. The FCE defaults to the receive mode. The time it takes to enable the transmit path is equivalent to the amount of the original packet that is lost, making the timing of this control path very important. It is likely that future work will attempt to further decrease the propagation delay of this signal path. The original 802.11b signal passes through the DC coupler and a 2.417 GHz band-pass filter before it enters a mixer with a local oscillator at 700 MHz. An image rejection filter is employed to capture the 1.717 GHz signal after the mixer, at which time the signal is passed to the transmit path, (which has been enabled. This signal is then amplified and down-converted to the original 2.4 GHz, 802.11b signal before it is received by the other 802.11b radio. This process is illustrated in Figure 12.

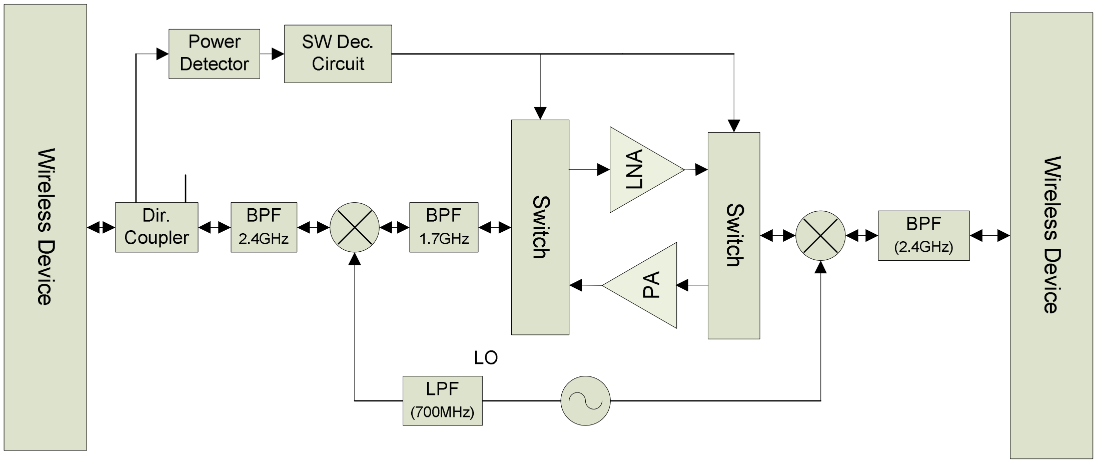
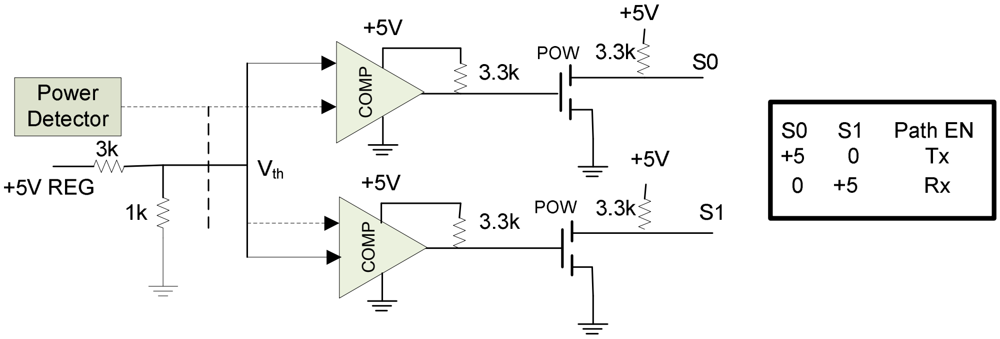


Figure 12. Overall 802.11b to 1.717 GHz FCE wired architecture.

The design of the switch decision circuit is made more complicated by the switches currently employed. The switches require inputs to ports S0 and S1 to enable proper path selection as characterized in Figure 12. To achieve this, a threshold voltage of 1.25 V is set using a series of resistors and a regulated 5 V source. The voltage output from the power detector is sent to the non-inverting input of one comparator and the inverting input of another comparator on the same chip. The threshold voltage is sent to the remaining comparator ports. The threshold is set at a transmit power of -5 dBm which is approximately 10 dB below the minimum transmit power of the 802.11b radio. Since the power coupled to the forward port when a signal is received is almost non-existent, setting the comparison voltage low makes sense to avoid the scenario where the transmit power is close to the threshold. Pull-up resistors are added after the comparators, then the outputs are sent to the gate of a series of power MOSFETs. These MOSFETs are necessary due to the high current drawn by the switches. This architecture of the switch decision circuit is shown in Figure 13.

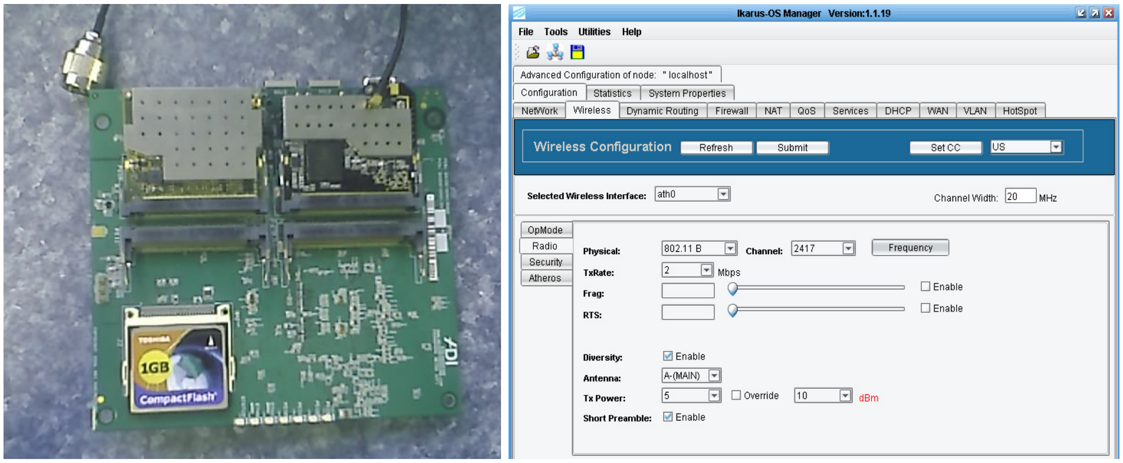


S0	S1	Path EN
+5	0	Tx
0	+5	Rx

Figure 13. Switch decision control logic design.

3.5.2 Radio and FTP Server Configuration

Two configurable 802.11 boards made by *ADI* were used as the wireless devices. The boards were configured using *Ikarus–OS Manager*. After initial configuration and setup, this software provides a user-friendly GUI that enables quick modification of the radio's behavior. Notable parameters that can be controlled include the transmit power, SSID, data rate, frequency (channel), and communication protocol. The radio is capable of acting as an 802.11a/b/g access point or client. The radio and configuration software are shown in Figure 14.



Advanced Configuration of node: " localhost"

US

Selected Wireless Interface: ath0

Channel Width: 20 MHz

Physical: 802.11 B

Channel: 2417

TxRate: 2 Mbps

Frag: ☐ Enable

RTS: ☐ Enable

Diversity: ☑ Enable

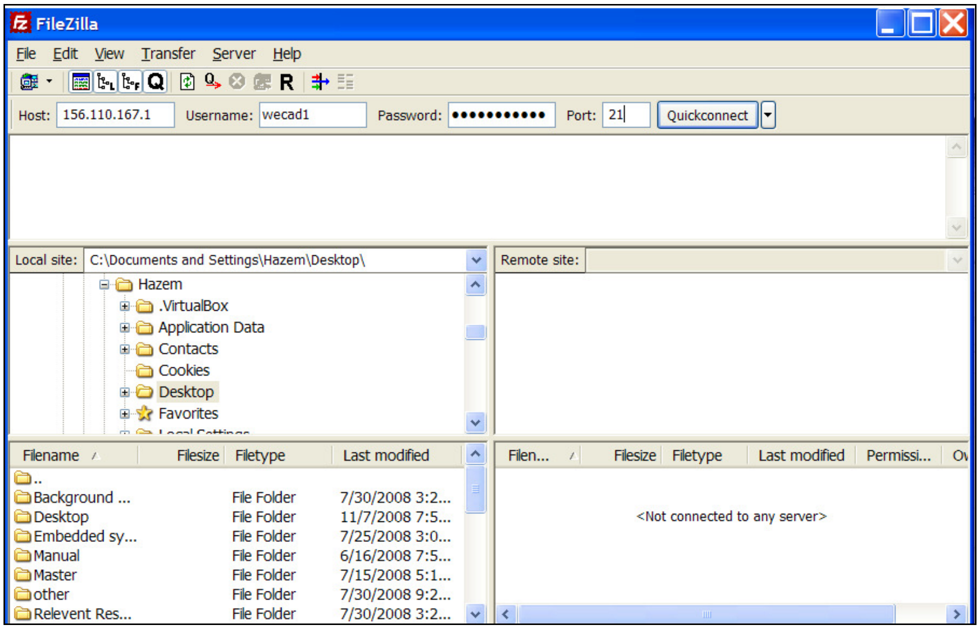
Antenna: A-(MAIN)

Tx Power: 5 ☐ Override 10 dBm

Short Preamble: ☑ Enable

Figure 14. Picture of the FDDI radio (left) and Ikarus-OS manager GUI (right).

After the radios were configured, it was necessary to obtain a means of transferring a file between the radios, which was accomplished using a free FTP solution, *Filezilla* (filezilla-project.org). This software provides a robust means of establishing FTP transfers over a secure wireless link. The *Filezilla* client was installed on a computer that was connected to one of the radios. The server software was installed on another computer connected to the other radio. After IP, user, port, and security information was established for both the client and server, an image file was successfully transferred wirelessly from one computer to another. The *Filezilla* user interfaces for both client and server are shown in Figures 15 and 16 respectively. It is important to note that the IP addresses of the server node must align with that of the host IP (156.110.167.X).



Host: 156.110.167.1 Username: wecad1 Password: ●●●●●●●●●●● Port: 21

Local site: C:\Documents and Settings\Hazem\Desktop\

Remote site:

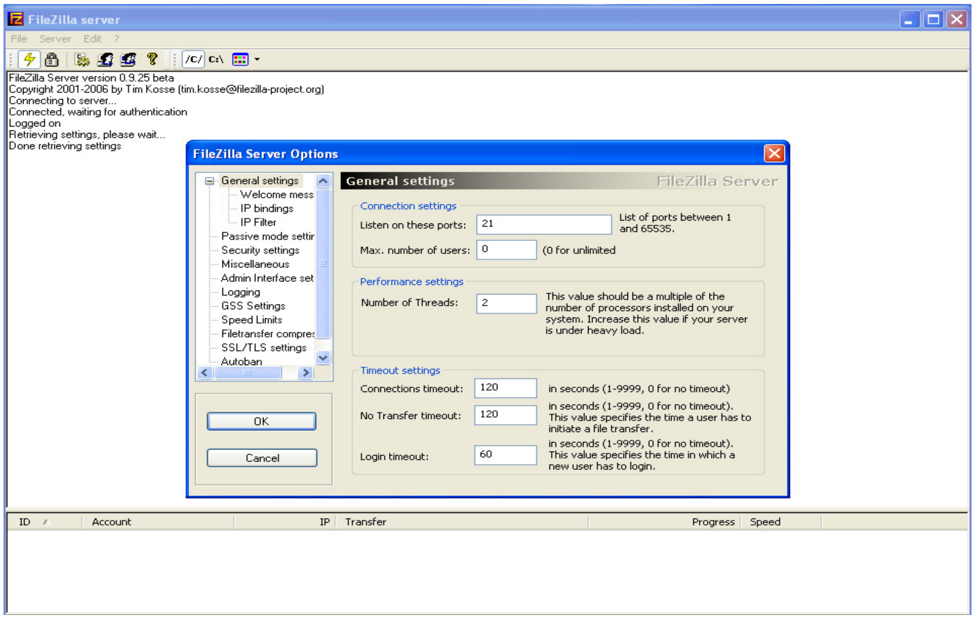
- Hazem
  - .VirtualBox
  - Application Data
  - Contacts
  - Cookies
  - Desktop
  - Favorites

Filename	Filesize	Filetype	Last modified
..			
Background ...		File Folder	7/30/2008 3:2...
Desktop		File Folder	11/7/2008 7:5...
Embedded sy...		File Folder	7/25/2008 3:0...
Manual		File Folder	6/16/2008 7:5...
Master		File Folder	7/15/2008 5:1...
other		File Folder	7/30/2008 9:2...
Relevent Res...		File Folder	7/30/2008 3:2...

Filen...	Filesize	Filetype	Last modified	Permissi...	Ow

<Not connected to any server>

Figure 15. Screenshot of the *Filezilla Client* application.



```
FileZilla Server version 0.9.25 beta
Copyright 2001-2006 by Tim Kosse (tim.kosse@filezilla-project.org)
Connecting to server...
Connected, waiting for authentication
Logged on
Retrieving settings, please wait...
Done retrieving settings
```

General settings

Connection settings

Listen on these ports: 21 — List of ports between 1 and 65535.

Max. number of users: 0 — (0 for unlimited

Performance settings

Number of Threads: 2 — This value should be a multiple of the number of processors installed on your system. Increase this value if your server is under heavy load.

Timeout settings

Connections timeout: 120 — in seconds (1-9999, 0 for no timeout)

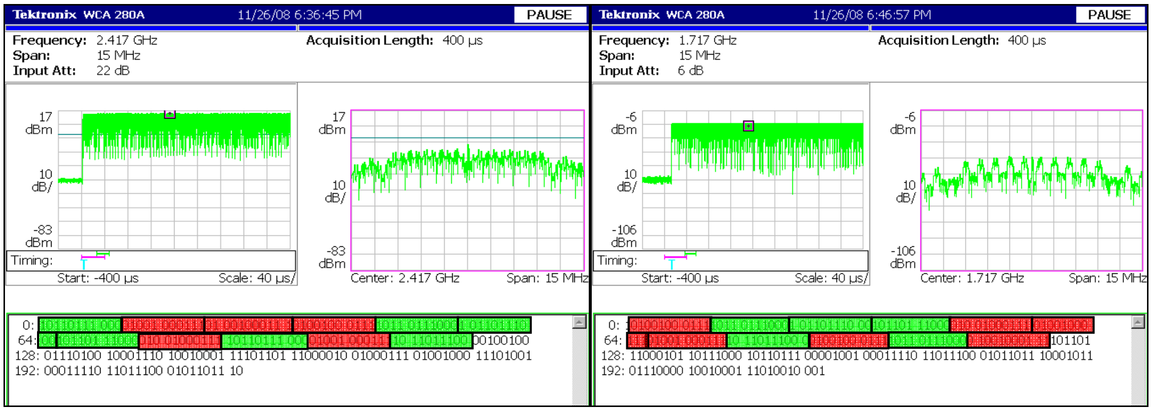
No Transfer timeout: 120 — in seconds (1-9999, 0 for no timeout). This value specifies the time a user has to initiate a file transfer.

Login timeout: 60 — in seconds (1-9999, 0 for no timeout). This value specifies the time in which a new user has to login.

Figure 16. Screenshot of the *Filezilla Server* application.

3.6 Evaluation of a DSSS System Using a Communication Analyzer

A proof-of-concept design of an 802.11b frequency translation system was developed, and measurements were taken to validate its performance at 1 Mbps. The test setup includes a signal generator to provide a local oscillator, an 802.11b AP to generate a periodic beacon signal, and a communication analyzer to determine if the original signal has been adequately received and decoded. Because 802.11b utilizes DSSS, verification is greatly simplified. The AP is set to the low data rate of 1 Mbps, and the analyzer is set to BPSK demodulation. If the analyzer can correctly decode the spreading sequences, then bit decoding would be straightforward for a typical receiver. The analyzer is set to decode 11 Mbps BPSK. In this scenario the decoded signal should consist of chipping sequences. The sequences of interest are shown in Table 1, and Figure 17 shows the proper decoding of the original 802.11b signal at 2.417 GHz and the signal down-converted to 1.717 GHz. The 2 Mbps data rate should work in exactly the same fashion except now the data is DQPSK modulated instead of DBPSK modulated. The only difference in the test methodology for this modulation type is to seek out four different 22-bit sequences rather than two different 11-bit sequences. The bit-mapping for 2 Mbps is shown in Table 2; however, a communication link must be established to generate 2 Mbps DSSS (the beacon is always 1 Mbps). Consequently this data rate has not yet been tested.



```
Frequency:  2.417 GHz          Acquisition Length:  400 µs
Span:       15 MHz
Input Att:  22 dB

0:   10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
64:  00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 10
```

```
Frequency:  1.717 GHz          Acquisition Length:  400 µs
Span:       15 MHz
Input Att:  6 dB

0:   10100100 01111011 01110001 01101110 00101101 11000010 01000111 01001000
64:  11111001 00011110 11011100 01100100 01111011 01110001 10010001 11101101
128: 11000101 10111000 10110111 00001001 00011110 11011100 01011011 10001011
192: 01110000 10010001 11010010 001
```

Figure 3. Decoding of spreading sequence at 2.417 GHz (left) and 1.717 GHz (right)

Table 1. 1 Mbps chipping sequence bit map.

Bits	Chipping Sequence
0	01001000111
1	10110111000

Table 2. 2 Mbps chipping sequence bit map.

Bits	Chipping Sequence
00	0011000011000000111111
11	1100111100111111000000
01	0110010110010101101010
10	1001101001101010010101

3.6.1 Forward Link Testing

One-way communication analysis is the first step in evaluating the validity of the FCE architecture. For purposes of discussion, we refer to the forward link as the link in which the signal passes directly from the wireless device to the directional coupler. The forward link is further described in Figure 18. A communication analyzer was employed to characterize the signal as it passes through components along the forward link. The results of this analysis can be found in Figures 19-21. It is clear from the figures that the signal was properly decoded, but they also show that the RF-LO isolation of the mixer was not sufficient. This can be seen in Figure 19 where the leakage to the LO port of the mixer is significant at 2.417 GHz. This leakage was also found through the splitter to the second mixing stage, causing multiple product frequencies that should not have been present at the output of the FCE. To resolve this issue, a filter was placed at the LO input of the mixer to increase the effective RF-LO isolation and eliminate this leakage.

Another problem, which can be seen in Figures 20 and 21, is that the delay in switching causes part of the packet to be lost. It also results in the transmitted signal being fed to the wrong port (Port 1) for several microseconds before the proper path is enabled. However, the isolation between the output and input port of the amplifier is high and prevents this problem from propagating against the receive path. The time delays and 2.4 GHz leakage are believed to be the primary reason why two-way communication through the wired FCE was not established.

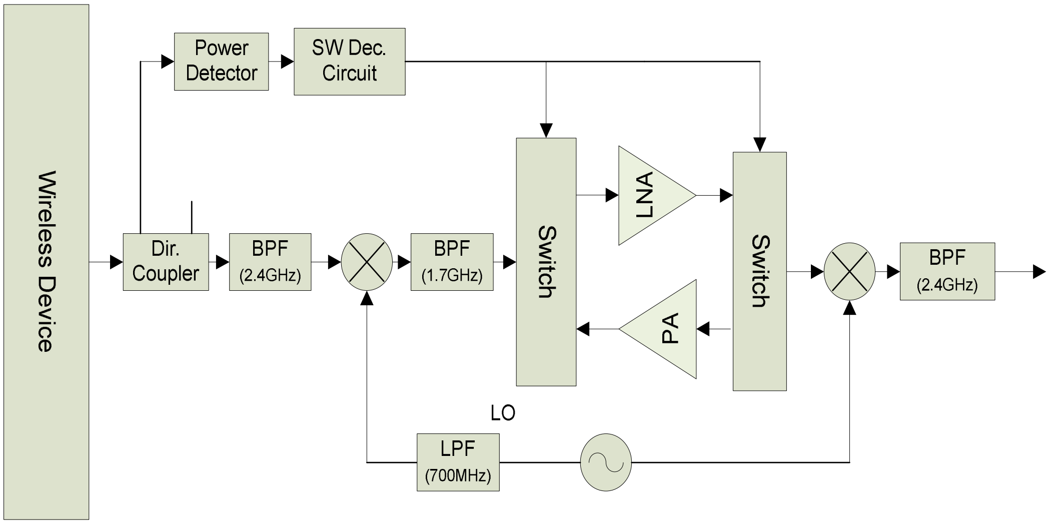
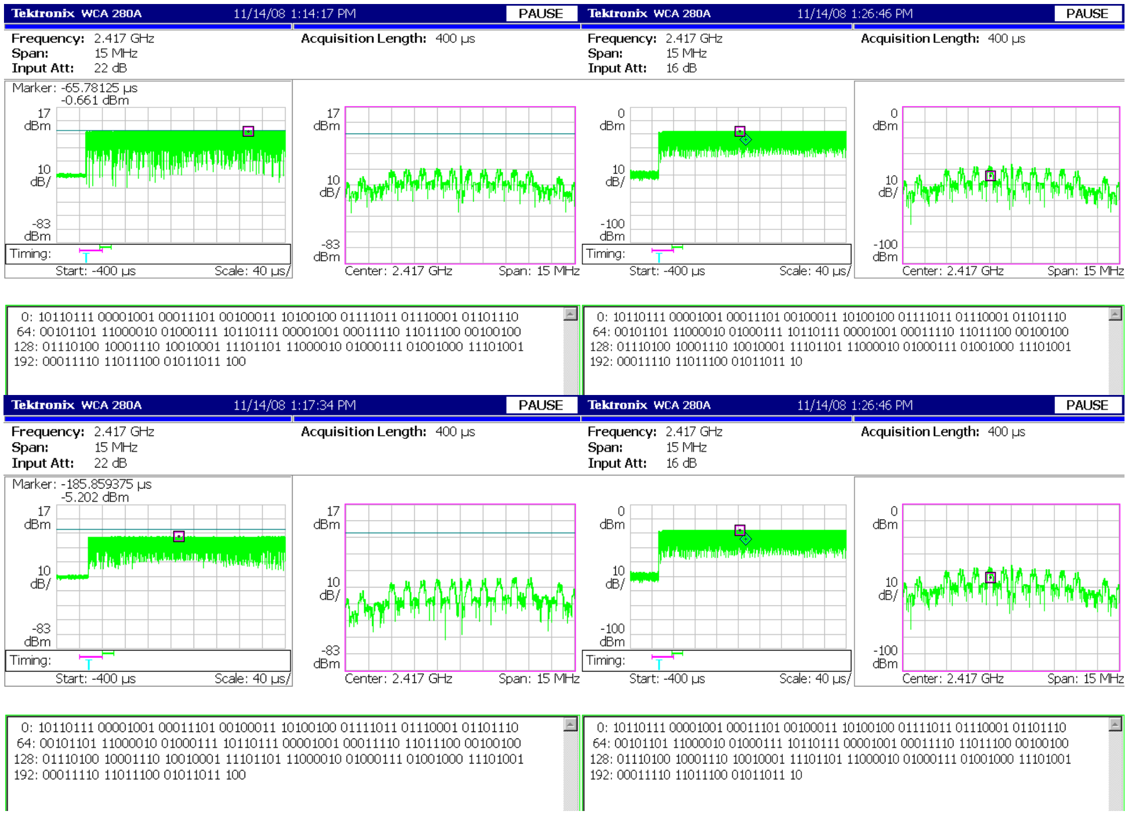


Figure 18. Forward link architectural description.



Tektronix WCA 280A 11/14/08 1:14:17 PM

Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 22 dB
Acquisition Length: 400 µs

Marker: -65.78125 µs
-0.661 dBm

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 100
```

Tektronix WCA 280A 11/14/08 1:26:46 PM

Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 16 dB
Acquisition Length: 400 µs

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 10
```

Tektronix WCA 280A 11/14/08 1:17:34 PM

Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 22 dB
Acquisition Length: 400 µs

Marker: -185.859375 µs
-5.202 dBm

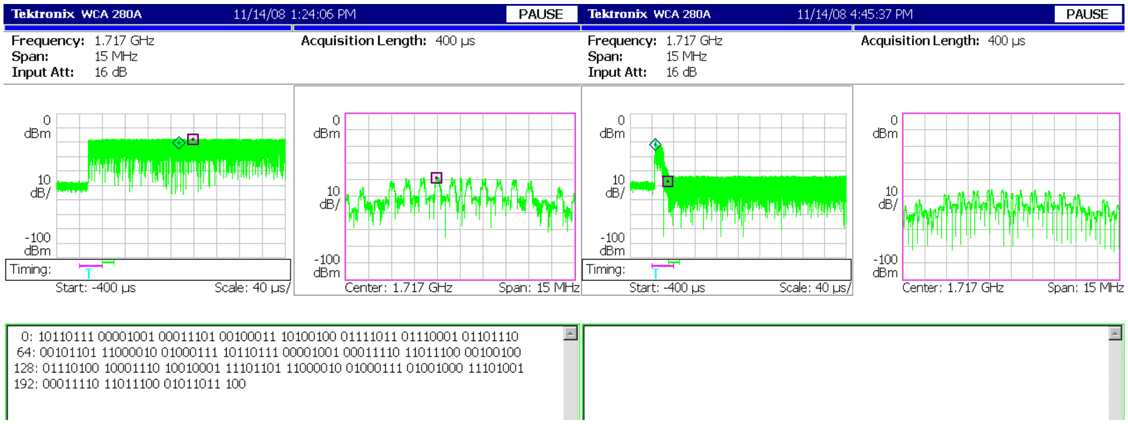
```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 100
```

Tektronix WCA 280A 11/14/08 1:26:46 PM

Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 16 dB
Acquisition Length: 400 µs

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 10
```

Figure 19. First stage of analysis in forward link. Analyzer output from 802.11b radio (top left), coupled to power detector (top right), from DC coupler and 2.4 GHz band-pass filter (bottom left), and leakage to LO port at 2.417 GHz (bottom right).



Tektronix WCA 280A 11/14/08 1:24:06 PM

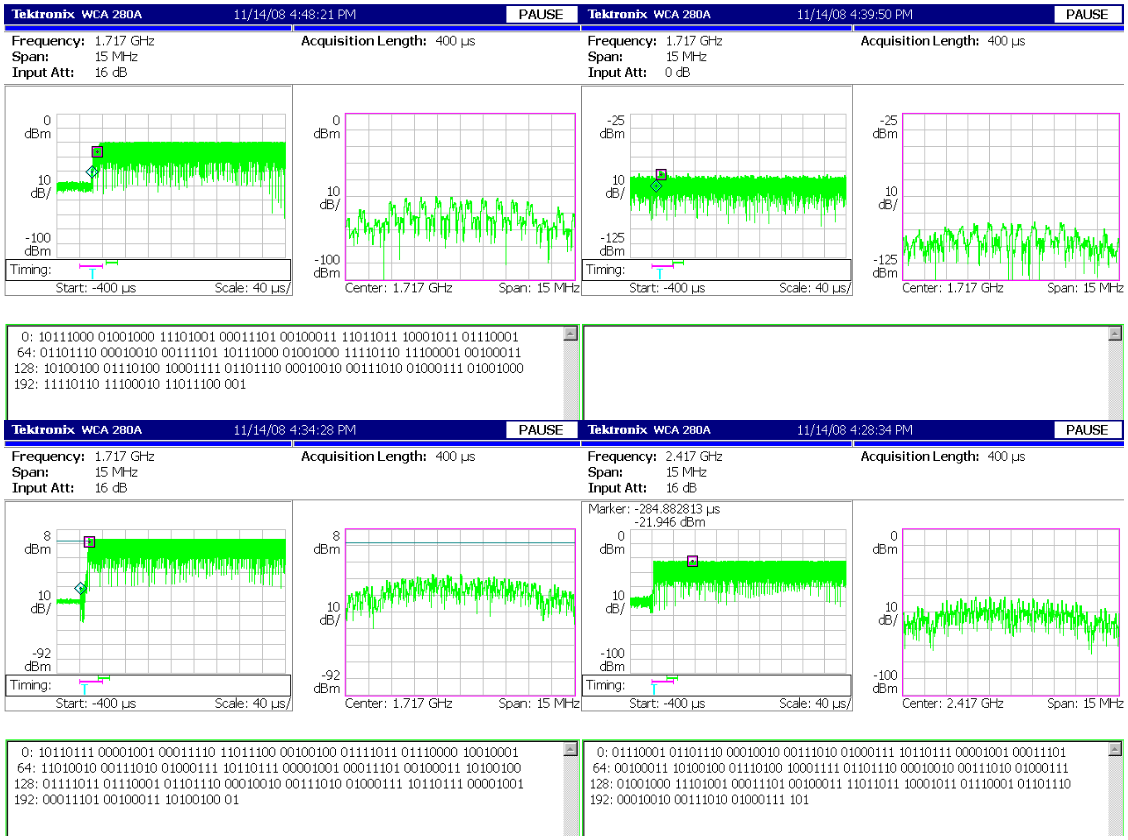
Frequency: 1.717 GHz
Span: 15 MHz
Input Att: 16 dB
Acquisition Length: 400 µs

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 100
```

Tektronix WCA 280A 11/14/08 4:45:37 PM

Frequency: 1.717 GHz
Span: 15 MHz
Input Att: 16 dB
Acquisition Length: 400 µs

Figure 20. Second stage of analysis in forward link. Analyzer output from first mixer (left) and port 1 of first switch (right).



```
Tektronix WCA 280A   11/14/08 4:48:21 PM   PAUSE
Frequency:  1.717 GHz      Acquisition Length:  400 µs
Span:       15 MHz
Input Att:  16 dB
Start: -400 µs   Scale: 40 µs/      Center: 1.717 GHz   Span: 15 MHz

  0: 10111000 01001000 11101001 00011101 00100011 11011011 10001011 01110001
 64: 01101110 00010010 00111101 10111000 01001000 11110110 11100001 00100011
128: 10100100 01110100 10001111 01101110 00010010 00111010 01000111 01001000
192: 11110110 11100010 11011100 001
```

```
Tektronix WCA 280A   11/14/08 4:39:50 PM   PAUSE
Frequency:  1.717 GHz      Acquisition Length:  400 µs
Span:       15 MHz
Input Att:  0 dB
Start: -400 µs   Scale: 40 µs/      Center: 1.717 GHz   Span: 15 MHz
```

```
Tektronix WCA 280A   11/14/08 4:34:28 PM   PAUSE
Frequency:  1.717 GHz      Acquisition Length:  400 µs
Span:       15 MHz
Input Att:  16 dB
Start: -400 µs   Scale: 40 µs/      Center: 1.717 GHz   Span: 15 MHz

  0: 10110111 00001001 00011110 11011100 00100100 01111011 01110000 10010001
 64: 11010010 00111010 01000111 10110111 00001001 00011101 00100011 10100100
128: 01111011 01110001 01101110 00010010 00111010 01000111 10110111 00001001
192: 00011101 00100011 10100100 01
```

```
Tektronix WCA 280A   11/14/08 4:28:34 PM   PAUSE
Frequency:  2.417 GHz      Acquisition Length:  400 µs
Span:       15 MHz
Input Att:  16 dB
Marker: -284.882813 µs
        -21.946 dBm
Start: -400 µs   Scale: 40 µs/      Center: 2.417 GHz   Span: 15 MHz

  0: 01110001 01101110 00010010 00111010 01000111 10110111 00001001 00011101
 64: 00100011 10100100 01110100 10001111 01101110 00010010 00111010 01000111
128: 01001000 11101001 00011101 00100011 11011011 10001011 01110001 01101110
192: 00010010 00111010 01000111 101
```

Figure 21. Final stage of analysis in forward link. Analyzer output from port 2 of first switch (top left), leaked to input of power amplifier (top right), output from LNA (bottom left), and output from second mixer at 2.417 GHz (bottom right).

