Optimizing Communications Systems: Finding Gains at the Least Marginal Cost

---

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

John C. Peng

December

2015

APPROVAL SHEET

The dissertation

is submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

AUTHOR

The dissertation has been read and approved by the examining committee:

Stephen G. Wilson

Advisor

Maite Brandt-Pearce

Bobby Weikle

John Lach

Stephen Patek

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, Dean, School of Engineering and Applied Science

December

2015

Acknowledgments

I would like to than my adviser Professor Stephen Wilson for his support and mentoring these past few years. Along with him, I would like to thank Professor Maïté Brandt-Pearce and Professor Toby Berger, to all of whom I owe my foundation in communications and information theory.

I would also like to thank Professor Bobby Wiekle, Professor John Lach, and Professor Stephen Patek for serving on my defense committee. I would also like to thank Professor Kamin Whitehouse and Professor Malathi Veeraraghavan for their help regarding the networking aspects of my dissertation.

I would like to thank my wife Cynthia for her support, without which this PhD would have been impossible. I am also indebted to my parents who have always supported me. I would also like to thank my colleagues and friends for their help and support.

I am also grateful for the financial support of the National Science Foundation.

Abstract

The analytical expression for the ultimate limit of communications efficiency has been known since Shannon's *A Mathematical Theory of Communication*, published in 1948. Since then, the digital revolution has provided the signal processing complexity needed to close the gap between this limit and practical communication systems. This dissertation on optimizing communications systems investigates the tradeoff between communications efficiency and signal processing complexity in light of the current state of art techniques in communications theory. It begins with techniques of calculating information rates for non-linear satellite channels. Information rate calculations give the absolute limit in communications efficiency and are used as a reference point to the optimization problem. The second part of this dissertation uses information rate calculations to examine the complexity versus efficiency tradeoff of a physical layer approach known as faster-than-Nyquist (FTN) signalling, which has been a popular area of research recently. Our results show FTN does not show significant advantages over a much simpler OFDM technique, which is already widely implemented in modern communications systems. These results inspired the question of *what method of optimizing communications systems could be obtained at the least marginal cost in complexity.* While techniques such as MIMO [9] still holds promise for future improvement at the physical layer, *application specific optimization may give the greatest performance gain at the least cost in signal processing complexity.* The final part of this thesis examines this statement and hopes to provide evidence of this conjecture.

Contents

List of Figures

List of Tables

Optimizing Communications Systems: Finding Gains at the Least Marginal Cost

John Peng*

December 15, 2015

*john.peng@email.virginia.edu

Chapter 1

Introduction

1.1 First Came the Theory

The research and development of communications and information theory bears a striking resemblance to a related area of research where the inception of an elegant theory overshadows its validation and application years later. James Clerk Maxwell predicted the existence of electromagnetic radiation along with his eponymous equations in 1863. Twenty-five years later, and 9 years after Maxwell's death, the existence of electromagnetic radiation was empirically verified by Henrich Hertz. In 1895, the year after Hertz's death, the application of electromagnetic waves in wireless communications was pioneered by Guglielmo Marconi [58].

Similarly, Claude Shannon in 1948 laid forth the foundation of communications and information theory in *A Mathematical Theory of Communication* [43]. In his treatise, the maximum bit rate for reliable communication over a given communications channel was analytically formulated. These rates allowed arbitrary complexity in the receiver and transmitter, and therefore practical systems can only approach this limit. As Maxwell's equations required the practical engineering of Hertz and Marconi to be verified and applied, the pursuit of Shannon's limit of reliable communication needed the digital revolution to provide the necessary signal processing power.

1.2 Optimization and Tradeoff

Figure 1.1 shows the fundamental tradeoffs involved in optimizing communications systems. The curve represents the maximum communications efficiency in an additive white Gaussian noise channel where only the points below the curve are achievable [38]. The ratio of energy per bit to noise power spectral density E_b/N_0 represents power efficiency, where lower values are more efficient. Spectral efficiency is represented in units of bits/Hz where higher values are deemed more efficient. Thus there is a fundamental tradeoff between power efficiency and spectral efficiency. Increasing complexity is required to approach the efficiency limits implied by the curve.

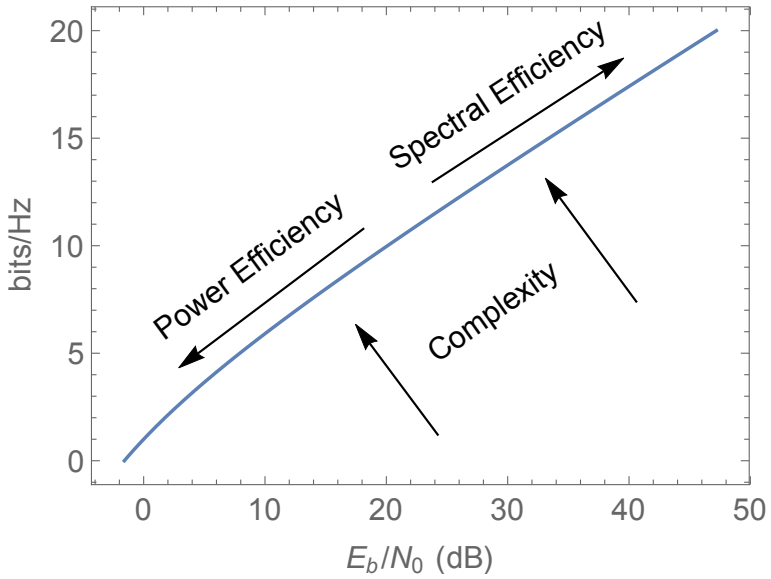

Figure 1.1: Tradeoffs in Optimizing Communications Systems

The integrated circuit was invented in 1958, and since then Moore's Law[1] has accurately predicted an exponential growth in computing power. As computing power has become cheaper, more elaborate and complex signal processing schemes were developed to push performance closer to the Shannon limit. Modern error correcting codes have been able to approach

[1]Though not a scientific law, Gorden Moore's prediction of exponential growth in affordable computational power has proved accurate over the last half century

within 1 dB of the Shannon limit on Gaussian noise channels. This monumental achievement raises the question of how to continue trading off complexity for communications efficiency, as the fundamental limit bars any further significant increase in efficiency

MIMO[2] technology is one of the answers. MIMO allows multiple channels to be created within the same bandwidth. This comes with an increase in the number of transmitters and receivers in a system as well as signal processing to distinguish the symbols that appear across the multiple channels. MIMO has been applied to many different communications channels such as wireless, optical, and DSL. Massive MIMO over millimeter wave frequencies has been proposed as the basis for 5G wireless.

While MIMO continues to hold promise of increasing communications efficiency, this thesis attempts to explore another area which may have been overlooked due the entrenched standards based on the layered communications stack. As will be explained, there is a confluence of technology and trends that makes this optimization particularly viable in the near future. We call this *application specific optimization* (ASO).

1.3 Organization and Contributions

This thesis is organized into three sections:

Computing Information Rates of Nonlinear Satellite Channels

Calculating information rates give a metric by which the efficiency of a practical communications system can be measured, and gives the margin of how much a communications system can be improved. The curve in Figure 1.1 holds only for the AWGN channel with Gaussian alphabet. Information rate calculations are more challenging for other more complex channels.

In this section we calculate information rates of reduced complexity receivers for the the non-linear satellite channel. We calculate rates at different memory depths, with and without the "matched" filter, and for one-way and bidirectional same frequency transmission. We also investigate designing reduced complexity receivers for the nonlinear satellite channel by marginalizing the channel law. (The work from this section were presented in *LATINCOM2013*, *GLOBECOM2013*, and *ICC2014*)

[2]Multiple Input / Multiple Output

Faster-than-Nyquist, and Comparison with OFDM

Here we compare the efficiency versus complexity tradeoff of *Faster-than-Nyquist* (FTN) signalling and *orthogonal frequency division multiplexing* (OFDM). To do so we apply these modulation techniques to the *Wifi G* and *Wifi B* spectral masks and calculate their respective information rates. While FTN has classically been applied to root-Nyquist pulses, we derive a pulse from the spectral mask and show that FTN techniques can still be applied. Although a worthwhile and interesting area of research, our analysis shows that any marginal improvement over another spectrally efficient modulation technique known as *orthogonal frequency division multiplexing* (OFDM) does not merit its implementational complexity. (The work form this section was presented in *LATINCOM2015*)

Application Specific Optimization

Our study of FTN suggests that further improvement in single channel communications efficiency may provide only marginal benefits at high complexity costs. This thought has inspired our research in ASO as a technique that seeks to unlock optimization gains which have up to now been shackled by entrenched layered protocol standards. We propose an approach to ASO that maintains compatibility with the Internet while providing performance gains at low marginal complexity costs as well as providing other benefits inherent in cloud computing. This novel architecture takes advantage of current trends in data center technology and network function virtualization. (The work form this section has been submitted to *NETSOFT2016* and *IEEE Transactions in Communications*)

Part I

Computing Information Rates of Nonlinear Satellite Channels

Chapter 2

Some Information Theory Background

This chapter presents the intuition behind information theory as a background to this dissertation. All three parts of this thesis rely on information rate calculations.

2.1 Entropy

Shannon formulated the foundation of information theory by appropriating the idea of entropy from statistical mechanics as a measure of information. Information messages can be represented as random variables, and a measure of information is the entropy of the random variable:

$$H(X) \triangleq E_X[p(x) \log_2 \frac{1}{p(x)}] \tag{2.1}$$

where X is a random variable representing information, $E_X[\cdot]$ is the expectation with respect to X, and p(x) is the probability distribution of X.

It is important to differentiate between Shannon's definition of information and our common notion of information. Here the definition of information is in the context of information that needs to be transmitted since it is unknown at the receiver. Thus the receiver only knows the possible values of the information as represented by a random variable and its distribution.

Notice that entropy is zero if the random variable is deterministic[1]. This is to say that if the receiver already knows what X will be, there is no transfer of information nor any need for communication. So if a copy of the *Encyclopedia Britannica* were present at both the transmitter and receiver, in terms of information theory there is no amount of information with respect to this fact since nothing needs to be communicated.

Each possible value that X may take on is known as a *symbol*, and the set of all possible symbols is known as the *alphabet*. Figure 2.1 shows the APSK-16 constellation as a practical example of an alphabet and its constituent symbols. APSK stands for amplitude and phase shift keying and is a modulation used in satellite communications under the DVB-S2 standard. There are 16 symbols in this alphabet where each symbol is a point on the complex plane. These complex values represent the amplitude and phase of a carrier for a bandpass waveform, much like a phasor in linear circuit analysis. In practice the distribution $p(X)$ is typically uniform over all the symbols in the constellation [2].

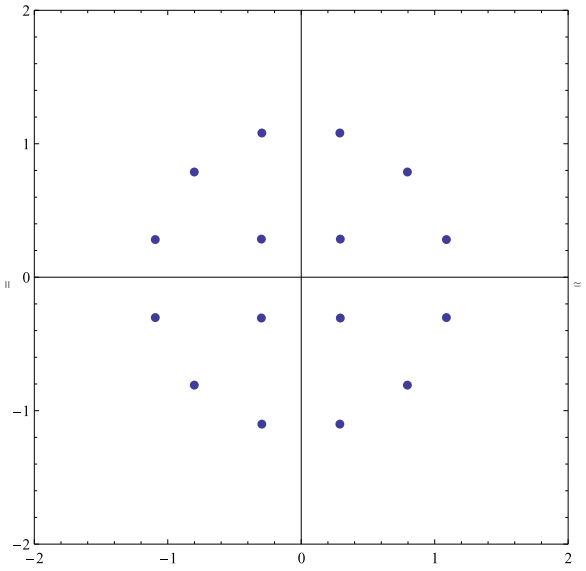

Figure 2.1: APSK-16 Constellation

[1] That is $p(x_1) = 1$ for some symbol x_1

[2] However this incurs a slight performance penalty with respect to the optimum performance theoretically attainable over a Gaussian noise channel

When the logarithm in (2.1) is base 2, the units of entropy is in bits. Thus $H(X)$ represents the average number of bits communicated when a symbol X is received. In other words, if the vector $\boldsymbol{x}_1^N \triangleq \{x_1, x_2, ...x_N\}$ is received with each symbol drawn independently from the distribution $p(x)$ the expected amount of information conveyed would be $NH(X)$ bits.

$H(X)$ also represents the minimum average number of bits needed to represent the random variable X. This is also to say that on average $NH(X)$ bits would be needed to represent a vector $\boldsymbol{x}_1^N$ of length N if each X_n in the vector were independent. An application of this would be to save information generated by X to some digital media. The most efficient mapping from vectors $\boldsymbol{x}_1^N$ to bits would use on the average $NH(X)$ bits.

Instead of using bits to represent symbol vectors $\boldsymbol{x}_1^N$, we could instead have a binary information source which are then encoded onto vectors $\boldsymbol{x}_1^N$ which are then transmitted over a communications channel. Just as there is an efficient mapping from vectors of X to bits, there is also an efficient mapping of bits to vectors of X such that the distribution of X is still $p(x)$ and the average number of bits encoded per symbol is $H(X)$. In the case of APSK-16, a binary bit source could be mapped four bits at a time to a constellation point symbol. As long as the bits are independent and identically distributed (IID), then the symbols are also IID. As most communications systems today are based on the binary logic of VLSI technology, this mapping is ubiquitous in practical implementations.

2.2 The Discrete-time Memoryless Channel

Thus X, or equivalently a bit source used to generate X, could represent an information source. And the average amount of information generated per realization of X is $H(X)$. However a communications channel typically does not faithfully represent each realization of X with perfect fidelity. Instead the channel can corrupt X according to the conditional distribution $p(y|x)$, which is known as the *channel law*. Now the receiver gets a random variable Y and now has to do its best to infer what the original X was. Figure 2.2 shows a discrete-time *memoryless* channel where each channel use is described by $p(y|x)$ is independent of the discrete time index n. X represents the transmitter's input into the channel, and Y is the output of the channel as seen by the receiver.

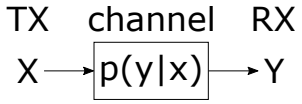

Figure 2.2: A Discrete-Time Memoryless Channel

For a memoryless channel

$$p(\boldsymbol{y}_1^N|\boldsymbol{x}_1^N) = \prod_{n=1}^{N} p(y_n|x_n)$$

An example of a discrete-time memoryless channel is the *additive white Gaussian noise* (AWGN) channel shown in Figure 2.3. $\mathcal{N}(0, \sigma^2)$ is normally distributed noise with variance σ^2. Since the noise is both white and Gaussian, the noise in each use of the channel is independent of each other. We also assume independence of the input symbols X between channel uses as well as with the noise. Thus the channel law can be expressed as

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2} \tag{2.2}$$

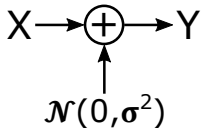

Figure 2.3: Discrete-Time AWGN Memoryless Channel

The output distribution $p(y)$ can be calculated from the input distribution $p(x)$ and the channel law by marginalizing the joint distribution [3]. In the case of the APSK-16 modulation in an AWGN channel:

$$\begin{aligned} p(y) &= \sum_x p(y|x)p(x) \\ &= \sum_{x=1}^{16} \frac{1}{16} \frac{1}{2\pi\sigma^2} e^{\frac{1}{2}\left(\frac{|y-x|}{\sigma}\right)^2} \end{aligned} \tag{2.3}$$

[3] While the input distribution $p(x)$ is discrete over the symbols in the alphabet, $p(y|x)$ and $p(y)$ are continuous distributions due to the AWGN

where $p(y|x)$ in this case is the circularly symmetric complex normal distribution with mean x and variance σ^2 in each dimension. While the channel in 2.2 represents a baseband channel where X and Y are real, in 2.3 the channel is complex representing the magnitude and phase of a carrier in a bandpass signal [4]. Plots of $p(y)$ for 20 dB SNR and 10 dB SNR are shown in Figures 2.4 and 2.5 respectively.

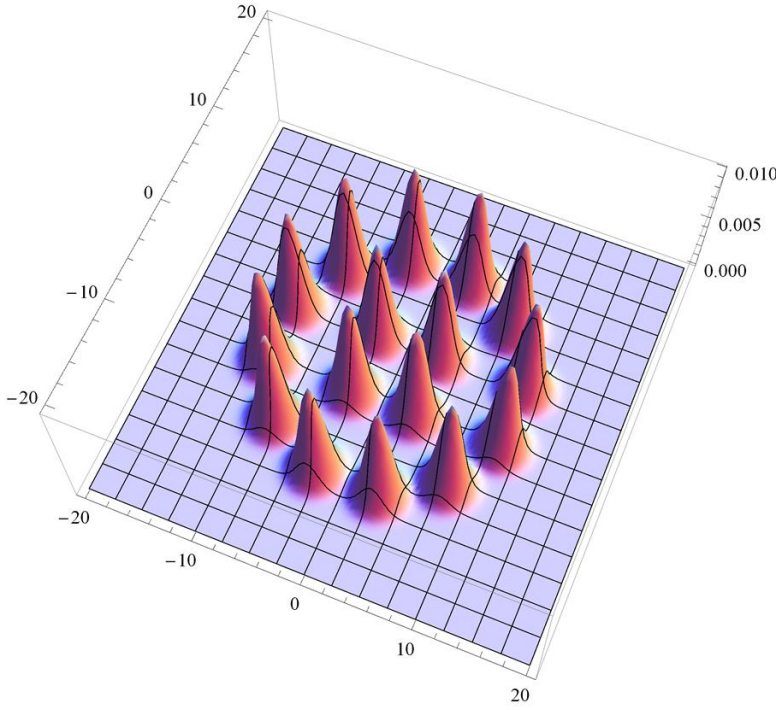

Figure 2.4: $p(y)$ for APSK-16 over an AWGN Channel at 20 dB SNR

2.3 Mutual Information

Shannon proved that given an input distribution $p(x)$, the maximum number of bits on average that a discrete-time memoryless channel could transfer per channel use was a difference in entropies. This difference is defined as

[4]The signal to noise ratio (SNR) in 2.2 is $E[X^2]/\sigma^2$, while in 2.3 it is $E[|X|^2]/2\sigma^2$

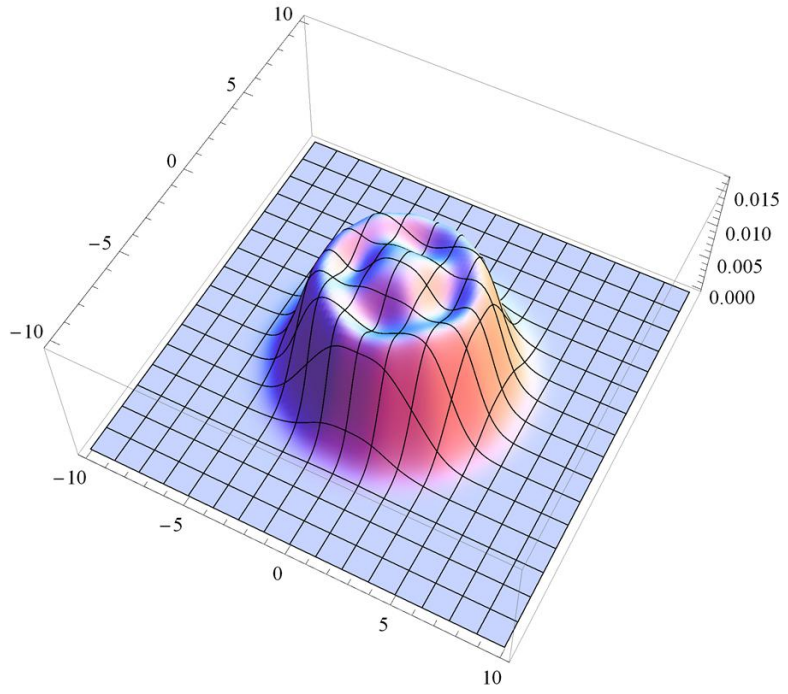

Figure 2.5: $p(y)$ for APSK-16 over an AWGN Channel at 10 dB SNR

the *mutual information*:

$$I(X;Y) \triangleq H(X) - H(X|Y) = E_x[\log_2 \frac{1}{p(x)}] + E_{x,y}[\log_2 \frac{1}{p(x|y)}] \quad (2.4)$$

$$= H(Y) - H(Y|X) = E_y[\log_2 \frac{1}{p(y)}] + E_{x,y}[\log_2 \frac{1}{p(y|x)}] \quad (2.5)$$

where $H(X|Y)$ is the conditional entropy of X given Y.

The conditional entropy $H(X|Y)$ can be interpreted as the amount of information needed to gain perfect knowledge of X given the knowledge of Y. This is the situation at the receiver where it has received Y but is missing the amount of information signified by $H(X|Y)$.

The mutual information $I(X;Y)$ represents the maximum average amount of information transferred per use of the communications channel given $p(x)$. It can be calculated by taking the measure of information inherent in the source, $H(X)$, minus the information that the receiver does not know, $H(X|Y)$ as in 2.4.

The mutual information can also be calculated as in 2.5. This difference can be interpreted as the amount of information inherent in the received signal $H(Y)$ minus the channel noise entropy $H(Y|X)$. This also appeals to our intuition as the amount of information in Y about X is the total amount of information in Y minus the "information" in Y that has nothing to do with X, $H(Y|X)$.

This suggests that the mutual information $I(X;Y)$ is the maximum amount of information about X that can be inferred from Y, and vice versa. And Shannon showed that this also represents the absolute limit of reliable communications – that is communications where the error rate can be driven to zero – when given an input distribution $p(x)$ and a channel law $p(y|x)$. This is done by encoding messages or bits not directly onto each symbol x_n, but onto a vector $\boldsymbol{x}_1^N$. As $N \to \infty$, the maximum number of messages that can be distinguished at the receiver after being corrupted by the channel is $2^{NI(X;.Y)}$. This results in an information rate of $I(X;Y)$ bits per channel use [5].

Figure 2.6 shows an example of the entropies related to the discrete-time memoryless channel. We would expect $H(Y)$ to be greater than $H(X)$ since Y includes the randomness of the channel. However the gap representing $I(X;Y)$ is the same as seen in (2.4).

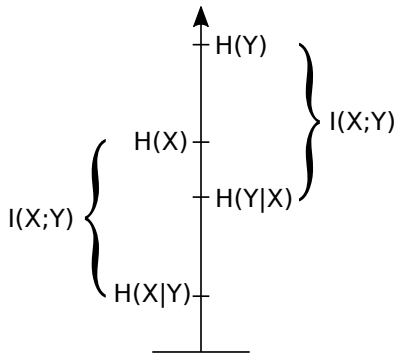

Figure 2.6: Mutual Information Illustration

[5] $log_2(2^{NI(X;Y)})/N = I(X;Y)$

2.4 Channel Capacity

In a typical engineering scenario the channel $p(y|x)$ is given while the channel input X and its distribution $p(x)$ are design variables. Shannon defined the channel capacity as the maximization of the mutual information over all input distributions:

$$C = \max_{p(x)} I(X;Y) \tag{2.6}$$

For the AWGN channel, Shannon showed that the input distribution $p(x)$ that maximizes the mutual information and thus gives the capacity is simply a Gaussian distribution $X \sim \mathcal{N}(0, \sigma_s^2)$. Then the capacity for a real baseband signal is

$$C = \frac{1}{2}\log_2(1 + \frac{\sigma_s^2}{\sigma_n^2}) \tag{2.7}$$

The channel capacity represents the maximum achievable bit rate for reliable communications over a given channel.

2.5 Practical Waveform Channels

Practical bandlimited AWGN waveform channels can be analyzed as discrete-time memoryless AWGN channels through a signal space transformation using *root-Nyquist pulse shaping* and *matched filtering*.