3.6.2 Reverse Link Testing

Reverse link testing yielded results that generally mirrored that of forward link testing. The signal flow in the reverse direction is shown in Figure 22. The signal strengths are slightly higher because a PA with higher gain was used in this path (PA and LNA were the only available components at time of testing). The signal was properly decoded at each step along the reverse path as illustrated in Figures 23-25.

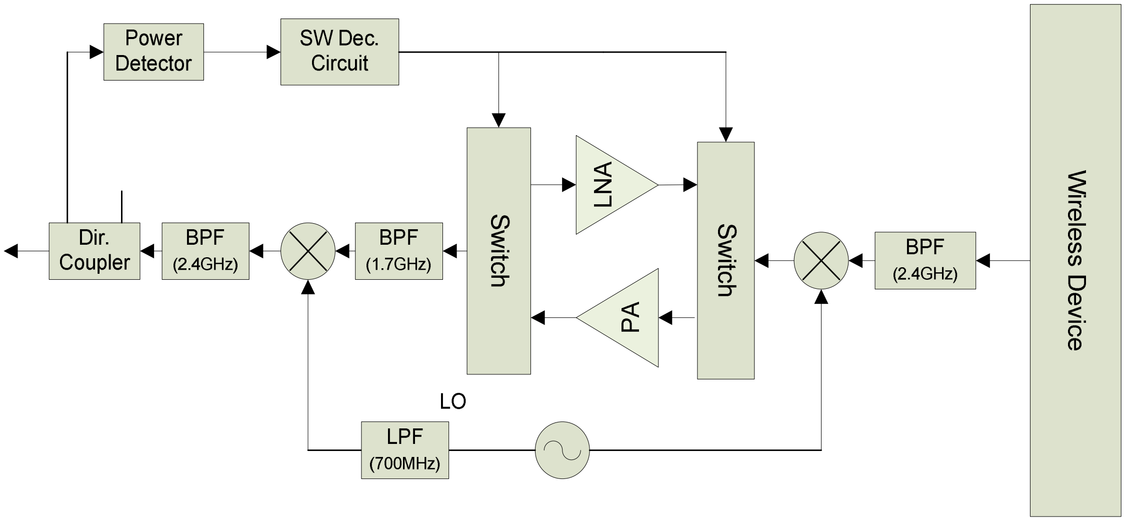
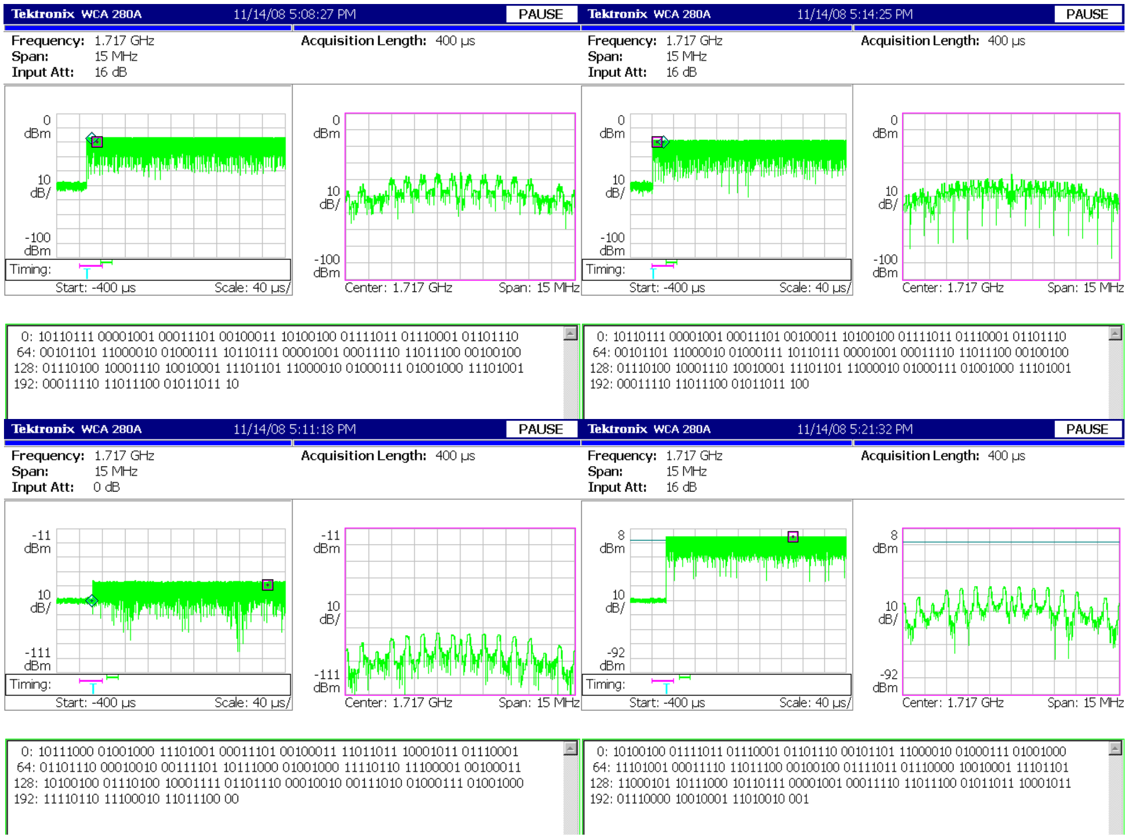


Figure 22. Reverse link architectural description.



```
Tektronix WCA 280A    11/14/08 5:08:27 PM    PAUSE
Frequency: 1.717 GHz    Acquisition Length: 400 µs
Span: 15 MHz
Input Att: 16 dB
Start: -400 µs    Scale: 40 µs/    Center: 1.717 GHz    Span: 15 MHz
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 10
```

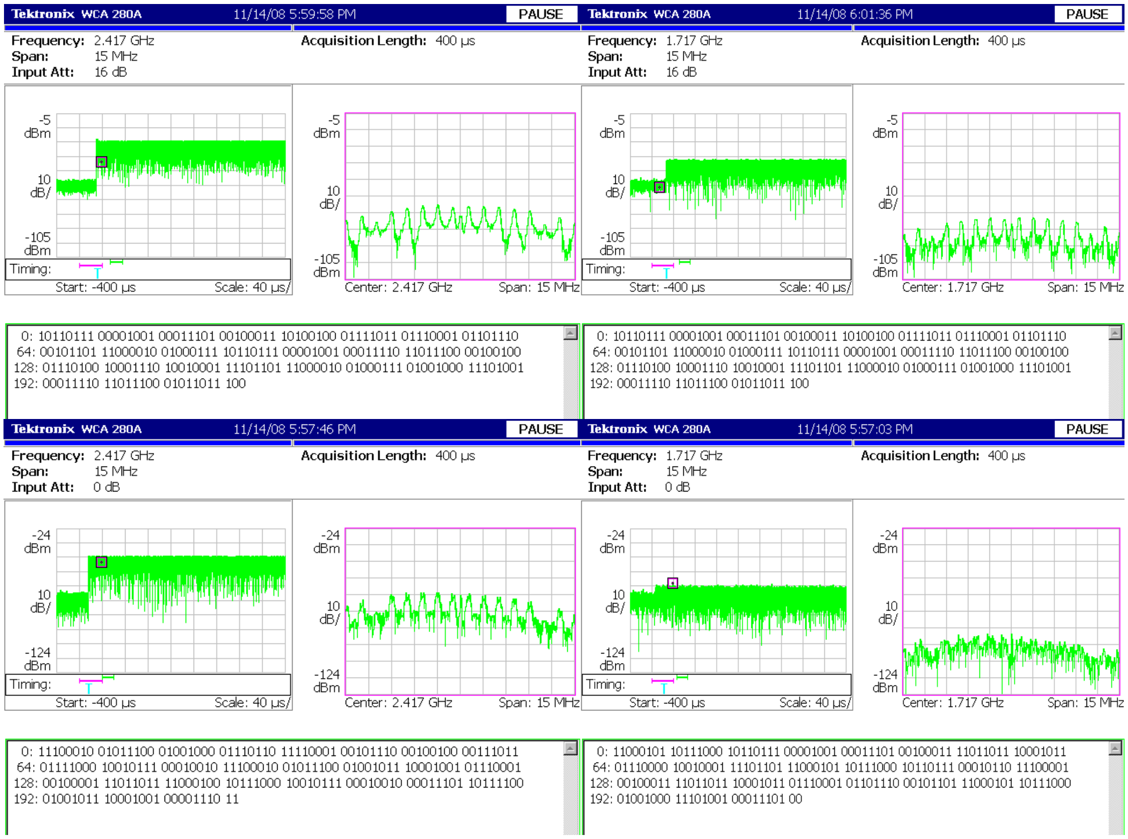
```
Tektronix WCA 280A    11/14/08 5:14:25 PM    PAUSE
Frequency: 1.717 GHz    Acquisition Length: 400 µs
Span: 15 MHz
Input Att: 16 dB
Start: -400 µs    Scale: 40 µs/    Center: 1.717 GHz    Span: 15 MHz
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 100
```

```
Tektronix WCA 280A    11/14/08 5:11:18 PM    PAUSE
Frequency: 1.717 GHz    Acquisition Length: 400 µs
Span: 15 MHz
Input Att: 0 dB
Start: -400 µs    Scale: 40 µs/    Center: 1.717 GHz    Span: 15 MHz
  0: 10111000 01001000 11101001 00011101 00100011 11011011 10001011 01110001
 64: 01101110 00010010 00111101 10111000 01001000 11110110 11100001 00100011
128: 10100100 01110100 10001111 01101110 00010010 00111010 01000111 01001000
192: 11110110 11100010 11011100 00
```

```
Tektronix WCA 280A    11/14/08 5:21:32 PM    PAUSE
Frequency: 1.717 GHz    Acquisition Length: 400 µs
Span: 15 MHz
Input Att: 16 dB
Start: -400 µs    Scale: 40 µs/    Center: 1.717 GHz    Span: 15 MHz
  0: 10100100 01111011 01110001 01101110 00101101 11000010 01000111 01001000
 64: 11101001 00011110 11011100 00100100 01111011 01110000 10010001 11101101
128: 11000101 10111000 10110111 00001001 00011110 11011100 01011011 10001011
192: 01110000 10010001 11010010 001
```

Figure 23. First stage of analysis in reverse link. Analyzer output from first mixer and 1.717 GHz filter (top left), port 1 of first switch (top right), port 2 of first switch (bottom left), and PA after switch and filter (bottom right).

Operation in the reverse direction is largely as expected, and in particular, the coupled power in this direction is greatly below threshold (Figure 24). However, a significantly strong 1.717 GHz signal can be seen at the output of the FCE in Figure 25. While this may not pose a significant problem for the 802.11b device because it has band-pass filters centered around 2.4 GHz, it will more than likely be addressed by adding additional filters in subsequent designs.



Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 16 dB
Acquisition Length: 400 µs

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 100
```

Frequency: 1.717 GHz
Span: 15 MHz
Input Att: 16 dB
Acquisition Length: 400 µs

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 100
```

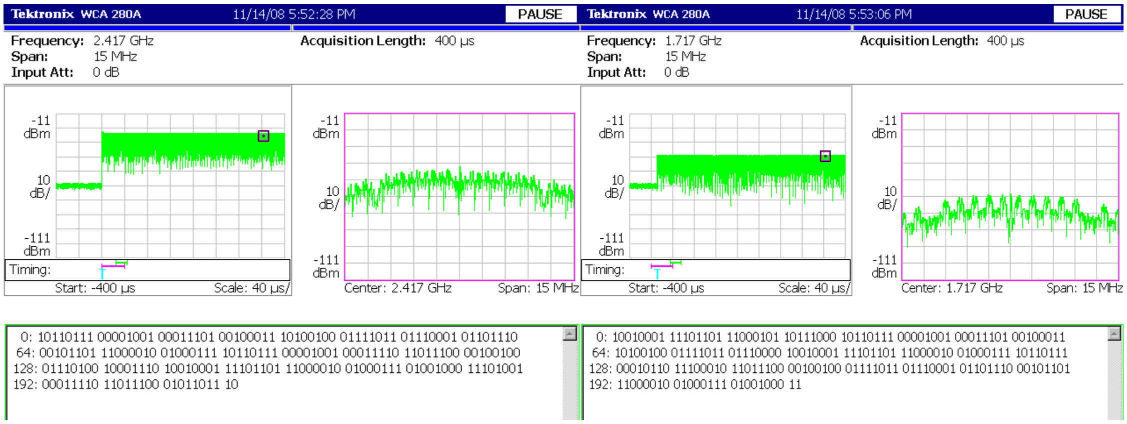
Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 0 dB
Acquisition Length: 400 µs

```
  0: 11100010 01011100 01001000 01110110 11110001 00101110 00100100 00111011
 64: 01111000 10010111 00010010 11100010 01011100 01001011 10001001 01110001
128: 00100001 11011011 11000100 10111000 10010111 00010010 00011101 10111100
192: 01001011 10001001 00001110 11
```

Frequency: 1.717 GHz
Span: 15 MHz
Input Att: 0 dB
Acquisition Length: 400 µs

```
  0: 11000101 10111000 10110111 00001001 00011101 00100011 11011011 10001011
 64: 01110000 10010001 11101101 11000101 10111000 10110111 00010110 11100001
128: 00100011 11011011 10001011 01110001 01101110 00101101 11000101 10111000
192: 01001000 11101001 00011101 00
```

Figure 24. Second stage of analysis in reverse link. Analyzer output from second mixer after filter at 2.417 GHz (top left), second mixer after filter at 1.717 GHz (top right), coupling to power detector at 2.417 GHz (bottom left), and coupling to power detector at 1.717 GHz (bottom right).



Frequency: 2.417 GHz
Span: 15 MHz
Input Att: 0 dB
Acquisition Length: 400 µs

```
  0: 10110111 00001001 00011101 00100011 10100100 01111011 01110001 01101110
 64: 00101101 11000010 01000111 10110111 00001001 00011110 11011100 00100100
128: 01110100 10001110 10010001 11101101 11000010 01000111 01001000 11101001
192: 00011110 11011100 01011011 10
```

Frequency: 1.717 GHz
Span: 15 MHz
Input Att: 0 dB
Acquisition Length: 400 µs

```
  0: 10010001 11101101 11000101 10111000 10110111 00001001 00011101 00100011
 64: 10100100 01111011 01110000 10010001 11101101 11000010 01000111 10110111
128: 00010110 11100010 11011100 00100100 01111011 01110001 01101110 00101101
192: 11000010 01000111 01001000 11
```

Figure 25. Final stage of analysis in reverse link. Analyzer output from FCE at 2.417 GHz (left) and 1.717 GHz (right).

4. Physical and MAC Layers of 802.11b/g

4.1 Physical Layer Aspects of the 802.11b Protocol

In order to understand the impact of a truncated 802.11b packet, it is necessary to briefly discuss the physical and MAC layers of this protocol. Two specific conditions must hold true for 802.11b to continue to function as intended. First, the physical properties of the transmitted signal must not be significantly changed due to factors such as noise, interference, or additional harmonics generated by the FCE. Second, the truncation of a small portion of the 802.11b preamble must not significantly impact synchronization, gain control, and channel estimation mechanisms.

802.11b operates in the popularized 2.4 GHz ISM band using DSSS modulation for the lower data rates and CCK modulation for higher data rate transmission. As with most wireless protocols, there is a training interval (preamble) that is used to enable frequency and timing synchronization, automatic gain control, channel estimation, and other measurements to compensate disparities between the receiver and transmitter and/or the effects of the wireless channel.

4.1.1 DSSS Modulation

Direct Sequence Spread Spectrum (DSSS) is a modulation technique used to spread information over a wide range of frequencies, thereby limiting its interference on other nearby wireless devices. Each symbol consists of eleven chips, where each chip corresponds to a logical 1 or 0. It is similar to BPSK except there is a mapping of eleven ones and zeros to one bit as opposed to the 1:1 ratio seen in BPSK. Decoding at the receiver can be achieved by correlating the received sequence with the chipping code.

The 802.11b protocol uses DSSS for its 1 Mbps and 2 Mbps transmission rates. The initial data stream is differentially encoded (using DBPSK modulation) before being mixed with the chipping sequence: 10110111000. This is sometimes accomplished by simply sending the chipping sequence and differentially encoded data stream through an XOR gate yielding two possible 11-chip sequences: 10110111000 or 01001000111. The chipping rate for 802.11b is 11 Mbps and base data stream rate is 1 Mbps. This translates to 1 symbol (11 chips) being transmitted every microsecond, or a 1 Mbps gross data rate For the 2 Mbps data rate specified in the standards, the data stream is sent through a 1:2 serial to parallel converter so that the two streams can be DQPSK modulated prior to the chipping sequence mapping, doubling the previous data rate. An overview of this behavior can be seen in Figure 26.

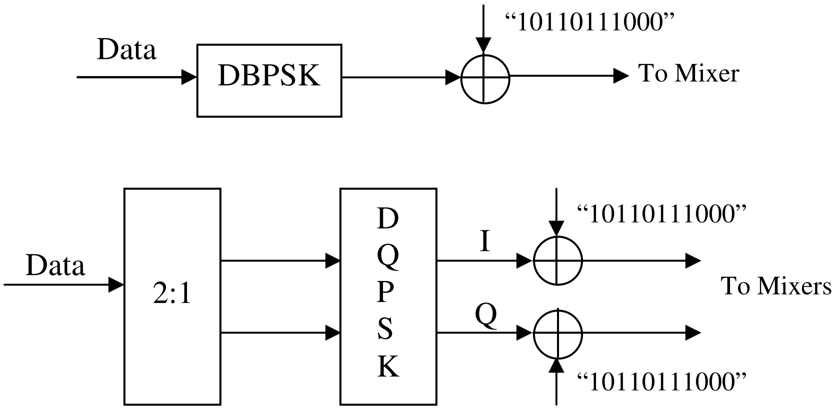


Figure 26. Example implementation of DSSS in 802.11b for 1 Mbps (top) and 2 Mbps (bottom).

4.1.2 CCK Modulation

Higher data rates of 5.5 Mbps and 11 Mbps are achieved using Complementary Code Keying (CCK) modulation. CCK uses a set of chips that have good correlation properties with one another (meaning they are nearly orthogonal to one another). For the 5.5 Mbps data rate, four bits are encoded in a symbol. The first two bits are used to select one of four, 8-chip-long CCK codewords. The second pair of bits is then used to DQPSK modulate the codeword. To maintain the spectral properties of 802.11b, the chipping rate is set to 1.375 Msym/s, which yields: 8*1.375 Msym/s= 11 Mchips/s. Recalling that these eight chips encode four bits of information, the net data rate is given by 4*1.375 = 5.5 Mbps. The only difference between this data rate and the 11 Mbps data rate is that six bits are used to select one of 64 codewords (rather than one of four), sacrificing some unwanted correlation for a higher data rate. This yields a data rate of 8*1.375Mbps = 11 Mbps. The procedures described above are summarized in Figure 27.

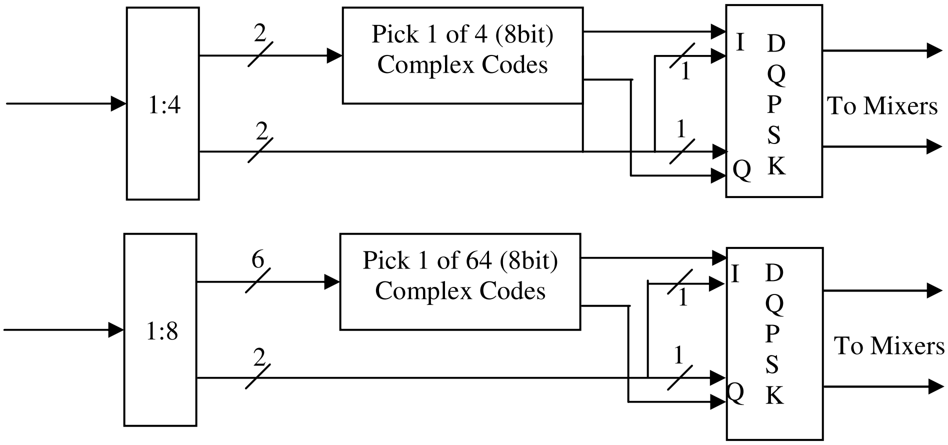
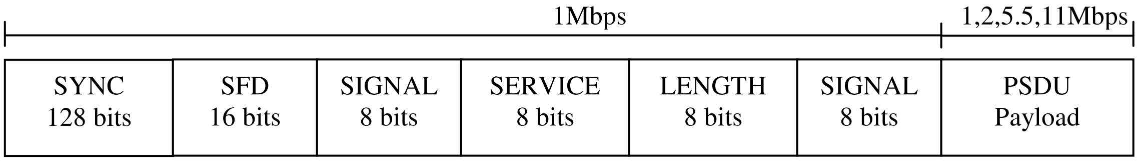


Figure 27. CCK modulation to achieve 5.5 Mbps (top) and 11 Mbps (bottom) in 802.11b.

4.1.3 Packet Structure

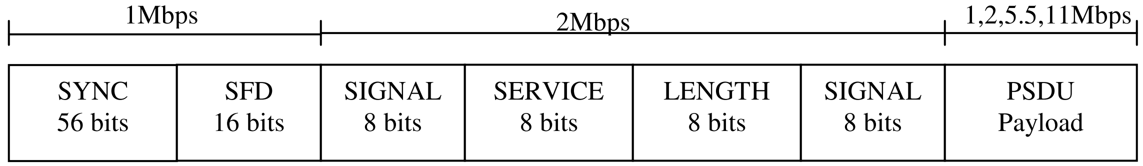
The packet structure in the 802.11b protocol takes one of two possible forms. The first contains what is known as the long preamble, a grouping of 128 synchronization bits with a 16-bit start of frame delimiter (SFD). These are then followed by additional overhead such as the length and service fields, as well as the transmitted data. Of particular interest are the initial fields of the packet as these sections will be truncated by any switching delay seen in the final design of the device. Of particular importance is the continuity of the SYNC field. This field contains alternating “1”s and “0”s such that if any portion of it is lost in transmission, essential synchronization and gain control functions can occur after the truncated bits without a change in operation.

The second packet structure is identical to the one described above except it utilizes a short preamble that consists of 56 SYNC bits and additional overhead is transmitted at 2 Mbps. This structure was developed to reduce overhead and improve effective throughput of an 802.11b wireless network. In so doing it has reduced the SYNC field to 44% of its original length. This means that failure to receive the preliminary bits in the preamble will have a more significant impact on the synchronization process than in the long preamble. This relationship will be explored more thoroughly in later chapters. A more thorough description of the packet types and their rates of transmission can be found in Figures 28 and 29 respectively.



1Mbps						1,2,5.5,11Mbps
SYNC 128 bits	SFD 16 bits	SIGNAL 8 bits	SERVICE 8 bits	LENGTH 8 bits	SIGNAL 8 bits	PSDU Payload

Figure 28. 802.11b long preamble packet structure.



1Mbps		2Mbps				1,2,5.5,11Mbps
SYNC 56 bits	SFD 16 bits	SIGNAL 8 bits	SERVICE 8 bits	LENGTH 8 bits	SIGNAL 8 bits	PSDU Payload

Figure 29. 802.11b short preamble packet structure.

4.2 Preamble Operations of 802.11b

The preamble of 802.11b contains information needed to perform frequency synchronization, timing synchronization, automatic gain control, and channel estimation. To this end a brief discussion on the processes by which these operations take place are provided in the subsequent subsections.

4.2.1 Frequency Synchronization

Frequency synchronizations techniques for DSSS are usually very similar to their narrowband counterparts. As with any received signal there can exist a phase and frequency offset that can adversely impact the decoded message. This problem is addressed by estimating the offset and adjusting the frequency/phase of the receiver's oscillator, thereby eliminating any discrepancy. Unlike narrowband signals, the frequency synchronization process occurs simultaneously with the spreading sequence synchronization needed to determine bit timing as discussed in the next section.

A number of techniques exist to achieve frequency synchronization. Without getting into great detail, two of the more popular techniques are the phased-locked loop and Costas loop. The phase-locked loop approach generates an error signal based on the phase/frequency difference between the received carrier and receiver's local oscillator, modifies the resulting error signal using a loop filter, and adaptively adjusts the frequency of the local oscillator until synchronization has been achieved. Conversely the digital Costas loop uses the output from the I and Q base-band signals to generate a phase error signal that is directly proportional to the phase error. The initial phase of the local oscillator can then be adjusted through software. With regards to the goals of this work, it is important to ensure that the conversion device does not significantly reduce the time-to-synchronize (TTS) of the base device or significantly impact the frequency offset between the received signal and the local oscillator of the base device.

4.2.2 Symbol Synchronization in 802.11b

As previously mentioned timing and frequency synchronization occur simultaneously in 802.11b transceivers. Symbol timing retrieval is made easier by the fact that DBPSK is used as the frequency does not necessarily have to be known to reach achieve symbol synchronization. In most circumstances, correlation measurements between the received training sequence of the preamble and the known spreading sequence are computed. This process is usually broken down into two stages known as coarse and fine tracking. Recalling the synchronization procedures discussed in Section 2.4, during the coarse tracking procedure the known sequence is correlated with the received sequence and a correlation value is computed. The sequence is then shifted a half duration of a chip at a time to retrieve all possible sequence overlaps within a half chip accuracy. The receiver is said to have reached coarse synchronization when it has selected the sequence orientation that maximizes the correlation value. Usually this value must also exceed a base threshold to ensure a valid signal is present.

Since the spreading sequence for 802.11b is 11 chips long, it requires a minimum of 22 shifts to achieve coarse synchronization. As each sequence in the preamble is transmitted at a rate of 1 Msym/s, the worst-case scenario requires a minimum of 22 μs to achieve coarse synchronization. This synchronization requirement can be relaxed if the correlations are performed in parallel; however, this approach is less used due to the increased cost, form-factor, and complexity it introduces. Fine synchronization is typically achieved by applying a delay-locked loop (DLL) approach to gradually align the receiver's spreading code with that of the training sequence.

The abovementioned process is similar to that of the PLL used in carrier synchronization except only the phase of the sequence (not the frequency) is altered. This can be achieved by using "early" and "late" correlation values to generate a composite correlation function [78] that after being low-pass filtered has a linear operating range. After some time the early and late correlations are driven to be equal and the "current" sequence has achieved fine synchronization. The speed and stability of convergence is directly related to the gain of the filter. The parameter of interest with regards to the converter design is the convergence time. As such convergence time (the parameter of interest with regards to the proposed converter) is largely design dependent and can vary depending on the receiver design.

4.2.3 Automatic Gain Control and Channel Equalization

Frequency and timing synchronization processes operate under the assumption that the signal has an appropriate amplitude range. To this end, automatic gain control (AGC) is used to ensure that this is the case. With regards to an 802.11b extension, two observations need to be made. First, provided the extension does not have an overall time-varying gain, the gain control operation of the base device should not be meaningfully affected. Secondly, the AGC can introduce an additional delay in reception in some scenarios as there are multiple thresholds that must be met to further signal reception. The AGC can have a maximum response time that generally ranges up to several microseconds, as is the case for the popular MAX2820 802.11b IF receiver chip developed by *Maxim IC* [79].

Similarly, the channel itself can alter the intended structure of the received signal due to inter-symbol interference (ISI) caused by multi-path. Frequency selective fading also plagues systems operating over a wide bandwidth, as is the case for 802.11b. Typical rms delay spreads for 802.11b environments are less than 100 ns [80] due to their intended indoor environment. This range of delay spreads can typically be handled by a RAKE receiver [80]. However, the competitive WLAN market has driven improved WLAN receivers that either replace the RAKE receiver with an equalizer or implement a combined design to improve its susceptibility to multi-path. Given the presence of an equalizer time is needed to train the filter coefficients before channel effects can be negated. Generally speaking, this should have a greater impact on fine timing synchronization than coarse or frequency synchronization due to the small delays involved, but can also extend the needed number of training symbols in the preamble of 802.11b. The equalization process usually runs concurrently with timing synchronization processes.

4.2.4 Comparison of Parallel and Serial Preamble Operations

In 802.11b the entire synchronization field has a uniform format of alternating (scrambled) “1”s and “0”s. This means that if a portion of this field is lost in transmission, the format of the SYNCH field remains the same regardless of the point in time at which reception resumes. As a consequence, if preamble procedures occur in parallel then there should be little or no difference in operation provided that there are enough remaining training symbols to complete the desired operations. If instead the SYNC field is further subdivided into intervals during which different processes take place serially, any individual process that is not given sufficient can cause reception to fail. For example, the 802.11a/g protocols are serial in nature because it is subdivided into short and long training symbols.

4.3 Operation of the 802.11b MAC in Data Transmission

The 802.11b protocol uses the standard 802.11 MAC with CSMA (and the optional CSMA/CA with RTS-CTS exchange). After joining a network, nodes employ CSMA to determine when the channel is free for transmission. They then wait for a predefined frame spacing, in most typical applications the DIFS which is 50 µs, before contending for access to the channel. Contention begins with 16 time slots that each last 20 µs. The nodes wait a random back-off number of slots and begin transmission with the node with the lowest wait time gaining access to the channel. The other nodes having heard this transmission delay their transmission to the next contention window. The contention process is summarized in Figure 30.

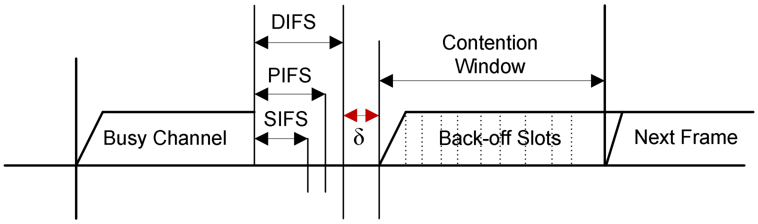


Figure 30. 802.11 MAC contention process.

In the event that transmission does not take place due to a collision, the contention window is doubled and a binary exponential back-off (BEB) time is allocated to the sending nodes. This process continues until the contention window reaches a maximum length of 256 slots. After a successful transmission the window is set back to its minimum value (usually 16 or 32 slots). For most practical purposes large contention only occur in very dense networks with a large number of competing nodes.

In networks with multiple nodes it may be advantageous to employ an RTS-CTS mechanism and the 802.11b standard allows for this option. After a successful contention slot has been selected, the transmitting node sends an RTS to the receiving node. The receiver waits for the SIFS (10 μs) and then sends a CTS signal to signify it is ready for reception. Nodes hearing the RTS or CTS update their Network Allocation Vector (NAV) so as not to transmit and interfere with the sending and receiving node for a predefined length of time. After receiving the CTS and waiting the SIFS, data transmission begins. This is followed by another SIFS and ACK from the receiving node. Figure 31 shows the MAC procedure and the impact on adjacent nodes in the network. The δ represents any additional time imposed by a long propagation or processing delay. For most indoor networks this can be assumed to be zero. From the figure it is clear that the RTS-CTS exchange helps to prevent collisions caused by the hidden terminal problem.

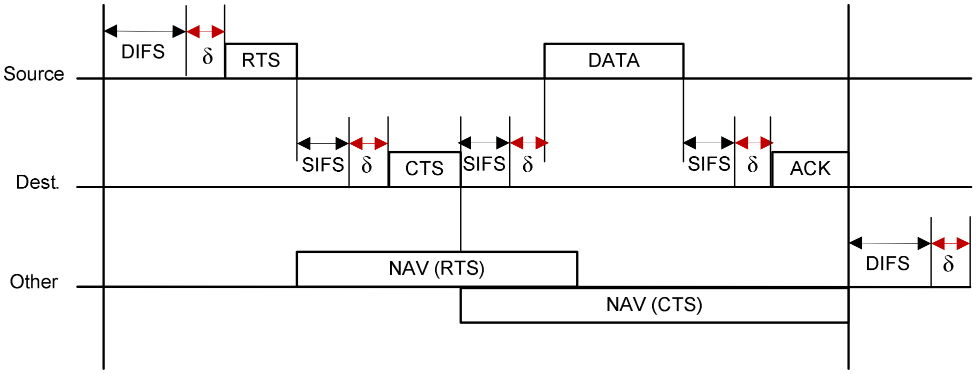


Figure 31. 802.11 MAC with RTS-CTS exchange.

4.3.1 Point Coordination Function and Time Sensitive Data Transfer

The point coordination function is used in time sensitive data transfer. It uses a point coordinator (PC) to establish polling based communications. The PC sends a beacon packet that contains information related to the contention-free period. It then begins to poll associated nodes which can respond with data and an acknowledgment (D/CF-ACK). These operations continue with a time separation of SIFS unless no acknowledgement from a polled device is received. In this case a wait period of PIFS is employed. This wait period, while greater than the SIFS, is less than the DIFS preventing non-contention free nodes from competing during a contention free interval. At the end of a contention free period the PC broadcast a CF-End packet and the DCF protocol is employed until the next contention free period.

4.3.2 Fragmentation

Fragmentation is the process by which a long MPDU is divided into a number of smaller MPDUs each with their own headers. Multiple smaller packets are then sent during the time allocated for the original long packet to be sent with each packet receiving an acknowledgement. This increases the throughput of transmissions occurring on a noisy channel. For purposes of this work, however, it is used as a tool to determine the cause of decreased throughput.

Consider the case in which in which a delay has caused a significant decrease in throughput. This implies that if the delay is caused by a noisy channel, fragmenting the packet should substantially improve the throughput. If this is not the case, there is strong evidence to suggest that the reduced throughput was caused by a MAC related issue. Fragmentation is used as a diagnostic tool for evaluating link performance in later sections.

4.4 Theoretical Throughput Analysis

Theoretical throughput analysis for 802.11b was conducted to get an understanding of reasonable throughput performance. It should be mentioned that the RTS-CTS exchange was removed because it is redundant for networks with a low number of nodes and because the impact of the SIFS is already accounted for in the acknowledgment packets. A payload of 1550 bytes/frame was assumed since many host computers limit their file size to this value [82]. For a given payload, data rate, and preamble type theoretical throughput values were computed. The results are shown in Tables 3-6. As the data rate is increased, the time gaps in the MAC have a greater impact on the system. This explains why for 1Mbps, an efficiency of almost eighty percent is achieved while at 11Mbps it is closer to fifty percent.

Table 3. 802.11b throughput analysis for 1 Mbps bit rate.

INPUT			**Bytes**	**Time (µs)**	**#**	**Total (µs)**
Payload (bytes)	1460	**Slot Duration**	NA	20	16	320.0
Rate (Mbps)	1.00	**SIFS**	NA	10	2	20.0
Preamble (S/L)	L	**DIFS**	NA	50	2	100.0
Extra Delay (µs)	0	**tp**	NA	0	2	0.0
OUTPUT		**RTS**	20	160	0	0.0
Eff. Throughput	0.85	**CTS**	14	112	0	0.0
		ACK	14	-	2	224.0
		PHY Header	24	-	4	768.0
		TCP Header	40	-	2	29.1
		MAC Header	36		2	576.0
		DATA	1460	-	1	11680.0
						13717.1

Table 4. 802.11b throughput analysis for 2 Mbps bit rate.

INPUT			**Bytes**	**Time (µs)**	**#**	**Total (µs)**
Payload (bytes)	1460	**Slot Duration**	NA	20	16	320.0
Rate (Mbps)	2.00	**SIFS**	NA	10	2	20.0
Preamble (S/L)	L	**DIFS**	NA	50	2	100.0
Extra Delay (µs)	0	**tp**	NA	0	2	0.0
OUTPUT		**RTS**	20	80	0	0.0
Eff. Throughput	1.56	**CTS**	14	56	0	0.0
		ACK	14	-	2	112.0
		PHY Header	24	-	4	768.0
		TCP Header	40	-	2	29.1
		MAC Header	36		2	288.0
		DATA	1460	-	1	5840.0
						7477.1

Table 5. 802.11b throughput analysis for 5.5Mbps bit rate.

INPUT	
Payload (bytes)	1460
Rate (Mbps)	5.50
Preamble (S/L)	L
Extra Delay (μs)	0
OUTPUT	
Eff. Throughput	3.33

	Bytes	**Time (μs)**	**#**	**Total (μs)**
Slot Duration	NA	20	16	320.0
SIFS	NA	10	2	20.0
DIFS	NA	50	2	100.0
tp	NA	0	2	0.0
RTS	20	29	0	0.0
CTS	14	20	0	0.0
ACK	14	-	2	40.7
PHY Header	24	-	4	768.0
TCP Header	40	-	2	29.1
MAC Header	36		2	104.7
DATA	1460	-	1	2123.6
				3506.2

Table 6. 802.11b throughput analysis for 11Mbps bit rate.

INPUT	
Payload (bytes)	1460
Rate (Mbps)	11.00
Preamble (S/L)	L
Extra Delay (μs)	0
OUTPUT	
Eff. Throughput	4.92

	Bytes	**Time (μs)**	**#**	**Total (μs)**
Slot Duration	NA	20	16	320.0
SIFS	NA	10	2	20.0
DIFS	NA	50	2	100.0
tp	NA	0	2	0.0
RTS	20	15	0	0.0
CTS	14	10	0	0.0
ACK	14	-	2	20.4
PHY Header	24	-	4	768.0
TCP Header	40	-	2	29.1
MAC Header	36		2	52.4
DATA	1460	-	1	1061.8
				2371.6

4.5 The 802.11g Protocol

Testing of the 802.11g protocol will be performed, but it is expected that a delay of several microseconds will make operation impossible. This is mainly due to the highly shortened preamble of ERP-OFDM which consists of only 12 symbols (10 long and 2 short) lasting a training time of approximately 16 μs. However, DSSS-OFDM utilizes the same synchronization process of 802.11b while enabling the payload to be transmitted at the higher payloads afforded by OFDM. Theoretical operation of this type of 802.11g node should therefore be possible.

The primary focus of this work remains the optimization of the performance by 802.11b devices. However, basic functionality of 802.11g devices will be tested to determine if the design can meet the needs of both designs. In the event that it does not, this testing may provide information needed to determine added design requirements that would facilitate converted 802.11g communications.

4.5.1 Description of OFDM

A typical frequency division multiplexing (FDM) system uses non-overlapping frequencies to transmit information simultaneously. Orthogonal frequency division multiplexing (OFDM) differs in that it utilizes very closely spaced frequency signals that are orthogonal to one another to dramatically increasing the bandwidth efficiency of the overall system. To generate orthogonal subcarriers, the well-known inverse Fourier transform (IFFT) is applied to N constellation points resulting in a constellation symbol being carried on each subcarrier. Data rates can then be increased/decreased by altering the number of constellation points used. Reducing the complexity of the constellation map also has the added benefit of increasing the probability of correct signal reception, making OFDM an excellent choice for adaptive modulation systems. Major drawbacks of OFDM include a high peak to average power ratio (PAPR) which can introduce non-linearity at the output of amplifiers, and the need to maintain orthogonal carriers requires very precise frequency synchronization.

4.5.2 ERP-OFDM Packet Structure

The ERP-OFDM packet structure enables the fastest data rates of the 802.11g frame formats and as such dominates the current Wi-Fi market. It contains many of the same fields of the 802.11b packet structure with the most notable difference being the much shorter preamble. Because OFDM utilizes 52 individual subcarriers it can convey information much faster than CCK modulation. The preamble is subdivided into ten short and two long symbols. The short symbols are used for signal detection, automatic gain control, diversity selection, and coarse frequency offset estimation [81]. After an idle period the two long symbols are used to fine tune the frequency estimation and perform channel estimation/equalization. The entire preamble is a total of 16 μs in length, a mere fraction of the long (128 μs) and short preamble (56μs) preambles of 802.11b.

As previously mentioned, it is clear that the proposed extension will not enable operate if ERP-OFDM is used as a delay on the order of 5 μs is anticipated. This more than halves the original length of time dedicated to the short symbol synchronization procedures. A structure of the ERP-OFDM packet structure and the impact of a transmission delay is illustrated in Figure 32. Ten short symbols of duration 0.8μs are separated from two long symbols lasting 3.2μs by a time gap of 1.6μs.

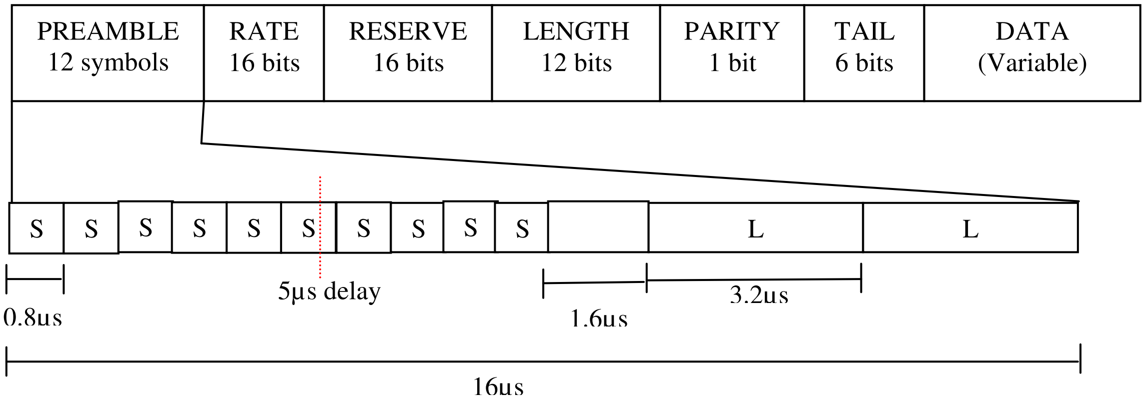


Figure 32. ERP-OFDM packet structure.

4.5.3 DSSS-OFDM Packet Structure

DSSS-OFDM has an identical preamble to 802.11b with the difference being that it has a modified PSDU. Retaining the same preamble format as 802.11b makes coexistence of 802.11b/g nodes much more straightforward as the MAC (destination node, etc.) information can easily be exchanged. The OFDM training information is appended to the beginning of the MPDU. It is of particular interest in this work because its location within the packet makes it immune to switching delays. This packet structure is illustrated in Figure 33.

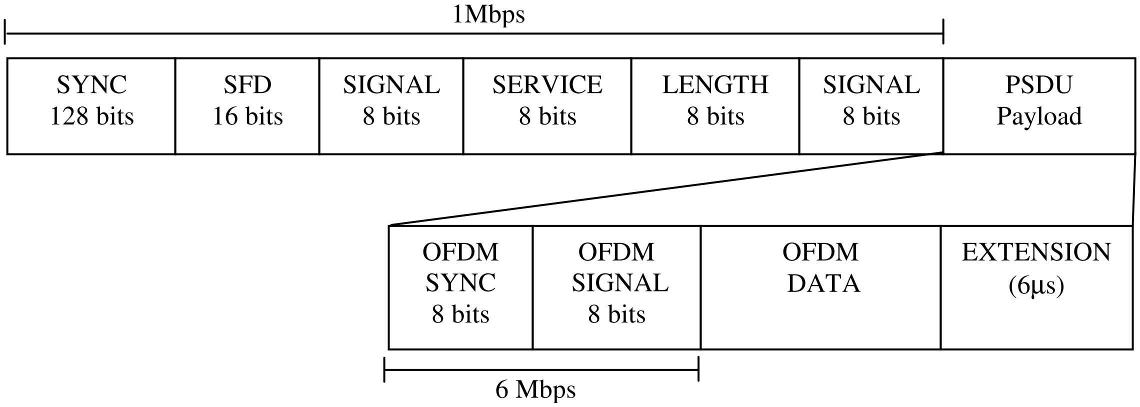


Figure 33. DSSS-OFDM packet structure.

4.5.4 MAC Comparison

The only significant difference in the 802.11g MAC when 802.11b nodes are present in the network is the 6 µs extension to the SIFS that is necessary to give the nodes the needed time to decode the more complex 802.11g signal. Since ERP-OFDM is used when no 802.11b nodes are present, the MAC operations in this case are irrelevant.

5. Wireless 802.11b Conversion Device Architecture and Test Setup

5.1 Overall Device Description

In this work, design and analysis of an 802.11b converter is presented along with the limitations with respect to time of two approaches that are examined. The overarching purpose of this study is to determine the feasibility of cognitive extensions to existing wireless architecture. It is proposed that by isolating the cognitive system, testing of the hardware path and its switching characteristics is sufficient to enable cognitive capabilities. With respect to the design, the 802.11b conversion device takes the output at the antenna of an 802.11b client/AP and converts it for operation at a new frequency. Separate paths are maintained for transmit and receive operations to meet isolation and gain requirements. Although specific focus is placed on 802.11b most of the design and concepts discussed in this section can easily be extrapolated to additional forms of communication.

5.1.1 Architecture Requirements

The extension should keep the signal as identical to the original signal as possible, limiting noise contamination and any harmonics produced by the mixers. The primary design requirement is that the throughput of the base device be maintained when connected to the extension. The delay when switching between the transmit and receive paths should be the main contributor to communication errors as the noise figure was kept very low. The prototype that was developed used off-the-shelf RF components that exhibited delays that were typically longer than their chip-level counterparts. In this manner the timing limitations could be explored while also allowing for rapid design changes.

5.1.2 Design Principles and Feasibility

Development of a cognitive receiver is a topic that has been thoroughly researched and it is not desired to rehash this body of work. The focus of this work is to determine if a cognitive receiver can be interfaced with configurable hardware to provide cognitive capabilities to a transceiver lacking these capabilities. Essentially, can any existing wireless system be made cognitive via an extension while maintaining the benefits of its MAC and physical layer. A block diagram of such a system is shown in Figure 34. To prove that such a system is feasible, extension designs for 802.11b were developed. It is assumed that changes to the physical layer by the cognitive system are only made directly after a packet has been transmitted. In this way the MAC provides

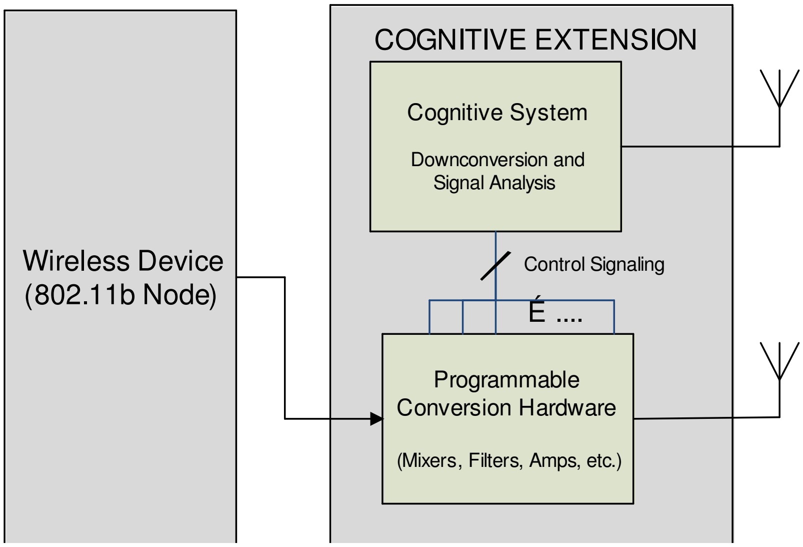

COGNITIVE EXTENSION
Cognitive System
Downconversion and Signal Analysis
Control Signaling
É ....
Wireless Device (802.11b Node)
Programmable Conversion Hardware
(Mixers, Filters, Amps, etc.)

Figure 34. Overview of cognitive extension principle.

a small protection time, like the SIFS in 802.11-based communication for example, during which transmission and reception is not occurring and the configurable hardware can safely be changed. Under this assumption, the feasibility of such a system can be proved possible by testing a fixed hardware conversion design. The major hurdle in such a design is the intelligent selection of the appropriate signal path, and much of this work will be devoted to analyzing this problem.

5.2 Design Approaches

Two different design approaches were investigated: a design that favors the base device MAC, and another that favors the physical layer of the base device. The vast majority of both designs are identical with the only major difference being in the receive path of the extension. Both receive paths were given an insignificant gain as the base device has built in amplifiers to compensate for low signal strengths. The physical layer based design consists of a low-noise amplifier and attenuator in series with one another. This combination acts like an isolator preventing the signal from flowing one direction while allowing it to flow the other. Thus when switching from the receive path to the transmitting path, the portion of the packet occurring during the switching is completely lost. It should also be noted that the small gain seen in the receive path is negligible as all testing was done within a small confined space rendering signal strength a non-issue.

Conversely, the MAC-based design is a simple, straight connection between the switches. The switches ensure that when the extension switches from the receive to the transmit path, the packet is only lost for a very small amount of time. Before the switch occurs transmission occurs through the receive path, resulting in a significant reduction in the transmit power. This setup favors the MAC because competing nodes can still hear a "weak" preamble during the switching process provided they are in range of the signal. The physical layer design, on the other hand, completely truncates the preamble, but does not alter the amplitude of the signal which can adversely affect preamble operations such as equalization and the setting of thresholds. The two approaches are summarized in Figure 35.

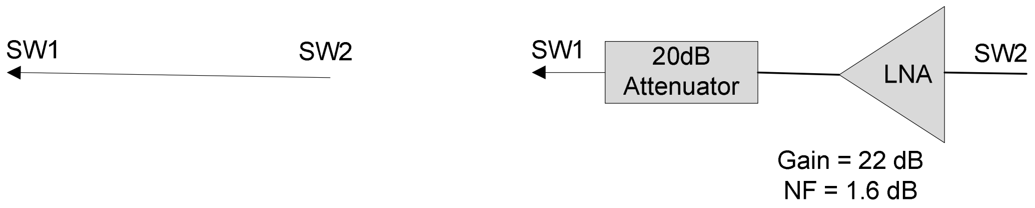


Figure 35. Weakened (left) and truncated (right) preamble receive paths.

The determination of gain in the receive path of the truncated preamble extension poses an interesting dilemma. The addition of an LNA not only prevents the weakened preamble architecture, but also adds 1 to 3 dB to the overall noise figure of the system. If the amplifier is removed, however, the attenuation occurring at the output of the mixer decreases the sensitivity of the receiver. Designers can optimize their extension for a given scenario. Because the test environment used in this work was confined and shielded from noise, the net receive path gains were kept close to one another (within 2 dB) to ensure unbiased results. It was also assumed that since testing occurred inside an anechoic chamber, an additional noise figure of about 2.7 dB would not have a meaningful impact on the results. This is confirmed in the next chapter as the truncated preamble approach outperforms the weakened-preamble design despite an increased noise figure.

5.2.1 Overlapping Architecture

As previously mentioned, most of the architecture of the weak preamble and truncated preamble designs is the same. The base device is attached via shielded SMA cabling to a directional coupler which directs a small portion of the signals power to a power detector. The reverse power port of the coupler is connected to a 50Ω terminator to prevent reflections. The output of the directional coupler is then fed to a 2.4 GHz band-pass filter that rejects image frequencies during reception. Next, the signal is mixed with a local oscillator to translate it to the desired transmit frequency and filtered (in this case at 1.7 GHz). It should be noted that in the case of a configurable frequency design, this can be accomplished by up-converting to a fixed frequency using a VCO, filtering appropriately, and down-converting to the intended transmit frequency. After the last filter, the signal is routed to the appropriate signal path as determined by the output of the power detector and the switching control circuit.

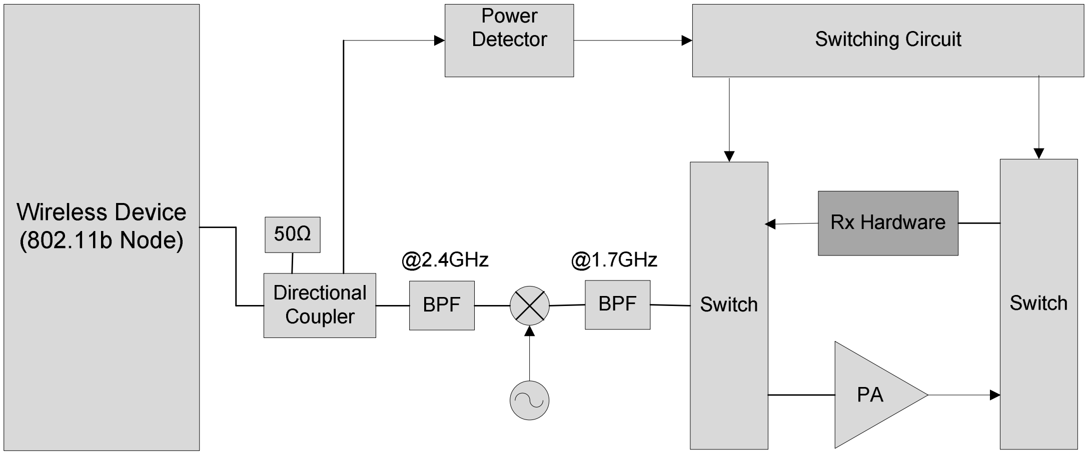


Figure 36. Overlapping architecture between weakened and truncated preamble designs.

If the signal is fed through the transmitting path, it is passed through a power amplifier to recover the gain lost during the mixing process. If the receiving path is selected, the signal is passed through one of the two designs presented) in Figure 35. The overlapping architecture design is illustrated in Figure 36.

The components add very little noise figure or insertion loss to the overall wireless node. Of particular importance are the switching delays, LNA noise figure, filter bandwidths, and insertion loss of the coupler. These parameters and others are summarized in Tabl 7. All of the components were purchased from the same manufacturer, *Mini-Circuits*.

Table 7. List of 50Ω components and relevant parameters.

Component	Function	Loss/Gain (dB)	Noise Fig. (dB)	Delay (μs)	Bandwidth (MHz)	Isolation (dB)
ZABDC10-25-HP-S	Directional Coupler	0.55	X	X	1500 - 2500	X
VBFZ-2575-S+	2.4 GHz BPF	1.72	X	X	2350 - 2800	X
VBFZ-1690-S+	1.7 GHz BPF	1.40	X	X	1455 - 1925	X
ZX05-42MH-S+	Mixer	9.8	X	X	5 - 4200	RF-LO 28 IF-LO 23
ZX47-40-S+	Power Detector	X	X	Rise: 0.40 Fall: 0.01	10 - 8000	X
ZX80-DR230-S+	Switches	0.9	X	2.0	0 - 3000	RF1-RF2 50 RF-Com 60
ZFL-2000	Power Amplifier	21.32	4.99	X	10 - 2000	X
ZX60-2522M-S+	Isolation Amplifier	23.50	2.95	X	500 - 2500	20

5.2.2 Switching Control Design

The switching control design was complicated by the need for excessive buffering and current limitations. The power detector can handle input powers between -40 and 20 dBm with a response time of about 400 ns. Its output voltage ranges from 0.6 V at the maximum power to 2.1 V at -40 dBm. The output of the power detector was fed to an LM234N operational amplifier chip acting as a buffer. Ground was connected to another buffer in a similar manner; however, because the rail limitation of the LM324N was roughly 1 V, the output of this buffer is also approximately 1 V. This was done to decrease the number of parts required. The outputs from these buffers were then fed to a difference amplifier with a gain of 4.7 to enable an effective output voltage ranging from roughly 0.5 V to 5 V.