A *Nyquist pulse* $p_{ny}(t)$ such as the sinc and raised cosine pulse have zero crossings at an offset T_0 and symbol intervals T, which occur at a rate known as the Nyquist rate:

$$p_{ny}(T_0 + kT) = \begin{cases} 1 & k = 0 \\ 0 & otherwise \end{cases}$$

where k is an integer.

This feature allows symbols $x[k]$ to be extracted from a pulse amplitude modulated signal $x_{ny}(t)$ without intersymbol interference (ISI) when sampled with the correct offset at the Nyquist rate:

$$x_{ny}(t) = \sum_n x_n p_{ny}(t - nT)$$
$$x_n = x_{ny}(T_0 + kT)$$

A root-Nyquist pulse is derived from a Nyquist pulse by spectral factorization. If $P_{ny}(f)$ is the Fourier transform of $p_{ny}(t)$, then the root Nyquist pulse is defined as $P_{rny}(f) = \sqrt{P_{ny}(f)}$. The root-Nyquist pulse has the property of being time orthogonal with itself when spaced at intervals of T:

$$\int_{-\infty}^{\infty} p_{rny}(t)p_{rny}(t+nT)dt = \begin{cases} 1 & n = 0 \\ 0 & otherwise \end{cases}$$

The root-Nyquist pulses spaced at intervals of T can be considered an orthonormal basis $\{\phi_n(t)\}$. Pulse shaping then is a mapping of a discrete time symbol sequence $\boldsymbol{x}_1^N$ onto a continuous time signal space spanned by this orthonormal basis:

$$x_{rny}(t) = \boldsymbol{x}_1^N \odot \boldsymbol{\phi}_1^N(t)$$

where $\odot$ denotes the inner product defined as

$$\begin{aligned} &= \sum_n x_n \phi_n(t) \\ &= \sum_n x_n p_{rny}(t - nT) \end{aligned} \tag{2.8}$$

Then the discrete symbols x_n can be recovered without ISI by:

$$x_n = \langle x_{rny}(t), \phi_n(t) \rangle \tag{2.9}$$

where $< \cdot >$ is the inner product is defined as

$$= \int_{-\infty}^{\infty} x_{rny}(t)\phi_n(t)dt \tag{2.10}$$

$$= \int_{-\infty}^{\infty} x_{rny}(t)p_{rny}(t - nT)dt \tag{2.11}$$

The inner product in 2.9 can be implemented as a matched filter $m(t)$ where $m(t) = p(T_0 - t)$. If we filter $x_{rny}(t)$ with $m(t)$ and sample at the right moments of $t = T_0 + nT$, then

$$x_{rny}(t) \star m(t) \mid_{t=T_0+nT} = \int_{-\infty}^{\infty} x_{rny}(\tau)m(T_0 + nT - \tau)d\tau \tag{2.12}$$

$$= \int_{-\infty}^{\infty} x_{rny}(\tau)p_{rny}(\tau - nT)d\tau \tag{2.13}$$

$$= \langle x_{rny}(t), \phi_n(t) \rangle \tag{2.14}$$

$$= x_n \tag{2.15}$$

The pulse shaping and matched filtering waveform channel is shown in Figure 2.7.

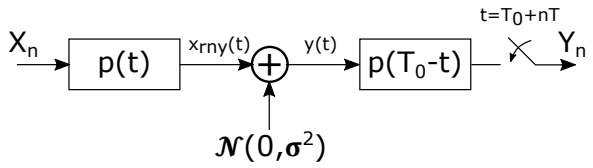

Figure 2.7: Pulse Shaping and Matched Filtering

As a consequence of the Karhunen–Loève theorem, AWGN noise at the output of the matched filter and sampled with recovered signals is independent from sample to sample [38]. Thus the combination of pulse shaping and matched filter reduces the continuous waveform channel to an equivalent vector channel where the mutual information calculations for the discrete-time memoryless channel in Section 2.3 can be applied. Thus the channel in Figure 2.7 is equivalent to the discrete time channel shown in Figure 2.3.

Quadrature amplitude modulation (QAM) is a passband communications technique which takes symbols from a complex alphabet, converts these discrete time symbols to a time domain waveform via pulse shaping, and then modulates a carrier's amplitude and phase according to complex waveform. APSK-16 is an example of QAM. The real and complex components in QAM can be construed as two baseband channels, and the QAM capacity can be calculated from (2.7) as follows:

The capacity per baseband channel is

$$C_{BB} = \frac{1}{2}\log_2\left(1+\frac{\sigma_s^2}{\sigma_n^2}\right) \tag{2.16}$$

Assuming that the pulse shaping function $p(t)$ and matched filter $\phi_k(t)$ have been normalized to unit energy, then the capacity of the QAM channel is

$$C_{QAM} = 2C_{BB} \tag{2.17}$$

$$= \log_2\left(1+\frac{E_s}{N_0}\right) \tag{2.18}$$

where $E_s = \sigma_s^2/2$ since the energy is divided between the two channels and $N_0 = \sigma_n^2/2$ which is the noise variance seen at the output of the matched filter. N_0 is the noise PSD.

The channel capacity can be compared to the mutual information for a practical alphabet and input distribution as a performance metric of the design. Figure 2.8 compares the Shannon capacity as calculated in (2.16) with the mutual information of various practical constellations in bits per channel use. The alphabet size of these constellations are summarized in Table 2.1. The gap between the Shannon bound and the mutual information for the practical constellations represents a performance penalty. This penalty can be kept low as long as the constellation size is sufficient for a given signal to noise ratio E_s/N_0.

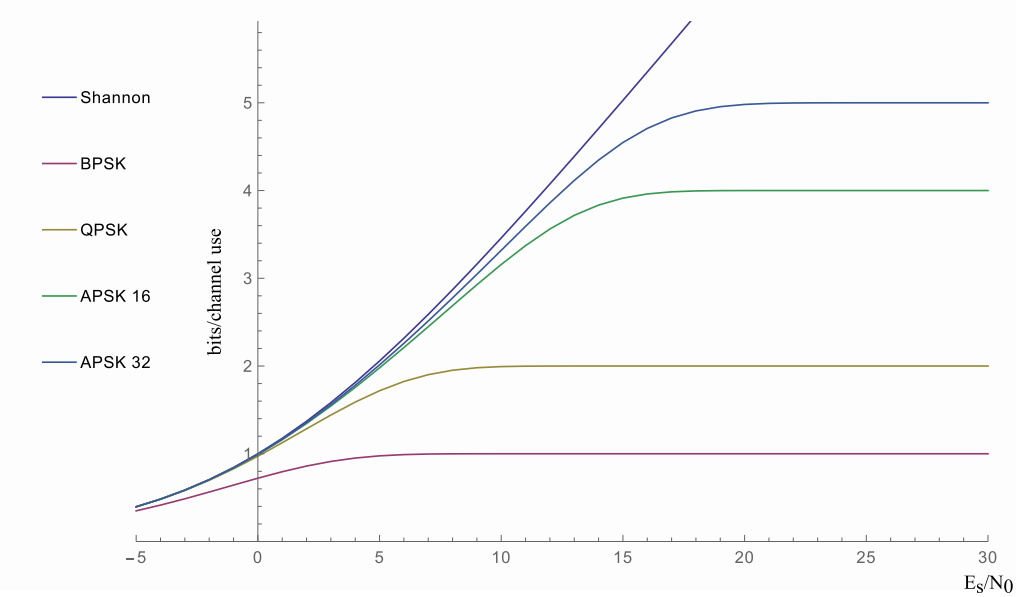

Figure 2.8: Bits Per Channel Use of Capacity vs. Practical Constellation Designs

Thus there is a linear transformation from the waveform signal space to an equivalent discrete-time AWGN vector channel. So the mutual information calculated for the equivalent discrete-time AWGN channel can be applied to the QAM waveform channel simply by multiplying the mutual information with the Nyquist symbol rate, which gives an information rate in bits per second for the channel. This bit rate can be compared to that of a practical system and an important performance measure.

The symbol rate is typically a function of the bandwidth of the channel. Shannon also derived the maximum spectral efficiency of an AWGN

Table 2.1: Constellations in Figure 2.8

Constellation	Symbols in Alphabet
Shannon capacity	Infinite
Binary phase shift keying (BPSK)	2
Quadrature phase shift keying (QPSK)	4
Amplitude phase shift keying (APSK) 16	16
Amplitude phase shift keying (APSK) 32	32

waveform channel. Using sinc pulse shaping, a symbol rate R_s equal to the bandwidth of the channel W is possible. In this case, the capacity in bits/s is:

$$\begin{aligned} C &= R_s \log_2(1 + \frac{E_s R_s}{N_0 W}) && \text{bits/s} \\ &= W \log_2(1 + \text{SNR}) && \text{bits/s} \end{aligned}$$

Dividing C by W gives the maximum attainable spectral efficiency:

$$C/W = \log_2(1 + \text{SNR}) \qquad \text{bits/s/Hz}$$

which is perhaps the most celebrated expression in all of communication theory.

That sinc pulse shaping results in the maximum spectral efficiency is a consequence of the sampling theorem. For QAM signalling, the sinc pulse spaced at the Nyquist rate $R_s = W$ forms a complete orthonormal basis for all time domain signals limited to a bandwidth W. Thus the mutual information of an equivalent DMC channel also applies to the bandpass waveform channel with sinc shaping since the discrete symbol vectors from the DMC can fully span the waveform signal space. The most efficient transmission scheme as implied by the mutual information has an equivalent representation in the waveform signal space via sinc pulse shaping, or ideal reconstruction in terms of the sampling theorem. Thus the mutual information or capacity of a discrete-time AWGN channel implies the maximum reliable bit rate possible in a bandlimited AWGN waveform channel by multiplying the mutual information by the bandwidth. This bit rate corresponds to pulse shaping with the sinc function.

Chapter 3

Information Rates of Non-linear Satellite Channels

The calculation of the mutual information for the memoryless discrete-time AWGN channels and the related information rates of bandlimited AWGN waveform channels are straightforward with modern computation tools. The calculation of the curves for practical constellations in Figure 2.8 involves a two-dimensional numeric integration over the complex plane for the calculation of $h(Y)$[1]. $h(Y|X)$ is the relative entropy of a Gaussian random variable which is known in closed-form [12]. These calculations assume a memoryless source and channel where (2.2) holds. When either the source or channel has memory – that is the output Y_n depends on more than one source symbols, for example X_n, X_{n-1}, and X_{n-2} – the calculation of the information rate can become much more complex. This is certainly the case with calculating the information rates of non-linear satellite channels.

3.1 Memory in Non-linear Satellite Channels

The satellite channel is a relay channel as shown in Figure 3.1[2]. The satellite acts as an intermediary which relays messages between two earth terminals. Typically frequency division duplexing is used where the uplink and

[1]The relative entropy $h(Y)$ pertains to a continuous random variable, but has the same form as $H(X)$ in (2.1) if the probability mass function is replaced with the probability density

[2]Figure adapted from Chenguang Xu

downlink are on different frequency bands. Because of their limited power budgets, satellites operate their amplifiers with higher power efficiency, but at the cost of greater non-linear distortion.

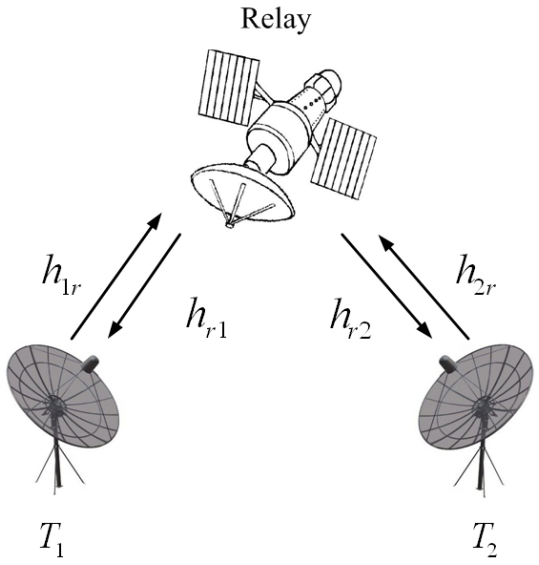

Figure 3.1: Satellite Relay Channel

The non-linear distortion of a traveling wave tube (TWT) amplifier found in many satellites can be modeled using the Saleh model [41]:

$$\begin{aligned} A(r) =& \frac{\alpha_a r}{1+\beta_a r^2} \\ \Phi(r) =& \frac{\alpha_\Phi r^2}{1+\beta_\Phi r^2} \\ g(x) =& A(|x|)e^{j(\angle x + \Phi(|x|)} \end{aligned}$$

where $\alpha_a = 2.15,\ \beta_a = 1.15,\ \alpha_\Phi = 2.16,\ \beta_\Phi = 9.10$

$A(r)$ represents output amplitude as a function of input amplitude and is plotted in Figure 3.2. This represents the amplitude-to-amplitude distortion as the amplitude distortion is only a function of the input amplitude. The peak amplitude output has been normalized to correspond to the unit input amplitude. $\Phi(r)$ represents phase shift as a function of input amplitude, and is also shown in Figure 3.2. This is the amplitude-to-phase distortion as the phase distortion is only a function of the input amplitude.

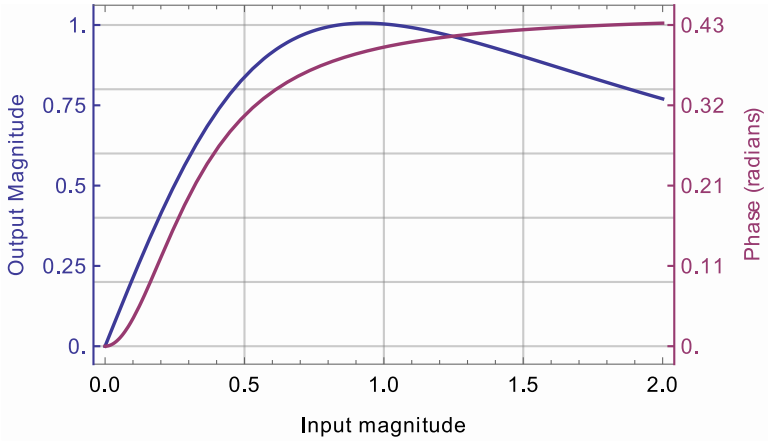

Figure 3.2: Saleh Model Characteristic

Figure 3.3 shows the non-linearity inserted between the pulse shaping and matched filteringin Figure 2.7. While the amplifier distortion according to this model is inherently memoryless, this non-linearity interferes with the operation of the pulse shaping and matched filtering originally designed provide ISI free detection(see Section 2.5). Due to this distortion, the "matched" filter is no longer matched to the transmit waveform. If this "matched" filter is used, in addition to the presence of ISI there will also be a loss of information since the output of the filter no longer provides sufficient statistics.

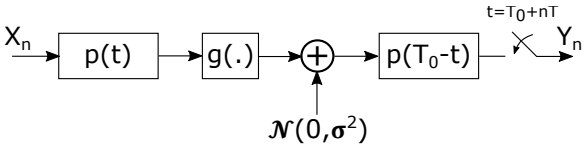

Figure 3.3: Non-linear Satellite Channel with Pulse Shaping and "Matched" Filtering

3.2 Entropy Rate Calculations for Finite State Channels

Information rate calculations of finite state channels with memory are entropy rate calculations:

$$
\begin{aligned}
I(\mathcal{X};\mathcal{Y}) &= h(\mathcal{Y}) - h(\mathcal{Y}|\mathcal{X}) \\
h(\mathcal{Y}) &\triangleq \lim_{N\to\infty} \frac{1}{N} h(Y_1^N) \qquad (3.1) \\
h(Y_1^N) &= -E_{Y_1^N}[\log p(y_1^N)] \\
&= -\int_{y_1}\int_{y_2}\ldots\int_{y_N} p(y_1^N)\log p(y_1^N)dy_1dy_2\ldots dy_N \\
p(y_1^N) &= \sum_{x_1}\sum_{x_2}\ldots\sum_{x_N} p(x_1^N)\prod_{n=1}^{N} p(y_n|x_{n-L}^n) \qquad (3.2)
\end{aligned}
$$

where $y_1^N = (y_1, y_2, \ldots y_N)$, and L is the memory order of the finite state channel.

The dependency of Y_n on X_{n-L}^n can be represented in terms of states. We can define

$$
\begin{aligned}
S_n &\triangleq X_{n-L+1}^n \\
S_{n-1} &\triangleq X_{n-L}^{n-1}
\end{aligned}
$$

With this definition the channel law can be expressed as

$$
p(y_n|x_{n-L}^n) = p(y_n|S_n, S_{n-1}) \qquad (3.3)
$$

The number of states is $|\mathcal{X}|^L$ where $|\mathcal{X}|$ is the alphabet size of the source X.

The complexity of these calculations is exponential in N and a large N is necessary for the convergence in 3.1. Instead of calculating the information rate from first principles, it is possible to estimate it using the Asymptotic Equipartition Property (AEP) [3, 36]

$$
\lim_{N\to\infty} -\frac{1}{N}\log p(y_1^N) \to h(\mathcal{Y}) \qquad (3.4)
$$

y_1^N is a long simulated realization of the random process Y_1^N which can be done with linear complexity with respect to N in lieu of the exponential

complexity in 3.2. The left side of 3.4 converges with probability one to the entropy rate.

3.3 Calculating Entropy Rates on a Trellis

If the channel has a finite state description then $p(y_1^n)$ can be calculated on a trellis. A trellis can be used to represents a state machine, with the nodes representing the states and the branches representing state transitions. Figure 3.4 shows two sections of a simple 4-state trellis for S_{n-1} to S_{n+1}. The trellis could be extended to both the left and the right by prepending or appending additional sections. Traditionally the trellis was used to describe convolutional codes, and decoding algorithms such as the Viterbi algorithm and BCJR algorithm are described in terms of the trellis. Here the trellis is used to model the channel memory with the state $S_{n-1} = x_{n-L}^{n-1}$ representing the previous inputs that have influence over the current output y_n. Once the output y_n is realized, the channel state transitions to a new state $S_n = X_{n-L+1}^{n}$.

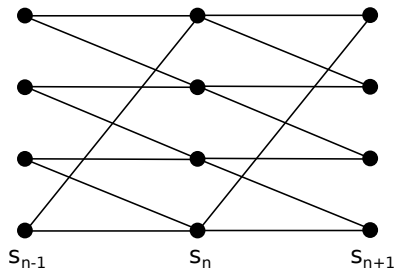

Figure 3.4: A 4-state Trellis

$p(y_1^n)$ can be calculated using the forward sweep of the BCJR algorithm [5], which is based on the following factorization:

$$p(y_1^n, s_n) = \sum_{s_{n-1}} p(y_n, s_n|s_{n-1}) p(y_1^{n-1}, s_{n-1}) \tag{3.5}$$

$$= \sum_{s_{n-1}} p(y_1^{n-1}, s_{n-1}) p(y_n|s_n, s_{n-1}) p(s_n|s_{n-1}) \tag{3.6}$$

This is a recursive algorithm where each iteration happens over one section of the trellis. The terms $p(y_1^{n-1}, s_{n-1})$ are associated with the nodes

towards the left of the trellis section. The terms $p(y_n|s_n, s_{n-1})p(s_n|s_{n-1})$ are associated with each branch in the trellis section, with $p(s_n|s_{n-1})$ representing the state transition probability, and $p(y_n|s_n, s_{n-1})$ representing the channel law. Then the nodes $p(y_1^n, s_n)$ at the right side of the trellis section can be calculated summing over the terms associated with the previous nodes multiplied by the terms associated with incoming branches. Once the algorithm has run for a sufficient length N for the convergence of the AEP in (3.4), then $p(y_1^n)$ can be calculated by marginalizing over s_n by summing over the node terms of the last trellis section.

While this method gives a linear complexity increase with the number of symbols N, the trellis size remains exponential with respect to the channel memory. Calculations for channels with large memory depth are still intractable. However one may arbitrarily specify a reduced complexity channel and with it calculate upper and lower bounds to the information rate [3].

The BCJR algorithm was designed to do maximum a posteriori probability (MAP) decoding of convolutional codes. Here we use the forward sweep to calculate a probability measure to estimate information rate, but the full BCJR over the same trellis would also be the optimum MAP detector for this channel. Thus there is a strong analog between calculating information rates of a channel by simulation and designing the receiver. Both have the same trellis structure.

3.4 Reduced Complexity Receiver Information Rates

If the real channel is characterized by a finite state trellis with the branch term $p(y_n|s_n, s_{n-1})$ then one can *arbitrarily* specify a reduced complexity channel characterized by $q(y_n|s'_n, s'_{n-1})$. s'_n represents a state variable from a reduced state space that is computationally tractable for the BCJR algorithm. Then we are able to calculate [3] an estimate for the upper bound of the information rate:

$$I(\mathcal{X};\mathcal{Y}) \leq -\frac{1}{N}\log q(y_1^N) - h(\mathcal{Y}|\mathcal{X}) \tag{3.7}$$

and an estimate for the lower bound:

$$I(\mathcal{X};\mathcal{Y}) \geq -\frac{1}{N}\log q(y_1^N) + \frac{1}{N}\log q(y_1^N|x_1^N) \tag{3.8}$$

The tightness of the bounds above depend on how closely $q(y_n|s'_n, s'_{n-1})$ approximates the true channel law $p(y_n|s_n, s_{n-1})$.

The lower bound of the achievable information rate stated above is also the achievable rate of a maximum likelihood (ML) receiver matched to our reduced complexity channel representation $q(y_n|s'_n, s'_{n-1})$ [3, 17, 18]. The random coding argument with an error exponent that contains the information rate represented in the above calculations has been used to prove this.

This result is particularly germane to the design of reduced complexity receivers since there is a one-to-one analog between the specification of $q(y_n|s'_n, s'_{n-1})$ and the structure of the ML receiver matched to it. Both imply the same trellis structure, and this makes defining the approximate channel and designing the reduced complexity receiver practically one and the same.

A convenience afforded by this result is that our study of reduced complexity receivers we can begin with focus on information rate calculations with the assumption that these rate curves can be approached by good codes and decoders.

In our investigation of non-linear reduced complexity receivers we use *marginalization* and *truncation*, or at times a combination of the two to obtain the reduced complexity design.

Reducing Complexity by Marginalization

One way to obtain a reduced complexity receiver is by reducing the state space by marginalizing the least influential symbols in the channel law. For example if a finite state channel law is $p(y_n|x_{n-3}, x_{n-2}, x_{n-1}, x_n)$, a reduced complexity channel law can be computed as:

$$p_m(y_n|x'_{n-1}, x'_n) = \sum_{x_{n-3}} \sum_{x_n} p(y_n|x_{n-3}, x'_{n-1}, x'_n, x_n)$$

Here the number of states has been reduced from $|\mathcal{X}|^3$ to $|\mathcal{X}|$, and $S'_n = X'_n$ and $S'_{n-1} = X'_{n-1}$. We would expect that the central symbols x_{n-2} and x_{n-3} to be more influential over y_n than the edge symbols x_{n-3} and x_{n-4} since the central symbols correspond to the central region of the pulse shape.

Although the number of states has been reduced, the description of the distribution $p_m(y_n|x'_{n-1}, x'_n)$ still has high complexity. In the case of an

AWGN channel, this distribution is a Gaussian mixture with $|\mathcal{X}|^2$ terms. For practical implementations it may be necessary to find a lower complexity representation such as a single Gaussian distribution. This can be done by calculating the mean μ_{p_m} and variance σ_μ^2 of the centroids of the Gaussian mixture. Then the mixture can be approximated as the Gaussian $\mathcal{N}(\mu_{p_m}, \sigma_\mu^2 + \sigma^2)$ where σ^2 is the variance of the AWGN. This technique is explored in Section 3.7.