The output from the difference amplifier was then sent to the opposite ports of two separate LM311 comparator chips to ensure equal timing in each branch. Similarly, a threshold voltage of 3.4 V was connected to the opposite ports of each comparator. Pull-up resistors were then used to connect 15 V to the base of a power MOSFET when the thresholds had been reached. The power MOSFETs were found to be necessary

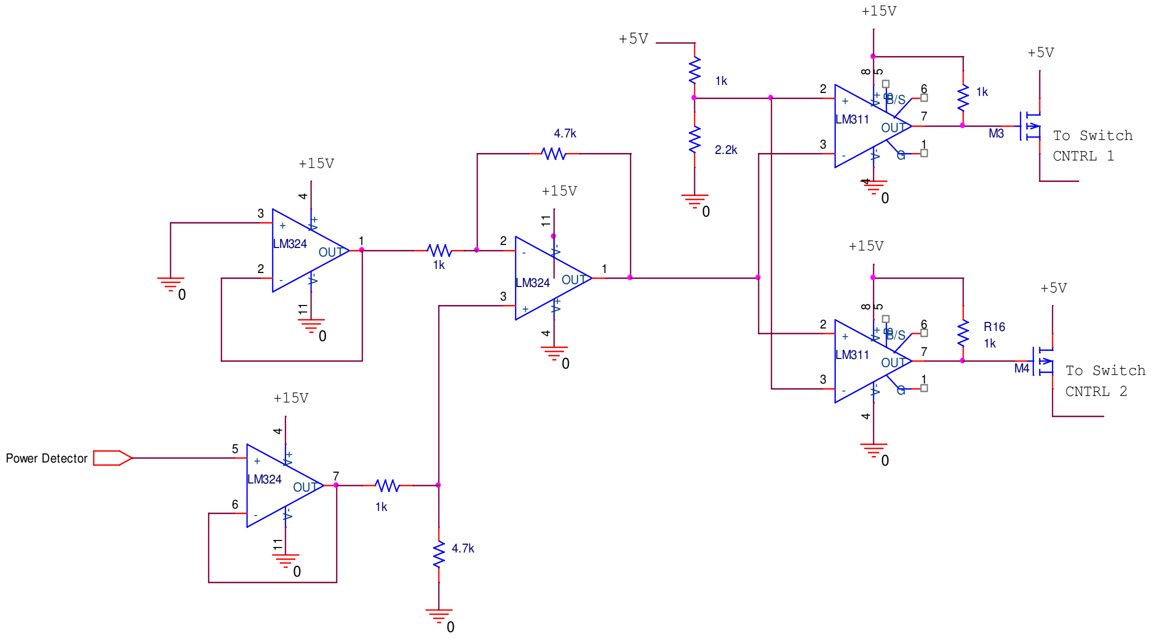


Figure 37. Schematic of switching control circuit.

throughout the design phase as the comparators could not supply the needed current to some of the switches that were being considered. Summarizing, when the power detector detects a signal, its output voltage adjusts accordingly. This voltage is then scaled to between roughly 0 and 5 V. A series of comparators are used to determine when a threshold has been reached triggering the appropriate transmit/receive path. The schematic for the circuit is shown in Figure 37.

5.3 Testing Environment

All testing of the converter extension was performed inside an anechoic chamber. The purpose of this was two-fold: to isolate the network from all other noise and 802.11 networks, and to prevent violation of FCC regulations at 1.7 GHZ. *IxChariot*, a commercially-available network performance evaluation software, was used to determine the throughput and packet loss performance of the overall network under various test conditions. A PC was setup outside the chamber running *IxChariot*. This computer was attached to the first AP Client (N1) whose output was then connected to a faceplate of the chamber. The interior faceplate connection was routed to the first frequency converter extension (FCE 1).

The basic structure described above is then repeated for the two remaining nodes with the major difference being that they are located completely inside the chamber. It is also important to note that N2 was configured as an AP during testing. All of the local oscillators used were obtained from signal generators located near the respective conversion extensions. A laptop running *IxChariot Endpoint* was located at the final node for a specific test setup. This is a supplementary program that is needed by

IxChariot to calculate the necessary network characteristics. This test environment is illustrated in Figure 38.

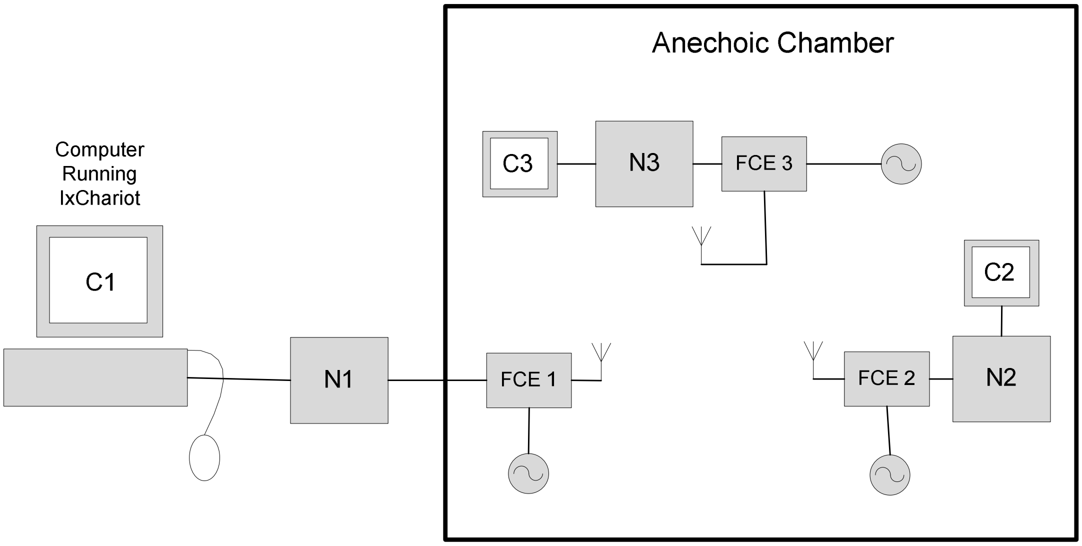


Figure 38. Description of test setup and environment.

5.3.1 Isolation of Nodes

Isolation of the nodes during testing is critical. The test results must be repeatable to draw any meaningful conclusions. If testing is performed outside of the isolated chamber, changes in the propagation environment and the presence or absence of noise/interference between subsequent tests can skew the results. It was also originally thought that the nodes may be capable of leaking the 2.4 GHz signal directly to one another. By placing N1 outside of the chamber and N2 inside the chamber, this possibility was completely eliminated for two node tests. Later testing showed that while significant leakage occurred during the wired tests discussed in Chapter 4, this was not true for the wireless scenario, enabling three node tests to be performed.

5.3.2 FCC Compliance

The FCC regulates the spectrum frequency and power limitations throughout the United States. The desired frequency of operation for the FCEs was roughly 1.72 GHz depending on the operating channel of the AP. This frequency is occupied by users who purchased bandwidth in the AWS-1 band in the recent auctions. Consequently, unlicensed usage is prohibited. Tests to ensure that FCC regulations are being met by a device are conducted within a certified anechoic chamber as it prevents emissions from leaving the chamber. Hence testing of the extensions within an anechoic chamber can also be conducted without violating the FCC regulations. The entire mapping of the AWS spectrum is shown in Figure 39.

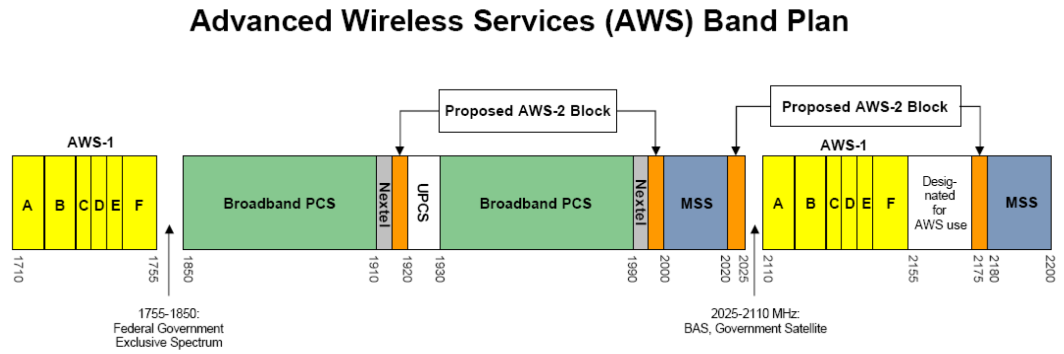


Figure 39. Description of the AWS frequency band [83].

5.3.3 Delay Line Design

The major variable that exists between various configurable hardware designs is the time that it takes to detect a transmitted signal and route the signal accordingly. To determine the limitations and primary concerns for the two design approaches considered, it becomes necessary to intentionally introduce additional delay into the system. This impacts the length of time that a node will receive a truncated or weakened preamble, and in so doing, the packet loss and throughput of the network. Ordinarily this could accomplished with an adjustable counter chip that triggers the switching at fixed delays. However, such a setup does not preserve the transition characteristics of the analog signal. Pure analog delays on the order of microseconds are not easily achieved. Exact delays, however, are not necessary as the primary interest is the general behavior of the nodes as delay is added. Of particular interest is whether the failures are MAC or physical layer related. Given this relaxed requirement, a fairly good analog delay can be obtained by concatenating buffer amplifiers in series.

The response time of a single LM324 is on the order of a few hundred nanoseconds. Concatenating a number of them together can result in delays of several microseconds. Furthermore, since additional LM334 op-amps acting as buffers exist in the switching control circuit, the slew rate of the chips is already accounted for in the design. The basic design of the switching delay line circuit is given in Figure 40. The output delays were taken at every fourth buffer, resulting in delays increments close to one microsecond. The output amplitude very slowly degrades as buffer amplifiers are added. To compensate for this effect, amplifiers with a gain slightly over unity were placed throughout the delay chain to ensure that the amplitudes at the output of every fourth buffer were comparable. Signal behavior at the output of each delay used can be found in the next chapter.

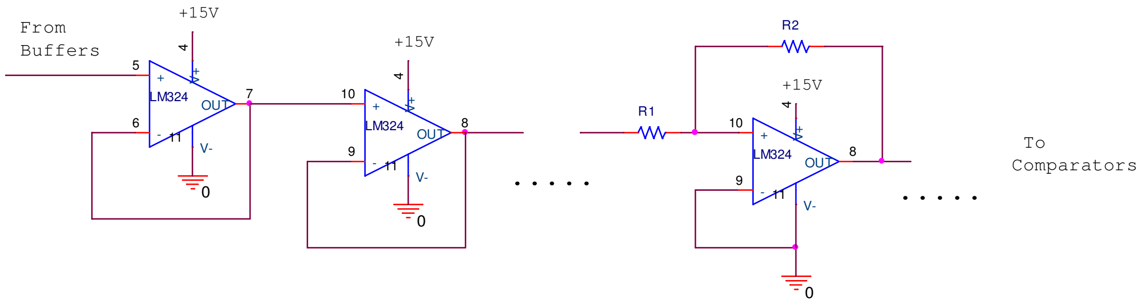


Figure 40. Basic delay-line configuration used during testing.

5.4 Design Summary

A frequency conversion extension for 802.11b devices was designed and presented in this chapter, with the overarching goal of assessing the feasibility of a cognitive extension 802.11 technology and wireless devices in general. Two different design approaches were considered. One approach used a design that favored the MAC layer by allowing the signal to be transmit a weak preamble through the receive path prior to switching. The other design favored the physical layer by truncating the preamble and not allowing synchronization, the setting of thresholds, and other initial operations to be biased by a weakened preamble. Most of the hardware is the same for both approaches utilizing coupled power detection to enable the appropriate signal path in a timely manner.

A switching control circuit was developed although the need for buffering and the high current drawn by some of the switches considered complicated its design. All testing was performed with the conversion extensions inside an anechoic chamber to prevent noise/interference and to adhere to FCC regulations. To test the limitations of the design approaches an analog delay line utilizing buffer amplifiers was developed and used to provide delays that maintained the transition characteristics of the signal prior to the adding the delay. The next chapter will analyze the inherent complications in the MAC and physical layer presented by this design.

6. Impact of the Design on MAC and PHY Layers

6.1 Physical Layer Implications

A prolonged switching delay introduced by the FCE can have an adverse effect on the preamble operations. This is particularly true in the weakened preamble approach as any amplitude sensitive operations could be biased if the weak signal falls outside the operating range of the system AGC. In the case of a truncated preamble, less time is available to train the receiver during the SYNC field of the preamble. Either of these problems results in a failure to recognize the start of frame delimiter. Without this knowledge inter-frame synchronization is lost including the MAC header information and transmitted data. Ultimately, this causes a packet loss decreasing the throughput of the system. In the sections that follow, specific physical layer issues that arise from the introduction of a switching delay are discussed.

6.1.1 Time-Related Complications

The scenario in which the receiver is not given sufficient time to synchronize and perform other necessary operations such as equalization is referred to as partial synchronization. In this case, the truncated SYNC field provides the needed information to synchronize in time or frequency, but has lost too many bits to fully complete the process. To account for the both low-speed and high-speed modes of operation in 802.11b, demodulation is achieved by correlating the received sequence with possible codes at that data rate. Even though DBPSK and DQPSK are used for the low data rates, it may still be advantageous to employ an independent correlator to avoid multiplying two noisy signals:

$$
\begin{aligned}
C &= \sum\nolimits_0^N [x(l)+n(l)][x(l-N)+n(l-N)] \\
&= \sum\nolimits_0^N [x(l)x(l-N)+n(l)n(l-N)+x(l)n(l-N)+x(l-N)n(l)] \quad . \qquad (65)
\end{aligned}
$$

This expression shows the noise components resulting from the multiplication of two subsequently received chipping sequences. Since both signals are contaminated by noise, three noise components contaminate their correlation. If instead a separate clean sequence is correlated with the received sequence only one noise component is present:

$$
\begin{aligned}
C &= \sum\nolimits_k^{k+N} [x(l)+n(l)][z(l)] \\
&= \sum\nolimits_k^{k+N} [x(l)z(l)+n(l)z(l)] \quad . \qquad (66)
\end{aligned}
$$

This approach would improve the receiver sensitivity by lowering the noise floor. Although the 802.11b standard specifies DBPSK and DQPSK for the lower data rates, it does not restrict the method by which the signal is to be decoded. Thus the FCE must account for the possibility that a correlator is used to decode both the high and low data rates.

Along this line of thought, consider a sequence that is received and after having reached the end of the SYNC field is only partially synchronized. The output of the correlator will not reach its optimum, and consequently may not exceed the power-dependent threshold needed for the demodulator to properly decode the SFD. By definition this will result in the packet being lost. To help visualize this scenario, sequences deviating by half of a chip are shown in Figure 41, illustrating the consequences of not reaching complete synchronization. This behavior holds true for both the long and the short preamble, though the impact on the short preamble should be much more noticeable given the fact that it has much fewer synchronization bits.

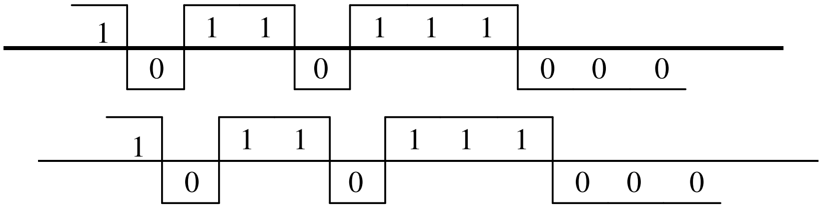


Figure 41. Illustration of partial synchronization of the demodulator.

As discussed in the background, a similar effect can occur when the receiver is attempting to synchronize with respect to the frequency/phase of the inbound signal. A time shortage may result in an improper frequency or phase estimation in the PLL, and a subsequently skewed base-band signal:

$$\begin{aligned} r(t) &= u(t)\cos(2\pi f_c t)\times\cos(2\pi[f_c - f_\delta]t + \varphi) \\ &= u(t)\cos(2\pi f_\delta t - \varphi) + u(t)u(t)\cos(2\pi[2f_c - f_\delta]t + \varphi) \end{aligned} \quad . \tag{67}$$

After a low-pass filter, this yields:

$$r(t) = u(t)\cos(2\pi f_\delta t - \varphi) \quad . \tag{68}$$

Clearly, inadequate frequency synchronization degrades the down-converted signal to a point at which communication may not be possible. Additional operations may also be impacted by a decreased training interval. For example, operating in a complex channel with a significant reduction in training time may produce poor equalization at the receiver. Similarly, gain estimation and diversity decisions must also be made before the end of the SYNC field.

6.1.2 Amplitude-Related Complications

Many operations in the receiver are sensitive to amplitude changes. The weakened preamble design considered intentionally introduces a spike in amplitude followed by a constant signal level. The switching process for the weakened preamble approach begins with a weak signal being transmitted through the receive path. Then while the switches are configuring to the transmit path the signal becomes even weaker until connecting with the transmit path. At this point there is a significant jump in amplitude caused by the power amplifier. Extending the delay increases the time the receiver is trained using a weak preamble before the switch occurs. It is therefore important to determine and discuss the amplitude dependent functions of the receiver.

The major functions that occur during the SYNC field of the preamble consist of signal detection, frequency and demodulator synchronization, diversity selection, and equalization. Signal detection is obviously not impacted at all by this approach unless the weakened preamble cannot reach the destination node. In this case the nodes will behave as if the preamble is truncated. Equalization and diversity should not be significantly affected, as the weak signal should still travel a similar multi-path route. It is possible that a multipath component could be weakened to the point that it reaches the noise floor before arriving at the receiver, changing the equalization coefficients or antenna selected, but this is unlikely for modest signal levels. Frequency synchronization is also not directly impacted. In many respects the weakened preamble approach is very similar to an amplitude shift-keyed symbol transition with the signal levels being drastically different. As can be seen in Figure 42, changes in amplitude do very little to alter the signal frequency and phase. This makes sense as a PLL loop integrates the error signal it obtains by mixing the unknown frequency with a synthesized frequency.

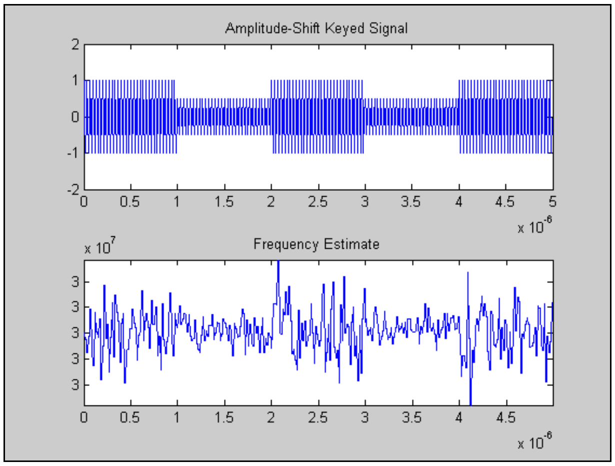


Figure 42. Negligible impact of amplitude variations on frequency estimation.

Before the amplitude change, the error continues to approach zero. Then, the amplitude change causes a spike in the error, that the PLL loop attempts to compensate for. Because it is already close in phase and frequency, the error signal is rapidly driven toward zero, causing very few problems.

Synchronization of the demodulator can be impacted by amplitude changes as the thresholds needed to achieve a SYNC lock and detect the SFD are both amplitude dependent. If the correlation value is set to high, the SFD might never be detected. Conversely if it is set to low, the threshold may be exceeded when the sequence is only partially correlated. It should also be mentioned that if the frequency synchronization is dependent on extracting the modulating sequence, then it may also be impacted by this threshold.

None of the problems mentioned above exist if the gain control of the receiver adequately compensates for the difference in the signal strength of the weakened and normal portion of the preamble. This means that any amplitude-dependent physical layer error is directly attributable to the gain control of the system. Unlike other wireless communication technology such as cellular phones, 802.11-based technology was originally designed for a much more static short range environment (wireless internet). Such an environment has a much stronger emphasis on multipath propagation, but at the same time is much less likely to experience phenomena such as deep fades. As a consequence the gain control systems for 802.11b may or may not be dramatically simplified, depending on the design. It is not uncommon for some designs to simply estimate the gain needed during the preamble and sustain this gain throughout the packet. This process would inevitably occur at the beginning of the SYNC field to ensure adequate gain for synchronization processes to occur. Under this scenario, the weakened preamble design would drastically underperform the truncated preamble approach.

If instead automatic gain control is used, this type of problem is much less likely, although problems may still exist. AGC is used to ensure fairly constant output amplitudes for a given range of input amplitudes. Recall that the signal goes from weak, to very weak, and then strong in rapid succession and remains strong afterwards. The response time of the AGC determines how rapidly it adjusts for amplitude changes. To rapid of a response would cause the AGC to try and perfect instantaneous amplitude spikes when it would be more beneficial to allow them to occur as the recovery time is costly. However, the response time should be fast enough to eliminate fades of a reasonable duration. If either the weakened or normal portion of the preamble falls outside the operating range of the AGC, a portion of the packet's amplitude will be skewed as it tries to compensate for the transition to the new mode of operation.

6.1.3 Problems Occurring Outside the Preamble

Although most of the problems introduced to the physical layer occur during the preamble, there is at least one significant issue that is present throughout the entire packet. Given the overall design architecture developed in Chapter 6, the signal path is determined by the transmit power coupled to a power detector. With the rapid response time of the power detector (400 ns rise time, 10 ns fall time), fluctuations in signal power may result in the receive path being enabled at some point during transmission losing a portion of the packet. The power output from an 802.11b node is fairly stable over time and does not present the power detector with this problem. However, 802.11g employs OFDM as its modulation scheme. This form of modulation is notorious for its peak-to-average power ratio (PAPR) which causes non-linearity issues in amplifiers. With regards to our concerns it may also establish a low power for sufficient time to disable transmission.

Several methods exist for smoothing the output of the power detector. The slew rate of the buffer amplifier will provide a slight reduction in the local signal variance. Rapid fluctuations at the output of the power detector can of course be eliminated altogether by implementing a low-pass filter. In doing so, however, the switching time will be increased as the transitions at the output of the power detector are greatly slowed by the addition of the filter. As a modest delay of 5 μs is already anticipated and obscures a significant portion of the 802.11g packet, it is highly unlikely that any additional delay would be tolerable.

6.2 MAC Layer Implications

The MAC layer problems resulting from the switching delay are much more straightforward and stem largely from the changes in the transmission timing and listening schedule. A modified version of the MAC procedures during a regular packet transmission is provided in Figure 43.

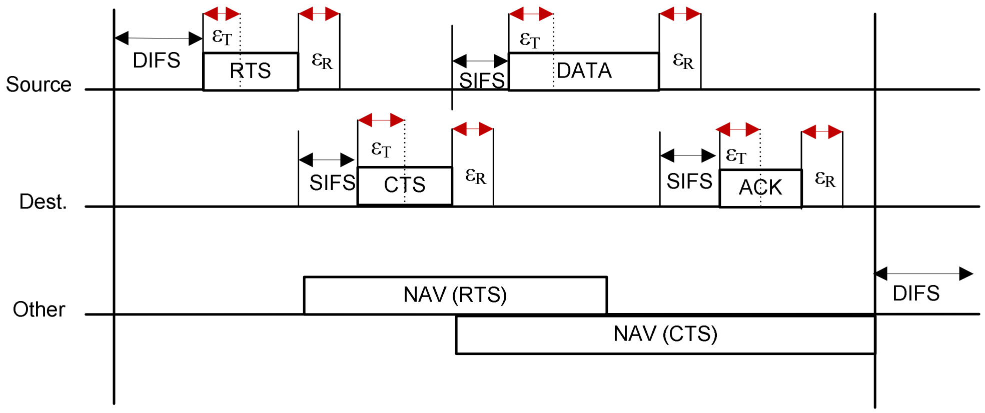


Figure 43. General MAC behavior when switching delays exist.

The receive-to-transmit delay is represented by ε_T. It is important to note that this delay severs/weakens a portion of the packet; it does not add to the delay of the overall packet transmission. Similarly, the transmit-to-receive delay is represented by ε_R. For ε_R to impact a two node network in any way, it must exceed ε_T + SIFS. This means that the transmit-to-receive delay can exceed the receive-to-transmit delay by about 10 μs under normal 802.11b operating conditions. The remainder of this section will discuss the specific impact these changes have under the assumption that this condition holds true.

6.2.1 Extension of SIFS, PIFS, and DIFS

From Figure 43 it can be seen that ε_T acts to extend all inter-frame spacing in the 802.11 MAC. Unlike the 6 μs virtual extension used in 802.11g, this extension does not result in decreased throughput as previously mentioned. Similarly, because the DIFS is five times the duration of the SIFS, a minimum delay of 40 μs would be required to cause outside radios to transmit during existing communication. In the event the nodes are operating in a prioritized burst mode, the delay would be limited to 20 μs (as the PIFS is 30 μs). At first glance, there appear to be no immediate problems presented by an extension to the IFS, however, the utilization of a poorly selected ACK timeout can complicate the use of an FCE.

The ACK timeout is a setting on a wireless AP or client that can be configured to optimize throughput on a network. This parameter is the time a node is allowed to wait to receive an ACK packet. Increasing the ACK timeout provides a longer transmission range at the expense of throughput. The receive-to-transmission delay extends the amount of time a node has to wait to hear an acknowledgement. If for example a node is configured to transmit in close range (less than 25 m), it will wait for a very small amount of time. Assuming it waits for 12 μs before retransmitting and the delay + SIFS is 15 μs, a collision will occur between the retransmitted and the ACK packets completely preventing communication. This may also result in decreasing the maximum range of transmission. Because the ACK timeout is a configurable parameter, it was set to a long range during testing to prevent this specific error from biasing test results.

6.2.2 Deviation from Ideal Slotted Transmission

Slotted transmission has been recognized as an effective means of collision prevention since the migration from ALOHA to slotted-ALOHA. Slotted CSMA-CA guarantees that no transmission is occurring if at any point in the slot duration a transmission is not heard. The receive-to-transmit delay imposed by the FCE introduces a period of time during a slot duration in which communication does not take place even though it will shortly thereafter. This can be thought of as a virtual hidden node problem since transmission is guaranteed to occur shortly. For example, if a node listens for half the slot then begins to configure its hardware for transmission and the effective delay was 12 µs, a collision would occur at the next slot. Obviously this problem is exacerbated by the addition of nodes in the network. The degree to which this will impact a network depends largely on how much of a slot duration is monitored and the signal detection time of individual nodes. Because 802.11b has a slot duration of 20 µs, the closer the delay gets to this value without exceeding it, the more likely a collision will occur. Figure 44 illustrates the slot scheme and problems posed by the delay.

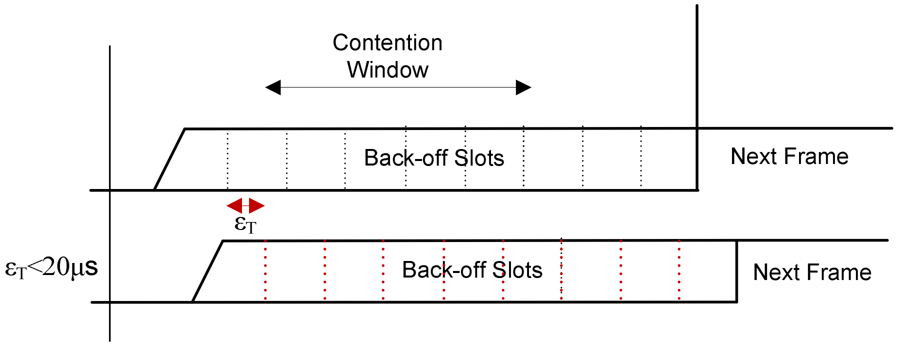


Figure 44. Deviation from ideal slotted contention window.

6.2.3 Extension of Beacon Interval

Beacons are periodically sent by the AP to clients so that their clocks can be updated and ensure that MAC timing throughout the network is synchronized. The interval between individual beacons is fixed and usually set to a value on the order of a one to two hundred milliseconds. The receive-to-transmit delay essentially extends the beacon interval as indicated in Figure 45. Because the beacon interval is much larger than the expected switching delay, the impact on beacon reception is negligible. If this were not the case, it may or may not cause a problem for nodes that use a power-saving mode. Nodes operating in this manner "go to sleep" after a period of inactivity and "wake up" periodically to get the beacon information. While the nodes are in a sleep state, all data meant for them are buffered in the access point. Under most ordinary circumstances, the receive-to-transmit delay will not have an impact on the network. If, however, the designer has chosen to implement a power saving function that only monitors for the presence of a beacon for a period of time less than this delay, the node will not receive the buffered data and will be kept in the sleep state until it needs to transmit data.

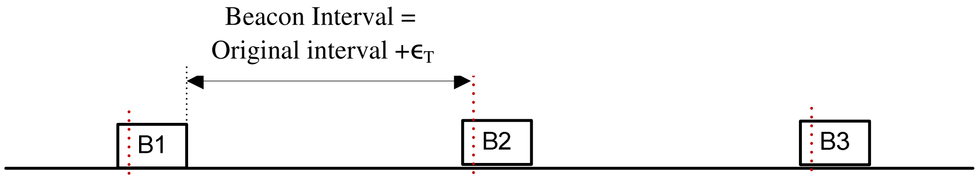


Figure 45. Extension of the beacon interval.

6.2.4 Communication with Other Devices

The case may arise where a user wants to interface the frequency converter extension with one node to enable communication with another node that has a fixed design (no extension). In this case the physical layer is not significantly impacted, but the MAC layer undergoes a number of interesting changes. Assume that the node with the extension is node A and the node with the fixed design is node B. When node A is transmitting to node B, the CTS and ACK preambles are left untouched while node B must account for the truncated/weakened RTS and DATA packets. The ACK timeout is also not an issue in this case, as there is no delay in receiving the acknowledgement. If the situation is reversed and node B is transmitting to node A this problem resurfaces as the preamble of the CTS and ACK packets are adversely affected. As the total number of packets being sent and received that have a modified preamble is reduced by a factor of two, this should cause a corresponding increase in throughput.

The real problem with this scenario occurs when the number of nodes in the network is increased. Due to the different delays the nodes exhibit during transmission, the slots of their corresponding contention windows do not align. As the number of nodes increases the probability of collision become more and more likely. Collisions can be made less frequent by reducing the switching delay. The closer the contention windows are to being identical, the more likely the transmission in a slot will be detected by either node prior to transmission.

7. Link Performance Analysis

7.1 *IxChariot* Test Configurations

IxChariot was used to determine the network performance impact of the 802.11b frequency converter. This is a powerful toolset that enables visualization of the throughput, packet loss, and additional network characteristics over time. Batch mode tests were run as they provided the greatest control over test simulations. The file sizes were chosen such that the tests lasted roughly five minutes. A few longevity tests were conducted to ensure that performance remained consistent over long lengths of time; however, they have been excluded here due to their redundancy.

IxChariot groups its testing into number of sample points and transactions per sample. For example a graph with one hundred sample points and five transactions per sample would transmit from node 1 (N1) to node 2 (N2) and node 2 to node 1 five times, before assigning a value to the sample point. This has the effect of averaging characteristics over time giving a more accurate representation of the network behavior.

7.1.1 Throughput Testing

Throughput testing was conducted using the TCP-IP network protocol as this is the most common application. Data is sent from N1 to N2 and then back to N1. The timestamps provided to N1 allow it to determine the total time elapsed when it has completed enough transactions to determine the throughput at a sample point. It then becomes a simple calculation of data received/time elapsed. It should be noted that in performing the calculations in this manner, uplink and downlink throughput are averaged with one another.

7.1.2 Datagram Loss Testing

For data loss testing, the UDP network protocol was necessary because *IxChariot* does not support packet (datagram) loss testing for the TCP protocol. Unlike most of the other measurements taken, datagram loss is computed one-way. Specifically, *IxChariot* provides the number of datagrams sent, number of duplicates sent, and the number of datagrams lost. In this way in can distinguish between a lost datagram and a lost ACK-datagram.

7.1.3 Lack of Conversion Delay

The architecture of the converter does not impose any delay itself, but rather truncates or weakens a portion of the preamble. Communication is either successful or fails, but no delay is seen unless a packet is lost and requires retransmission. *IxChariot* measures one-way delay using the RTP protocol, which ignores packet losses. This coupled with the fact that response times are poor indicators of actual delay has made throughput and packet loss the primary focus of performance testing.

7.2 Performance of a One-Hop Network

The test setup described in the previous chapter was used to perform *IxChariot* analysis for a one-hop scenario. Given the close proximity and static environment of the test chamber, throughput was high and packet loss was non-existent. For comparison purposes, baseline tests were conducted to indicate how the 802.11b conversion device network performance compares with the ADI boards with no extension.

7.2.1 Baseline Testing

Baseline tests were conducted using the ADI boards described in Chapter 2 and 2.4 GHz antennas. It should be mentioned that one of the boards was isolated outside the test chamber to prevent direct leaking to the other board. Throughput results for 1, 2, 5.5, and 11 Mbps are provided in Figures 46 - 49. While the numbers are close to the theoretical projections in Chapter 5, there are discrepancies that are most likely attributable to processing delays and/or a reduction in the payload size.

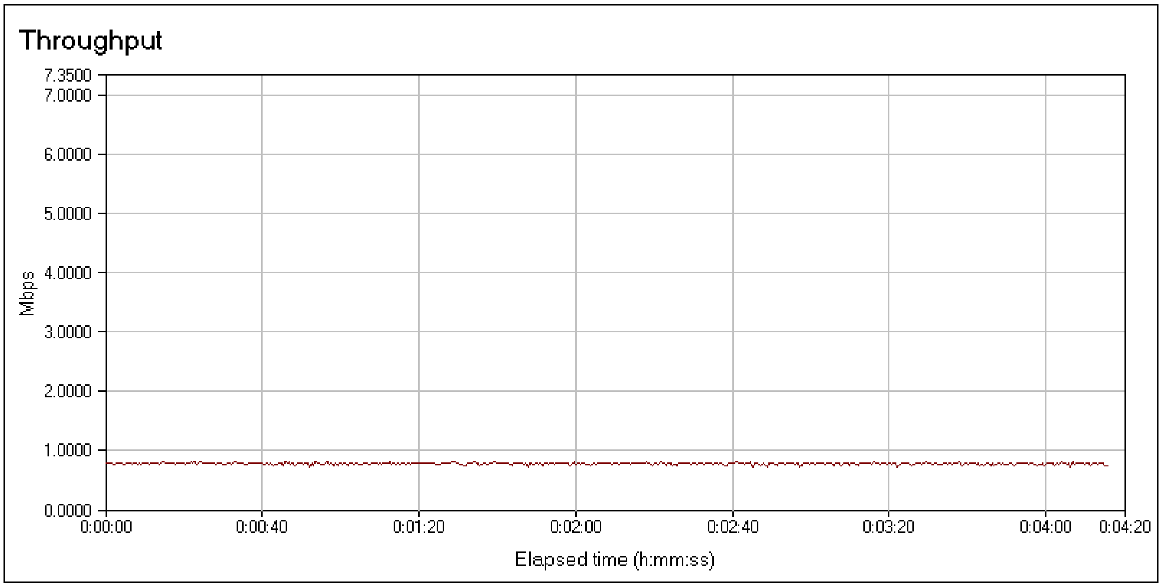


Figure 46. Long preamble baseline throughput for 1 Mbps bit rate.

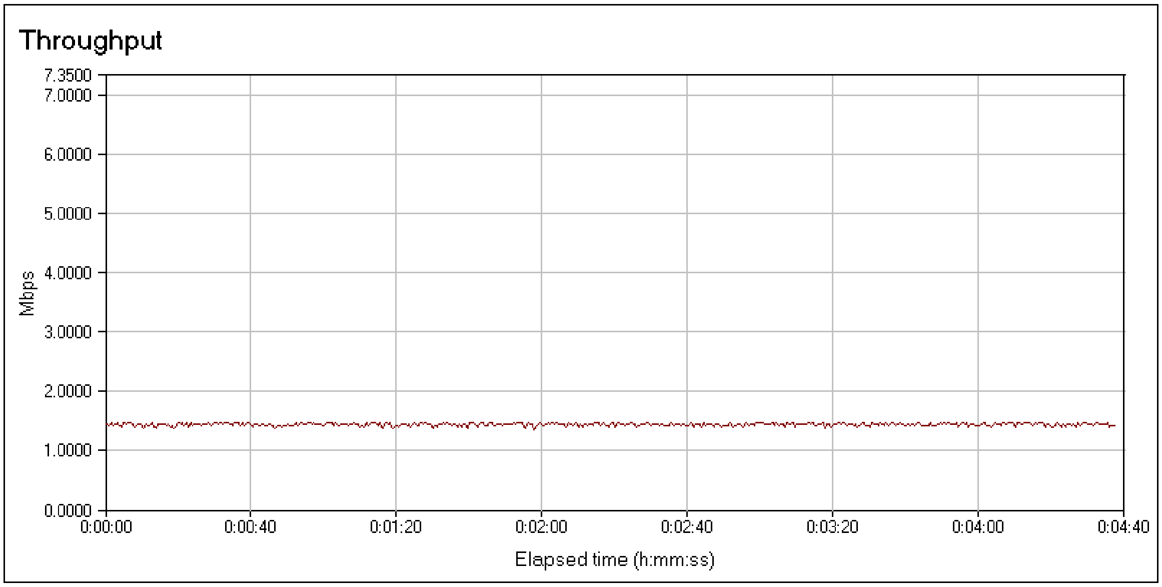


Figure 47. Long preamble baseline throughput for 2 Mbps bit rate.

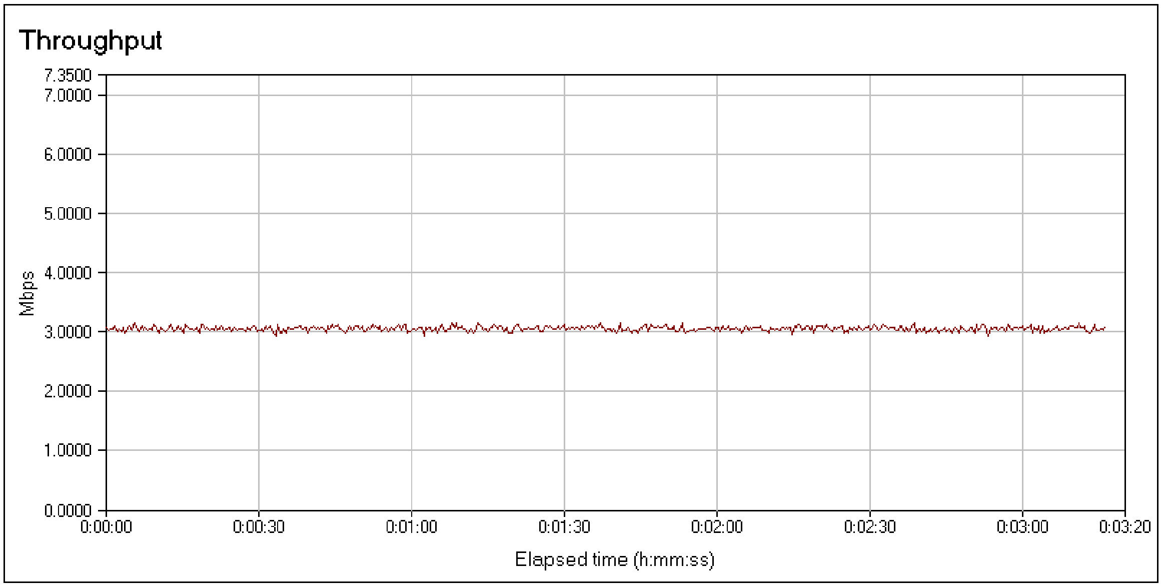


Figure 48. Long preamble baseline throughput for 5.5 Mbps bit rate.

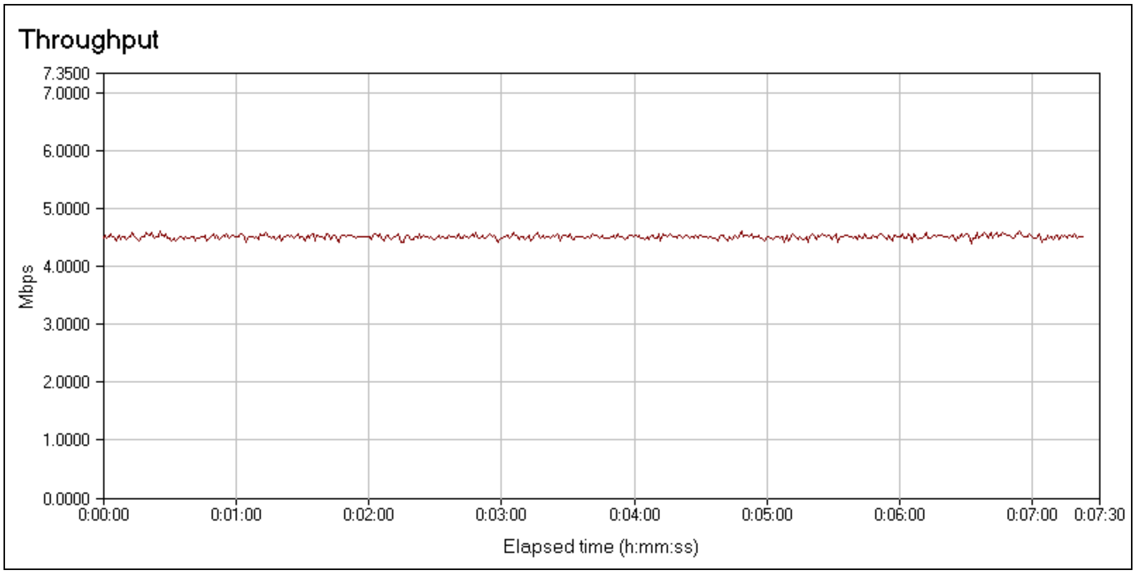


Figure 49. Long preamble baseline throughput for 11Mbps bit rate.

7.2.2 One-Hop Throughput Results

Replicating the baseline tests with the frequency converter extension produced almost identical results with regards to throughput. A comparison of the average throughput of the baseline and converter is given in Table 8 while the throughput results for the converter can be found in Figures 50 - 53. The results indicate that the extension can provide near optimal data rates for a two-node setup.

Table 8. Average throughput results.

Data Rate (Mbps)	Baseline (Mbps)	Converter (Mbps)
1.0	0.78	0.74
2.0	1.44	1.40
5.5	3.05	3.04
11	4.51	4.51

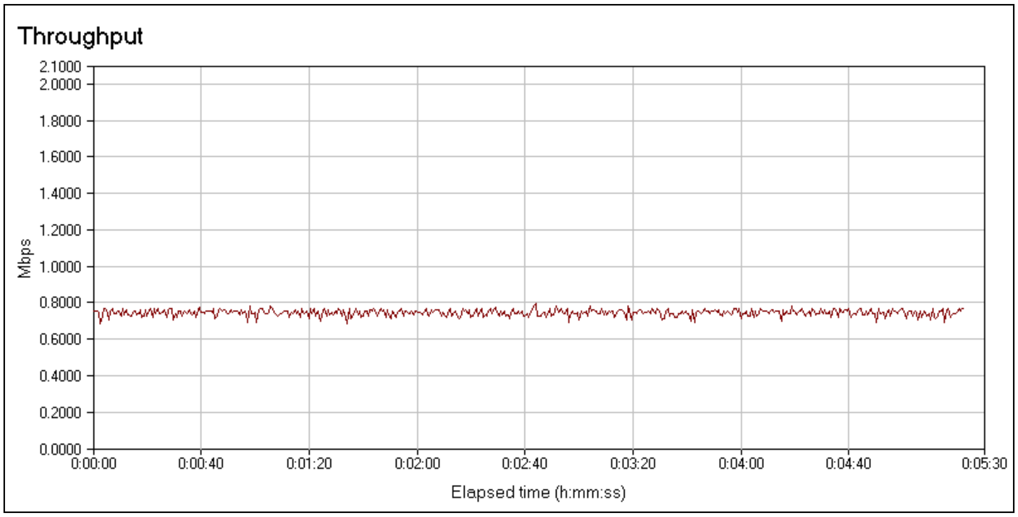


Figure 50. Long preamble device throughput for 1 Mbps bit rate.

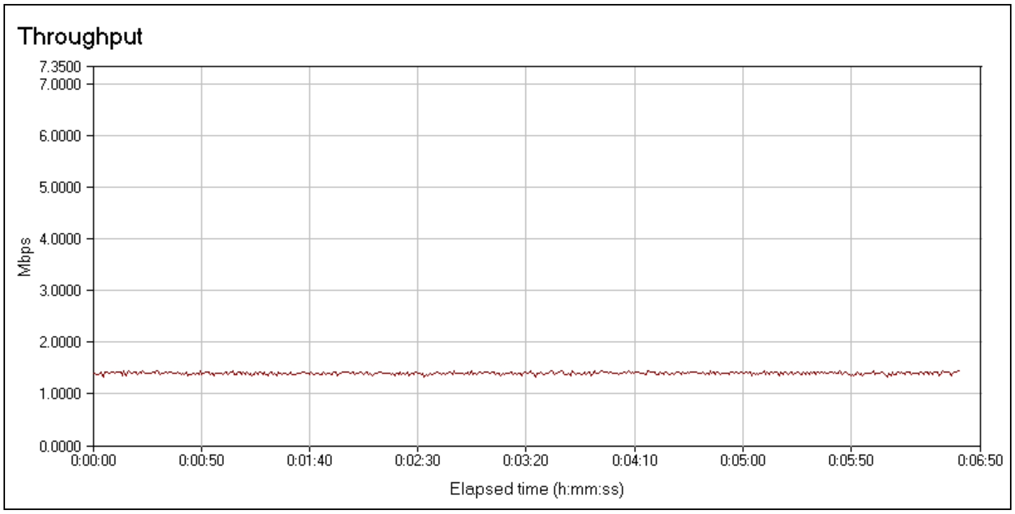


Figure 51. Long preamble device throughput for 2 Mbps bit rate.

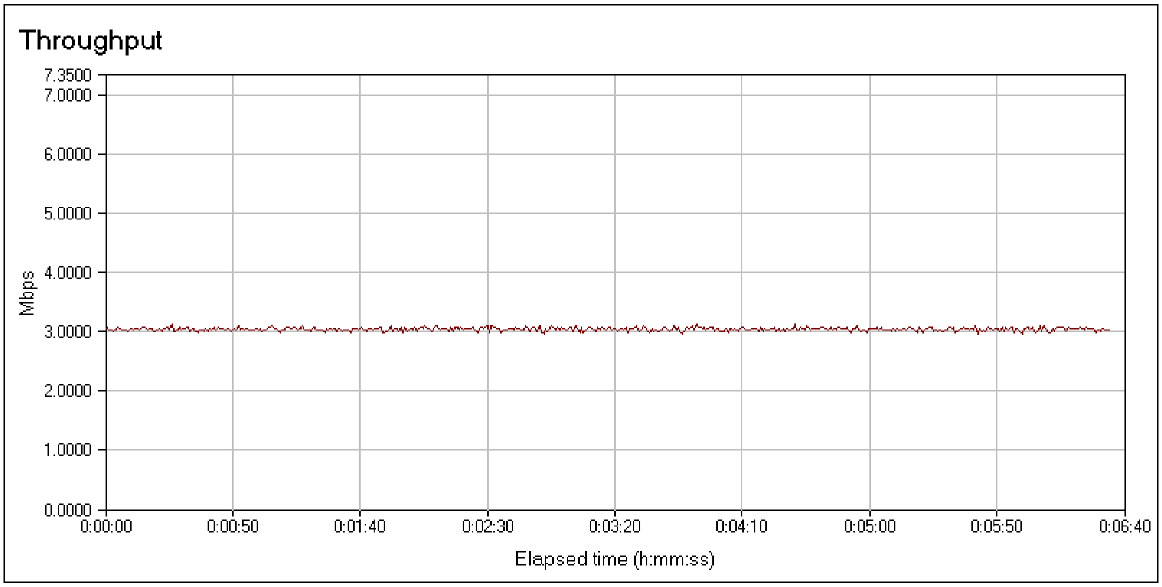


Figure 52. Long preamble device throughput for 5.5 Mbps bit rate.

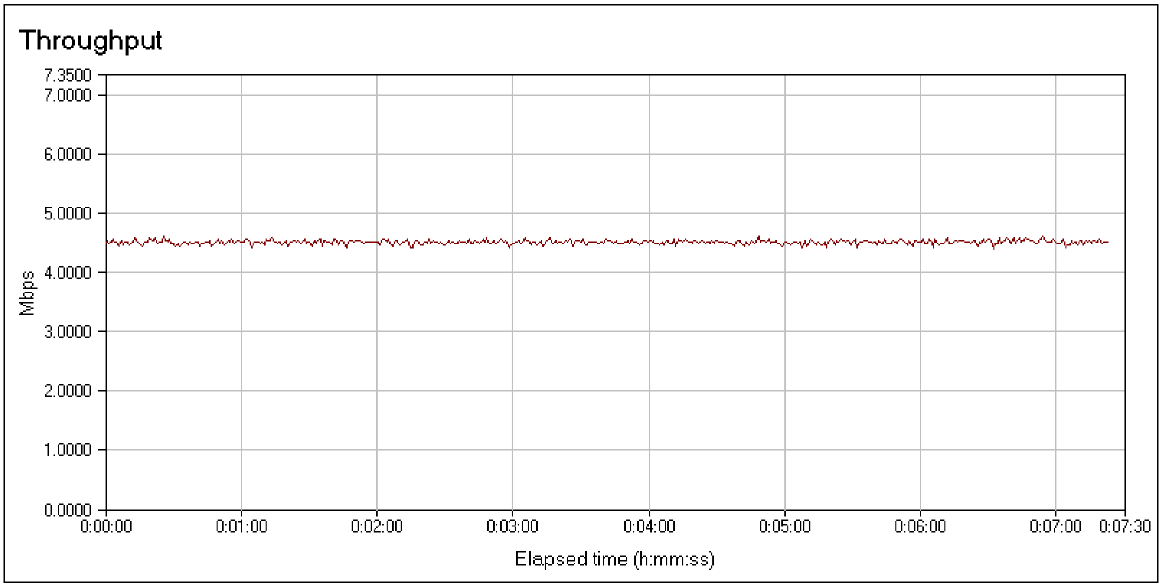


Figure 53. Long preamble device throughput for 11Mbps bit rate.

7.2.3 One-Hop Packet Loss Results

Given the close proximity of the nodes and controlled environment provided by the anechoic chamber, it is no surprise that the nodes exhibited no packet loss for the baseline tests. However, they also exhibited no packet loss for the converter tests as well. This would seem to indicate that the long preamble has sufficient time to recover fully from the small delay imposed by the converter extension.

7.3 Performance of a Two-Hop Network

An additional node was added as a client to from a three-node network. Since the nodes were not operating in ad-hoc mode, all communication was forced through the node acting as an AP. For comparison purposes, baseline tests were conducted to indicate how the 802.11b conversion device network performance compares with the ADI boards with no extension when a two-hop network is employed.

7.3.1 Baseline Testing

Baseline tests were performed in a similar manner to the one-hop network using the TCP network protocol. The resulting throughput versus time graphs can be found in Figures 54 - 57. As one might expect the throughput is roughly half of that of the one-hop network.

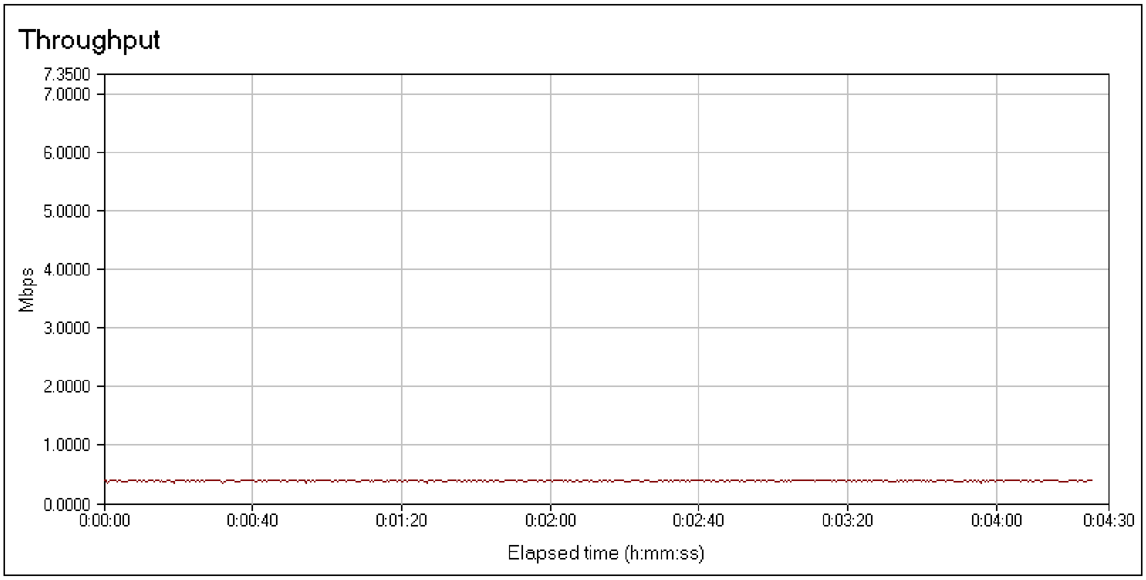


Figure 54. Long preamble baseline two-hop throughput for 1 Mbps bit rate.

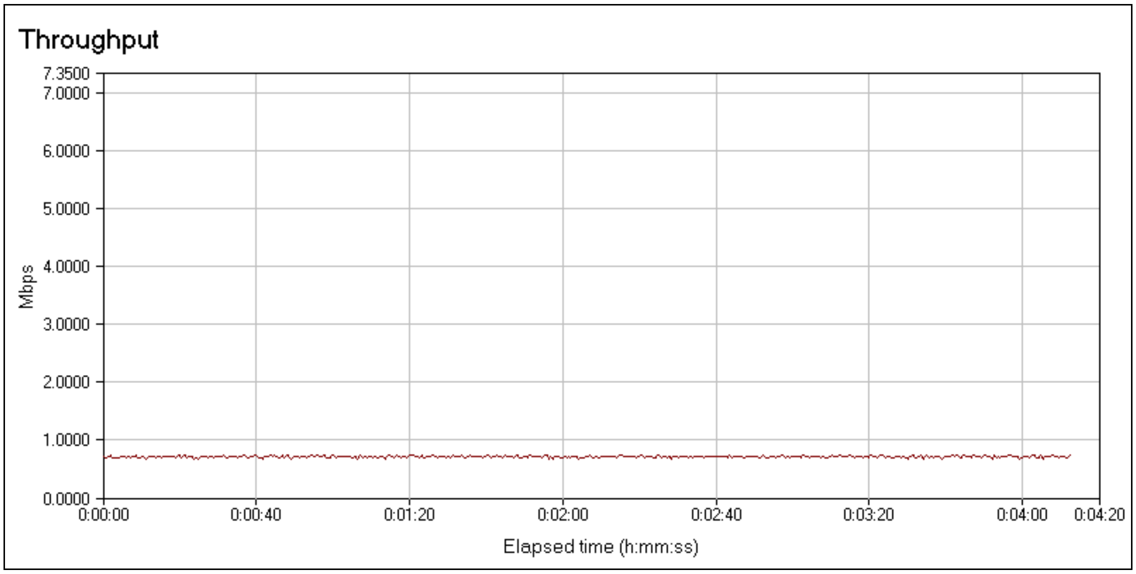


Figure 55. Long preamble baseline two-hop throughput for 2 Mbps bit rate.

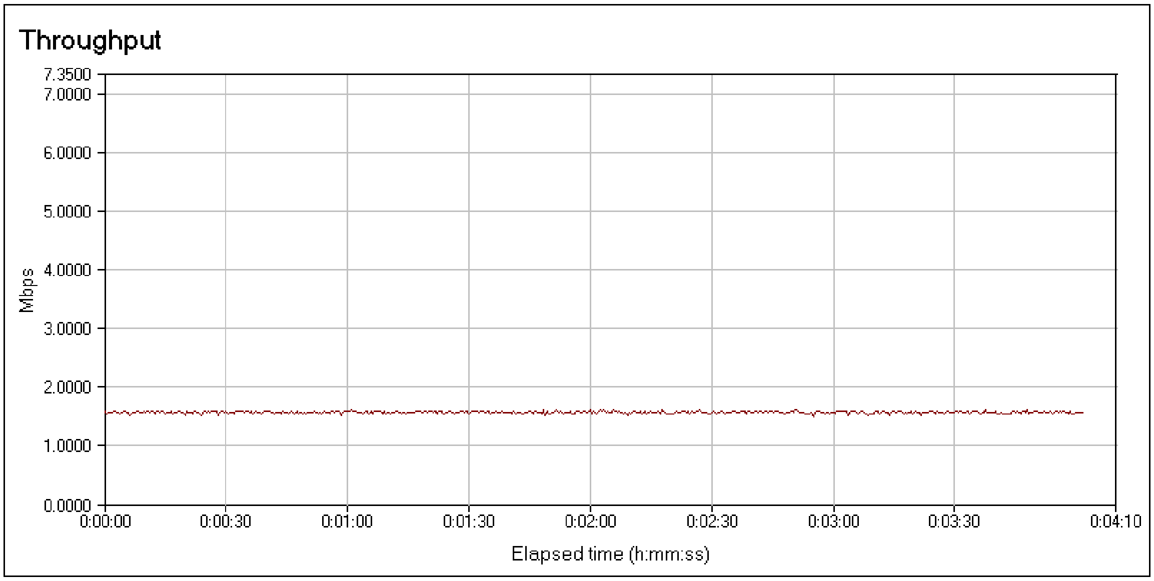


Figure 56. Long preamble baseline two-hop throughput for 5.5 Mbps bit rate.