Reducing Complexity by Truncation

Another way to obtain a reduced complexity receiver is to simply truncate without marginalization the effect of the least influential symbols during pulse shaping before the non-linearity. If the edge symbols are truncated and the non-linearity is mild, then since the constellation is symmetric we would expect that the distribution calculated by truncation to be close to that calculated by marginalization.

3.5 One-way Transmission with "Matched" Filter Detection

Figure 3.5 shows the satellite relay channel for one-way transmission with "matched" filtering. $\overline{U}$ is the bit source, and $\hat{U}$ is the estimated bits at the output of the detector D. The $E|\Pi|M$ block stands for channel encoding, interleaving, and modulation. $h(t)$ is the pulse shaping filter and $h(-t)$ is the "matched" filter, which is no longer truly matched to $h(t)$ because of the non-linear distortion caused by $g(\cdot)$. The non-linearity is modeled using the Saleh model as explained in Section 3.1 h_{1r} is the transfer function from the transmitting terminal to the satellite, and h_{r2} is the transfer function from the satellite to the receiving terminal. $\hat{X}$ is the transmit sequence, and $\hat{Y}$ is the receive sequence. $N(t)$ is AWGN.

Figure 3.6 (adapted from [56]) shows a realization of $\overline{Y}$ which is the output of the sampler after "matched" filtering in the absence of noise. A root-raised-cosine (RRC) pulse shape is used with an excess bandwidth parameter of $\beta = 0.25$. The dispersion of the samples is due to the non-linearity disturbing the orthogonality of the pulses, causing ISI.

An *input backoff* (IBO) of -2 dB was used here. By backing off the input the amplifier operates over less of its non-linear region. In Figure 3.2 an IBO

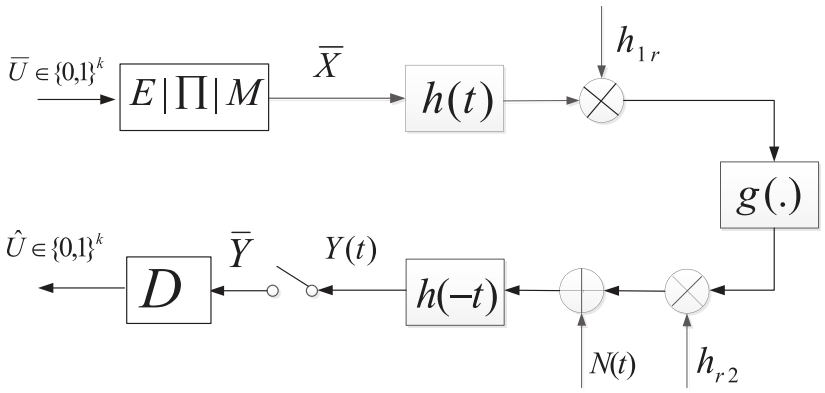

Figure 3.5: One-way "Matched" Filtering System Diagram

of 0 dB corresponds to an average input power level of 1, which has been normalized to correspond to the peak instantaneous output power. Backing off the input mitigates the non-linear distortion, however eventually it also begins to lower the output power which also affects the performance of the satellite link.

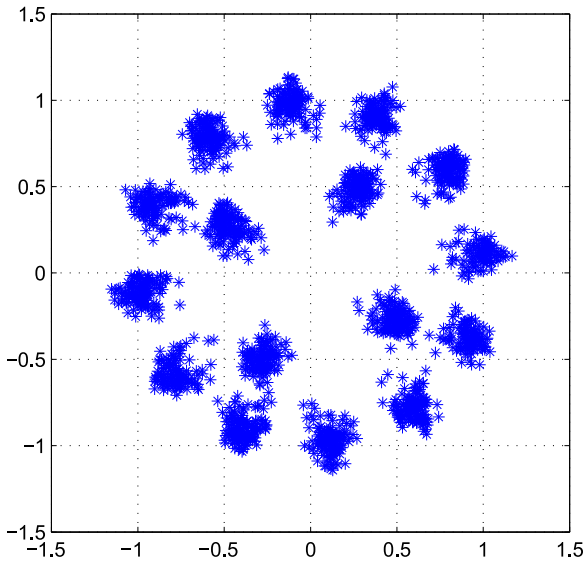

Figure 3.6: RRC Matched Filter Output Without AWGN with IBO = -2 dB

The finite state channel law $p(y_n|x_{n-L}^n)$ for the "matched" filter system in Figure 3.5 is the normal distribution $\mathcal{N}(\mu, \sigma^2)$ where

$$\mu = \int_{-\infty}^{\infty} g\left(\sum_{m=n-L}^{n} x_m p(t-mT)\right) p(t-(n-L+k))dt \tag{3.9}$$

and $\sigma^2 = N_0/2$ is the noise variance.

L is the memory depth induced by the non-linearity, and k is an integer such that x_{n-L+k} is the input symbol that has most influence over y_n. If $g(\cdot)$ is a mild non-linearity, we would expect that the center symbol x_{n-L+k} would have the most influence over y_n. We would expect that this symbol is in the middle of the vector x_{n-L}^n because the RRC matched filter is greatest in the middle.

In our numerical results, an RRC pulse having a length of $L_p = 8$ symbol periods at 8 times oversampling was used. The full memory order of the matched filtering system is $2(L_p - 1) = 14$. This is the number of adjacent symbols that affect the output Y_n in addition to the most influential symbol Y_{n-L+k}.

When using a "matched" filter configuration, it makes sense to marginalize or truncate down to a reduced complexity channel law that has an odd number of symbols. For example, based on the index in (3.9), the three symbols that have the most influence over y_n are x_{n-L+k}, which corresponds to the symbol over the main lobe of the "matched" filter, and $x_{n-L+k-1}$, which is the symbol immediately to the left of the main lobe, and $x_{n-L+k+1}$, the symbol immediately to the right. The other symbols can be marginalized or truncated. This results in a reduced complexity channel law with a reduced memory $L' = 2$, which is investigated in Section 3.5.

Non-Return-to-Zero (NRZ) Pulse Shaping

The NRZ pulse shape is a rectangle function with the width at the symbol rate. This avoids ISI even in the presence of the non-linearity since the pulses do not overlap. However this is not a practical pulse shape for spectral efficient communications since the occupied bandwidth is infinite. However calculating the information rates using NRZ pulse shaping is easy and a useful comparison against information rates of more bandwidth-efficient pulse shapes.

Figure 3.7 shows a realization of $\overline{Y}$ for NRZ pulse shaping without noise. Notice that in comparison to Figure 3.6 there is no ISI. However when compared to Figure 2.1, one can see the effect of the amplitude-to-amplitude and amplitude-to-phase distortion (see Section 3.1) where the radius of the rings and their phases have been distorted accordingly. The ring ratio is the ratio of the outer ring of the constellation to the inner ring. At the input to the non-linearity it is 2.85, as recommended in the DVB-S2 standard. This ratio is clearly changed by the non-linearity.

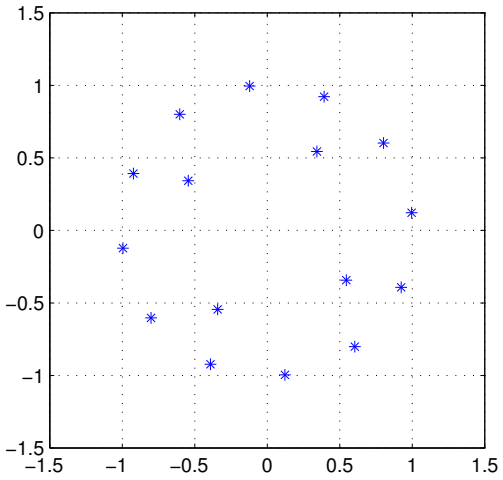

Figure 3.7: NRZ Matched Filter Output Without AWGN

In the case of this specific pulse, to calculate information rates we do not need to resort to the simulation techniques from Section 3.3. Instead the numerical integration techniques from Section 2.5can be used over the distorted constellation shown in Figure 3.6.

The information rates calculated for NRZ pulse shaping is shown in Figure 3.8, assuming uplink noise is negligible at the input to the nonlinearity. Here the information rates are plotted for different input backoff (IBO) levels. The IBO levels are in reference to the input level of the Saleh model that corresponds to E_{sat} which is the received energy for a signal at saturated power output. Based on the normalization of the Saleh model in Section 3.1, the IBO reference value at 0 dB corresponds to an input at

unit energy. The complex noise variance at the receiver output is N_0, the one-sided noise spectral density.

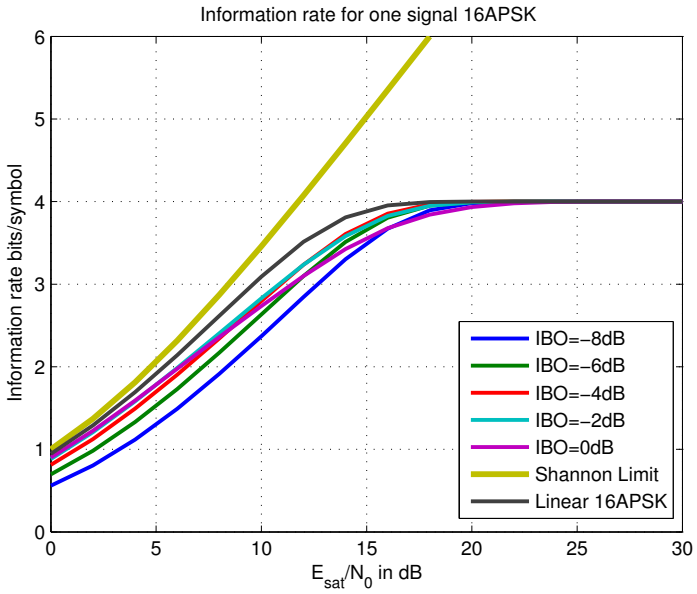

Figure 3.8: NRZ Matched Filter Information Rates

As required for 16-APSK modulation, the high-SNR asymptote approaches $\log_2 16 = 4$ bits/symbol at all input backoffs (IBO), but the proper choice of input backoff can optimize resources. For example, with coded transmission onto 16-APSK at 3 bits/symbol, input backoff of -2 dB is optimal, and requires $E_s/N_0 = 11$ dB. Notice also that for lower SNR (or rates), even closer to saturation is (slightly) better, but input backoff = -2 dB is remarkably robust. For comparison in Figure 3.8, we show the achievable rate for unconstrained Gaussian channel and for 16-APSK, interpreting E_{sat} as *average* energy per symbol. Much of the gap between the linear 16-APSK result and best nonlinear curve is due to the difference in average output power between linear and nonlinear cases, about 1.1 dB at IBO=-2 dB. Under an equal average power comparison, linear and nonlinear achievable rates are quite similar, indicating that nonlinear distortion is a relatively small effect compared to additive noise over the region of interest, at least for this NRZ pulse.

Memoryless Detection for RRC Pulse Shaping

In the case of RRC Pulse shaping and "matched filtering", It is possible to use memoryless detection where there is no compensation for the ISI. The ML detector of a memoryless channel is

$$\hat{X} = \underset{x_m}{\arg\max} \quad p(Y|X = x_m) \tag{3.10}$$

If the channel is AWGN, then the ML detector amounts to picking the symbol x_n which minimizes the Euclidean distance $|y - x_n|$.

For a channel with memory the above detection rule could be modified as

$$\hat{X} = \underset{x_m}{\arg\max} \quad q(Y|X = x_m) \tag{3.11}$$

where q could be the marginal of the channel law:

$$p_m(y|x) = \sum_{x_{n-L}} \sum_{x_{n-L+2}} ... \sum_{x_{n-L+k-1}} \sum_{x_{n-L+k+1}} ... \sum_{x_n} p(y_n|x_{n-L}, x_{n-L+1}, ..., x_{n-L+k-1}, x, x_{n-L+k+1}, ...x_n)$$

In this case $q(y|x)$ is a Gaussian mixture with centroids μ_x at points determined by the pulse shaping filter $p(t)$, non-linearity $g(\cdot)$, and "matched" filter $p(-t)$ as seen in (3.9). Figure 3.6 can be interpreted as a sample of the centroids of $p_m(y) = \sum_x p_m(y|x)p(x)$. Each of the 16 clusters of points represent a sample of the centroids of $p_m(y|x)$.

When the SNR is low, the Gaussian mixture will resemble a single Gaussian with a mean μ_{p_m} that is the average of the centroids, and a variance $\sigma^2_{p_m} = \sigma^2_\mu + \sigma^2$ where σ^2_μ is the variance of the centroids and σ^2 is the variance of the AWGN. This suggests a memoryless reduced complexity receiver that does minimum Euclidean distance detection with the metric $|y - \mu_{p_m}|$.

While in Section 3.3, the entropy rate estimation is done on a trellis, it is also possible to use the AEP (3.4) to estimate information rates for a reduced complexity receiver that does not take advantage of the memory. In this case, a trellis is not used, however the principles in Section 3.4 regarding reduced complexity receiver information rates still apply. In this case, the reduced complexity channel law is $p_m(y|x)$, from which we can estimate $H(\mathcal{Y}|\mathcal{X})$, and $p_m(y)$ can be used to estimate $H(\mathcal{Y})$.

To do this, a long sequence x_1^N is generated. Then from this, y_1^N is generated by simulating the pulse shaping, non-linearity, AWGN, "matched" filtering, and sampling. Then the entropy rates can be estimated as

$$\hat{h}(\mathcal{Y}|\mathcal{X}) = \frac{1}{N}\sum_{n=1}^{N} p_m(y_n|x_n)$$

$$\hat{h}(\mathcal{Y}) = \frac{1}{N}\sum_{n=1}^{N} p_m(y_n)$$

Then the information rate for the above reduced complexity ML receiver matched to $p_m(y_n|x_n)$ can be estimated as

$$\hat{I}(\mathcal{X};\mathcal{Y}) = \hat{h}(\mathcal{Y}) - \hat{h}(\mathcal{Y}|\mathcal{X})$$

In the following results, a combination of truncation and marginalization was used. The memory of the system was truncated from from 14 to 4, leaving two symbols on both sides of the most influential symbol. Then marginalization and Gaussian approximation was used to obtain $p_m(y_1|x_1, x_2)$. Then the AEP was applied as in the one way case to compute the information rates.

The information rates in bits per channel use as a function of E_s/N_0 is plotted for the case of RRC pulse shaping and a memoryless reduced complexity receiver is shown in Figure 3.9. The RRC pulse has a roll-off factor of 0.25, which is in the midrange of the roll-off factors specified in the DVB-S2 standard.

Again the curves are plotted for several IBO levels. A point of interest from Figure 3.9 is that to obtain $R = 3$ bits/symbol, we require E_{sat}/N_0 of 11.2 dB, and at this point IBO = -2 dB seems optimum. Thus memoryless detection for RRC yields slightly worse information rates than the memoryless NRZ case at the same IBO.

Memory-2 Detection for RRC Pulse Shaping

Truncation was used to generate a reduced complexity trellis, where the pulse shaping only included the indicies $n - L + k$, $n - L + k - 1$, and $n - L + k + 1$. This results in a memory-2 trellis with $|\mathcal{X}|^2 = 16^2 = 256$ states.

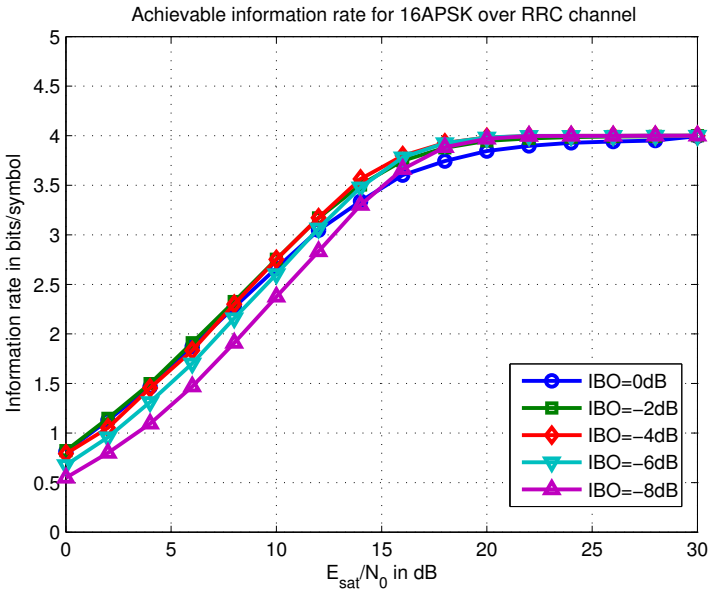

Figure 3.9: Memoryless Reduced Complexity Information Rates (RRC pulse shaping)

Figure 3.10 shows the information rates calculated for a reduced complexity Memory-2 receiver. The information rates were calculated on a trellis as explained in Section 3.3. Here the required SNR to obtain 3 bits/symbol drops by about 0.4 dB to 10.8 dB as compared with memoryless detection for IBO = -2 dB. This is a consequence of better modeling of the deterministic channel effect.

3.6 Two Bidirectional Users with "Matched" Filter Detection

Figure 3.11[3] (adapted from [35]) shows the system diagram of a two users using the same frequency for bidirectional communications. Each user transmits a symbol sequence ${X^{(1)}}_1^N$ and ${X^{(2)}}_1^N$ respectively that are pulse shaped and combined at the satellite antenna. The combined signal passes through the non-linearity $g(\cdot)$. Then the signal is received by both terminals as

[3]Figure adapted from Chenguang Xu

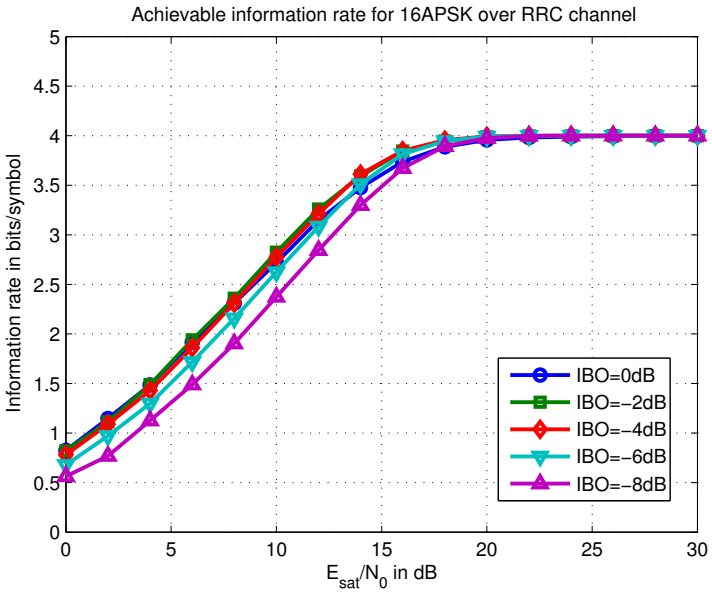

Figure 3.10: Memory-2 Reduced Complexity Information Rates (RRC pulse shaping)

$Y^{(1)}{}_1^N$ and $Y^{(2)}{}_1^N$ respectively. Since each terminal knows its own transmit sequence, it is able to cancel the echo of its own transmit sequence and extract the intended data. This can potentially double the spectral efficiency by reuse of the same spectrum for bidiretional data transfer.

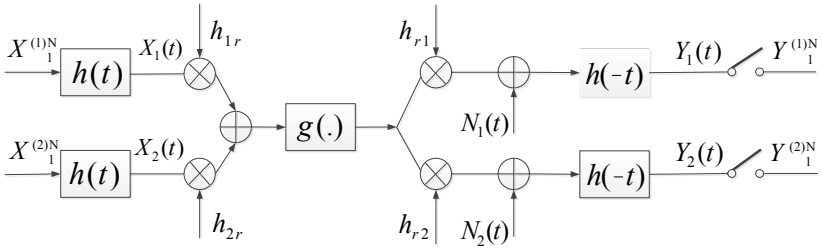

Figure 3.11: Two Bidirectional Users "Matched" Filtering System Diagram

To calculate information rates in the case of on-frequency bidirectional transmission, we need to estimate $I(\mathcal{X}^{(1)};\mathcal{Y}^{(1)}|\mathcal{X}^{(2)})$ which gives the information rate of user 1 transmitting $X^{(1)}{}_1^N$ to user 2 via the satellite which is

received as ${Y^{(1)}}_1^N$. ${Y^{(1)}}_1^N$ contains information from both users since both ${X^{(1)}}_1^N$ and ${X^{(2)}}_1^N$ are combined at the satellite. Likewise $I(\mathcal{X}^{(2)};\mathcal{Y}^{(2)}|\mathcal{X}^{(1)})$ gives the information rate of user 2 transmitting to user 1 in this bidirectional scenario.

The calculation of $I(\mathcal{X}^{(1)};\mathcal{Y}^{(1)}|\mathcal{X}^{(2)})$ is a difference in entropy rates:

$$I(\mathcal{X}^{(1)};\mathcal{Y}^{(1)}|\mathcal{X}^{(2)}) = \lim_{N\to\infty} h({Y^{(1)}}_1^N|{X^{(2)}}_1^N) - \lim_{N\to\infty} h({Y^{(1)}}_1^N|{X^{(1)}}_1^N, {X^{(2)}}_1^N) \tag{3.12}$$

where the second term the noise entropy.

As with one-way transmission, the state description of

$$p({y^{(1)}}_n|{x^{(1)}}_{n-L}^n, {x^{(2)}}_{n-L}^n)$$

needed to calculate

$$h({Y^{(1)}}_1^N|{X^{(2)}}_1^N)$$

is exponential with respect to the memory depth L. The complexity of the trellis is only a function of ${x^{(1)}}_{n-L}^n$, and has $|\mathcal{X}|^L$ states, the same as in the one user case. As before we need to use a reduced complexity channel law

$$p({y^{(1)}}_n|{x'^{(1)}}_{n-L'_1}^n, {x^{(2)}}_{n-L}^n)$$

with reduced memory depth L'. Again, this can be done by either truncation or marginalization.

NRZ Pulse Shaping

NRZ pulse shaping can be used in the two user case which results in a memoryless channel despite the non-linearity. As with one-way transmission, the entropy calculations can be done via numerical integration. These calculations were done by Xu in [57]. Figure 3.12 shows the information rates for two synchronous users with equal average power.

Memoryless Detection for RRC Pulse Shaping

As with the above results in one-way transmission, a truncation and marginalization was used. The memory of the system was truncated from from 14 to 4, leaving two symbols on both sides of the most influential symbol. Then marginalization and Gaussian approximation was used to obtain

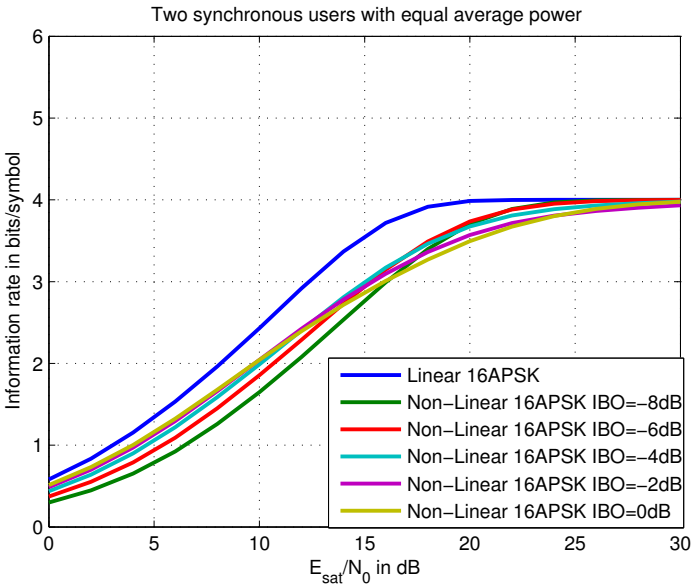

Figure 3.12: NRZ Matched Filter Information Rates for Two Users with Equal Power

$p_m(y_1|x_1, x_2)$. Then the AEP was applied as in the one way case to compute the information rates.