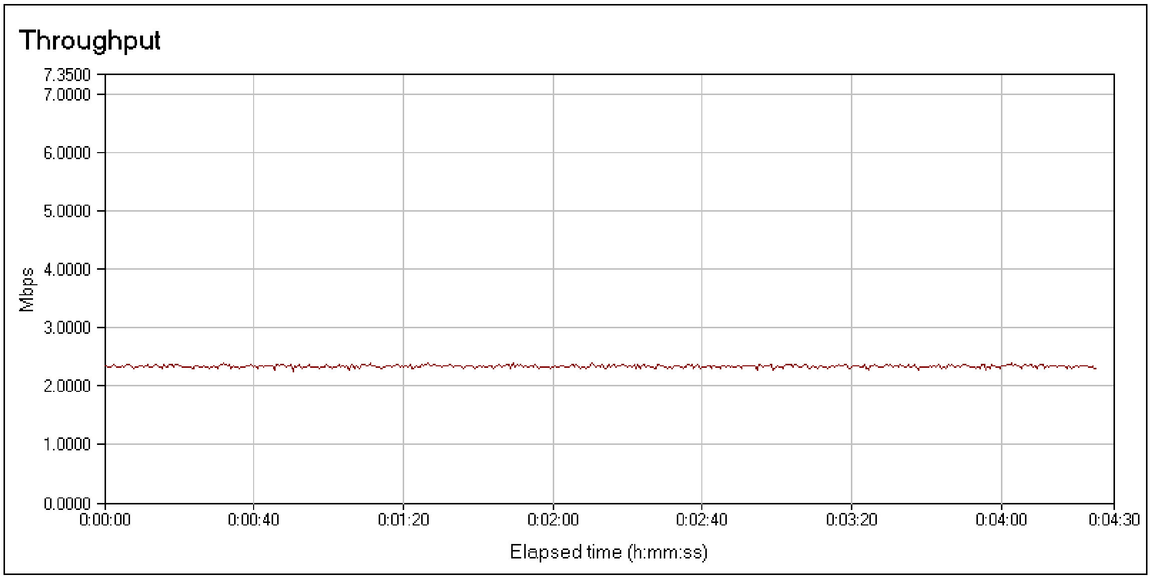


Figure 57. Long preamble baseline two-hop throughput for 11 Mbps bit rate.

7.3.2 Two-Hop Throughput Results

A frequency converter extension was added to each of the two client nodes and to the AP node, and additional throughput testing was performed. The average throughput results for the two-hop baseline and frequency converter tests are given in Table 9.

Table 9. Average two-hop throughput results.

Data Rate (Mbps)	Baseline (Mbps)	Converter (Mbps)
1.0	0.39	0.38
2.0	0.71	0.71
5.5	1.57	1.53
11	2.34	2.30

These numbers are slightly greater than half the one-hop scenario, most likely because the message need not be fully decoded after the first hop. The throughput graphs for the converter tests are illustrated in Figures 58 - 61.

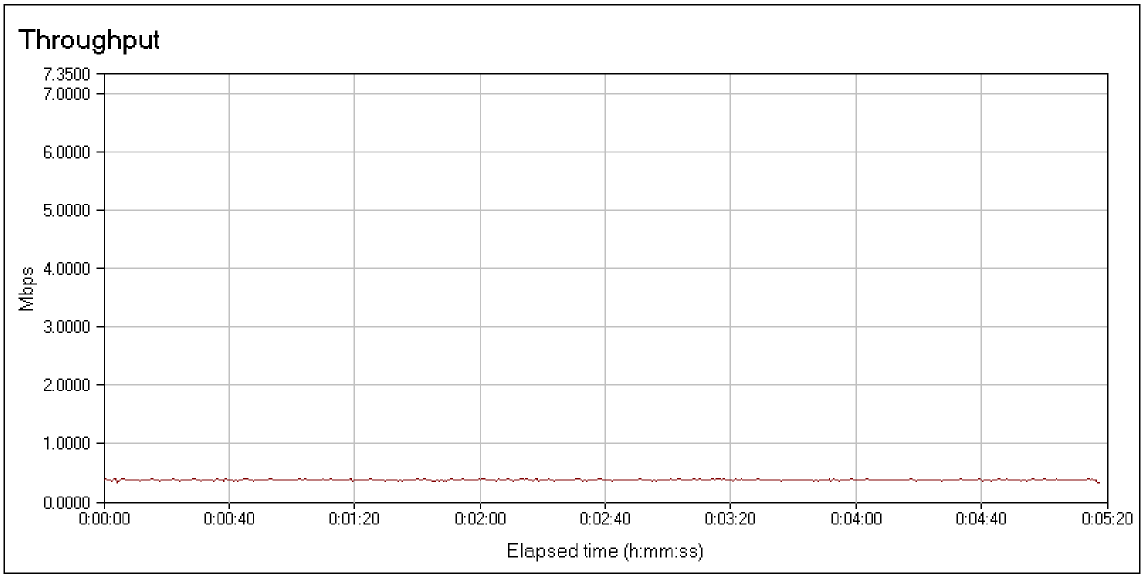


Figure 58. Long preamble device two-hop throughput for 1 Mbps bit rate.

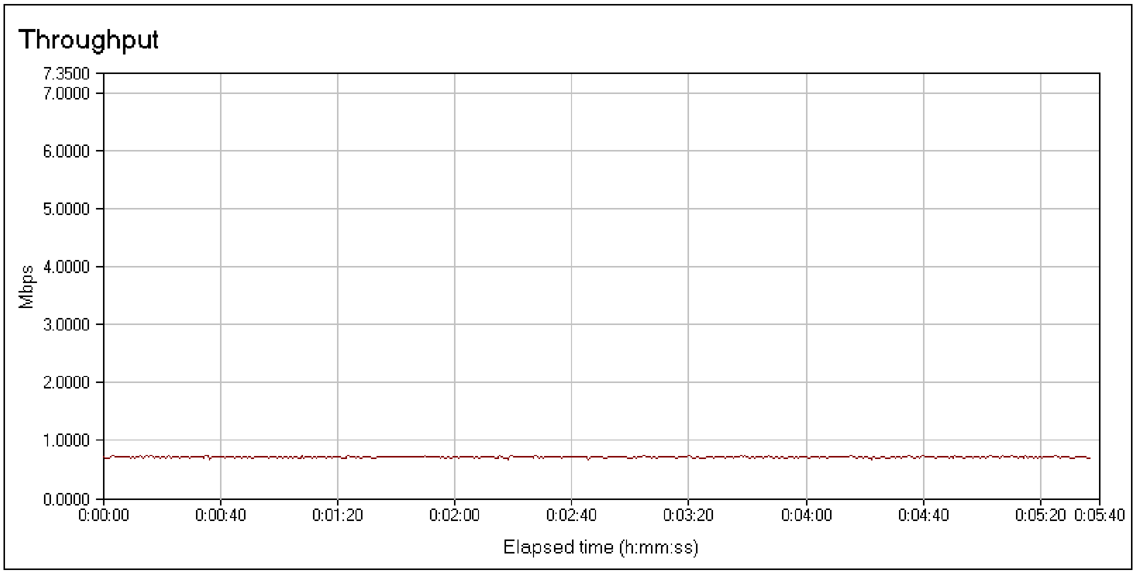


Figure 59. Long preamble baseline two-hop throughput for 2 Mbps bit rate.

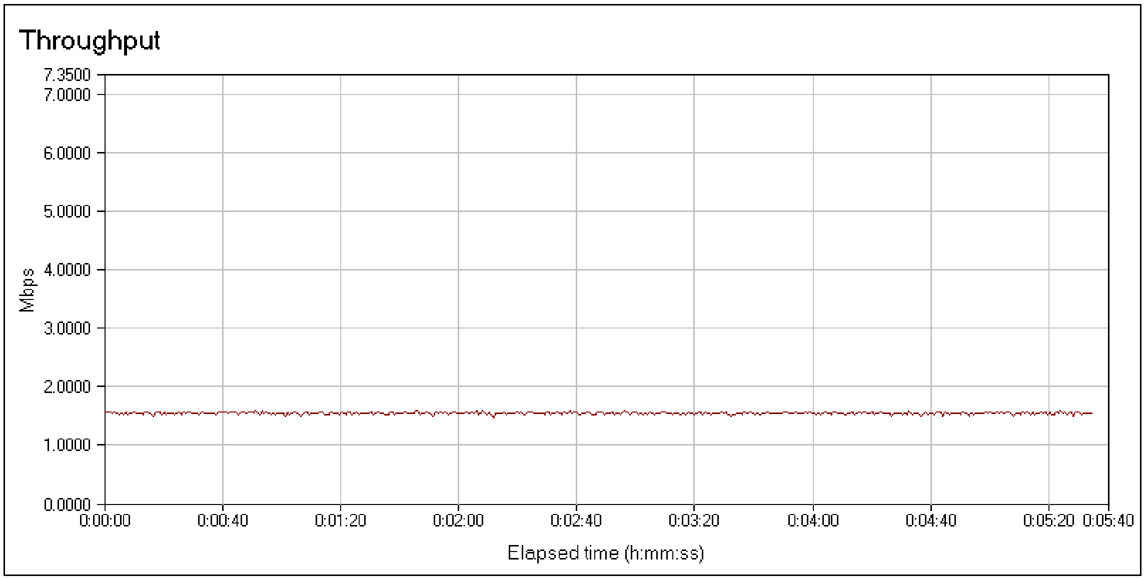


Figure 60. Long preamble baseline two-hop throughput for 5.5 Mbps bit rate.

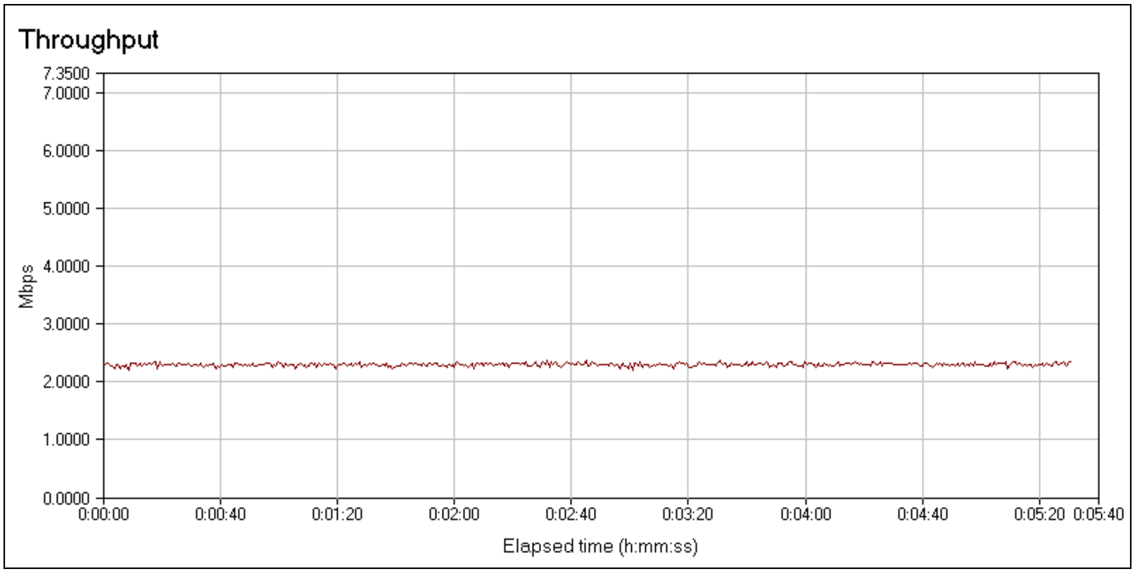


Figure 61. Long preamble baseline two-hop throughput for 11 Mbps bit rate.

7.3.3 Two-Hop Packet Loss Results

Although a three-node network makes packet collisions more likely, the close proximity of the nodes prevents any hidden or exposed terminal problems. As a consequence, much like the one-hop scenario, no packet loss was seen in the baseline or frequency converter extension testing.

7.3.4 Impact of Short Preamble

The nodes were unable to communicate when a short preamble was employed. The base switching delay imposed by the device was roughly 6 µs, which is ten percent of the entire preamble. It is thought that a smaller switching delay, particularly one lasting on the order of nanoseconds, would enable short preamble operations.

7.3.5 Impact on the ACK Timeout

As anticipated, the ACK timeout had no impact on the network throughput performance when a weakened preamble was employed. Because ACK transmission only occurs after a packet has been sent, the receiving node will already be defaulted to the receive path. Hence, directly after receiving the packet, it sends an ACK with a weakened preamble. The timing involved is unchanged, though the alteration of the ACK preamble may result in the packet being lost.

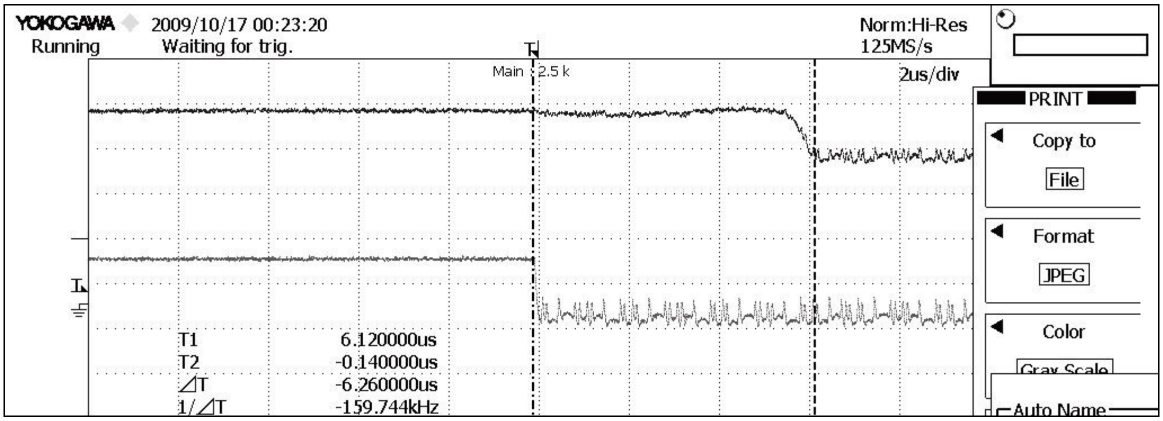
There are two procedures that typically occur during the ACK timeout evaluation. The last step to ensure that a received transmission is the intended acknowledgement is not relevant to this discussion. However, the first step is to determine the presence of a signal. This analysis is completed very rapidly, as the SIFS and propagation time have small durations. Therefore, when low ACK timeout distances (less than 300 m) were used during testing of the truncated preamble approach, communication failed. Thus a large timeout value was set during all testing performed in this chapter to ensure resulting throughput/packet loss of the truncated and weakened preamble designs were not unfairly biased.

7.4 Impact of Additional Switching Delay

The delay of both design approaches was found to be 6.1 μs. Using the delay buffers described in Chapter 6, additional delay was added to the switching circuit of the converter extension. Analysis of the impact on the throughput and packet loss was performed for several delay lengths at 1, 2, 5.5, and 11 Mbps. It should be noted that the delay of the power detector (approximately 400 ns) must be added to the delays mentioned in this section to obtain the overall delay imposed by the switching circuit.

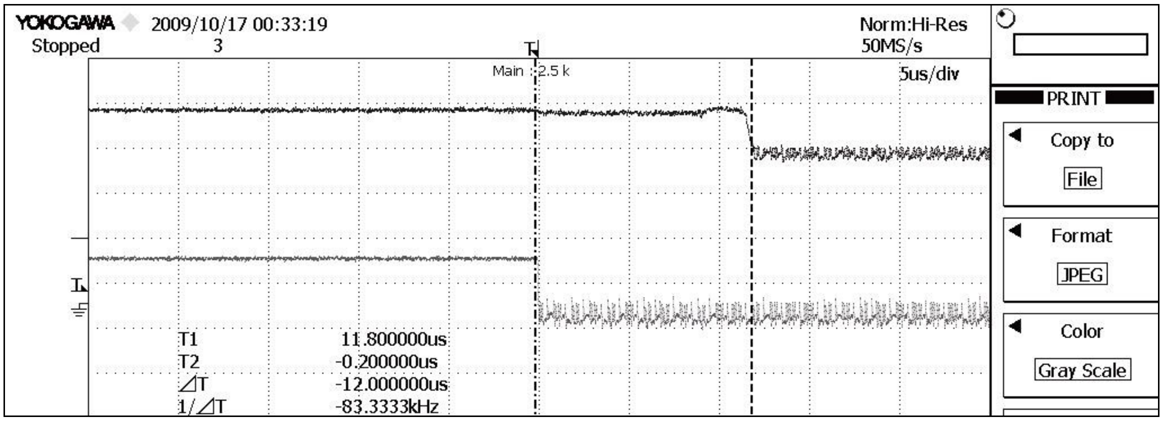
7.4.1 Effective Delays

The effective delays were measured by placing a power detector with a receive antenna directly next the antenna of the conversion extension and monitoring the output of this power detector and the power detector of the CE. Assuming they have roughly the same response time, the only additional delay imposed by the device is the response time of the original power detector as previously mentioned. Both the receive-transmit path (Rx-Tx) and the transmit-receive path (Tx-Rx) delays were recorded. It should also be noted that the buffer chips introduced some delay variation, mostly at the last few chips. A few examples of how the delay and delay variations were measured can be found in Figures 62 - 67. Additionally, Table 10 provides a summary of the delay characteristics imposed at the output of each buffer chip.



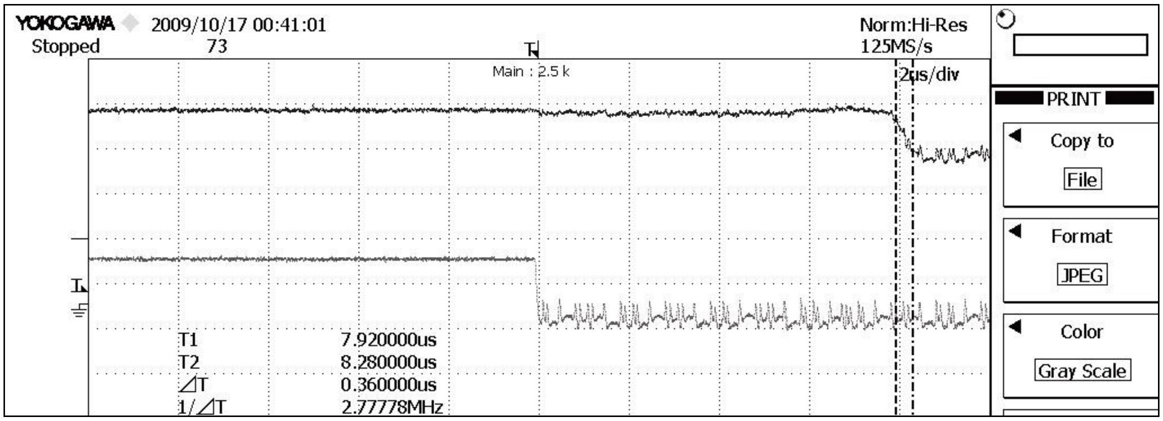
```
2009/10/17 00:23:20
Running        Waiting for trig.
Norm:Hi-Res
125MS/s
Main 2.5 k
2us/div
T1      6.120000us
T2     -0.140000us
⊿T     -6.260000us
1/⊿T   -159.744kHz
```

Figure 62. Rx-Tx delay after first buffer chip.


YOKOGAWA 2009/10/17 00:33:19
Stopped 3
Norm:Hi-Res
50MS/s
Main : 2.5 k
5us/div

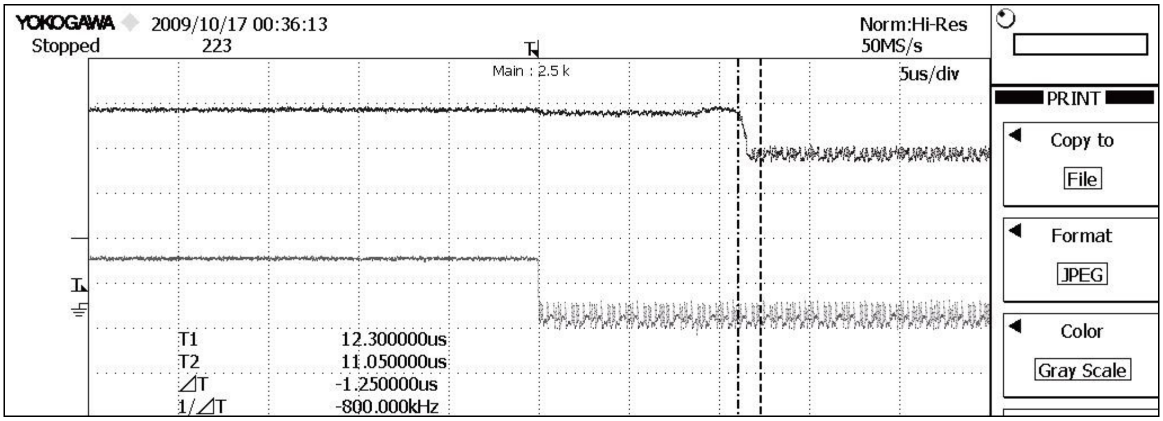
T1 11.800000us
T2 -0.200000us
⊿T -12.000000us
1/⊿T -83.3333kHz

Figure 63. Rx-Tx delay after sixth buffer chip.


YOKOGAWA 2009/10/17 00:41:01
Stopped 73
Norm:Hi-Res
125MS/s
Main : 2.5 k
2us/div

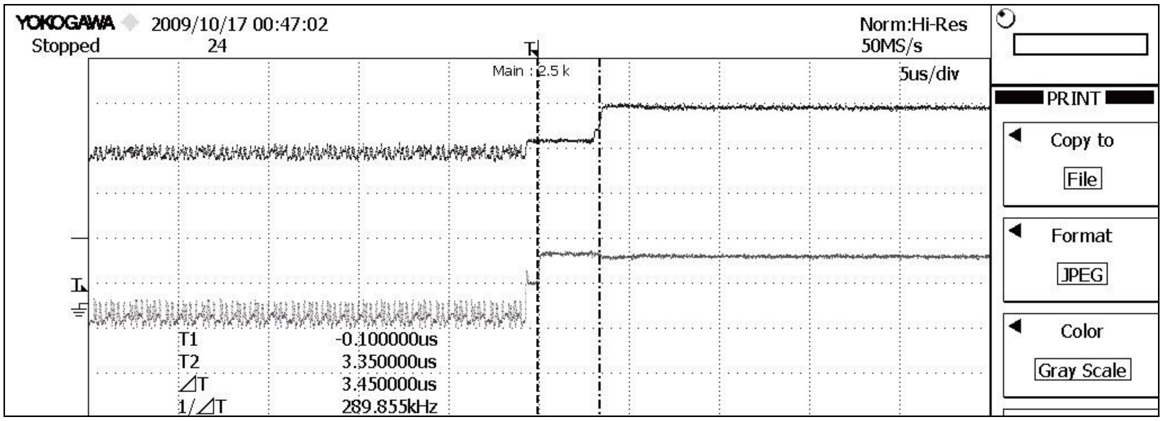
T1 7.920000us
T2 8.280000us
⊿T 0.360000us
1/⊿T 2.77778MHz

Figure 64. Maximum deviation of Rx-Tx delay (between vertical lines) after third buffer chip.


YOKOGAWA 2009/10/17 00:36:13
Stopped 223
Norm:Hi-Res
50MS/s
Main : 2.5 k
5us/div

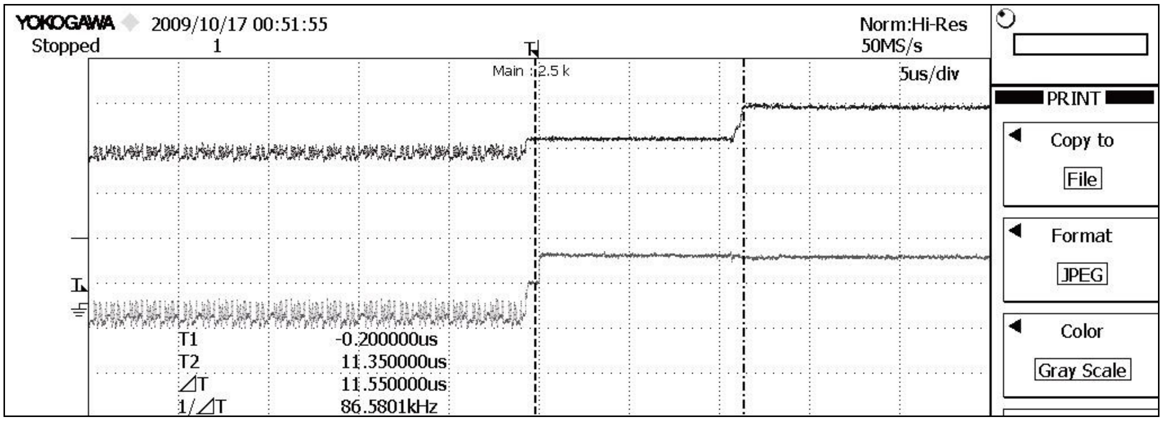
T1 12.300000us
T2 11.050000us
⊿T -1.250000us
1/⊿T -800.000kHz

Figure 65. Maximum deviation of Rx-Tx delay (between vertical lines) after sixth buffer chip.


YOKOGAWA 2009/10/17 00:47:02
Stopped 24
Norm:Hi-Res
50MS/s
Main : 2.5 k
5us/div

```
T1     -0.100000us
T2      3.350000us
⊿T      3.450000us
1/⊿T    289.855kHz
```

Figure 66. Tx-Rx delay after first buffer chip.


YOKOGAWA 2009/10/17 00:51:55
Stopped 1
Norm:Hi-Res
50MS/s
Main : 2.5 k
5us/div

```
T1     -0.200000us
T2      11.350000us
⊿T      11.550000us
1/⊿T    86.5801kHz
```

Figure 67. Tx-Rx delay after sixth buffer chip.

Table 10. Delays at buffer amplifier outputs.

Chip #	Typical Rx-Tx Delay (μs)	Max Rx-Tx Variation (μs)	Typical Tx-Rx Delay (μs)	Max Rx-Tx Variation (μs)
1	6.3	<0.1	3.5	<0.1
2	7.1	<0.1	4.9	<0.1
3	8.5	0.2	6.5	0.25
4	10.1	0.45	8.0	0.60
5	11.2	0.55	10.0	0.70
6	12.0	0.75	11.6	0.75

Based on the results, there is a significant difference in the Tx-Rx and Rx-Tx delay, particularly at smaller delay times. The Tx-Rx delay was generally much smaller for small delays but approached the Rx-Tx time for larger delays. It should also be noted that the delay variation increases when additional switching delay is added. This variation was measured by allowing the oscilloscope to run with a refresh rate of about 2s and adjusting the cursors to adjust for the maximum deviation. While a digital delay using counters could have been employed to obtain exact delays, it was deemed more important to preserve the analog delay maintaining the physical characteristics of the output at the power detector.

7.4.2 Impact on Throughput and Packet Loss

The additional switching delay does not impact the lower data rates until the output of the fifth buffer chip. The throughput suffers a small degradation that becomes much worse at the output of the sixth buffer chip. This is true for both the 1 Mbps and 2 Mbps data rates. Because the throughput degrades for different delays at the higher data rates, the errors occur in the physical layer. The weakened preamble has not trained the receiver adequately for proper reception. Subsequent packet loss testing showed that at this amount of delay duplicate packets are sent (due to a lost ACK) and packets are lost during transmission from N1 to N2. The general throughput behavior for various delays is illustrated on the next few pages in Figures 68 – 73.

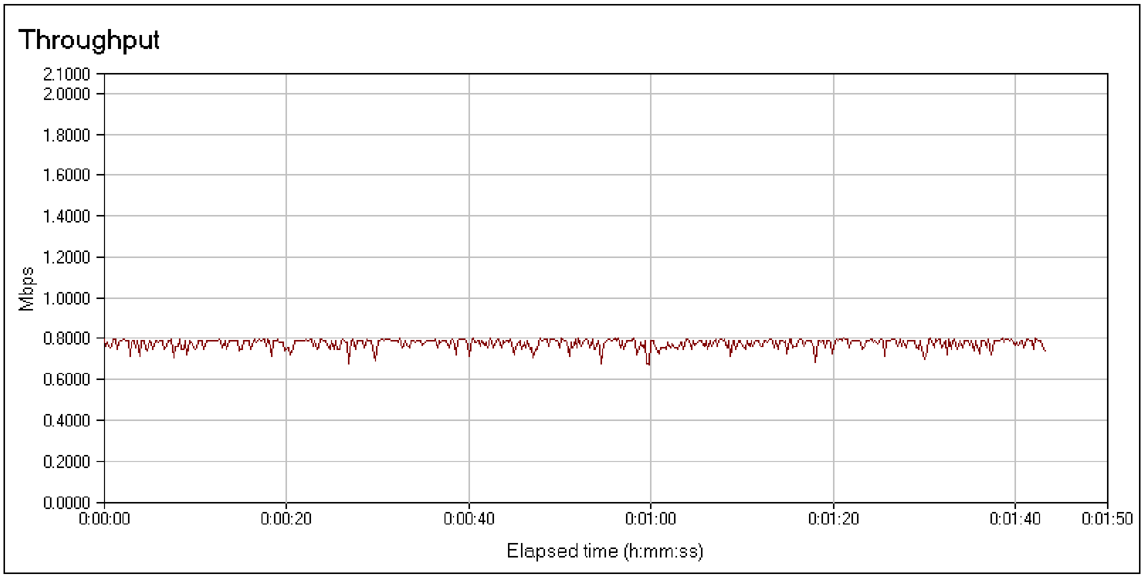


Figure 68. Throughput at 1 Mbps with four buffer chips added to switching delay.

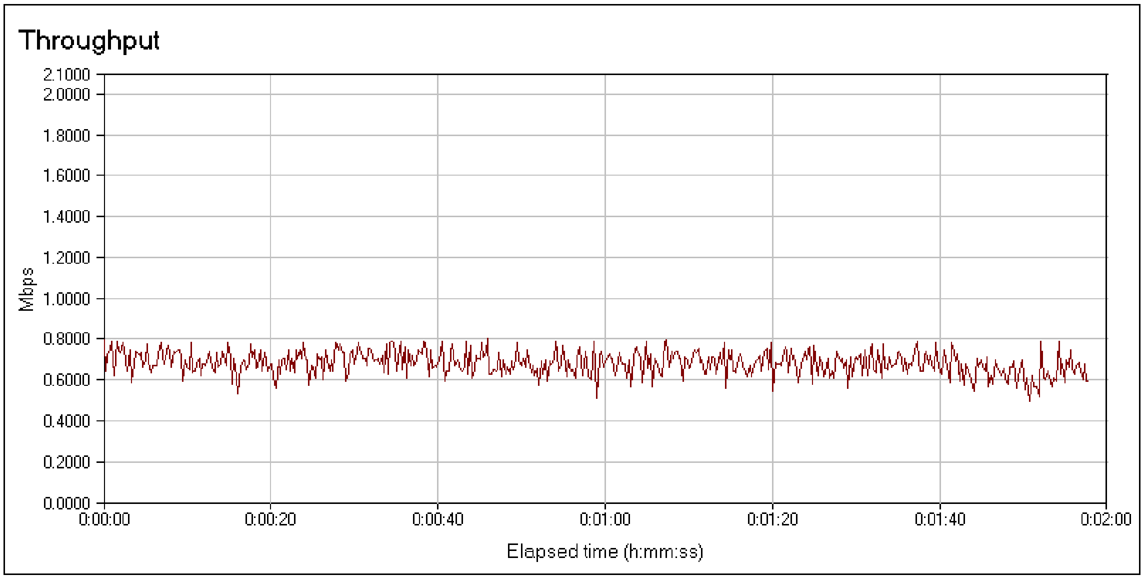


Figure 69. Throughput at 1Mbps with five buffer chips added to switching delay.

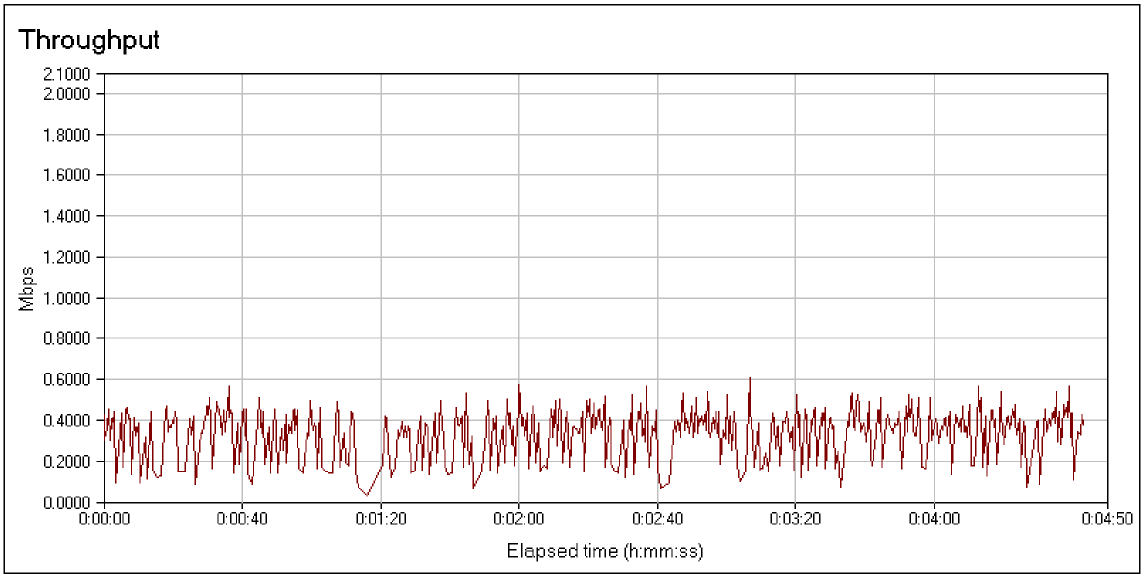


Figure 70. Throughput at 1 Mbps with six buffer chips added to switching delay.

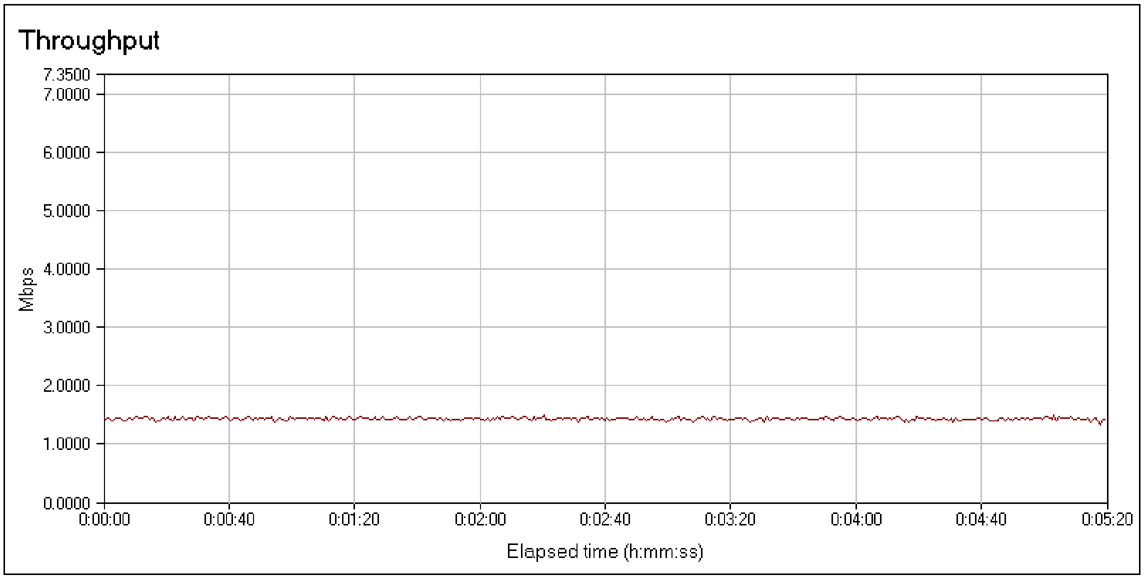


Figure 71. Throughput at 2 Mbps with four buffer chips added to switching delay.

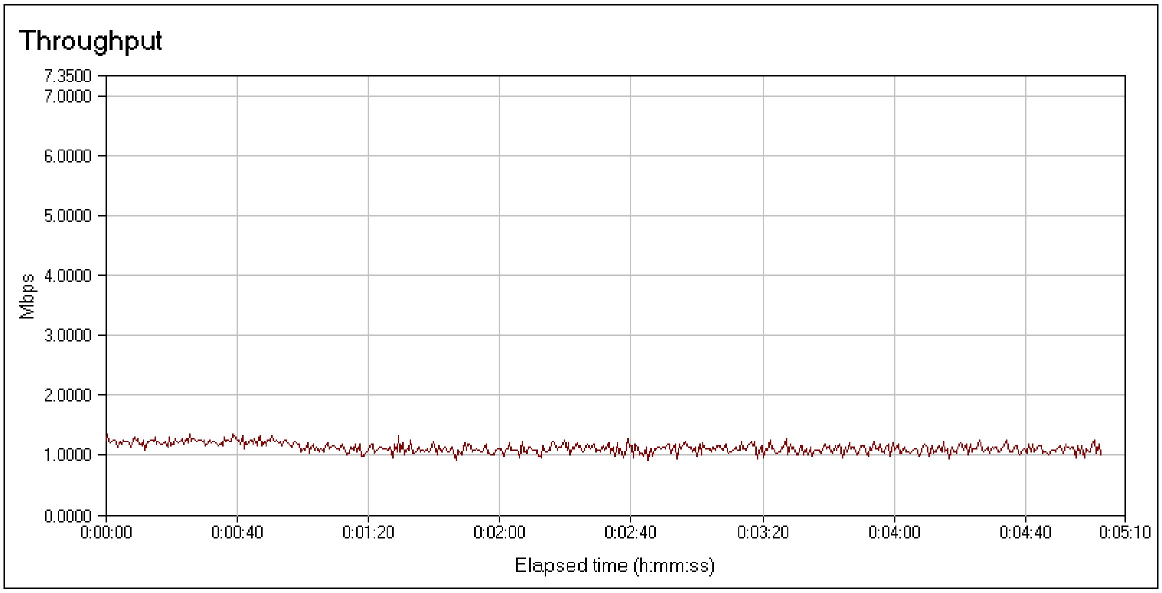


Figure 72. Throughput at 2 Mbps with five buffer chips added to switching delay.

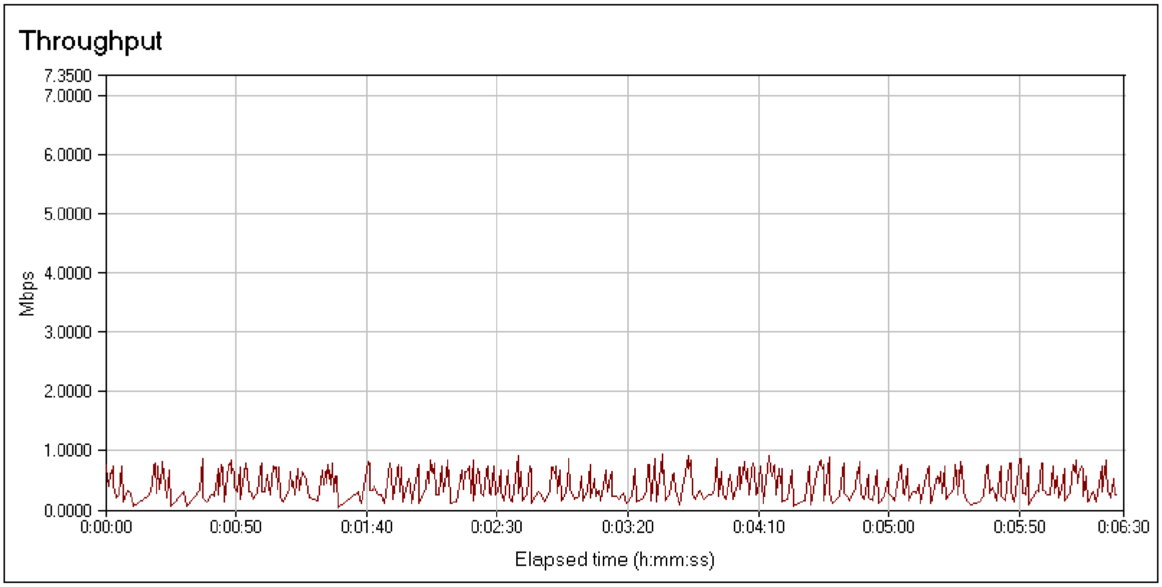


Figure 73. Throughput at 2 Mbps with six buffer chips added to switching delay.

The higher data rates also show decreased throughput with an increase in switching delay. However, this degradation is much more abrupt and occurs at smaller delays (output of third buffer chip). The lack of symmetrical behavior between the high and low data rates suggests that a physical layer problem exists and is caused by the added delay circuit. Because adding switching delay has no impact on the overall noise figure of the system, the source of the degradation is therefore attributable to any or all of the preamble operations including AGC, equalization, or frequency/symbol synchronization. The physical layer problems at the higher data rates, however, do not eliminate the possibility that the lower data rates throughput degradation is MAC-related. Figures 74 - 79 show the throughput at data rates of 5.5 Mbps and 11 Mbps for various delays, and Tables 11 - 14 provide a comparison of packet loss for the test scenarios developed in this section.

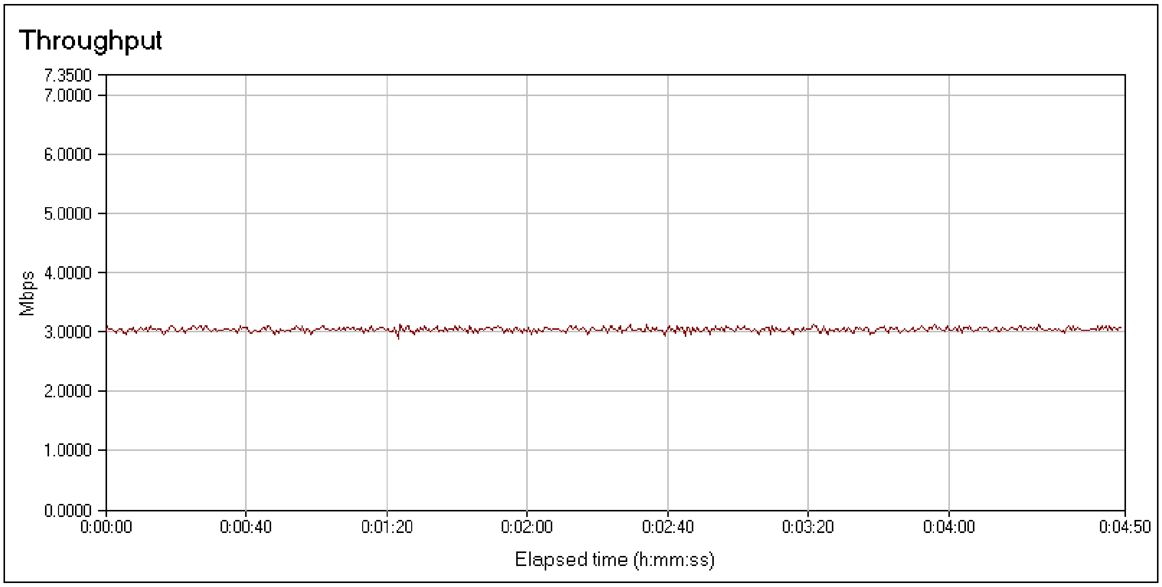


Figure 74. Throughput at 5.5 Mbps with one buffer chip added to switching delay.

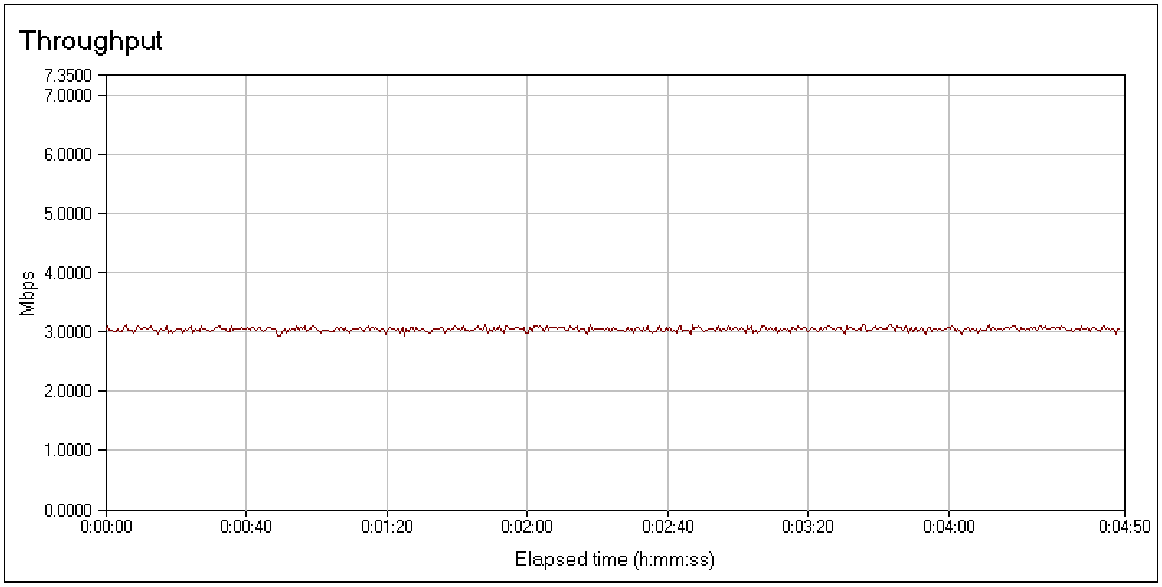


Figure 75. Throughput at 5.5 Mbps with two buffer chips added to switching delay.

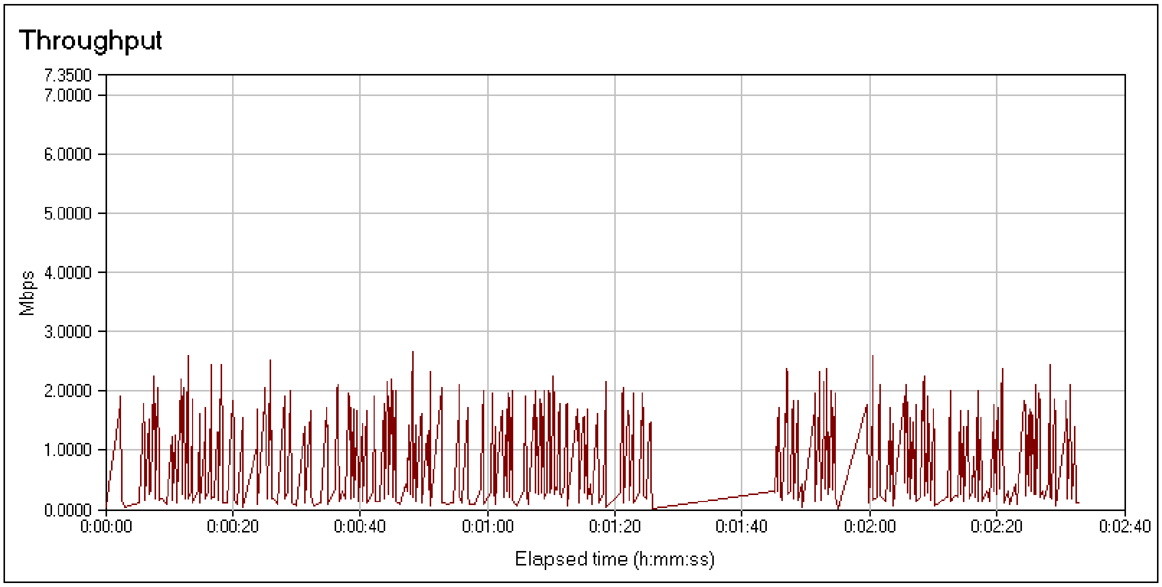


Figure 76. Throughput at 5.5 Mbps with three buffer chips added to switching delay.

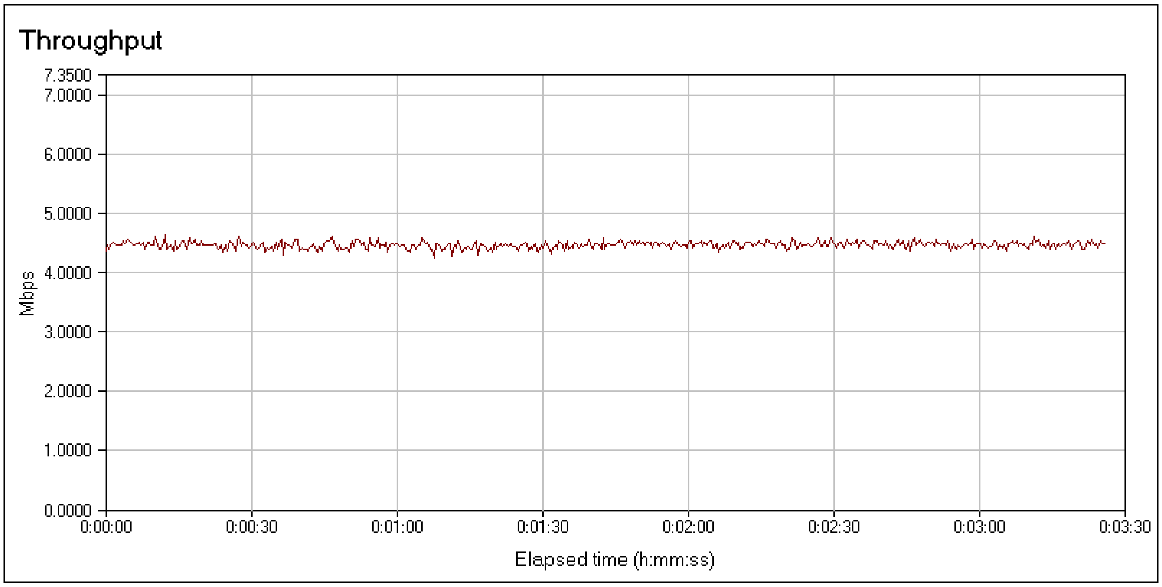


Figure 77. Throughput at 11 Mbps with one buffer chip added to switching delay.

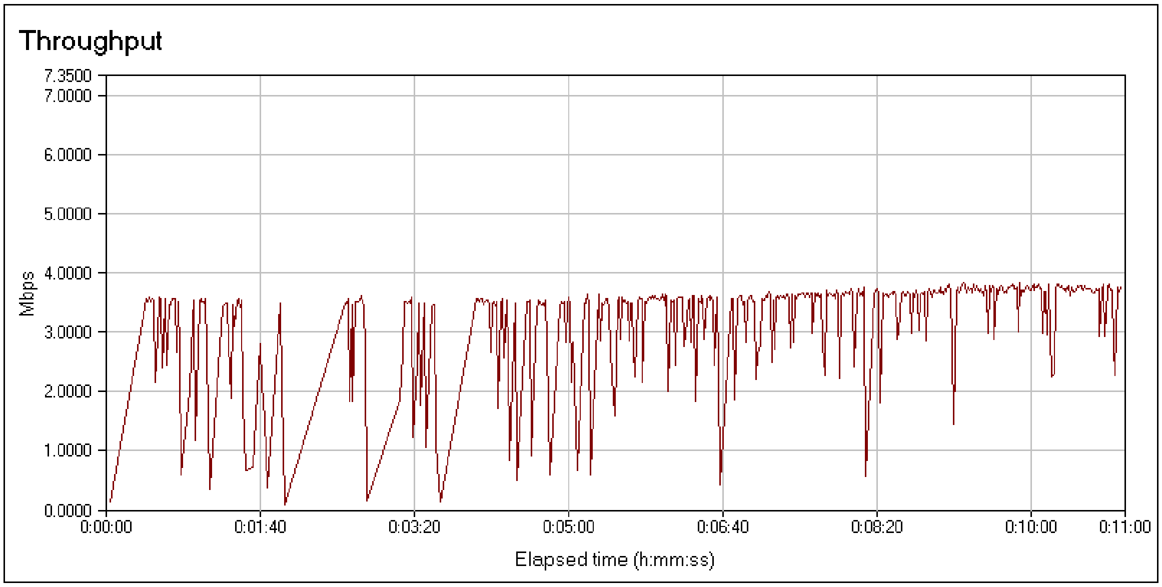


Figure 78. Throughput at 11 Mbps with two buffer chips added to switching delay.

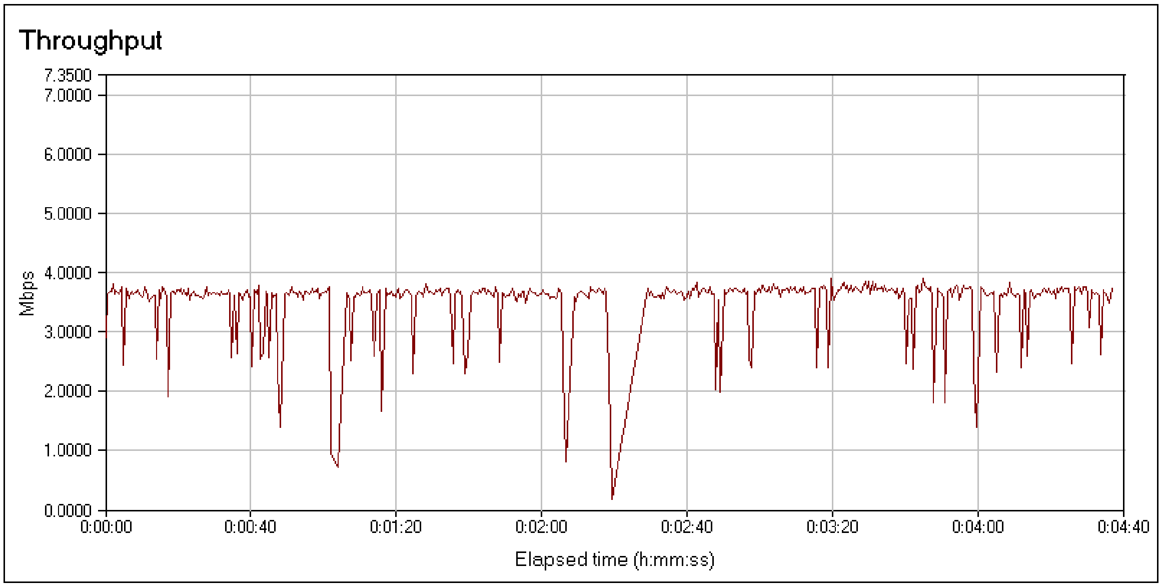


Figure 79. Throughput at 11 Mbps with 2.5 buffer chips (10 buffers) added to switching delay.

Table 11. Packet loss vs. delay at 11 Mbps.

Chip #	Packets Sent	Duplicates Sent	Lost	Loss (%)
1	50008	0	0	0
2	50008	7	0	0
3	Communication Fails			
4	Communication Fails			
5	Communication Fails			
6	Communication Fails			

Table 12. Packet loss vs. delay at 5 Mbps.

Chip #	Packets Sent	Duplicates Sent	Lost	Loss (%)
1	48001	0	0	0
2	48001	0	0	0
3	4739	1238	1238	26.1
4	Communication Fails			
5	Communication Fails			
6	Communication Fails			

Table 13. Packet loss vs. delay at 2 Mbps.

Chip #	Packets Sent	Duplicates Sent	Lost	Loss (%)
1	24002	0	0	0
2	24002	0	0	0
3	24002	0	0	0
4	24002	1	0	0
5	18517	16	0	0
6	6621	774	745	12.7

Table 14. Packet loss vs. delay at 1 Mbps.

Chip #	Packets Sent	Duplicates Sent	Lost	Loss (%)
1	11001	0	0	0
2	11001	0	0	0
3	11001	0	0	0
4	11001	0	0	0
5	11007	6	0	0
6	4854	353	342	7.6

7.4.3 Impact of Fragmentation

Breaking up the packet into multiple smaller packets sent during the normal transmission time, or fragmentation as it is known, resulted in an increased throughput provided that the fragment size was chosen to be fairly large (greater than one third the normal packet size). This reaffirms the position that packet loss is occurring in the network. It should be noted that if the packet is broken up too small, the additional overhead eliminates the improved throughput seen by fragmenting. Throughput results for two levels of fragmentation are shown below. Figure 80 shows poor throughput performance when fragment size of 480 bytes is used while Figure 81 shows substantially more stable throughput when a fragment size of 700 bytes is used.

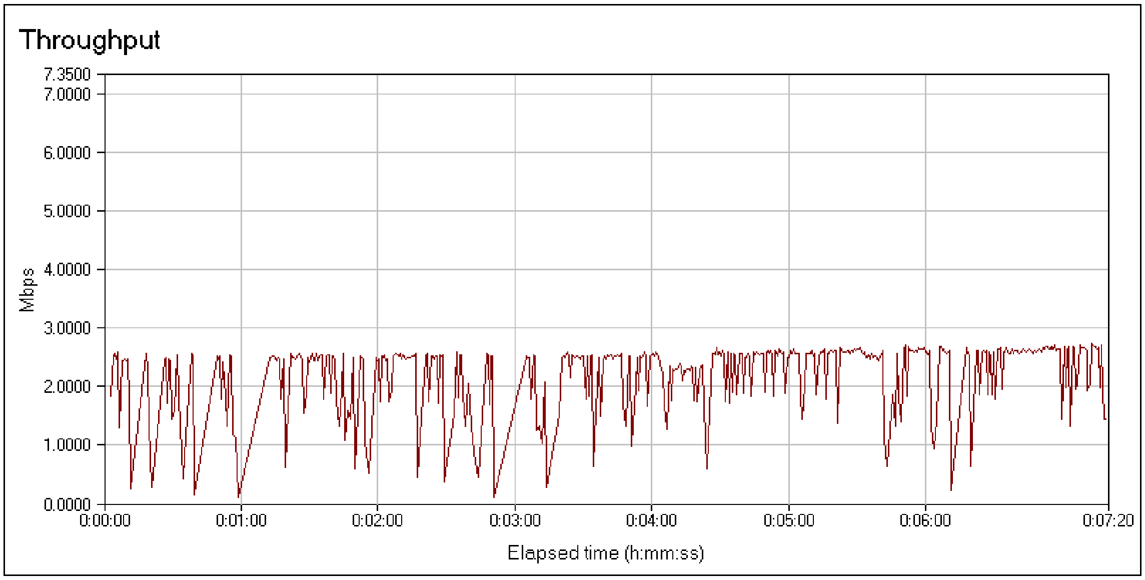


Figure 80. Throughput at output of second buffer delay (11 Mbps) with 480 byte fragment size.