Figure 3.13 shows the estimated information rates for this setup with various IBO levels. We observe that to achieve 3 bits/modulator symbol the required SNR is 16.1 dB, with an optimal IBO of about -6 dB. Compared with similar results for single-user RRC transmission, the SNR must be increased by about 4.5 dB, and the backoff increased to compensate for larger *peak to average ratio* (PAPR) with two signals.

Memory-2 Detection for RRC Pulse Shaping

As in one-way transmission, truncation was used to generate a reduced complexity trellis, where the pulse shaping only included the indexes $n - L + k$, $n - L + k - 1$, and $n - L + k + 1$. This results in a memory-2 trellis with $|\mathcal{X}|^2 = 16^2 = 256$ states.

Figure 3.14 presents the results obtained with this memory-2 model. Incorporation of two-symbol memory into the model gives a better channel

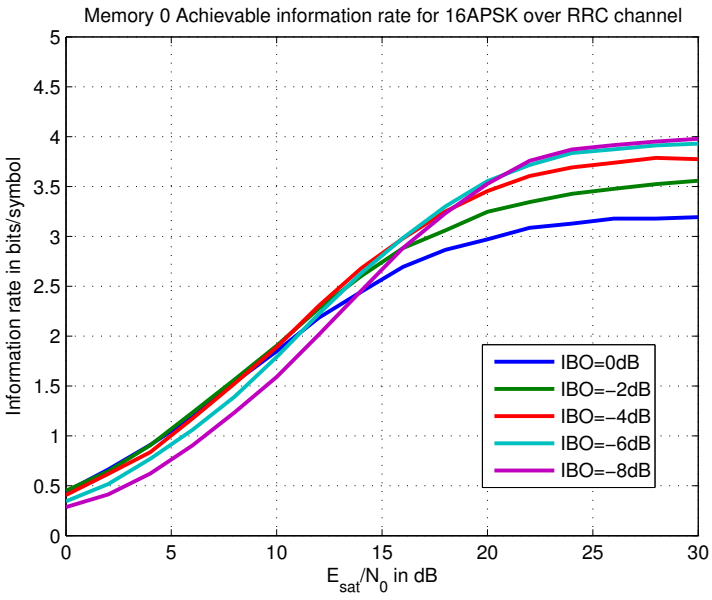

Figure 3.13: Bidirectional Memoryless Reduced Complexity Information Rates (RRC pulse shaping)

model, and proper decoding will slightly increase achievable rates, or lower SNR needed to obtain a given achievable rate. For example, to obtain R=3 bits/symbol/user, we now require only 15.2 dB, primarily because the optimal backoff can increase to -4 dB. In essence, proper modeling (and processing) of the nonlinear ISI allows a higher amount of nonlinearity.

3.7 One-way Transmission with Waveform Detection

In Section 3.5, a "matched" filter and sampling shown in Figure 3.5 was used to reduce the continuous waveform to discrete samples Y_n. While this architecture is common in linear communications systems, this causes loss of information as explained in Section 3.1. This information can be preserved by removing the "matched" filter and performing detection in the waveform domain. Figure 3.15 shows such a system without a "matched" filter.

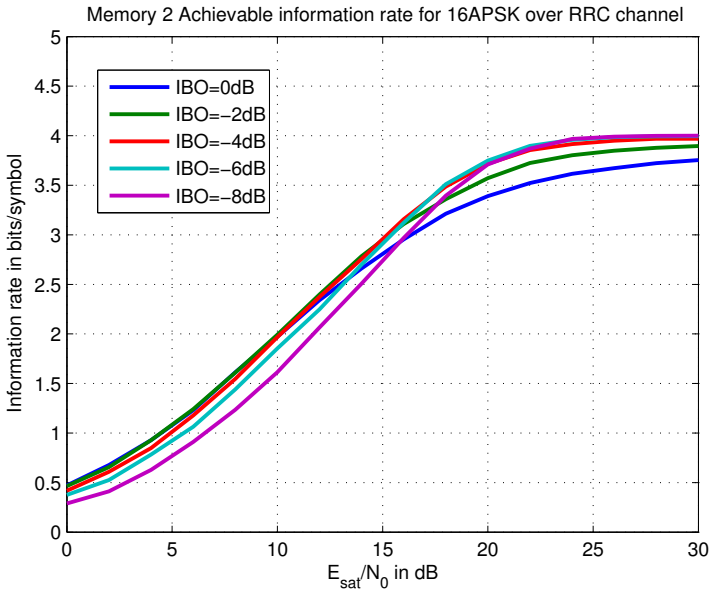

Figure 3.14: Bidirectional Memory-2 Reduced Complexity Information Rates (RRC pulse shaping)

Memory-3 Detection

Next, we discard the matched filter and sample the received signal at eight times the symbol rate. We omit the OMUX filter in this experiment. The results here are based on marginalizing a channel with pulse shaping filter length eight symbols (so $L = 7$) to a reduced-complexity representation with $L = 3$.

To characterize the memory of the system in the absence of a matched filter we divide the pulse into 8 basis chips [6], each the length of one symbol as shown in Figure 3.16. An oversampling rate of 8 was chosen to encompass, without aliasing, the spectral regrowth caused by the nonlinearity.

Thus y_n from the channel law factor $p(y_n|s_n, s_{n-1})$ associated with each trellis branch (see Section 3.3) is changed to a vector $\boldsymbol{y}_n$. The expected value $E[\boldsymbol{Y}_n|S_n = s_n, S_{n-1} = s_{n-1}]$ is the output of the memoryless nonlinearity with an input comprised of a linear combination of the basis chips. If we enumerate the basis chips as row vectors $\phi_1, \phi_2, ...\phi_8$ and form a generator matrix $G = [\phi_1; \phi_2; ...\phi_8]$, an 8 (pulse length) by 8 (oversampling rate)

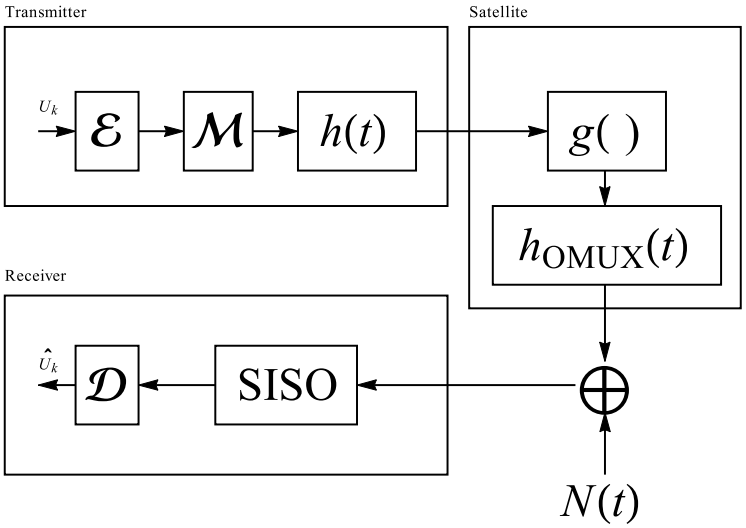

Figure 3.15: One-way Waveform Detection System Diagram

matrix. Then:

$$\begin{aligned} E[\boldsymbol{Y}_n | S_n = s_n, S_{n-1} = s_{n-1}] &= \mu_p(s_n, s_{n-1}) \\ &= \mu_p(x_{n-L}^n) \\ &= g(x_{n-L}^n G) \end{aligned} \tag{3.13}$$

where x_{n-L}^n is interpreted as a row vector and $g(\cdot)$ is the memoryless nonlinearity which in our case is the Saleh model.

A logical way to obtain a reduced complexity receiver is to marginalize the symbols associated with the chips that have the least amplitude. Based on our pulse and basis chips, to reduce the memory order $L = 7$ to $L' = 3$, we would marginalize s_1, s_2, s_7, s_8 according to (7). The result is a Gaussian mixture which we will approximate with a spherically-symmetric Gaussian vector of length 8, with a variance per real dimension that is the same across all branches in the trellis.

The expected chip projections associated with the reduced complexity

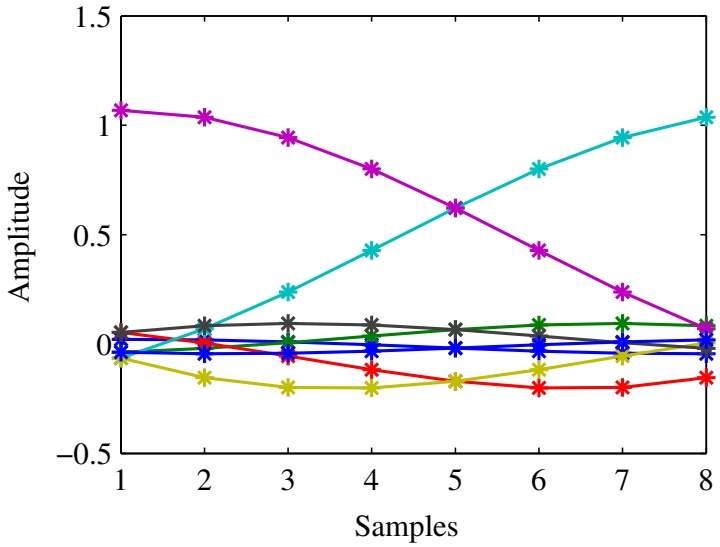

Figure 3.16: Eight Basis Chips

receiver is calculated by:

$$\begin{aligned} E[Y_m|S'_m = s'_m, S'_{m-1} = s'_{m-1}] &= \mu_q(s'_m, s'_{m-1}) \\ &= \mu_q(x'^m_{m-L'}) \\ &= \sum_{x_1}\sum_{x_2}\sum_{x_7}\sum_{x_8} \frac{\mu_p(x_1, x_2, x'_1, x'_2, x'_3, x'_4, x_7, x_8)}{|\mathcal{X}|^4} \end{aligned}$$

where $|\mathcal{X}|$ is the size of the alphabet which in our case was 16.

These were stored into a lookup table, and the single common variance was calculated empirically by calculating the variance of the error between a long simulated realization from the real channel, and a waveform generated from $\mu_q(s_m, s_{m-1})$.

Figure 3.17 shows the information rates calculated from this method. Figure 3.18 shows a magnification of the same results at 3 bits per symbol. For reference there is a plot of achievable rates for a linear channel, as well as the rates of a non-return-to-zero (NRZ) pulse shaped configuration. Under NRZ or rectangular pulse shaping there is no memory, but the constellation still gets distorted by the nonlinearity.

When compared to the memoryless matched filtered case in 3.5, we can see that the the SNR requirement at 3 bits per symbol has improved to

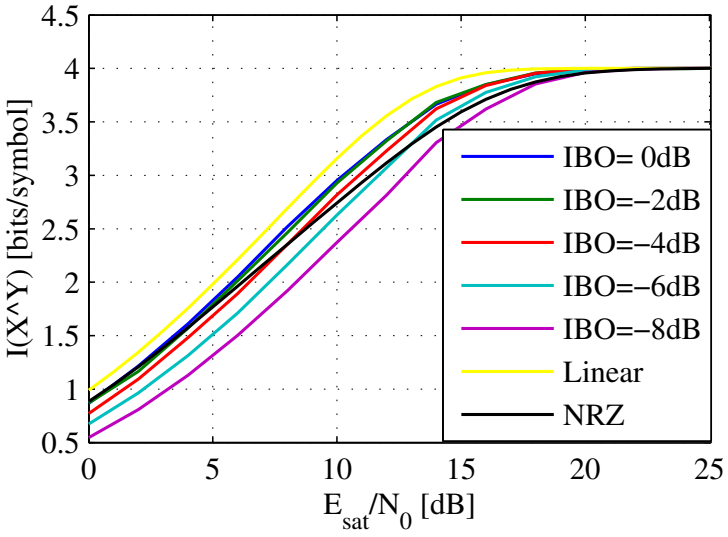

Figure 3.17: Information Rate for Memory-3 Reduced-State Receiver

about $E_{sat}/N_0 = 10.2$ dB which is a 1 dB improvement. The curve that has the highest rate is now IBO=0 dB; essentially by better probabilistic modeling of the channel, with oversampling, we are able to 'decode' the increased distortion associated with operation at saturation. The memoryless receiver sees these nonlinearity effects as intersymbol interference, but a trellis-based receiver is able to compensate for some of this nonlinearity and hence perform at higher rates in the presence of the nonlinearity.

It is important to note that our linear model produces a larger average output power than the nonlinear model, so the comparison with linear channel results is not exactly fair. We measured the output power at IBO=0dB to be 0.7 dB below that of our linear amplifier. The required SNR for the linear model is $E_s/N_0 = 9.4$ dB. So the gap between the linear model and the IBO=0 dB curve is mostly due to output power difference, which suggests that the $L = 3$ receiver is able to compensate for most of the nonlinearity, even at IBO=0 dB.

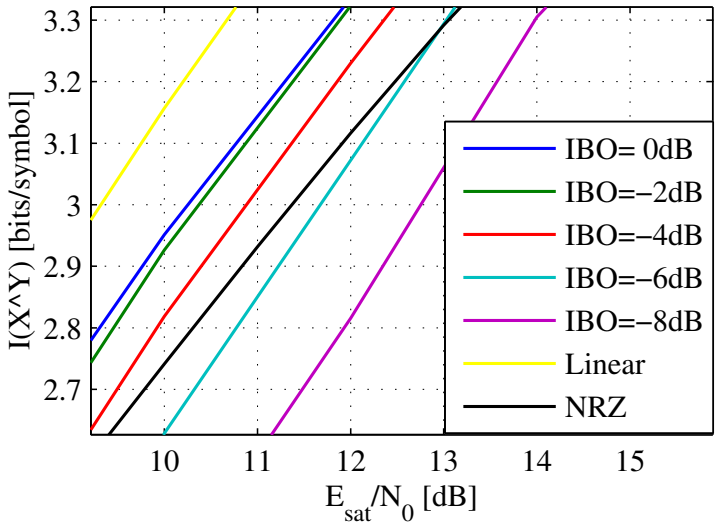

Figure 3.18: Information Rate for Memory-3 Near 3 Bits/Symbol

Marginalization with Generalized Gaussian Vector Approximation

Here we investigate whether a general Gaussian vector approximation to the marginals in (5) would perform better than the spherically symmetric approximation we used in the previous section. To investigate this, we applied an *output multiplexing* (OMUX) filter (see Figure 3.15) to the output of the nonlinear amplifier with $BT_b = 1.38$ allowing downsampling from 8 times oversampling to 2 times oversampling without aliasing effects. This was done to avoid the generation of a 16×16 covariance matrix which would be needed at 8 times oversampling. This filter was incorporated in the calculation of the approximation of $q(y_m|s'_m, s'_{m-1})$.

Now, instead of just finding $\mu_q(x'^m_{m-L'})$ as the mean from

$$\mu_p(x_1, x_2, x'_1, x'_2, x'_3, x'_4, x_7, x_8)$$

for all x_1, x_2, x_7, and x_8, we also find the covariance matrix $C_q(x'^m_{m-L'})$ based on these μ_p values and store them with μ_q for each branch in the trellis. During the trellis sweep we use the general form of the multivariate normal distribution with the appropriate μ_q and C_q covariance matrix for each branch.

However, with this finer-tuned channel law, we did not see any appreciable improvement. Figure 3.19 shows the two curves for IBO=0 dB from the two methods are basically indistinguishable. We attribute this to the dominance of additive receiver noise over most of the SNR range shown.

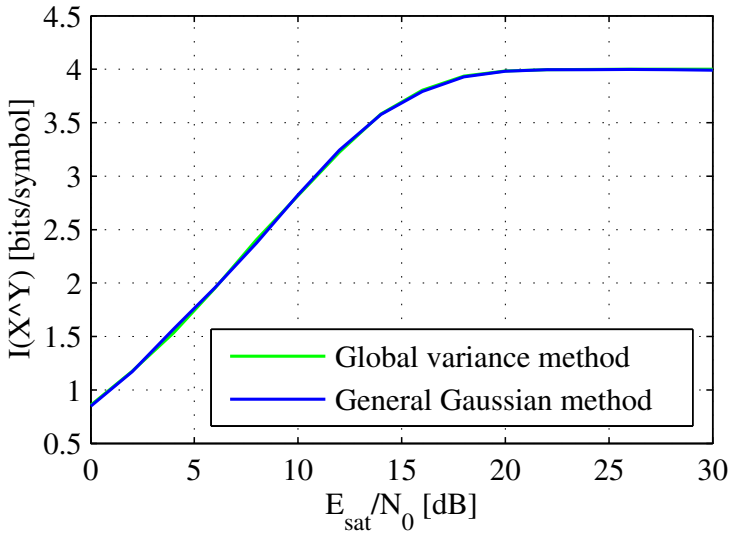

Figure 3.19: Comparison of Single Common Variance and General Gaussian Approximation

Part II

Faster-than-Nyquist, and Comparison with OFDM

Chapter 4

Introduction

Many applications in wireless technology specify power spectrum masks that emissions must satisfy, primarily in order to avoid significant interference to adjacent channels, and perhaps to allow for nonlinear amplification effects. Examples of such masks are found in WiFi, UMTS standards, and in satellite communication. These masks imply an information-theoretic limit on the reliable communication rate, given some received power level P_r and Gaussian noise level.

Assuming a standard white Gaussian noise model for the channel, along with an ideal dispersionless transmission, the constrained capacity [1] can be shown to be [1]

$$C_{con} = \int_0^\infty \log_2[1 + \frac{P_r|H(f)|^2}{N_0}]df \quad bits/second \tag{4.1}$$

where $|H(f)|^2$ is the spectral mask constraint, normalized so that its one-sided spectral integral is 1. In a familiar special case where the mask is an ideal rectangle having bandwidth B Hz, we have $|H(f)|^2 = 1/B$ and the resulting constrained capacity is

$$C_{con} = B\log_2(1 + \frac{P_r}{N_0 B}) \quad bits/second \tag{4.2}$$

There has been renewed interest recently in transmission schemes for maximizing information rate subject to bandwidth constraints. Time-domain

[1] Constrained in the sense that the transmitted power spectrum is shaped according to $|H(f)|^2$

FTN operation [1] is a single-carrier QAM approach that intentionally signals faster than the 'Nyquist rate', or faster than a system designed to have zero intersymbol interference (ISI) would use. This introduction of ISI can actually allow an increase in information rate, in bits/symbol and bits/second, whenever the pulse shaping design has 'excess bandwidth', i.e. the occupied bandwidth exceeds that associated with sinc-pulse shaping. The gains are larger when the excess bandwidth increases; on the other hand for very tight pulse shaping designs, FTN operation cannot provide any gain over the traditional brickwall Shannon formula in (4.2) above, see [40].

In this chapter we compare the use of FTN with OFDM in scenarios with mask constraints, using the WiFi 802.11 b and g masks as design examples. The 802.11b standard is based on direct sequence spread spectrum and has a rectangular mask over the 22 MHz central zone. It requires attenuation of -30 dB relative to the peak power spectral density from 11 MHz out to 22 MHz from the center. At 30 MHz from the center the mask requires a relative power of -50 dB.

On the other hand, the 802.11g standard is based on OFDM with a mask that allows an 18 MHz central zone with decibel-linear decay to -20 dB relative power at 11 MHz from center, followed by another decibel-linear decay to -40 dB at 30 MHz from the center. Beyond 30 MHz the mask limits emissions at -40 dB. These two masks are shown in Figure 4.1.

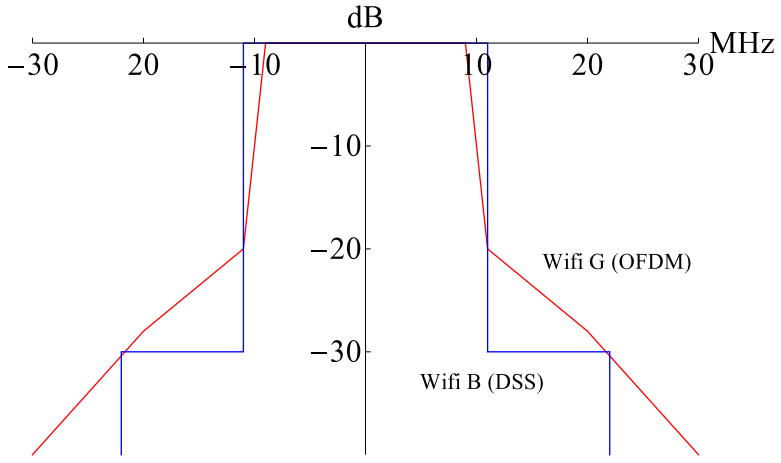

Figure 4.1: Spectral Masks for WiFi Standards

Traditionally the single-carrier QAM approach for these mask constraint would pick a transmit pulse that is root-Nyquist for signaling rate of maybe 16 Msps, with excess bandwidth factor 3/8. This design would fit under both masks and have an occupied bandwidth of 22 MHz. Zero ISI ensues, assuming an ideal channel. Similarly, OFDM designs, as in 802.11g, would space 64 tones (not all active subchannels) over the range of 20 MHz, with guard intervals on upper and lower band edges to meet the mask constraint. Neither design fully utilizes the potential of the mask-constrained channel; the 'excess bandwidth' allowed by the mask can significantly increase information rate especially for high SNR.

Both FTN and more aggressive OFDM designs can increase the achievable information rate over these baseline designs, as discussed below. We first review FTN and its implementation, then concentrate on OFDM, with the belief that OFDM has implementation advantages for attaining the constrained capacity implied by the mask.

4.1 FTN Background

A FTN pulse modulated signal has the form

$$x_{ftn}(t) = \sum_n x_n p_{rny}(t - nT/s)$$

where $p_{rny}(t)$ is a root-Nyquist pulse with orthogonality at time intervals of T see(2.5). FTN pulse shaping has the same form as (2.8) except it contains a speedup factor s, with $s = 1$ signifying the Nyquist rate. x_n are the information bearing symbols from some alphabet.

Minimum Distance

The development of FTN began with Mazo's study of minimum distance between Nyquist pulse modulated signals at rates faster than the Nyquist limit [29]. Mazo found that the minimum squared Euclidean distance, which is defined as

$$d^2_{min} = \min_{\boldsymbol{x^{(1)}}_1^N, \boldsymbol{x^{(2)}}_1^N} \int_{-\infty}^{\infty} \left(x^{(1)}_{ftn}(t) - x^{(2)}_{ftn}(t) \right)^2 dt \tag{4.3}$$

$$\text{where } \boldsymbol{x^{(1)}}_1^N \neq \boldsymbol{x^{(2)}}_1^N \tag{4.4}$$

remained the same even for some values of $s > 1$, i.e. when ISI is allowed. Mazo found that for sinc pulse shaping s could be increased by 25% without changing d^2_{min}. Since there is a direct relation between d_{min} and the uncoded error rate, this suggests that for uncoded sequences, the symbol rates could be increased by 25% without significantly affecting the uncoded error rate. This comes at a cost of increased receiver complexity as ISI exists and a more complicated receiver such as a Viterbi detector is needed to resolve the ISI.

Increased Number of Dimensions

While the minimum distance is useful to compute a lower bound for uncoded error rates, it is not useful in predicting *coded* information rates. Codes by design increase the distance between transmit sequences as not all possible transmit sequences are used, and the squared distance between valid sequences is greatly increased. Thus the minimum distance sequences $x_1(t)$ and $x_2(t)$ found by the minimization in (4.3) would typically not both appear in a codebook, and thus the minimum distance as defined here would not apply in a coded system.