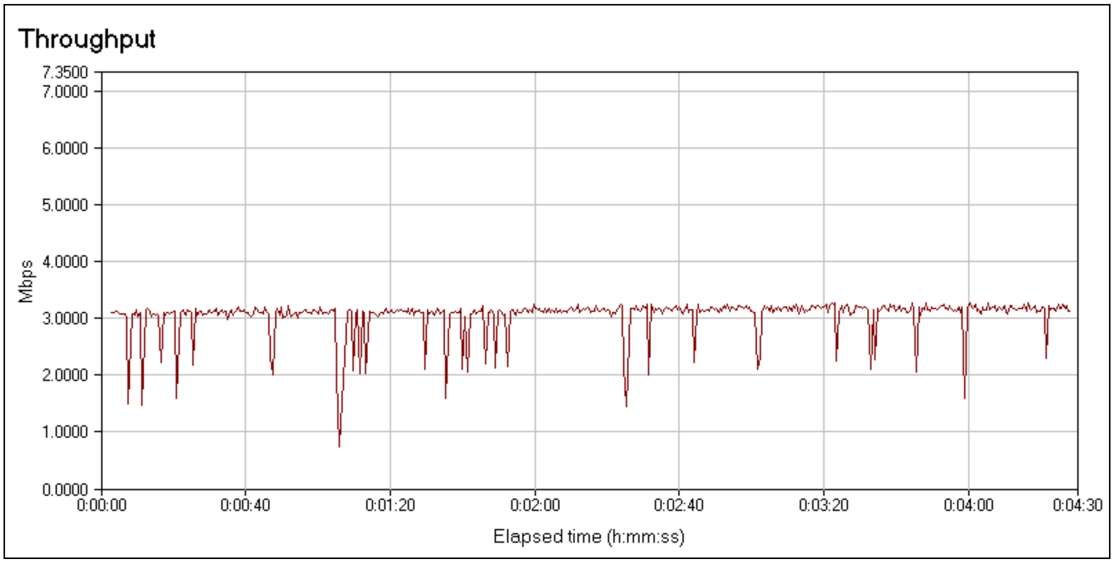


Figure 81. Throughput at output of second buffer delay (11 Mbps) with 700 byte fragment size.

7.5 Comparison of Truncated and Partially-Weak Preamble

Testing up to this point was conducted on nodes using the partially weak preamble hardware design, as it was thought that the base devices physical layer would be more resistant to a change in amplitude then the complete absence of a portion of the packet. The results, however, show quite the opposite. While operation with the short preamble remained unattainable, the truncated preamble architecture did not suffer from any significant throughput degradation under any of the testing scenarios previously discussed. This was true even at the highest delay that was tested (roughly 12 μs), as can be seen in Figure 82. This shows that for low density networks, decay in the throughput is caused by physical layer operations, not the MAC. Furthermore, it would indicate that only amplitude sensitive physical layer processes impact the system including signal detection, AGC, frequency synchronization, equalization, and the setting of thresholds.

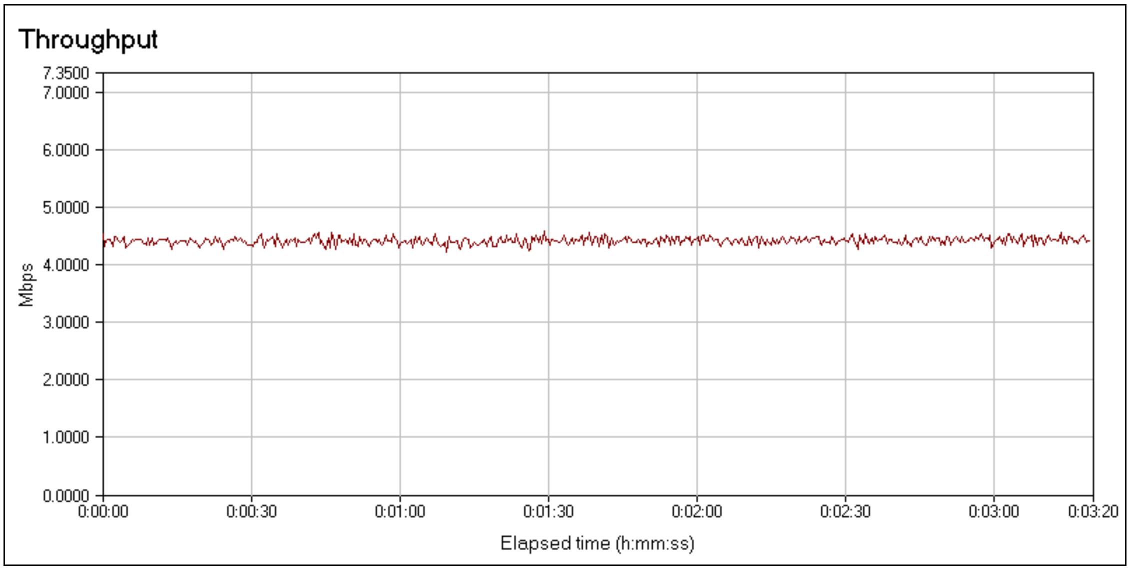


Figure 82. 11 Mbps throughput at maximum delay for truncated preamble architecture.

7.6 Testing with 802.11g

The conversion device was unable to establish an 802.11g link for any transfer rate. This was true even when an 802.11b node was placed within the network to force the nodes to employ the DSSS-OFDM packet structure. Considering that over half of the short-symbol training sequence is modified, it is not surprising that communication is not possible. It is also believed that the high PAPR coupled with the fairly quick response time of the power detector employed caused the switching to be interrupted in the middle of the packet. Figures 83 and 84 show examples of the output of the power detector when 802.11g is used. Of particular interest is the presence of an interruption in transmission in Figure 84 with duration less than the SIFS. This confirms that OFDM power detection in 802.11g does pose additional design constraints.

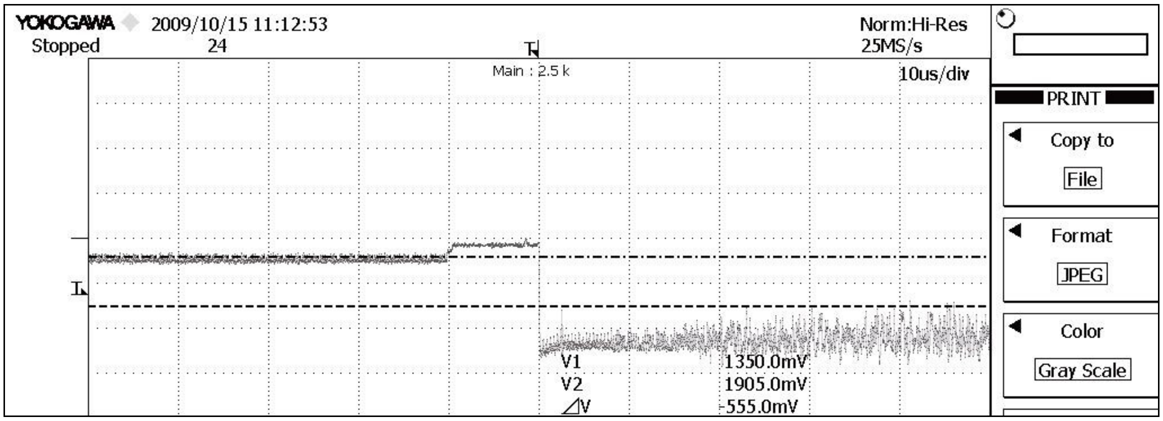


Figure 83. Example output of power detector when 802.11g is employed.

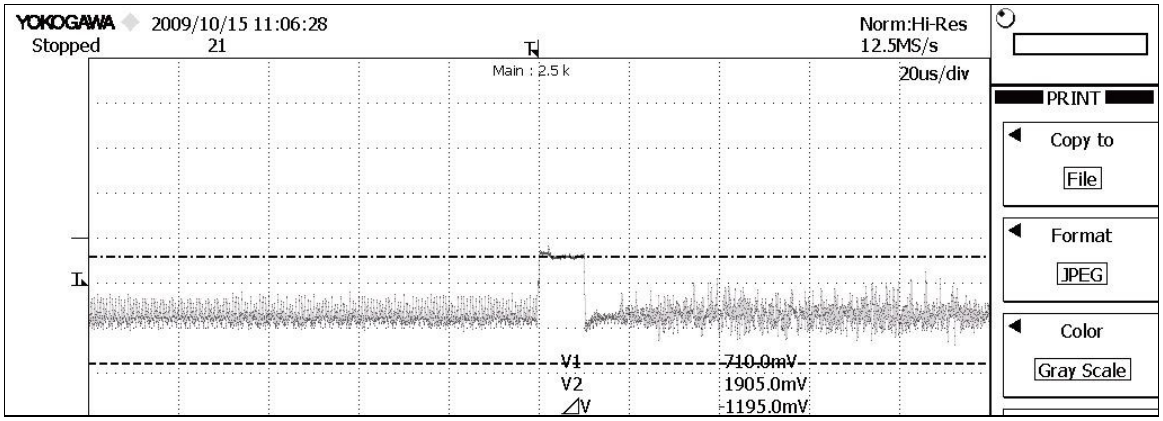


Figure 84. Second example of the output at the power detector when 802.11g is employed.

7.7 Summary of Results

Both design architectures enabled frequency conversion at 1700-1750 MHz. The resulting throughput was also found to be comparable with baseline tests performed on the ADI engineering nodes. As the delay was increased, the weakened-preamble architecture suffered from significant throughput degradation. The same, however, could not be said for the truncated-preamble architecture, which withstood delays in excess of 12 μs without a decrease in throughput. Based on this result, it was concluded that the packet losses seen originated in the physical layer, as any MAC layer issues would have had a greater impact on the truncated preamble architecture. It should be noted, however, that this may not hold true for high density networks as discussed in Chapter 7.

Both designs had limitations as neither could support short preamble operation or 802.11g. It is thought that this is a result of the dramatically shortened training periods both of these implementations employ and, in the case of 802.11g, may also have been due to the quick response time of the power detector and large amplitude fluctuations caused by OFDM.

8. Conclusion

An 802.11b frequency conversion extension was designed and tested to assess the feasibility of enabling cognitive capabilities in existing wireless communication systems. The structure and limitations of a software-defined radio were described in Chapter 2 in order to establish the configurable architecture that typically accompanies a cognitive system. Thorough background research was conducted on the applications of cognitive radio including blind modulation, symbol rate, and frequency estimation. However, the current focus of cognitive radio research is on intelligent sharing of spectrum resources. To enable this function in existing devices, it is necessary to convert the output signal to different frequencies without significantly altering the physical and MAC layer performance of the base device. This necessitates a good understanding of these layers in the 802.11b protocol.

The 802.11b protocol uses DSSS, employing DBPSK/DQPSK for the lower data rates and CCK for the upper rates. At the MAC Layer, CSMA is used in conjunction with an RTS-CTS exchange to eliminate the hidden node problem. The design of the conversion extension alters the performance of processes at both layers. The output power of the base device is coupled to a power detector and switching circuit in order to intelligently enable the correct transmit/receive path. The switching time introduced by this design causes the training procedure in the SYNC field to operate under a reduced training time and extends the inter-frame spacing of the MAC. A preliminary wired architecture connecting two extensions was designed and tested to expose flaws, including improper switch isolation and RF leakage of the 2.4 GHz signal. A wireless design was then developed that used two different receive paths designs: one that truncates the preamble thereby favoring the PHY-layer (by not providing false training information), and another that utilizes a weakened portion of the SYNC field leaving the close range MAC behavior undisturbed.

Results showed that both of the considered designs had throughput and packet loss statistics comparable with the base device. However, communication using the short preamble was not achieved, which is likely due to the fact that a 6 μs delay represents more than ten percent of the SYNC field. As delay was added to the system using a buffer delay-line, it became clear that the truncated preamble approach was superior, as it sustained delays up to roughly 14 μs. The use of fragmentation confirmed that the decrease in throughput was directly attributable to packet loss. Although 802.11g was not the focus of this work, testing was attempted on this protocol without success. Still, the theoretical signal detection problems caused by a high PAPR were confirmed through testing.

The results of this work show that it is feasible to produce a cognitive extension to an 802.11b device, and as the switching delay is made increasingly smaller, a cognitive extension to a wireless system in general. The primary requirement for an extension to be applicable is a half duplex system, or alternatively a full duplex system employing separate transmit and receive antennas. Making use of the physical and MAC layer procedures in the base device reduces the design time and cost while simplifying the overall architecture. Furthermore, a single cognitive extension can be designed for operation with multiple wireless platforms. Only minor changes in the presented design would be needed to achieve complete frequency and power configurability. These would include the additions of a programmable gain control loop in the transmit path and an extra conversion stage to a frequency well outside of the operating range of the extension (coupled with appropriate filtering). The extra conversion stage simplifies filtering after the final conversion stage by making the translating frequency very large, and thus allowing the extension to work for a wide range of frequencies. While current cognitive research is largely focused on the emerging 802.22 standard, it is expected to expand into higher frequency/bandwidth scenarios in the future. The analysis developed in this work shows that the design of a complete cognitive radio is usually unnecessary and redundant, as an extension can fully meet the needs of the network.

9. References

[1] R. I. Lackey, and D. W. Upmal, “Speakeasy: the military software radio,” *IEEE Communications Magazine* 33, no. 5 (1995): 56-61.

[2] P. G. Cook, and W. Bonser, “Architectural overview of the SPEAKeasy system,” *IEEE Journal on Selected Areas in Communications* 17, no. 4(April 1999): 650-661.

[3] J. Melby, “JTRS and the evolution toward software-defined radio,” in *Proceedings MILCOM 02',* vol. 2 (Oct 7-10 . 2002): 1286-1290.

[4] Joint Tactical Radio Systems (JTRS) Joint Program Office*, Software Communications Architecture Specification*, 2006.

[5] J.-S. Lee, J.-H. Park, S.-W. Kim, Y. Li, and H-G. Ryu, “Implementation of DSP-based digital receiver for the SDR application,” in 5th Int'l Symp. on Multi-Dim. Mobile Comm. (2004): 6-10.

[6] Y. Lin, H. Lee, M. Who, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, “SODA: A high performance dsp architecture for software defined radio,” *IEEE Micro* 27, no. 1 (Jan.-Feb. 2007): 114-123.

[7] H. Uchikawa, K. Umebayshi, and R. Kohn, “Secure download system based on software defined radio composed of FPGAs,” in 13th IEEE Int'l. Symp. on Personal, Indoor, and Mobile Radio Comm. (2002): 437-441.

[8] G. Schelle, J. Fifield, and D. Grimwald, “A software defined radio application utilizing modern FPGAs and noc interconnects,” in Int'l Conf. on Field Programmable Logic and Apps. (2007):. 177-182.

[9] A. DiStefano, G. Fiscelli, and C. G. Giaconia, “An FPGA-based software defined radio platform for the 2.4GHz ISM band,” *Ph. D. Research in Microelectronics and Electronics* (June 2006): 73-76.

[10] G. Girau, A. Tomatis, F. Dovis, and P. Mulassano, “Efficient Software Radio Implementations of GNSS Receivers,” in IEEE Int'l. Symp. on Circuits and Systems (2007): 1733-1736.

[11] G. Atia, S. Aeron, E. Ermis, and V. Saligrama, “On throughput maximization and interference avoidance in cognitive radio,” in 5th IEEE Consumer Comm. and Networking Conf. (2008): 963-967.

[12] X. Zhou, X. Zhao, and P. Cheng, “An interference avoidance schem for ofdm cognitive radio,” in Int'l Conf. on Comm. Technology (2006): 1-5.

[13] F. V. Hooft, “A heterogeneous software defined radio architecture for electronic signal interception, identification, and jamming,” in IEEE Military Communications Conf., (2003): 1178-1183.

[14] C. Bergstrom, S. Chuprun, S. Gifford, G. Maalouli, “Software defined radio (SDR) special military applications,” *Proceedings MILCOM 02',* vol. 1 (Oct. 2002): 383-388.

[15] F. Ge, Q. Chen, Y. Wang, C. W. Bostian, T. W. Rondeau, and B. Le, “Cognitive radio: from spectrum sharing to adaptive learning and reconfiguration,” in IEEE Aerospace Conf. (2008): 1-10.

[16] S. Geirhofer, L. Tong, and B. M. Sadler, “Dynamic spectrum access in the time domain: modeling and exploiting white space,” *IEEE Communications Magazine* 45, no. 5 (2007): 66-72.

[17] Z. Ji, and K. J. R. Liu, “Dynamic spectrum sharing: a game theoretical overview,” *IEEE Communications Magazine* 45, no. 5 (2007): 88-94.

[18] Q. Zhao, and B. M. Sadler, “A survey of dynamic spectrum access,” *IEEE Signal Processing Magazine* 24, no. 3 (May 2007): 79-89.

[19] Lyrtech©, "Small form factor sdr evaluation module/ development platform user's guide," 2007.

[20] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *Institute of Engineering and Technology (IET) Communications* 1, no. 2 (April 2007): 137-156.

[21] J. Chen, Y. Kuo, F. Fu, C. Li, and J. Li, "Digital modulation identification based on software radio platform," in 6th Int'l Conf. on Parallel and Distributed Computing (2005): 945-949.

[22] D. Asano, and M. Ohara, "Automatic modulation identification using a frequency discriminator," *IEICE Transactions on Communication,* vol. E91-B, no. 2 (February 2008): 575-578.

[23] D. Boiteau, and C. L. Martret, "A general maximum likelihood framework for modulation classification," in IEEE Int'l Conf on Acoustics, Speech, and Signal Processing (1998): 2165-2168.

[24] P. C. Sapiano, and J. D. Martin, "Maximum likelihood PSK classifier," in IEEE Military Comm. Conf. (1996): 1010-1014.

[25] J. A. Sills, "Maximum-likelihood modulation classification for PSK/QAM," in IEEE Military Comm. Conf. (1999): 217-220.

[26] H. Chung-Yu, and A. Polydoros, "Likelihood methods for MPSK modulation classification," *IEEE Trans. on Communications*, (Feb.-Apr. 1995): 1493-1504.

[27] K. S. Shanmugan, and A. M. Breipohl, *Random signals: detection, estimation, and data analysis*, (Jon Wiley and Sons, Inc., 1988): 488-490.

[28] J. Lopatka, and R. Bukowski, "A use of instantaneous frequency estimators for radio signals identification," in IEEE AFRICON (1999): 1139-1142.

[29] I. Rashid, H. Maqbool, Mehmood-ur-Rehman, and F. Nadir, "Digital modulation identification by basic modulation parameters," in 9th International Multitopic Conference IEEE INMIC (2005): 1-6.

[30] D. Grimaldi, S. Rapuano, and L. D. Vito, “An automatic digital modulation classifier for measurement on telecommunication networks,” *IEEE Trans. on Instrumentation and Measurement* 56, no. 5 (Oct. 2007): 1711-1720.

[31] H. Ishii, T. Suzuki, H. Hosoya, and T. Kamisawa, “Automatic modulation identification for non-linear digital modulation (based on software radio techniques),” in 14th IEEE Int'l. Symp. on Personal, Indoor, and Mobile Radio Communication Proceedings, (2003): 1237-1241.

[32] L. D. Vito, S. Rapuano, and M. Villanacci, “An improved method for the automatic digital modulation classification,” in IEEE Instrumentation and Measurement Technology Conf. Proceedings (2008): 1441-1446.

[33] P. Prakasam, and M. Madheswaran', “Automatic modulation identification of QPSK and GMSK using wavelet transform for adaptive demodulator in SDR,” in Int'l Conf. on Sig.Proc., Communication, and Networking (2007): 507-511.

[34] K. C. Ho, W. Prokopiw, and Y. T. Chan, “Modulation identification of digital signals by wavelet transform,” *IEEE Proc.-Radar, Sonar Navig.* 147, no. 4(Aug. 200): 169-176.

[35] K. C. Ho, W. Prokopiw, and Y. T. Chan, “Modulation identification by the wavelet transform,” in IEEE Military Comm. Cont. (1995): 886-890.

[36] K. C. Ho, H. Liu, and L. Hong, “On improving the accuracy of a wavelet based identifier to classify CDMA signal and GSM signal,” in Proc. of IEEE Int'l.l Symp. on Circuits and Systems (ISCAS 1999): 564-567.

[37] H. Liu, and K. C. Ho, “Identification of CDMA signal and GSM signal using the wavelet transform,” in 42nd Midwest Symposium on Circuits and Systems (1999): 678-681.

[38] G.-R. Kwon, J.-S. Lee, J.-D. Jin, and S.-J. Ko, “Noise-robust modulation identification method for adaptive receiver based on software defined radio,” *IEEE Trans. on Consumer Electronics* 53, no. 3 (August 2007): 1211-1216.

[39] A. Swami, and B. M. Sadler, “Hierarchical digital modulation classification using cumulants,” *IEEE Trans. on Communication* 48, no. 3 (March 2000): 416-429.

[40] L. Liu, and J. Xu, “A novel modulation classification method based on high order cumulants,” in Int'l Conf. on Wireless Comm., Networking, and Mobile Computing (WiCOM 2006): 1-5.

[41] H.-C. Wu, M. Saquib, and Y. Zhifeng, “Novel automatic modulation classification using cumulant features for communications via multiplath channels,” *IEEE Trans. on Wireless Communications* 7, no. 8 (August 2008): 3098-3105.

[42] C. M. Spooner, “On the utility of sixth order cyclic cumulants for rf signal classification,” in 35th Asilomar Conf. on Signals, Systems, and Computers (2001): 890-897.

[43] O. A. Dobre, “Higher-order cyclic cumulants for high order modulation classification,” in IEEE Military Comm. Conf. (2003): 112-117.

[44] T. T. Wang, “The segmented chirp z-transform and its applications in spectrum analysis,” *IEEE Trans. on Instrumentation and Measurement* 39, no. 2 (April 1990): 318-323.

[45] I. Sarkar, and A. T. Fam, “The interlaced chirp z-transform,” in Int'l Conf on Sig. Processing and Comm. (2004): 46-50.

[46] C. Li, X. Zhang, H. Li, Q. Zhang, and Z. Tan, “Carrier frequency estimation of unconventional BPSK signal based on the DFT,” in 8th Int'l Conf on Sig. Proc. (2006).

[47] J. A. Sills, and Q. R. Black, "Frequency estimation from short pulses of sinusoidal signals," MILCOM 96' Conf. Proceedings 3 (Oct. 1996): 979-983.

[48] M. P. Fitz, “Planar filtered techniques for burst mode carrier synchronization,” in Proc. IEEE GLOBECOM '91 (Dec. 1991): 365-369.

[49] M. Luise, and R. Reggiannini, “Carrier frequency recovery in all-digital modems for burst transmissions,” *IEEE Trans. on Communications* 43, no. 3 (Mar. 1995): 1169-1178.

[50] B. Dongming, Z. Gengxin, and Y. Xinying, “A maximum likelihood based carrier frequency estimation algorithm,” in 5th Int'l Conf. on Signal Processing (ICSP 2000): 185-188.

[51] L. Wu, L. An, and B. Liu, “An iterative ML-based carrier frequency estimation algorithm,” in Int'l Conf. on Comm. Tech (ICCT 2006): 1-3.

[52] U. Mengali, and M. Morelli, “Data-aided frequency estimation for burst digital transmission,” *Trans. on Communications* 45, no. 1 (1997): 23-25.

[53] Y. Jin, and H. Ji, "Cyclic autocorrelation based blind parameter estimation of PSK signals," in Int'l Conf. on ITS Telecommunications Proceedings (June 2006): 1293-1296.

[54] M. Cabrera, and M. A. Lagunas, “Eigen based methods to jointly estimate frequency and timing in PSK and MSK signals,” in ICASSP (1992): 413-416.

[55] G. Li, “A stable efficient adaptive notch filter for direct frequency estimation,” *IEEE Trans. on Signal Processing* 45, no. 8 (Aug. 1997): 2001-2009.

[56] M. Mojiri, and A. R. Bakshai, “An adaptive notch filter for frequency estimation of a periodic signal,” *IEEE Trans. on Automatic Control* 49, no. 2 (Feb. 2004): 314-318.

[57] M. Ta, and V. Debrunner, “Adaptive notch filter with time-frequency tracking of continuously changing frequencies,” in IEEE Int'l. Conf. on Acoustics, Speech and Sig. Processing (2008): 3557-3560.

[58] H. C. Ho, and P. C. Ching, “Adaptive algorithm for direct frequency estimation,” *IEEE Proc. on Radar, Sonar, and Navigation* 151, no. 6 (Dec. 2004): 359-364.

[59] Y. Tachwali, W. J. Barnes, and H. Refai, “Configurable symbol synchronizers for SDR applications,” *Journal of Networking and Computer Applications* 32, no. 3 (May 2009): 607-615.

[60] W. J. Barnes, and Y. Tachwali, “A configurable symbol synchronizer for digital systems,” in IEEE Global Comm. Conf. (Dec. 2008): 1-5.

[61] L. Mazet, and P. Loubaton, “Cyclic correlation based symbol rate estimation,” in Asilomar Conf. on Signals, Systems, and Computers (1999): 1008-1012.

[62] W. Yin, and K. Wang, “A New Method to Symbol Rate Estimation of MPSK Signals,” in Congress on Image and Sig. Processing (CISP 2008): 394-398.

[63] Y. T. Chan, J. W. Plews, and K. C. Ho, “Symbol rate estimation by the wavelet transform,” in Int'l. Symp. on Circuits and Systems (1997): 177-180.

[64] X. Jun, W. Fu-ping, and W. Zan-ji, “The improvement of symbol rate estimation by wavelet transform,” in Int'l Conf. on Comm., Circuits, and Systems (2005): 100-103.

[65] T. Ghirmai, “Sequential Monte Carlo method for fixed symbol timing estimation and data detection,” in Conf. on Information Sciences and Systems (2006): 1291-1295.

[66] T. Ghirmai, M. F. Bugallo, J. Miguez, and P. M. Djuric, “Joint symbol detection and timing estimation using particle filtering,” in Int'l Conf. on Acoustics, Speech, and Sig. Processing (ICASSP 2003: 2003, : IV -596 to IV-599.

[67] M. Flohberger, W. Kogler, W. Gappmair, and O. Koudelka, “Symbol rate estimation with inverse fourier transforms,” in Int'l Workshop on Satellite and Space Communication (2006): 110-113.

[68] H. Xu, Y. Zhou, and Z. Huang, “Blind roll-off factor and symbol rate estimation using IFFT and least squares estimator,” in Int'l Conf. on Wireless Comm., Networking and Mobile Computing (2007): 1052-1055.

[69] TexasInstruments. "CC2590 Datasheet," http://focus.ti.com/lit/ds/symlink/cc2590.pdf. (Nov. 14, 2008)

[70] RFLinx. "Antennafier 915 UD Series 2.4GHz to 915 MHz Converter Datasheet," http://www.rflinx.com/pdf/ds/cat/133.pdf. (Nov. 15, 2008)

[71] A. Jakobschuk, "Amplifier noise figure optimization," *Proc. of the IEEE* 56, no. 9 (Sept. 1968): 1631-1632.

[72] N. Garmendia, and J. Portilla, "Study of pm noise and noise figure in low noise amplifiers working under small and large signal conditions," in Int'l Microwave Symp. (2007): 2095-2098.

[73] P. B. Basyurt, and N. Tarim, "An x-band SiGe low-noise amplifier with high gain and low noise figure," in 3rd In't Symposium on Comm., Control, and Sig. Proc., (2008): 1103-1106.

[74] Maxim-IC. "Application note 2875: three methods of noise figure measurement "; www.maxim-ic.com/appnotes.cfm/appnote_number/2875 (Nov. 15, 2008).

[75] A. Geens, and Y. Rolain, "Noise figure measurements on nonlinear devices," *IEEE Trans. on Instrumentation and Measurement* 50, no. 4 (2001): 971-975.

[76] C.-S. Cheng, S.-W. Lin, C.-C. Wei *et al.*, "A high isolation 0.15um depletion-mode pHEMT SPDT switch using field-plate technology," in Asia-Pacific Microwave Conf. (2007): 1-4.

[77] P. Sun, L. Wang, P. Upadhyaya *et al.*, "High isolation 10GHz to 20 GHz SPDT switch design using novel octagonal pin diode structure," in IEEE Workshop on Microelec. and Electron Dev. (2005): 38-41.

[78] "The Delay Locked Loop." New Wave Instruments Resouces, James A. Vincent, 1993 Web http://www.newwaveinstruments.com/resources/reprints/advanced_topics/pn_code_tracking/the_delay_locked_loop/dll.html. (Nov. 11, 2009)

[79] Maxim-IC, “2.4GHz 802.11b Zero-IF Transceivers”, Datasheet Rev. 5, May 5 2005. Web. http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821A.pdf

[80] Vocal Technologies Ltd., “802.11b White Paper”. Website http://www.vocal.com/white_paper/802.11b_wp1.doc (Nov. 11, 2009).

[81] S. Blionas, K. Masselos, C. Dre, F. Ieromnimon, T. Pagonis, A. Pneymatikakis, A. Tatsaki, T. Trimis, A. Vontzalidis and D. Metafas "DESIGN Story: A Hiperlan2/IEEE802.11x reconfigurable SoC for." *Workshop* INTRACOM S.A., Athens, Greece 2002.

[82] Gast, Matthew. "When Is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput - O'Reilly Media." Technology Books, Tech Conferences, IT Courses, News - O'Reilly Media. O Reilly Media, 8 Aug. 2003. Web. http://www.oreillynet.com/pub/a/wireless/2003/08/08/wireless_throughput.html. (Nov. 11, 2009).

[83] FCC, “Advanced wireless services band plan,” Data sheet. Web. wireless.fcc.gov/services/aws/data/awsbandplan.pdf. (Nov. 11, 2009)