One could however compute the minimum distance between codewords in a codebook. Even then some codes with poor minimum distance have been shown to have good performance; and the performance of a code depends more on distance properties among all the codewords rather than just the worst pair [28]. Thus while error probabilities based on minimum distance are a tight lower bound for uncoded sequences, this does not extend to coded sequences. So the advantage of FTN signalling for coded sequences operates on a different mechanism than the minimum distance argument.

The reason for improved information rates with FTN over Nyquist signalling is that for a given transmission length, FTN signalling can generate signals that span more dimensions than signals at the Nyquist rate, as long as the pulse used is not the sinc. At the Nyquist rate, the number of dimensions equals the number of pulses in the sequence as the train of Nyquist pulses are themselves an orthonormal basis $\mathcal{B}_N$ for Nyquist rate signalling. When the same pulse is used with FTN signalling, $\mathcal{B}_N$ can no longer span the new set of FTN signals, and a new basis $\mathcal{B}_{FTN}$ can be formed from $\mathcal{B}_N$ using the Gram-Schmidt procedure such that $\mathcal{B}_N \subset \mathcal{B}_{FTN}$. The additional dimensions infer a faster information rate.

For the sinc pulse $\mathcal{B}_N = \mathcal{B}_{FTN}$, and there is no advantage for FTN with the sinc pulse. This reconciles FTN concepts with Shannon's limit for spectral efficiency given in (4.2). Shannon's derivation is based on the sampling theorem where the sinc pulse spaced at the Nyquist rate is an orthonormal basis for all signals constrained to the square mask with width B. Thus the codewords based on the sinc pulse are able to span the full signal space defined by the ideal rectangular spectrum and average power constraint. This explains why FTN with the sinc pulse does not lead to faster information rates since any waveform based on FTN sinc signaling could still be represented at the Nyquist rate as a consequence of the sampling theorem. In other words FTN transmission with the sinc pulse does not generate any new dimensions. A formal presentation of the forgoing ideas is given in Section 3.3 of [40].

More Distance Between Codewords

There is another way to understand how FTN increases the information rate. Let us consider an orthonormal basis $\mathcal{B}_{Nsinc}$ which is the sinc pulse at the Nyquist rate over the bandwidth B for some time interval $\mathcal{T}$ which we set to the length of a codeword. $\mathcal{B}_{Nsinc}$ represents the maximum number of dimensions over $\mathcal{T}$ which is equal to the number of sinc pulses at the Nyquist rate that fit in $\mathcal{T}$. Ignoring the transients at the edges of $\mathcal{T}$, $\mathcal{B}_{Nsinc} = 2B\mathcal{T}$

Now consider a root-Nyquist pulse $p(t)$ with excess bandwidth parameter β where the Nyquist rate of the pulse is $B/(1+\beta)$. The occupied bandwidth is B and therefore $\mathcal{B}_{Nsinc}$ is able to span any signal generated by $p(t)$ at any symbol spacing T/s.

Based on the arguments of dimensionality above, speeding up the pulse rate allows codewords to use points in the signal space spanned by $\mathcal{B}_{Nsinc}$ under the same spectral and power constraints that are not available at the Nyquist rate. This allows a better spacing of codewords which give a higher performance than is possible at the Nyquist rate.

FTN Performance

Here we give an example of the performance benefit of FTN signalling. In Figure 4.2 are plots of the frequency spectrum for three baseband signals. The first is that of a sinc pulse shaping at the Nyquist rate with a bandwidth of $B = 1/2$. The plot of capacity as a function of E_s/N_0 is shown on 4.3.

This signal represents the optimal performance theoretically obtainable for an AGWN channel with bandwidth $B = 1/2$ (see Section 2.5).

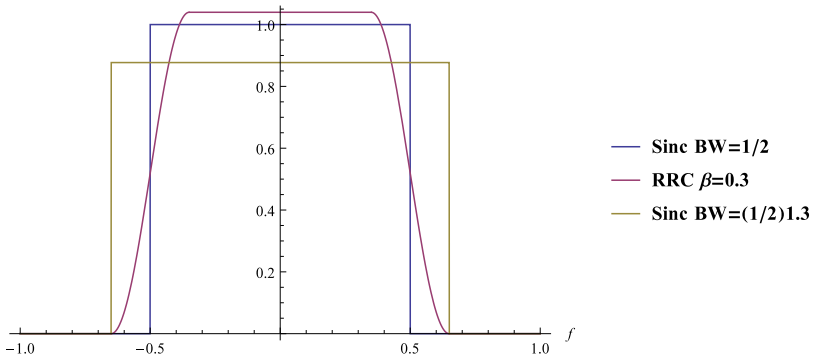

Figure 4.2: FTN Spectrum

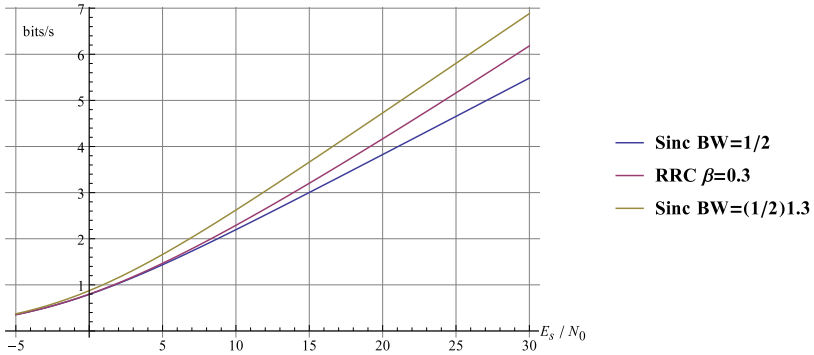

Figure 4.3: FTN Capacity

However pulse shaping with the sinc pulse is impractical because the pulse decays slowly and truncation produces significant spectral leakage. Also, significant ISI is generated when there is a timing error in the sampling of the matched filter output [21]. For these reasons the RRC pulse is often used in practice.

The RRC pulse is also root-Nyquist, and when used at the Nyquist rate it gives the same performance as the sinc pulse at the same symbol rate. However it uses more spectrum as stipulated by an excess bandwidth parameter β. In Figure 4.2 the frequency spectrum of an RRC pulse designed

at the same symbol rate with $\beta = 0.3$ as the sinc pulse with $B = 1/2$ is shown. The occupied bandwidth for this signal is $B = 1/2(1.3) = 0.65$. When used at the Nyquist rate, the information rate for this system is the same as that of the sinc pulse with $B = 1/2$. However if we use FTN with $s > 1$ for this pulse we can approach the FTN constrained capacity calculated by (4.1) using the RRC spectrum shown in Figure 4.3. However to take advantage of these higher bit rates, the incurred ISI needs to be decoded with a more complex receiver.

Finally, to show that Shannon's expression for capacity is not violated, the capacity of an AWGN channel with $B = 1/2(1.3) = 0.65$ is also plotted. Thus the FTN rates is upper bounded by this ultimate limit, but shows an improvement over Nyquist rate signalling.

4.2 FTN Implementation Issues

There are several significant implementation difficulties that FTN research tends to overlook. We list

- Receiver complexity to decode the ISI, at least for optimal processing, grows exponentially with the memory length of the induced ISI channel, which in turn is proportional to the over-clocking factor s. The optimal processor is a Viterbi processor for uncoded signaling, and a BCJR-style MAP soft decoder for coded FTN operation, followed by an outer decoder. Attainment of high spectral efficiency normally requires larger alphabet, making the optimal receiver complexity even larger. There has been work on reduced complexity receivers to either equalize or to shorten the channel response, [19],[49] but some attendant loss in information rate is implied.

- Synchronization, both for symbol timing and carrier phase tracking, is complicated by the lack of a well-defined receiver eye pattern, hence the inability to make tentative decisions to drive symbol timing loops and/or decision-directed carrier tracking loops. Nonlinear operations on the received FTN signal can produce spectral lines that contain timing information, but performance is certainly worse than with Nyquist transmission.

- As the overclocking parameter s increases, the QAM signal's peak-to-average power ratio (PAPR) grows, making linear transmitter design more problematic. (This is also an issue with OFDM discussed below.)
- Most of the work on FTN to data has focused on binary inputs to the transmitter, with s sufficiently large that the constrained capacity (associated with Gaussian codebooks) can be achieved. Increasing s implies a rather long effective channel response that implies large receiver complexity. For example, if we hope to attain capacity of say 4 bits/Nyquist rate interval with binary FTN, the overclocking parameter must be at least 5, complicating the receiver design (synchronization and decoding.)
- Operation on non-ideal channels will change the end-to-end ISI characteristic, and make optimal receiver complexity even larger, assuming ISI can be learned.
- In a coded transmission scenario, the FTN decoder must produce soft-outputs to the outer channel decoder, with iterative exchange between decoders to operate near channel capacity limits. For these practical reasons, we revisit OFDM, popular for many bandwidth-efficient applications (LTE, WiFi, ADSL, etc.).
- We proceed to show that OFDM can achieve the constrained capacity implied by any spectral mask. The design requires careful selection of the number of OFDM subchannels, along with band-edge engineering of the energy profile to satisfy the mask. OFDM does not have the primary difficulty of FTN, namely exponentially-growing receiver complexity, but instead relies on FFT hardware for a reasonably simple implementation that provides ISI-free operation and easy interface with coding hardware. Furthermore another OFDM advantage is low complexity equalization at the cost of the *cyclic prefix* (CP).

4.3 OFDM Background

In OFDM [39] the transmitter accepts blocks of N data symbols per OFDM symbol duration T_f seconds, and forms the inverse FFT to produce a discrete-time complex sequence which is then converted to continuous-time and upconverted, or vice versa. Our formulation employs frequency-domain

zero-padding with $(M-1)N$ zeros prior to the IFFT, to allow easier CT reconstruction free of aliasing. We refer to this as M-fold padding.

The discrete-time domain signal for a single OFDM frame is then

$$x_{ofdm}[n] = \sum_{k=0}^{N-1} x_k e^{j2\pi kn/MN}, \quad n \in \{0, ...MN-1\} \tag{4.5}$$

where x_k are the message symbols. The time domain sequence has a rate $f_s = MN/T_f$. At this point practical designs will prepend a CP to allow for easy channel equalization, but we omit this step for simplicity.

The DTFT of this time-domain signal is

$$\begin{aligned} X_{ofdm}(e^{j\Omega}) &= \sum_{k=0}^{N-1} x_k \sum_{n=0}^{MN-1} e^{-jn[\Omega - \frac{2\pi k}{MN}]} \\ &= \sum_{k=0}^{N-1} x_k e^{-j\frac{MN-1}{2}[\Omega - \frac{2\pi k}{MN}]} \frac{\sin[\frac{MN}{2}(\Omega - \frac{2\pi k}{MN})]}{\sin[\frac{1}{2}(\Omega - \frac{2\pi k}{MN})]} \end{aligned} \tag{4.6}$$

A continuous stream of OFDM symbols is merely a concatenation of such signals. Using this DTFT and defining the power spectrum as the limiting expected value of the magnitude-squared of the DTFT, normalized by time, it is fairly easy to show that the power spectrum for the DT signal is proportional to

$$\mathcal{P}(e^{j\Omega}) \propto \sum_{k=0}^{N-1} \sigma_k^2 \frac{\sin^2[\frac{MN}{2}(\Omega - \frac{2\pi k}{MN})]}{\sin^2[\frac{1}{2}(\Omega - \frac{2\pi k}{MN})]} \tag{4.7}$$

where σ_k^2 is the energy profile in frequency assigned to the various subchannels. A typical profile is constant with zero guard bands on either band edge; alternatively we might adopt a flat central zone with a taper near the band edge.

The *power spectral density* (PSD) in (4.7) is simply the sum of translated squared 'digital sinc' functions, with spacing in Ω given by $2\pi/MN$ and null-to-null width π/MN. The latter is smaller as N increases for a given overall bandwidth, which explains why the PSD drops more sharply out-of-band as N increases, as seen below.

It is this PSD that we focus on below, but one should realize that the final PSD for the CT signal is, assuming an ideal interpolator from DT to CT,

$$\mathcal{P}_{CT}(f) \propto \frac{T_f}{MN}|H(e^{j2\pi f T_f/MN})|^2 \times \sum_{k=0}^{N-1} \sigma_k^2 \frac{\sin^2[\pi(fT_f - k)]}{\sin^2[\pi(fT_f - k)/MN]}, \quad |f| \leq \frac{MN}{2T_f} \tag{4.8}$$

where $H(e^{j\Omega})$ is the frequency response of a DT digital filter to reduce high-frequency sidelobe content without introducing significant distortion.

Sidelobes of a Tone: the 'Digital Sinc' versus the Ideal Sinc

In this and the next section we study the behavior of the spectral properties of the sidelobes produced by an individual tone as well as the overall modulated OFDM symbol. These sidelobes are relevant to fitting the transmit signal under a spectral mask.

The M-fold padding effectively oversamples the time-domain waveform by M. The digital sinc function representing a tone in (4.6) is a close to the ideal sinc function when M and N are sufficiently large. The ideal sinc function would be the spectral density of a tone if the ideal rectangular window of length T_f were applied to the reconstructed IFFT output. Because the ideal rectangular function has an infinite range of frequencies, it is impossible to represent it in discrete time. However with sufficient oversampling, the difference can be mitigated. The sidelobes of the digital sinc are worse than the ideal sinc, but the differences only become evident farther away where the sidelobes have already been significantly attenuated.

The digital sinc power spectrum has the following form when the peak power spectral density is normalized to 0 dB:

$$\mathcal{P}_{tone,dsinc}(f) = \frac{1}{(NM)^2} \frac{\sin(\pi T_f f)^2}{\sin\left(\frac{\pi T_f f}{MN}\right)^2} \tag{4.9}$$

With the same normalization the ideal sinc has the following spectrum:

$$\mathcal{P}_{tone,sinc}(f) = \frac{\sin(\pi T_f f)^2}{(\pi T_f f)^2} \tag{4.10}$$

The denominators of the above expressions give the envelope of the sidelobes. The sidelobes touch the envelope at $f = (\frac{1}{2}+m)\Delta_f$ where $\Delta_f = 1/T_f$, and m signifies the m-th sidelobe. Figure 4.4 shows $\mathcal{P}_{tone,dsinc}(f)$ and its sidelobe envelope $\frac{1}{(NM)^2}\frac{1}{\sin\left(\frac{\pi T_f f}{MN}\right)^2}$ plotted against the frequency normalized by Δ_f. At these frequencies close to the tone there is no difference between the digital sinc and ideal sinc and their related envelopes regardless of M.

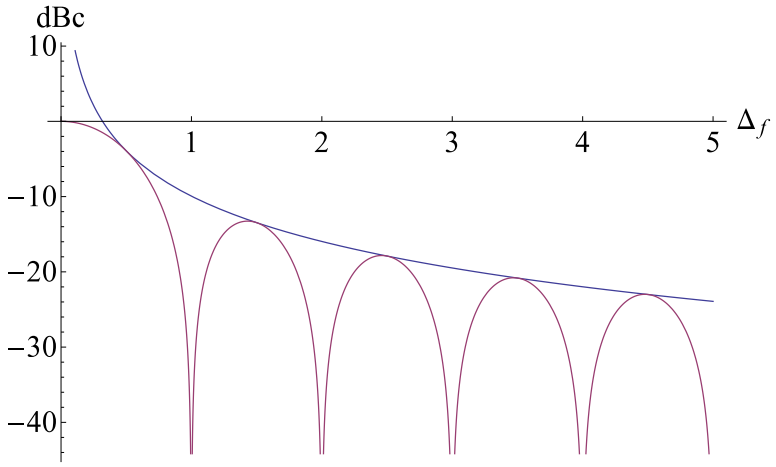

Figure 4.4: Single-Tone Sidelobe Envelope

Figure 4.5 shows the sidelobe envelopes of various values of M, the oversampling rate compared to the envelope of the ideal sinc for $N = 1024$ and $B = 22$ MHz. For $M = 4$, the point at which the digital sinc envelope differs 1 dB from the ideal happens at 20 MHz away from the tone frequency and at a power level of around -70 dBc. With $M \geq 2$ the digital sinc is virtually identical to the ideal for frequencies within 5 MHz of the tone.

Furthermore the digital sinc spectrum has a cyclic nature where the sidelobes wrap around at the folding frequency and appear at the other end. The sinc function has an infinite spectrum, while the digital sinc is limited by the sampling theorem to half the sampling rate. This behavior can also be understood as the cyclic convolution due to windowing in the discrete time domain. Again this is mitigated by oversampling as the sidelobes decay over the nulled frequencies of the M-fold padding before appearing at the other side.

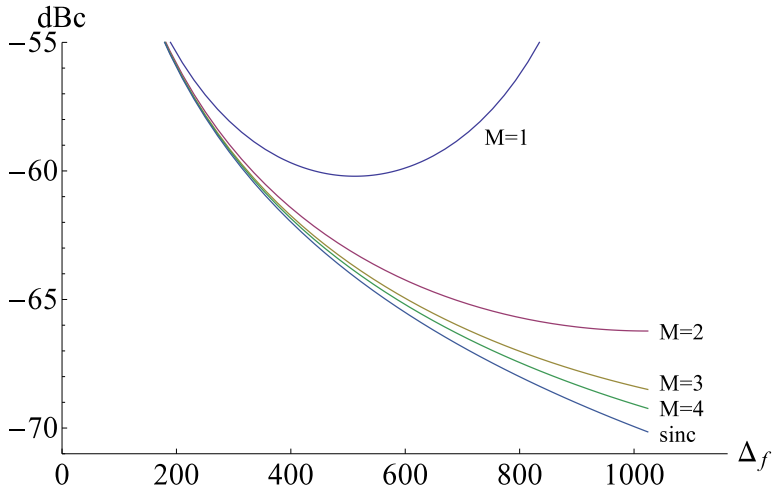

Figure 4.5: Single-Tone Sidelobe Envelope vs. M ($N = 1024$, $B = 22$ MHz)

Another difference is that the sidelobes of the digital sinc are limited to half the sampling rate as a consequence of the sampling theorem. Thus if the IFFT of the modulator is taken over a bandwidth B, then the sidelobes are limited to a bandwidth of MB. In practice these sidelobes will be filtered by some by some $H(e^{j\Omega})$ as in (4.8) or perhaps by some analog filter.

OFDM Sidelobes

Since we are studying information rates under spectral mask constraints, we analyze the PSD characteristics of OFDM sidelobes. If we assume independence of the data X_k^1 in (4.5), the power spectral density of the sidelobes of the *entire* OFDM symbol is a summation of that of the tones comprising the symbol. We begin with the ideal sinc from (4.10), where the peak PSD of each tone is normalized to 0 dB:

$$\mathcal{P}_{ofdm,sinc}(f) = \sum_{k=0}^{N-1} \left(\frac{\sin[\pi T_f (f - \Delta_f k)]}{\pi T_f (f - \Delta_f k)} \right)^2 \tag{4.11}$$

The envelope of OFDM sidelobes can be calculated from the envelope of the individual tones. For convenience, we center the frequency axis on

the last tone of the OFDM symbol:

$$\mathcal{P}_{env,sinc}(f) = \sum_{k=0}^{N-1} \frac{1}{\left(\pi T_f (f + \Delta_f k)\right)^2} \tag{4.12}$$

$$= \frac{1}{(\pi T_f)^2} \sum_{k=0}^{N-1} \frac{1}{(f/\Delta_f + k)^2} \tag{4.13}$$

Taking $N \to \infty$, the above series converges as

$$\lim_{N\to\infty} \mathcal{P}_{env,sinc}(f) = \frac{1}{(\pi T_f)^2} \sum_{k=0}^{\infty} \frac{1}{(f/\Delta_f + k)^2} \tag{4.14}$$

$$= \frac{1}{(\pi T_f)^2} \psi^{(1)}(f/\Delta_f) \tag{4.15}$$

where $\psi^{(1)}$ is the *polygamma function of order 1*, defined as the second derivative of the logarithm of the gamma function:

$$\psi^{(1)}(z) \triangleq \frac{d^2}{dz^2} \ln \Gamma(z) \tag{4.16}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(z+k)^2} \tag{4.17}$$

Thus we have a closed-form expression of the sidelobe envelope of a complete OFDM symbol as $N \to \infty$. For finite N the sidelobe envelope can be calculated as:

$$\mathcal{P}_{env,sinc}(f, N) = \frac{1}{(\pi T_f)^2} \left(\psi^{(1)}(f/\Delta_f) - \psi^{(1)}(f/\Delta_f + N)\right) \tag{4.18}$$

For large N the second term can be neglected and the expression in (4.14) is a tight upper bound of the sidelobe envelope. The correction for finite N becomes more necessary when an accurate estimate is needed for sidelobes further away from the OFDM symbol.

As in Section 4.3, the points at which the sidelobes touch the envelope are at $f = (\frac{1}{2} + m)\Delta_f$ for the m-th sidelobe.

This analytic expression for the sidelobe envelope is useful in determining the number of tones that need to be deleted or filtered to meet a certain

spectral mask or requirement. When designing a sidelobe filter, the sidelobe envelope can be determined using this method instead of by brute force calculation via (4.11).

The use of a CP lowers the sidelobes. Calculation of the sum of the sidelobe envelopes gives

$$\mathcal{P}_{env,cp}(f) = \sum_{k=0}^{N-1} \frac{1}{\left(\pi(T_{cp}+T_0)(f+\Delta_f k)\right)^2} \tag{4.19}$$

$$= \frac{1}{(\pi T_0)^2}\left(\psi^{(1)}(f/\Delta_f) - \psi^{(1)}(f/\Delta_f + N)\right) \tag{4.20}$$

where T_{cp} is the length of the CP, $T_{cp}+T_0$ is the length of the complete OFDM symbol, and $\Delta_f = 1/T_0$.

However for OFDM with a CP, summing the envelope of the tones is no longer a tight bound on the envelope of the OFDM symbol since the sine numerators of the tone sinc functions are no longer exactly in phase and do not add constructively over all frequencies. However this expression may still be useful for situations where T_{cp} is much less than T_0.

Chapter 5

Mask Constrained Information Rates

In the following sections we investigate the application of FTN and OFDM to increase the information rates under mask constraints. Figure 5.1 shows the constrained capacities of the two WiFi masks compared to various practical implementations.

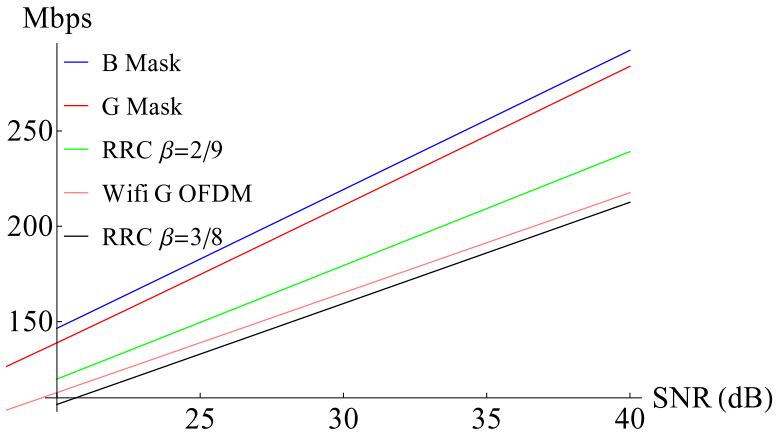

Figure 5.1: Constrained Capacities of the Mask Compared with Single Carrier Nyquist Signalling

The constrained capacity for the b mask was calculated using (4.2) using only the central zone 22 MHz bandwidth. Similarly the capacity for the g mask was calculated over the same central bandwidth using (4.1). In prac-

tical systems we would expect adjacent channels beyond the central zone which makes use of spectrum beyond this zone impractical. The allowance of the masks for spectral leakage beyond the central zones are mainly to account for OFDM sidelobes and amplifier non-linearity.

The capacity for Nyquist signalling is shown for root raised cosine (RRC) pulse designs for excess bandwidth parameters of $\beta = 2/9$ and $\beta = 3/8$. Both of these represent practical single carrier designs over an occupied bandwidth of 22 MHz. The symbol rate for these designs are $R_s = 18$ Msps and $R_s = 16$ Msps respectively. The PSD for RRC $\beta = 2/9$ lies within the b mask but violates the tapering in the central zone of the g mask, while the PSD for RRC $\beta = 3/8$ complies with the g mask.

Finally, the constrained capacity of an OFDM system specified according to the WiFi g standard is plotted as well. It has $N = 64$ over a 20 MHz bandwidth with 52 active subcarriers. The PSD of this system obviously lies beneath the g mask for which it was designed. The constrained capacity curve of this system lies between that of the above single carrier designs.

5.1 The WiFi 802.11g Spectral Mask

First we investigate the use of FTN and OFDM with a more aggressive design to approach the capacity implied by the g mask. Both these techniques are able to close the gap to capacity from the information rates for the single carrier designs and the standard WiFi g implementation.

FTN Approach

The FTN approach to the spectral mask constraint is to pick a pulse shape to shape the PSD to the mask. The PSD is dependent only on the pulse shape, and not on the transmission rate, assuming independent input symbols. In Figure 5.2, we show a transmit pulse (linear-phase choice) whose QAM signal meets the g mask, *independent* of signaling rate. This pulse was obtained by numerical conversion of the mask to the time-domain. The pulse is constrained to a 22 MHz bandwidth since adjacent channels are presumed to exist beyond this bandwidth.

This pulse shape will not in general be a root-Nyquist pulse at any signaling rate, so FTN is somewhat a misnomer in this case. However the same principles apply: By accelerating, or over-clocking the modulator

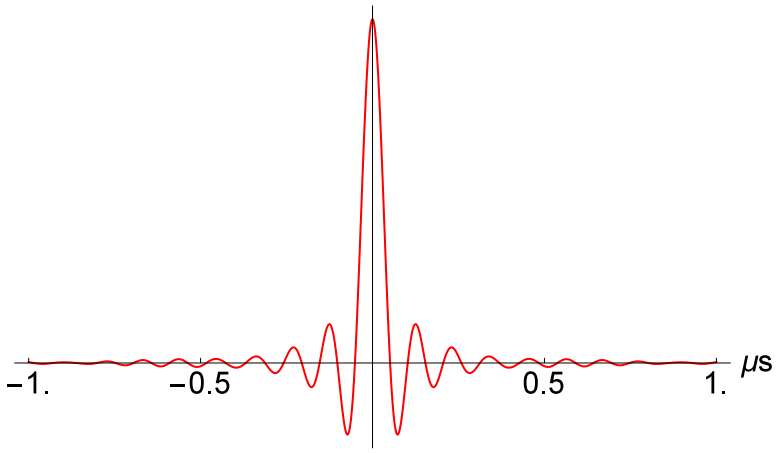

Figure 5.2: Pulse Meeting 802.11g Mask, Suitable for FTN, Truncated to 2 Duration

relative to some typical, small-ISI rate, and properly decoding the resulting ISI, the *G Mask* constrained capacity shown in Figure 5.1 can be obtained with sufficiently large over-clocking factor s. Based on the flat portion of the mask we somewhat arbitrarily assign a nominal rate $R^* = 18$ Msps. Then we overclock this nominal rate by the factor s in attempt to approach the mask constrained capacity. Whatever the signaling rate, a filter matched to this pulse and sampled at the appropriate rate provides sufficient statistics for optimal processing in uncoded or coded scenarios.

The information rate for a Gaussian alphabet using a pulse $p(t)$ with Fourier transform $H(f)$ is [40, 54]

$$I = sR^* \frac{1}{2\pi} \int_{-\pi}^{\pi} \log_2 \left(\frac{P_r}{sN_0R^*} P(e^{j\Omega}) \right) d\Omega$$

where $P(e^{j\Omega})$ is the DTFT of $g[n]$:

$$P(e^{j\Omega}) = \sum_n g[n] e^{-j\Omega n}$$

and $g[n]$ is the autocorrelation of $p(t)$ sampled at the FTN rate sR^*:

$$g[n] = \int_{-\infty}^{\infty} p(t)p(t - n/sR^*)dt$$

Figure 5.3 shows these information rate calculations for various values of s. For $s \geq 22/18 = 1.22$ we find that the information rates have converged to the mask constrained capacity. This corresponds to the fact that the nominal symbol rate is 18 Msps while the bandwidth of $H(f)$ was chosen to be 22 MHz. Similarly, given a root-Nyquist pulse with some excess bandwidth factor β, overclocking by a factor $s = 1+\beta$ is sufficient to attain the constrained capacity limit in bits/second, at least if large alphabet codebooks are allowed [40] [54].

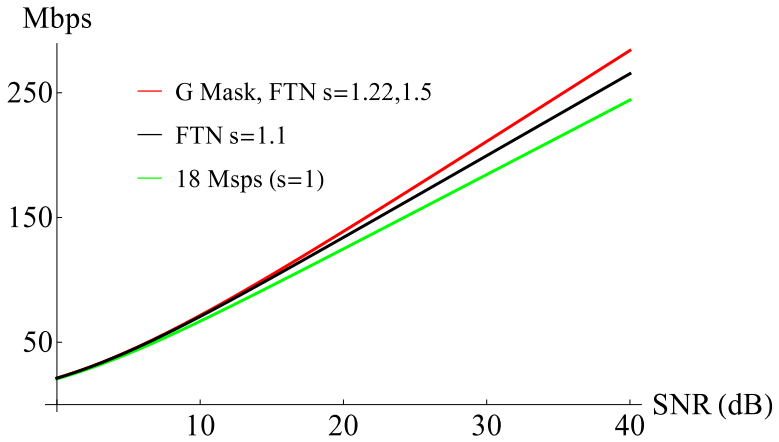

Figure 5.3: WiFi g FTN Information Rates

For high performance decoding of FTN signals, a *soft input soft output* (SISO) equalizer can be used such as the BCJR algorithm which can be combined with an iterative decoder. The standard BCJR algorithm uses metrics that expect white noise. To this end a noise whitening filter can be used before processing by BCJR. The noise whitened channel response $f[n]$ can be calculated from $g[n]$ by minimum phase spectral factorization, and is shown in Figure 5.4 out to 30 coefficients.

Figure 5.5 shows the *signal to residual ISI ratio* (SRISIR) due to truncation of $f[n]$ for use in the BCJR algorithm. The number of states in the BCJR trellis is M^L where M is the number of constellation points and L is the memory order, where $L = N_f - 1$ where N_f is the number of coefficients in $f[n]$. If we would like to operate at an SNR of 30 dB and 40 dB we would could only truncate down to $L = 14$ and $L = 27$ respectively.

The impractical complexity of the full BCJR to approach the con-

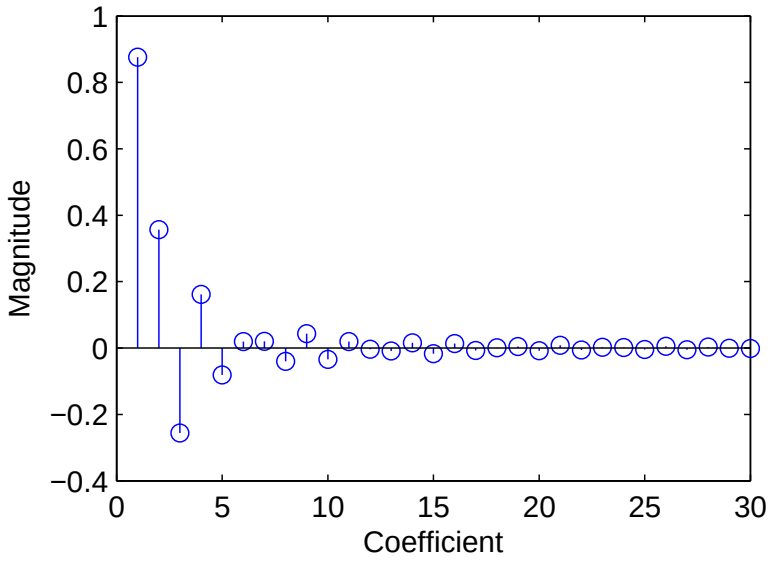

Figure 5.4: Mask Pulse Noise Whitened Channel Response

strained capacity of the g mask is shown in table 5.1. As seen in Figure 5.1, FTN shows its greatest absolute advantage over Nyquist signalling at high SNR. At 30 dB and 40 dB SNR, the FTN information rates are 18% and 19% greater than Nyquist signalling at 18 Msps respectively. At $s = 1.22$ with a symbol rate of 22 Msps, large constellation sizes M are needed to encode the required number of bits. The M shown are minimum sizes as even larger constellations are necessary to accommodate channel coding. It may be possible to increase s and thereby lower the M required, however this induces higher ISI and increases the L necessary to maintain an acceptable SISIR. Clearly straight-forward implementation of BCJR with truncation of $f[n]$ is impractical.

Aside from truncating $f[n]$, there are other ways of reducing the complexity. It is possible to use an impulse-shortening filter, however this will color the noise causing a mismatch with the metrics of the BCJR algorithm and incur a performance penalty [1]. [2] proposes a reduced complexity *M-BCJR* algorithm, but only considers binary alphabets.

Thus straightforward FTN techniques to approach the constrained capacity of the *WiFi-G* at high SNR are impractical, and at lower SNR, the advantages of FTN diminish. While reduced complexity FTN methods are

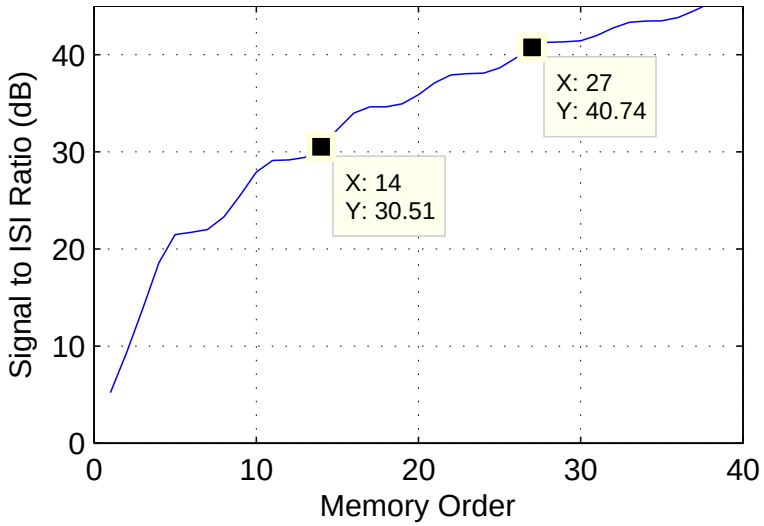

Figure 5.5: Mask Pulse Truncation Signal to Residual ISI Ratio (SRISIR)

Table 5.1: Mask Pulse FTN BCJR Complexity

SNR	Information Rate	Symbol Rate	Bits per Symbol	Minimum Constellation Size M	Memory Order L	BCJR States
30 dB	211 Mbps	22 Msps	9.59	2^{10}	14	1.4×10^{42}
40 dB	284 Mbps	22 Msps	12.9	2^{13}	27	4.6×10^{105}

an active area of research, we will show in the next section that the well understood techniques of OFDM can approach the mask constrained capacity with relative ease.

OFDM Approach

The design for OFDM requires choice of N and the power profile such that the PSD falls under the mask. Typically the first corner in the sidelobe region is the most difficult to meet, and traditional design defines null channels at both band edges so the sidelobe sum in (4.8) is within the mask. (The parameter M is also a design choice, but assuming $M \geq 2$ the effect is rather minor, and is only chosen to help with DT to CT conversion.)

Flat Profile

It is evident from study of (4.8) that for a given design bandwidth in Hz, use of a larger N (and thus larger T_f) is helpful in that the power spectrum for each subchannel becomes narrower. With equal-energy profiles, (what we call flat profiles) this in turn possibly allows more active subchannels to crowd the band edge while still meeting the mask. More active tones obviously increases capacity, for a given overall power, or alternatively, the OFDM PSD is more closely approaching the continuous-frequency PSD mask[1].

When adopting a design bandwidth 22 MHz, Figure 5.6 illustrates the PSD for $N = 64$, 128, and 256 with flat profile. In each case edge channels have been trimmed so that the PSD meets the mask. The allowed *number of active subchannels* in these cases is $N_a = 46$, 104, and 208. N_a can be accurately estimated using (4.18) instead of iteratively graphing (4.11) and manually checking that PSD meets the mask.

Notice that with flat profile designs, the ratio N_a/N cannot increase when N exceeds 128, as meeting the first (-20 dB) mask corner dominates the constraint. In other words, once the density of subcarriers reaches a critical value, the continuous-frequency PSD no longer changes, and fixed-interval guard bands must be maintained so that the sum of subchannel sidelobes meets the mask[2].

Tapered Profile

The fact that the mask is not vertical in the main-lobe region allows tapering of the energy profile near the band edges to admit even more active subchannels while still meeting the mask. Numerical maximization of the information rate with the mask PSD constraints can be used to determine the tone power allocations. Due to the symmetry of the mask, half of the tones $N/2 - 1$ representing the positive frequencies were subject to the maximization. The constraints were taken at steps of half the tone spacing $\Delta_f/2$ from 0 to 13 MHz. Beyond 13 MHz it is assumed that a transmit filter would be employed to guarantee compliance to the mask.

[1] We note that attaining the constrained capacity also requires near-Gaussian (large alphabet) constellations for high rates

[2] The mask constraint is not met at ± 20 MHz frequencies, but this can be easily solved with a non-distorting digital filter following the IFFT

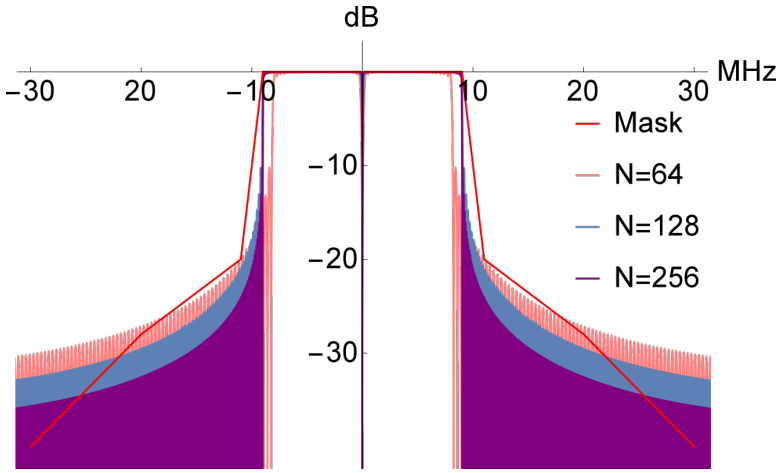

Figure 5.6: PSD for DT OFDM Signal with $N = 64$, 128, 256, $M = 4$, Flat Profile

It turns out that the optimal power profile is a function not only of the mask but also the SNR. Figure 5.7 shows the optimized PSD profiles of an $N = 64$ OFDM signal. The peak PSD of the mask has been normalized to unity. Without loss of generality, interpreting the mask as the PSD constraints at the receiver, optimization for two extreme SNR cases are shown: one with a noise power spectral density of $N_0 = 0$ dB (SNR = -1.44 dB) and $N_0 = -40$ dB (SNR = 38.0 dB)[3].

While water-filling gives the solution to maximizing the information rate subject to an average power constraint [12], here the constraint is not power but rather the PSD as stipulated by the mask. Furthermore water-filling does not involve a tradeoff between bandwidth and power, and is free to allocate a fixed amount of power over the entire bandwidth. Maximization of information rates subject to a mask on the other hand involves a power versus bandwidth tradeoff that is a function of SNR.

The tradeoff between power and bandwidth in relation to SNR can be understood in terms of operation at high SNR in the *bandwidth limited region* where the bitrate $R > B$, and at low SNR in the *power limited region* where $R < B$ [38]. The log function in (4.2) results in diminishing returns as power is increased. At high SNR in the bandwidth limited region, these marginal gains are so low that trading off power for bandwidth becomes

[3]The noise power in the SNR calculation is taken as $N_0 B$, where $B = 22$ MHz

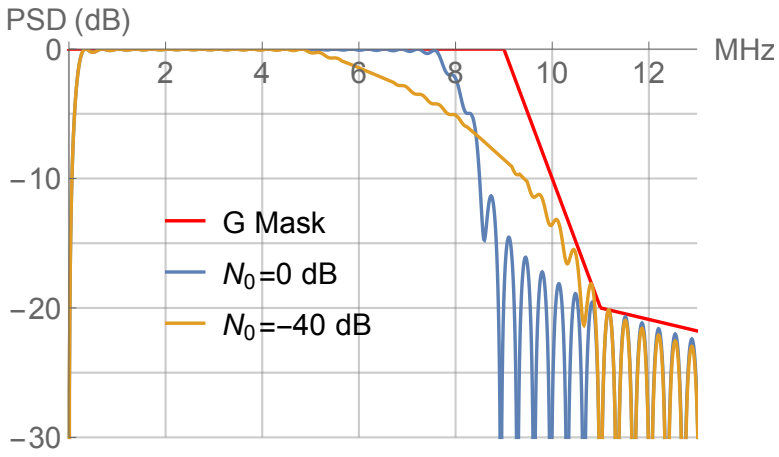

Figure 5.7: OFDM $N = 64$ $M = 4$ PSD for Tapered Low SNR ($N_0 = 0$ dB) and High SNR (N_0 = -40 dB) Profiles

attractive. Similarly at low SNR in the power limited region trading off bandwidth for power results in higher performance. The optimization of power profiles subject to a mask constraint involves these tradeoffs.

Notice that in the high SNR profile the optimization results in a higher bandwidth usage at the cost of power. The power output of the high SNR profile is 0.52 dB below the low SNR profile, and 1.3 dB below the maximum power implied by the mask. Table 5.2 summarizes the relative powers of the different profiles considered in this section.

Table 5.2: Relative Power of Tapered Profiles

N	Profile	Power Relative to Mask
64	Low SNR	- 0.77 dB
64	High SNR	- 1.30 dB
128	Low SNR	- 0.07 dB
128	High SNR	- 0.15 dB

The low SNR profile favors greater power over bandwidth. It forgoes the spectrum allowed at the lower corners of the mask since using this area of the mask precludes power allocation at the upper mask corner. At low SNR, any power allocated to the lower mask corner is buried far below the noise floor, and little is lost by excluding this allocation. Allocating power

to the upper corner increases the total power since the mask allows greater power here than at the lower corner. Furthermore this actually increases the effective bandwidth seen at the receiver since the power density at the top of the mask is close enough to the noise PSD to make a difference, whereas the high SNR profile drops the bandwidth across the top to allocate power to the lower corner.

The resulting channel capacity for different power profiles is easily computed by taking the capacity available per orthogonal subchannel, and summing over the active subchannels, accounting for any possible taper on the energy profile. The capacity plots below assume Gaussian 2-D variables for the input message symbols to the IFFT, but this can be approached at higher rates with progressively larger alphabets.

Plotting the information rate verses actual SNR shows that rates for the high SNR profile is always better than the low SNR profile at a given SNR. However this is not a fair comparison since the optimization constraints were not based on average power but rather the mask constraint. We want to allow certain profiles to tradeoff bandwidth for power without penalizing these optimizations. To this end we compare the two profiles based on on a nominal SNR:

$$SNR_{nom} = \frac{\int_{-11\text{MHz}}^{11\text{MHz}} M(f)df}{N_0(22\text{MHz})} \tag{5.1}$$

where the signal power is taken to be the maximum power implied by the mask PSD $M(f)$. Since everything is constant in this equation except N_0, this amounts to comparing the two profiles at the same noise PSD. This could also be interpreted as comparing the two profiles that are constrained to the same mask PSD at the transmitter and received over the same pathloss. Figure 5.8 shows this plot. Notice that at low SNR, the low SNR profile performs ever so slightly better. However the differences are obvious at high SNR.

The differences between the high and low SNR profiles diminish as N is increased as the optimization is able to better hug the mask with smaller tones. Figures 5.9 and 5.10 show similar profiles for $N = 128$. The power versus bandwidth tradeoff is still evident between the upper and lower corners, but their effect on information rates is less pronounced than at $N = 64$. In fact the low SNR profile is only 0.08 dB greater in power than the high SNR profile. At $N = 256$ there is hardly any difference between high SNR and low SNR optimizations.

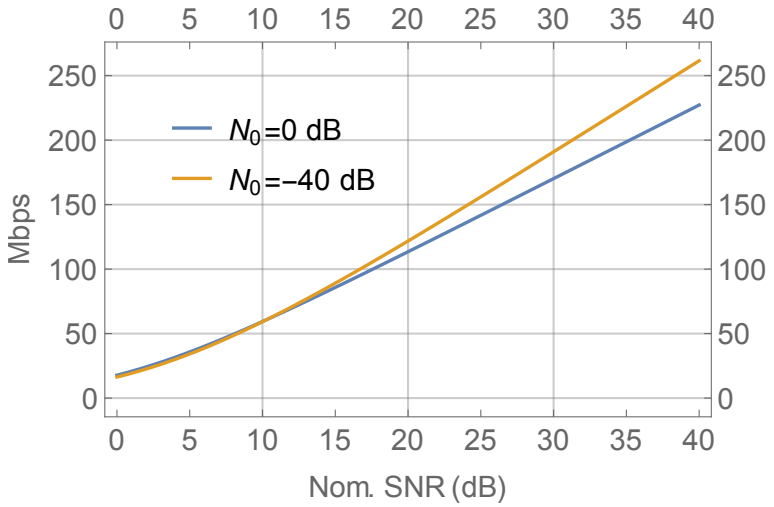

Figure 5.8: OFDM $N = 64$ $M = 4$ Mask Constrained Capacities for Tapered Low SNR ($N_0 = 0$ dB) and High SNR (N_0 = -40 dB) Profiles

Figure 5.11 compares the performance of the flat and tapered profiles under the g mask. Only the high SNR profiles are shown for the tapered cases. The information rates are again plotted against the nominal SNR in (5.1). Note the superiority of the tapered designs, with the increased slope of the tapered OFDM capacity reflecting the more efficient use of the mask-constrained spectrum. The tapered $N = 64$ design (optimized for high SNR) performs better than even the flat $N = 128$ profile at high SNR. Even at a modest levels of $N = 128$, the optimized tapered profile has approached the mask constrained capacity. Advantages of lower values of N include lower complexity and more immunity to Doppler frequency shift due to motion.

For high SNR, the increase in achievable information rate is about 20% in a fair power and noise level comparison. Flat profiles benefit from increasing N to 128, but no further gains are possible for this mask; with this design the number of active subcarriers must be kept significantly less than N, however, to meet the mask, reflected in the poorer capacity in 5.11. There seems no good reason not to adopt a more aggressive tapered OFDM recipe, since the spectral emission constraint is satisfied.

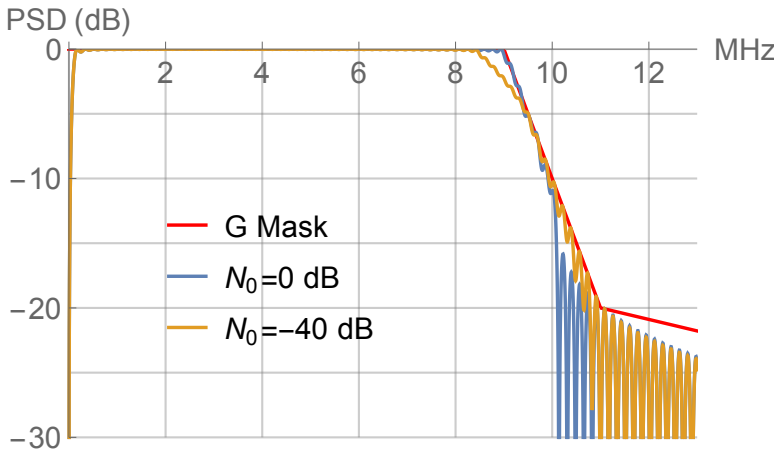

Figure 5.9: OFDM $N = 128$ $M = 4$ PSD for Tapered Low SNR ($N_0 = 0$ dB) and High SNR ($N_0 =$ -40 dB) Profiles

5.2 The WiFi 802.11b Spectral Mask

FTN Approach

For the FTN case the b mask implies the sinc function. Speeding up the sinc pulse beyond the Nyquist rate does not lead to higher information rates. Furthermore difficulties in using the sinc pulse are well known. This leads to the adoption of an excess bandwidth pulse shape such as the RRC in practice. Thus FTN is not a good approach for this spectral mask.

OFDM Approach

OFDM is a practical way to approach the capacity implied by the mask. As with the g mask, we first explore the flat profile. Then we go on to the numeric optimization techniques which results in tapered profiles that tradeoff power and bandwidth at different levels of SNR.

Flat Profile

In deriving flat profiles for the b mask, we can use the polygamma formulation of the OFDM sidelobe envelope in Section 4.3. We calculated that approximately 101 subcarriers need to be deleted from each side such that

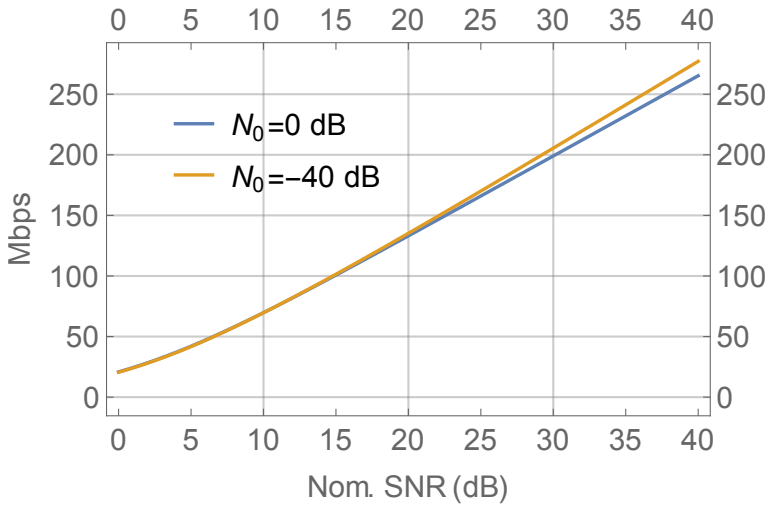

Figure 5.10: OFDM $N = 128$ $M = 4$ Mask Constrained Capacities for Tapered Low SNR ($N_0 = 0$ dB) and High SNR ($N_0 =$ -40 dB) Profiles

the sidelobes fall beneath the -30 dB corner of the b mask. This number of sidelobe deletions holds for large N. This suggests that the constrained capacity increases as a function of N since the percentage of sidelobes deleted decreases as N increases. The fraction of active subcarriers due to the deletion of 204 subcarriers[4] is tabulated as a function of N below.

N	Fraction of active subcarriers
1024	0.801
2048	0.9
4096	0.95
8192	0.975
16384	0.988
32768	0.994

The fraction of active subcarriers also represents the fraction of the mask constrained capacity attained by the OFDM modulation. Thus OFDM with large N is able to approach the mask capacity. $N = 32768$ is an FFT size that is already being used in the DVB-T2 standard.

[4] 101 subcarriers deleted from both sides, and tones at DC and the folding frequency are not used

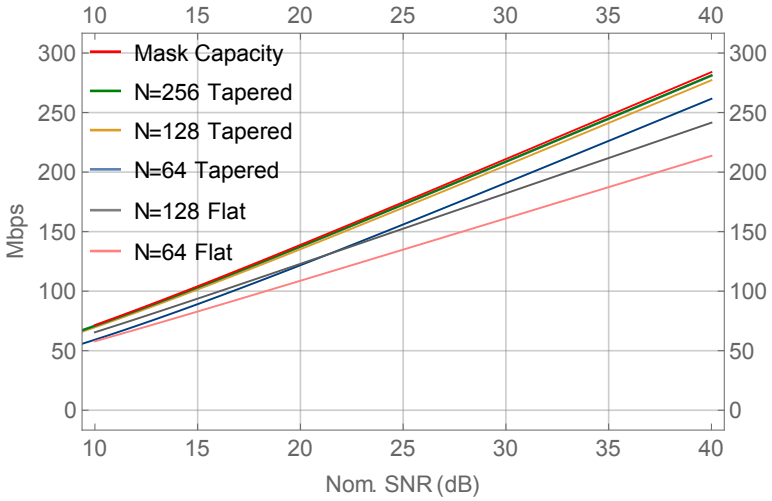

Figure 5.11: Capacity Comparison for OFDM Designs

Tapered Profile

While the mask resembles a brick wall spectrum, optimizing the information rate subject to the mask at different SNR as done in Section 5.1 still results in a tapered profile that features a tradeoff between power and spectrum. In fact these tradeoffs are even more pronounced than for the b mask. Repeating the same optimizations for $N = 64$ and $N = 128$ we obtain the results in Figures 5.12, 5.13, 5.14, and 5.15

The tradeoff of power for spectrum at high SNR is evident which results in a similar vacancy of the upper corner for occupancy at the lower corner, resulting in greater bandwidth. Table 5.3 shows the relative power of each of the profiles with respect to the full power implied by the mask. Compared to the relative powers of the less aggressive G Mask in Table 5.2, these results are much lower. The trend of profiles converging to the mask power as N increases can be seen here, but it is clear that much higher N is needed for convergence. The convergence of the low SNR and high SNR profiles with increasing N is also apparent.

While the results from this section and Section 5.2 show that the B Mask capacity can be approached with increasing N, in the following section we propose some additional techniques that could be investigated in future research.

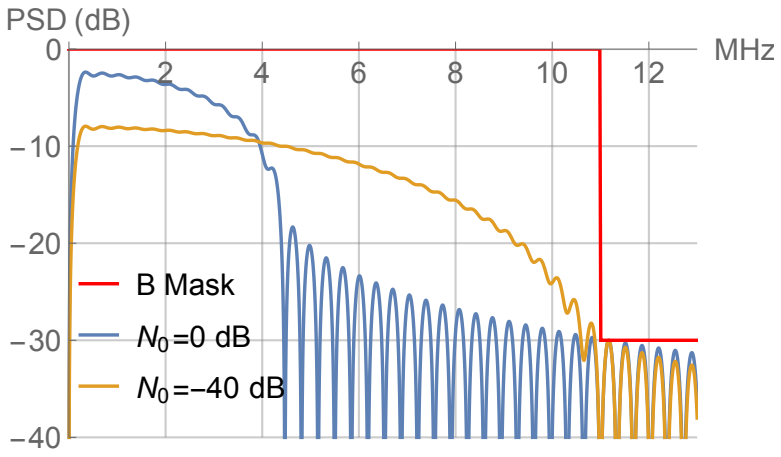

Figure 5.12: OFDM $N = 64$ $M = 4$ PSD for Tapered Low SNR ($N_0 = 0$ dB) and High SNR (N_0 = -40 dB) Profiles

Table 5.3: Relative Power of Tapered Profiles

N	Profile	Power Relative to Mask
64	Low SNR	- 8.53 dB
64	High SNR	- 11.19 dB
128	Low SNR	- 5.73 dB
128	High SNR	- 8.27 dB

5.3 Future Research

Deleting tones at the edges in the flat profiles is perhaps the most simple way of mitigating OFDM sidelobes. We also have explored tapered profiles that converge to the mask constrained capacity more quickly. In addition to avoiding spectral leakage outside of the mask, it is also possible to filter the sidelobes at the transmitter.

A filter with a flat gain and linear phase over the desired signal bandwidth can filter the sidelobes while passing most of the signal undistorted. However OFDM is well suited to equalize linear distortion, this being perhaps one of its greatest features. Thus a general filter could be used over OFDM signal which can be combined with tone scaling to meet some mask constraint.

This more aggressive filtering will introduce some inter-tone interfer-

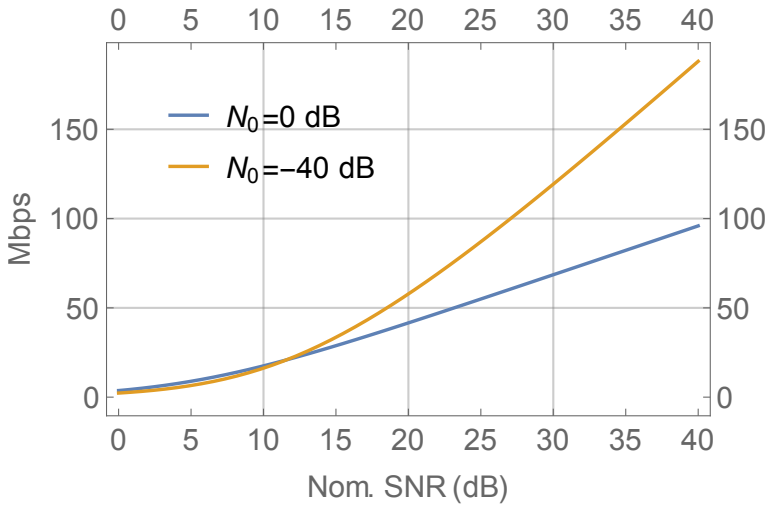

Figure 5.13: OFDM $N = 64$ $M = 4$ Mask Constrained Capacities for Tapered Low SNR ($N_0 = 0$ dB) and High SNR ($N_0 =$ -40 dB) Profiles

ence. However standard OFDM equalization can be used to combat this distortion and a CP the length of the filter impulse response would ensure good performance of the equalizer. The overhead of the CP can be mitigated by increasing N or even using a channel shortening filter as done in digital subscriber lines. A potential problem with channel shortening in single carrier communications such as in Nyquist and FTN signalling is the coloring of the noise which creates a mismatch with the well known receiver structures that assume white noise (see Section 5.1). However noise coloring is not a problem with OFDM, again due to the ease of equalization. There are also even more advanced techniques where sidelobes are canceled [11].

Thus OFDM has a whole host of properties and techniques that can be exploited to approach the capacity under a mask constraint. While we have shown that tapering at even modest N can attain to the mask capacity of the g mask, these other techniques could be explored for the more aggressive brick wall allocations like the b mask. The combination of these techniques with our method of optimized tapering profiles may allow convergence to the mask capacity at lower N than shown with the flat profiles in Section 5.2.

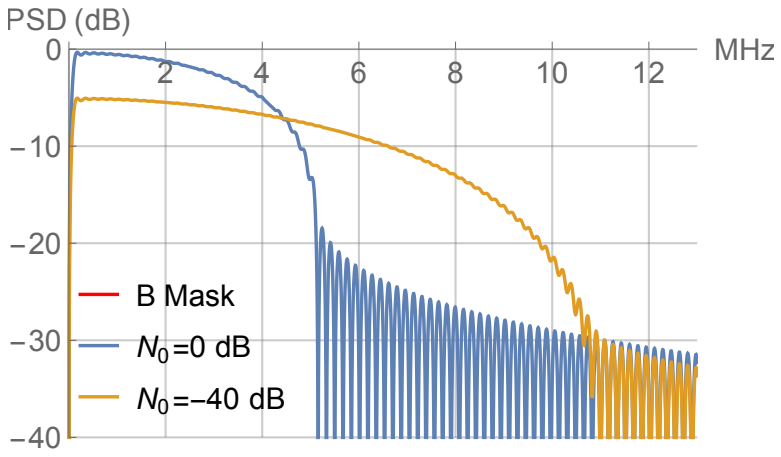

Figure 5.14: OFDM $N = 128$ $M = 4$ PSD for Tapered Low SNR ($N_0 = 0$ dB) and High SNR ($N_0 =$ -40 dB) Profiles

5.4 Conclusion

In meeting a spectral mask constraint, two possible transmission strategies are FTN (single-carrier QAM) and OFDM with aggressive energy profile and sufficiently large FFT size N that sharpens the spectral rolloff to help meet mask constraints. Considering complexity of implementation, especially the receiver complexity and synchronization difficulties of FTN, it seems clear that OFDM is a preferred choice. We believe this conclusion holds for other masks than the ones considered here.

For OFDM, large N , combined with power tapering, is suggested by our study, which has a small (factor of $\log_2 N$) complexity penalty per message symbol. Also, larger N will increase the peak-to-average power ratio and the sensitivity to Doppler offset. Nonetheless solutions exist for both. On the other hand, larger N helps amortize the cyclic prefix overhead (not considered here). As usual, channel equalization for non-ideal channels is greatly simplified with OFDM. Finally, coupling channel coding with OFDM is much easier as well.

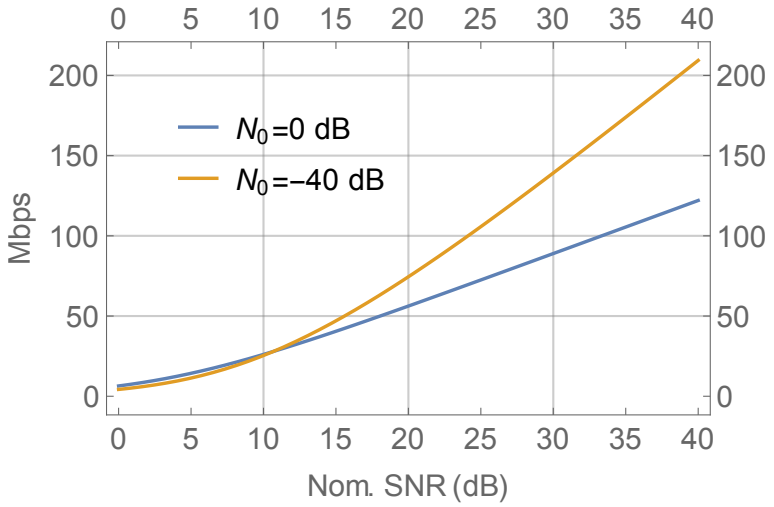

Figure 5.15: OFDM $N = 128$ $M = 4$ Mask Constrained Capacities for Tapered Low SNR ($N_0 = 0$ dB) and High SNR ($N_0 =$ -40 dB) Profiles

Part III

Application Specific Optimization

Chapter 6

Introduction

The optimization of communication systems involves the tradeoff of complexity for efficiency. Power and spectral efficiency are two common goals in communications engineering. Given limitations in power and spectrum, we would like to maximize the amount of information transfer over the system. This of course comes at a cost in complexity, and the greatest performance at the least marginal cost in complexity is of interest.

In the past, improvements at the physical layer have been responsible for most of the gains in communications efficiency. The integrated circuit was invented in 1958, and since then Moore's Law has accurately modeled exponential growth in computing power. As computing power has become cheaper, more elaborate and complex signal processing schemes were developed to push physical layer performance closer to the Shannon limit, which dictates the absolute limit in the efficiency of reliable communications. Modern error correcting codes such as *Low-density Parity Check Codes* (LDPC) and Turbo Codes have been able to approach within 1 dB of the Shannon limit. This monumental achievement raises the question of how to continue trading off complexity for communications efficiency, as the fundamental limit bars any further significant increase in efficiency.

While advances in the physical layer have approached the fundamental limit, it is well known that the layered approach to protocol design and the end-to-end principles of Internet architecture, while providing benefits of modularity and abstraction, limit communications efficiency [4]. In particular protocol layering precludes *application specific optimization*[1] where the

[1]We need to distinguish between the word *application* with respect the application

functionality traditionally associated with the lower layers can be optimized based on the application.

Some of the performance enhancing techniques made possible by ASO are:

Optimizing Channel Coding, Modulation, and Media Access Control

Functions of the physical layer such as channel coding and modulation as well as the media access control (MAC) can be tailored to the application, and thereby achieve performance and efficiency gains. For example, streaming recorded video should be handled differently than gaming traffic. Recorded video streaming is a high bit rate application that also has a high tolerance to latency, while real-time gaming typically has relatively low data rates but demands low latencies. This suggests the use of a channel code with long codewords for video streaming, while the gaming traffic would use shorter codewords. This could be implemented in an orthogonal frequency division modulation (OFDM) system where most of the tones would be allocated high throughput traffic where the codeword spans multiple OFDM symbols, while a few tones could be dedicated to low latency traffic that could be decoded at the reception of each OFDM symbol. Furthermore the *media access control* (MAC) function could intelligently schedule the traffic and tone allocation based on the application requirements.

Optimizing Source Coding

Another important feature of ASO is that the source coding at the application layer[fn::In information theory, can be adapted to the channel conditions known at the physical layer. Today's Internet video streaming depends on the client software to estimate the speed of the connection to decide between different bitrate representations of the video. Poor estimates of the bandwidth causes content already downloaded to be wasted as the client switches between different representations. In a study of mobile clients more than 35% of the downloaded content was discarded [31, 30] due

layer of a protocol stack and a *software application* which may be a commercial software package. Later we will consider the partitioning of a software application as a means to implement ASO. At present we are concerned with the optimization of different protocol applications such as voice, video, and web traffic.

to this reason. In an ASO system, the source coding can be dynamically adjusted with intimate knowledge of the channel conditions and this kind of waste could be avoided.

Joint Source Channel Coding and Decoding

Joint source channel coding (JSCC) is the joint design of the source coding and channel coding. Typically these two functions are designed independently. Shannon's separation theorem states that the optimal performance theoretically attainable can be achieved by using optimal source coding independent of optimal channel coding. However optimal source coding and channel coding is difficult, and a joint design can result in less complexity [14]. Furthermore this separation results in designs that are not resilient to occasional errors. An example of joint source channel coding is unequal error protection in video where more important parts of the compressed data are protected with more redundancy in the channel coding.

Joint source channel decoding (JSCD) is closely related. Practical compression algorithms leaves residual redundancy at the output of the source coder which can then be exploited by the decoder in conjunction with the injected redundancy of the channel coding. This can be implemented as part of an iterative decoder which is used in modern error correcting codes.

Thus ASO allows the functions of the lower layer to adapt to the application at the highest layer, and also allows the source coding at the application layer to be tuned to the channel conditions known at the lowest layers. It also allows for JSCC and facilitates JSCD which can provide further performance and efficiency gains not yet seen in practical systems.

6.1 Background Information

In this section we discuss access networks and protocol layering, giving a motivation and background to ASO.

Access Networks

Access networks such as digital subscriber line (DSL), digital cable, and cellular wireless are the bottlenecks to faster broadband Internet access. This challenge has been called the *last mile problem* as the access network is responsible for the last leg of the transmission link from the core network

to the end terminal. The medium for access networks is predominantly based on copper lines and wireless transmission, which have less bandwidth than the core network which is based on fiber optic communications. There is excess capacity in the core network [22], and the core network can handle more traffic than the access network can generate or receive. Therefore faster Internet speeds depend on better access network performance. *Our ASO solution focuses on improving the efficiency of the access network.*

TCP/IP and Protocol Layering

TCP/IP is the dominant networking protocol today. Like many other communications protocols, it is based on a layered architecture. The layered architecture is not unique to protocol design. The concept of layering in telecommunications protocols was probably borrowed from operating system architecture. It is simply a method to implement abstraction and encapsulation in the design of complex systems, and is not central to the solution of the engineering problem. In fact object-oriented design – another means of providing modularity and abstraction – superseded layering in operating system research and development.

In a *layered* architecture, layers are only allowed to interact directly with the layers immediately above or below it. The advantage of such a design is so that layers can be designed, implemented, and tested more independently of one another. Although this provides convenience in the design and verification process, it also results in a performance degradation due to overhead and preclusion of ASO. Because the lower layers are separated from the highest level application layer, the network which only operates on the lower protocol layers does not know what application is related to the data it is transporting. Because it does not know the application, it cannot optimize the traffic accordingly. That is it cannot perform ASO. This shortcoming of the network architecture was well understood from the outset of the Internet. The *end-to-end argument* [15] promoted a "dumb" network solely responsible for routing packets the their destination without any knowledge of the application. This was a conscious engineering trade off to simplify the network.

While this architecture has worked well to form the Internet over disparate link layer technologies, the ever increasing demand for higher data rates has drawn attention to these inherent inefficiencies. Research such as cross layer optimization, JSCC, and JSCD attempt to address these issues.

Surprisingly, while these inefficiencies are widely known to exist, there is scarce data quantifying how much could be gained by ASO. This may be because of the widespread opinion that TCP/IP has become entrenched as the protocol of the Internet, and there is nothing that can be done. Our research challenges this notion.

As will be explained, there is a confluence of technologies and trends that will allow such optimization to be made over both wireless and wireline access networks. These trends across computing and networking will enable ASO while still supporting backward compatibility to the ubiquitous TCP/IP protocol.

6.2 Organization

We present a holistic and multidisciplinary approach to ASO. Our concern is not just ASO in and of itself, but also how such optimization can be done and yet maintain backward compatibility with the Internet. Furthermore we consider how these techniques can reach economies of scale necessary for successful commercialization and adoption.

In Section 7 we consider practical methods to circumvent protocol layering over the access network, while maintaining Internet connectivity through support of TCP/IP. While most of the literature on cross layer design seeks to modify the existing protocol stack, we consider approaches [4] which completely eliminates the existing stack over the access network.

Three methods are investigated here: 1) the *performance enhancing proxy* (PEP), 2) new ASO *application programming interfaces* (API), and 3) subscriber and application partitioning across the access network. The subscriber partitioning method of circumventing protocol layering is to our knowledge a novel application of data centers and network function virtualization.

In Section 8 we consider an ASO enabled replacement of TCP/IP over the access network. However this is not a formal proposal of a new protocol. Instead, in Section 8.2 we consider the possibility that the time may be ripe for a turning point in telecommunications history where strictly defined protocols give way to a new age of malleable software defined protocols. While active networks and the notion of programmable networks have been in the literature since the 90's [16], recent trends in *cloud computing* and

network function virtualization (NFV) may pave the way to translate these old ideas into new practices for ASO.

Finally in Section 9 we analyze the potential performance improvement that ASO could provide to the ADSL access network in the case of recorded video streaming, which comprises the majority of Internet traffic.

Chapter 7

Practical Approaches to ASO

In this section we investigate architectures that provides ASO and other benefits by circumventing protocol layering over the access network, while maintaining Internet connectivity through support of TCP/IP. While most of the literature on cross layer design seeks to modify the existing protocol stack, we propose a revolutionary approach [4] which completely eliminates the existing stack over the access network.

7.1 Approach 1: Performance Enhancing Proxies

Performance enhancing proxies (PEP) [8] can be used to intercept and optimize TCP/IP traffic destined for an endpoint. This technique is used, for example, in satellite communications where the long latency due to propagation delay causes problems with the TCP acknowledgment mechanism. Figure 7.1 shows a PEP situated between the network and the satellite link. The PEP is used to send acknowledgments back to the Internet while the satellite link uses a modified protocol to account for the latency. This is known as a split connection implementation of a PEP.

By using a similar topology, a proxy located in the infrastructure side of the access network can use a full protocol stack to decode the application using techniques such as *deep packet inspection*. Deep packet inspection is a technique which allows an intermediate node in the route to decode

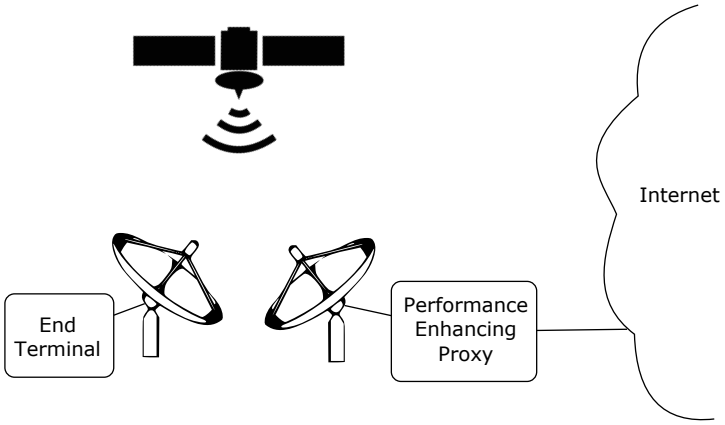

Figure 7.1: Performance Enhancing Proxy

information at the session and application layers of the TCP/IP stack[1]. This is done by implementing parts of the stack on the intermediate node. With knowledge of the application, the proxy can allow ASO over the access network.

Figure 7.2 shows a radio access network with a performance enhancing proxy which is able to decode the application of the traffic intended for the mobile terminal (MT). The software on the terminal would need to be modified to receive the optimized communications as well as translate the inbound traffic from its optimized form back to normal TCP/IP interface that the software applications expect.

This solution violates the end-to-end principle of TCP/IP since it involves breaking the TCP/IP link into two segments: one from the terminal to the proxy, and from the proxy to the destination in the Internet. However this is unavoidable if ASO is to be incorporated.

Another issue is the termination of encryption at the proxy. Thus if the proxy is compromised, then the communications link is no longer secure. It is however possible to encrypt the link from the proxy to the terminal where the two segments are encrypted separately.

[1]Typically an intermediate node only operates up to the network layer

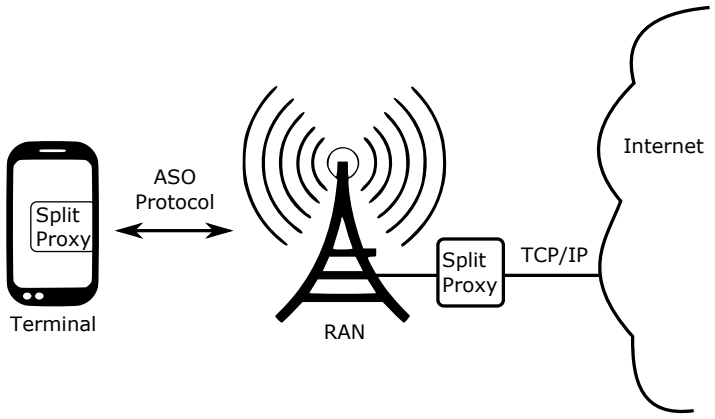

Figure 7.2: Radio Access Network with PEP

Custom Access Network Protocol

It is possible to use a custom protocol in the link from the proxy to the terminal, while maintaining backward compatibility since the link from the proxy to the Internet destination is still TCP/IP. In Figure 7.2, the wireless protocol does not need to be restricted to TCP/IP. It could be a custom ASO protocol. Since the bottleneck is in the access network, we should perform ASO over this last link. Since the proxy is in the infrastructure of the access network, the communications between the proxy and the Internet destination remains *unoptimized* TCP/IP which is not critical since it is over the high speed optical core network[2]

Video Optimization

In the case of video optimization, the proxy would be responsible for identifying video traffic and potentially transcoding it for optimized transmission over the access network. Currently *HTTP Adaptive Streaming* (HAS) is the dominant means to transport video over the Internet. The proxy could

[2]Here we assume that the Internet destination is a commercial server residing on either the core network or at least an optical access network, and not on a slower copper based or wireless access network. This is typical of commercial servers and content delivery networks.

detect HAS traffic and then transcode it for optimal transport. Once the application is known, the techniques listed in Section 1 can be applied.

Challenges and Problems with the PEP Approach

In the case of video optimization the PEP proxy would need to implement a full TCP/IP protocol stack as well as parts of the client application, such as video decoding if transcoding is done. The complexity of the proxy would probably be on the same order of magnitude as the subscriber endpoint itself. However the proxy could be implemented using cloud computing and virtualization techniques where a virtual machine could be dedicated as a proxy for the endpoint. However given this architecture, we propose in Section 7.3 a better approach to ASO.

Another drawback to this PEP approach is the need to convert the ASO traffic back to TCP/IP at the subscriber terminal for existing TCP/IP based software applications. Some optimization is inevitably lost in this conversion. For example, if the subscriber is using HAS for video streaming, the software application is responsible for estimating the network speed and selecting the video bitrate. There is not a direct mechanism for the channel conditions to be relayed to the application. Thus to take full advantage of ASO, software applications need to use ASO APIs. This is described in the following section.

7.2 Approach 2: ASO APIs

Instead of using deep packet inspection in a PEP, another way for the network to discern the application is the use of ASO APIs that application developers could use to take advantage of ASO. Figures 7.3 and 7.4 shows the use of ASO APIs at the infrastructure side and subscriber side respectively.

The use of these APIs by software applications would signal to the network requests for particular ASO transmission schemes. In addition, the Internet destination could also use a corresponding API to optimize the source coding right at the origin. This would avoid the need to transcode at the proxy, which would reduce the burden on the infrastructure as well as provide some ASO benefit over the core network.

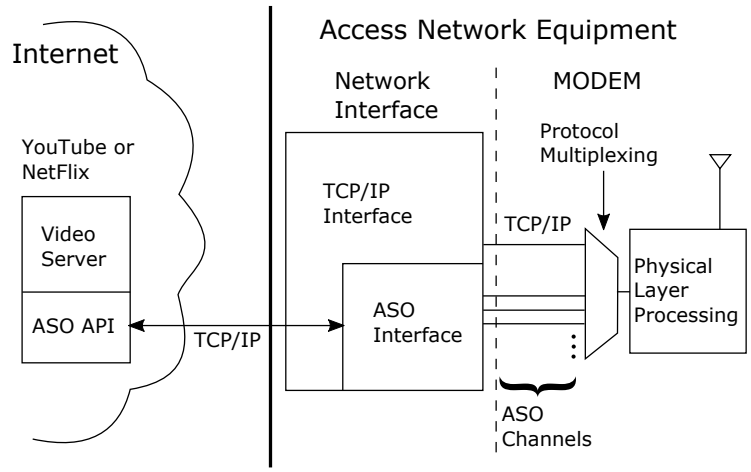

Figure 7.3: ASO APIs at the Infrastructure Side

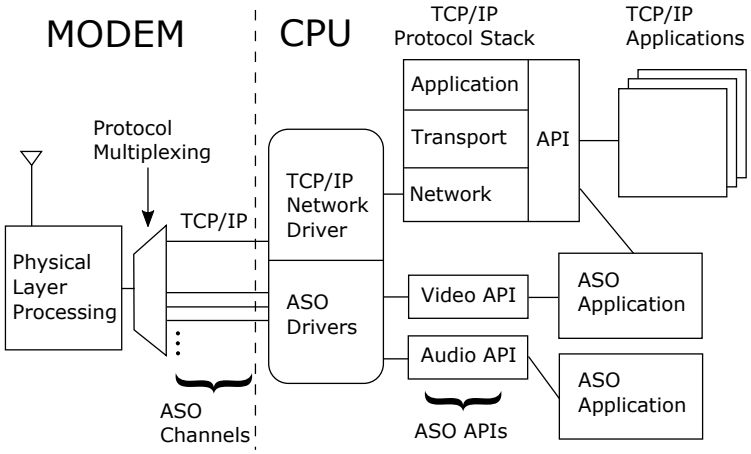

Figure 7.4: ASO APIs at the Subscriber Side

Protocol multiplexing could be used to support TCP/IP concurrently with ASO protocols. An example of protocol multiplexing can be seen in the *EtherType* field of the Ethernet frame where different network layer protocols can be specified over the same link. Here, traditional TCP/IP and different ASO protocols enhancing different applications can be concurrently supported.

Challenges and Problems with the ASO API Approach

There is a question of whether a service such as YouTube or Netflix would be motivated to use ASO APIs on their servers to optimize video delivery to a particular access network. While these services may be less motivated, *content delivery networks* (CDN) who provide their services to the access networks, would perhaps be willing to support this optimization[3] There is an interesting solution to this problem if cloud computing techniques were used to provide the ASO interface. In this case the CDN could be collocated in the same data center as the ASO interface. In this case the access service provider could run its own CDN using the ASO APIs.

A major drawback to the ASO API approach is that software applications seeking to improve would need to be rewritten to use the APIs. The following Section 7.3 seeks to address this issue.

7.3 Approach 3: Subscriber and Application Partitioning

This technique of ASO addresses some of the weaknesses in the more straightforward approaches to ASO considered above. As will be explained in Section 8.2, this approach leverages some recent trends in *cloud computing*, *software defined networks* (SDN), and *network function virtualization* (NFV).

Here we propose partitioning the subscriber and user applications across the access network infrastructure. This is known in the literature is known as *mobile cloud computing* (MCC) where virtual machines in the cloud assist resource constrained mobile terminals [25, 42]. Although the idea of MCC was originally conceived for wireless terminals, we apply the same concept to other access networks to support ASO.

Mobile Cloud Computing

Figure 7.5 shows the architecture of a typical cellular network. The cellular network interfaces with the Internet using TCP/IP, and the TCP/IP endpoint is located at the MT.

[3]We thank Professor Kamin Whitehouse for this idea

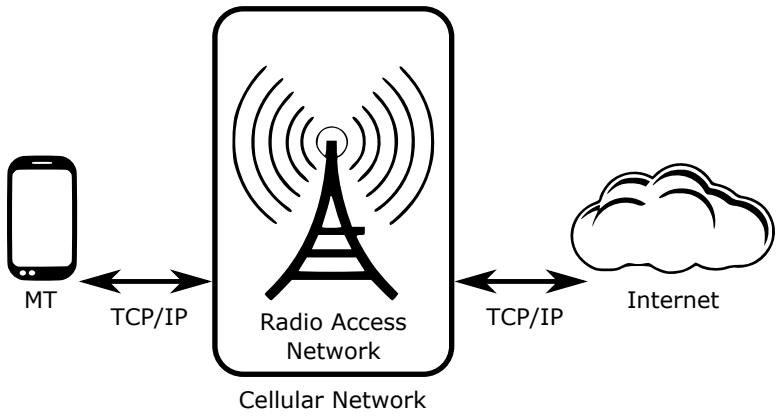

Figure 7.5: Traditional Cellular Network Architecture

Figure 7.6 shows a cellular network that is assisted by a mobile cloud, with the link between the terminal and *mobile cloud* (MC) still being traditional TCP/IP. The advantages of MCC stem from computational offloading: better battery life for the end terminal, faster execution in the cloud, and potentially less traffic over the access network since traffic from the MC to the Internet does not pass through the access network. Thus the subscriber unit now consists of an end terminal and a virtual machine component running in the MC joined together through the access network and Internet.

Mobile Cloudlet for ASO

While the literature regarding MCC typically construes the communication between the terminals and cloud as TCP/IP, this is not necessary if the cloud is collocated with the access network. A *cloudlet* perhaps more accurately describes this configuration. This definition is given by [42]:

> A cloudlet is a trusted, resource-rich computer or cluster of computers that's well-connected to the Internet and available for use by nearby mobile devices.

Figure 7.7 shows a radio access network that has ASO facilitated by with mobile cloudlet assistance. By partitioning the subscriber across the access network and placing the TCP/IP stack in the infrastructure in the

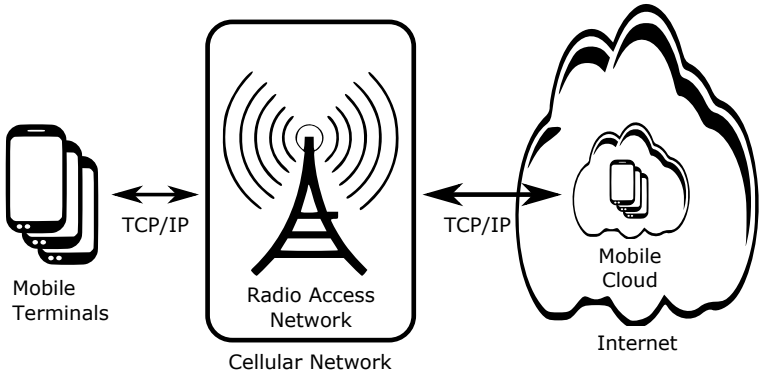

Figure 7.6: Mobile Cloud-Assisted Cellular Network Architecture

mobile cloudlet, we are free to perform ASO over the access network. To the rest of the Internet, this device appears to be a typical TCP/IP endpoint. However all the techniques of ASO can be implemented over the access network for performance gains.

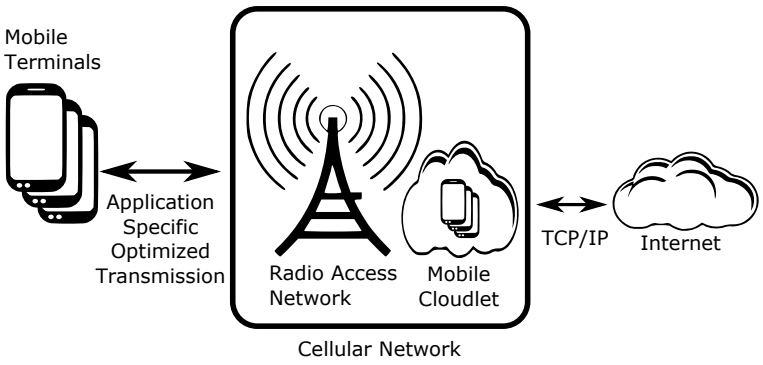

Figure 7.7: Application Specific Optimization with Mobile Cloudlet

The Cloud Radio Access Network (C-RAN)

This combination of an access network with a cloudlet is already happening with the *cloud radio access network* (C-RAN), which has advantages aside from ASO [55] which are discussed in Section Network Functions on General Purpose Computing Hardware. In the C-RAN architecture, the baseband processing traditionally done in the base stations (BS) are pushed further back into the network infrastructure.

Figure 7.8 shows a simplified diagram of a traditional radio access network (RAN). The antennas are connected to the base station by what is called fronthaul. Then there is a backhaul carrying user traffic to the access network provider's core network. Thus the base station which does the baseband signal processing is located at the edge of the network.

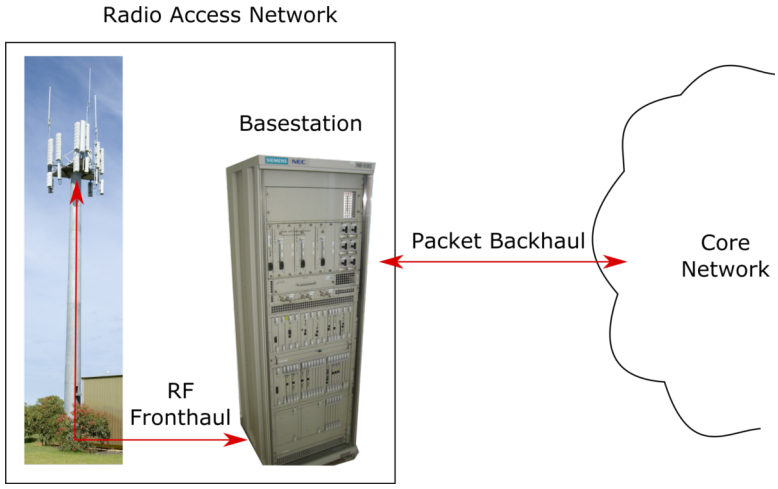

Figure 7.8: Traditional Wireless Network Infrastructure

The C-RAN pushes the baseband processing further into the network as shown in Figure 7.9. Instead of the RF fronthaul to the base station, the C-RAN uses an optical fronthaul to bring the signal further back into the network infrastructure. The baseband signal processing is done in the cloud within a data center running virtualized base stations.

Thus the RAN itself is evolving towards a cloud based architecture, and adding a mobile cloudlet to it simply amounts to provisioning additional virtual machines within the same data center. Thus the connection between the C-RAN and mobile cloudlet could just be the same local high speed

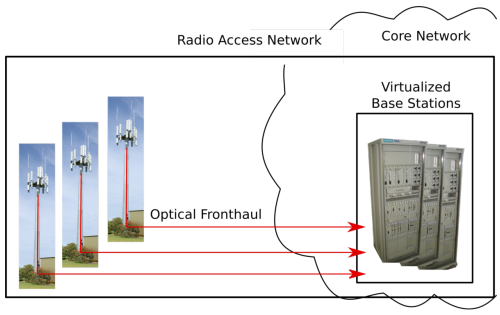

Figure 7.9: Cloud Radio Access Network (C-RAN)

network, such as Infiniband [26] connecting the nodes of the data center. Thus the access network, cloudlet, and ASO enabled CDNs could all be collocated within the same data center.

To our knowledge this use of MCC in this way to support ASO of the access network is novel. It seems that the MCC literature all assume that the communication between the mobile terminal and the MC is TCP/IP.

ASO with Access Network Traffic Reduction

ASO in this architecture is not only able to provide ASO performance gains for different types of traffic, but also is able to eliminate some traffic over the access network. For example file sharing traffic could be eliminated from the access network if the files were stored in the infrastructure cloudlet. The end terminal could then be configured as an ASO enabled thin client which only uses the access network to relay the data necessary for operation of the graphics user interface. Different partitionings of the subscriber and application across the access network results in different combinations of device energy usage and bandwidth utilization [25].

Section Partitioning the Web Browser shows how such a partitioning can avoid a particularly inefficient operation in the *Hyper-text Transport Protocol* (HTTP).

Legacy Applications

The partitioning of the subscriber unit involves a new distributed operating system which can run partitioned applications involving at least two processors: one on the end terminal and another in the infrastructure cloudlet. Likewise it may seem that this partitioning of the software application requires rewriting it, however dynamic partitioning of existing software applications is an active area of research in MCC [48, 47].

Furthermore ASO could be supported in legacy applications by intercepting certain operating system calls. System calls to output audio or video serve to identify the application. The implementation of these calls in the operating system could then be rewritten to support ASO. Because the virtual machine is running in the resource rich cloud, real-time audio and video transcoding to support JSCC is possible. Thus while ASO partitioning requires a new operating system, legacy applications could enjoy ASO by rewriting the implementation of certain application specific operating system calls.

New ASO Applications

While this subscriber partitioning scheme may potentially support ASO for legacy applications, new applications can be written using an ASO API. This will provide advantages in the dynamic partitioning of the software application. We consider this topic in Section 8.1.

Other Advantages

Because the subscriber unit exists on the cloud, this decouples its existence from the end terminal. If the end terminal is lost, the service provider can disconnect it from the account while preserving the state and private information stored on the mobile device. This makes the end terminal easy to replace. Furthermore if left at home, it is conceivable that one can log into the mobile device from a computer connected to the internet. Or even one could borrow or rent another end terminal.

Partitioning the Web Browser

To illustrate application partitioning we consider the partitioning of a web browser and show how this leads to traffic reduction and higher performance

via ASO.

HTTP was designed initially to convey static web pages. The protocol is text based and was not originally designed for communications efficiency. As the protocol evolved with the Internet, mechanisms were added to support interactive web pages. This evolution ushered in the so-called Web 2.0[4]. One of the mechanisms employed to support dynamic web pages is Asynchronous Javascript and XML (AJAX).

One dynamic feature implemented by AJAX is automatic text completion when using a search engine. This feature drops down suggested search options as individual key presses are transmitted as they are typed. Use of this feature on the *New York Times* website[5] shows 1.4 kilobytes used to transmit each key press. An optimized protocol would use only a few bytes at the most.

Although this is a very specific use case, it serves a good illustration of the inefficiencies presented by the evolution of legacy protocols to provide new features for which they were not originally designed. These issues compound with the fact that the original protocol was not designed for spectral and power efficiency. Perhaps it was assumed that the Internet would be mainly composed of wireline terminals connected to the electrical grid where bandwidth and power are at less of a premium than for battery powered wireless terminals. This assumption clearly does not hold with the rise of smartphones and tablets, and will prove altogether false with the Internet of Things on the horizon.

One way to significantly reduce the inefficiencies due to AJAX in the access network is to partition the web browser application. If the web browser can be partitioned between the terminal and a cloudlet in the access network infrastructure, then this partition can be strategically chosen to minimize the traffic between the terminal and network[6]. Once partitioned, source coding methods such as entropy coding can be used to compress the function calls between modules that are on opposite ends of the partition.

Figure 7.10 shows the use of this technique to reduce the number of bytes from 1.4 kilobytes to a few bytes to encode a dynamic keypress. Here

[4]According to Wikipedia, Web 2.0 describes World Wide Web sites that emphasize user-generated content, usability, and interoperability. [53]

[5]As of 5/10/2015

[6]Note that this is but one possible form of optimization with mobile cloud computing. It is also possible to partition to minimize processing on the end terminal which saves battery power in the case of mobile terminals [25]

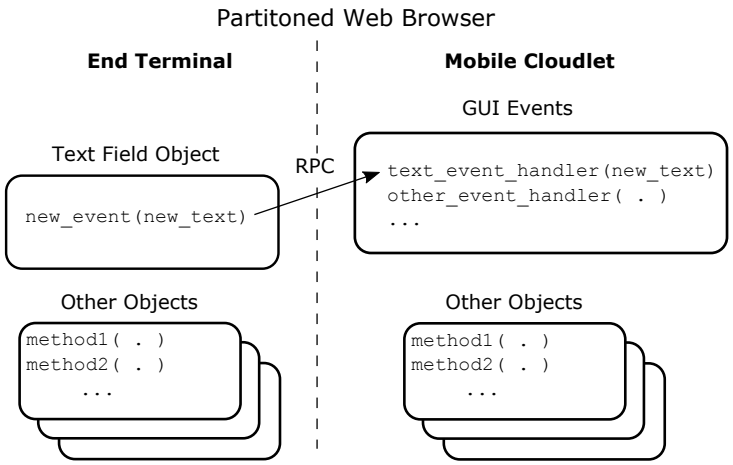

Figure 7.10: Web Browser Partitioning

we assume an object-oriented software architecture which is common with graphics user interfaces. The text field object on the end terminal has a registered event handler located in the mobile cloudlet. When a key press is detected at the user terminal, the event handler is invoked by a structure representing the function call and its argument. These structures can then be compressed using entropy coding techniques such as Huffman or arithmetic coding. A few bytes should be sufficient for this transfer.

Note that traffic over the access network has in this case been reduced by roughly two orders of magnitude, but the traffic from the mobile cloudlet to the web server is still dictated by the HTML and TCP/IP protocols. However since the bottleneck and resource constraint is in the access network, we achieve our goal of optimizing over the access network and yet maintain backward compatibility with the Internet protocols.

Chapter 8

The ASO Access Network Protocol

Having shown that it is possible to circumvent TCP/IP and protocol layering over the access network, in this chapter we propose a mechanism for an ASO enabled access network protocol. Furthermore in Section 8.2 we examine the possibility of software defined protocols where the access network protocol can evolve with the malleability of software, and a possible departure from strictly defined protocol standards.

8.1 RPC Based ASO Protocol

Now we consider the design of the ASO protocol over the access network in our subscriber partitioning approach. Since we are partitioning the operating system and software application the obvious choice would be a *remote procedure call* (RPC) based protocol that ties together the partitioned subscriber unit over the access network.

The Remote Procedure Call

An RPC is simply a function call to a remote system. The following RPC is a function call that has *parameter objects* that are sent to the remote system with the call, and *return objects* that are returned at the completion of the call:

```
[return_object1, return_object2, ...]
```

```
= rpc(param_object1, param_object2, .. );
```

RPC calls can represent function calls and memory access. An RPC with no return objects could represent an unacknowledged packet.

In fact some popular Internet application layer protocols are designed as RPC calls. For example HTTP uses the *GET* and *POST* methods – which are essentially RPC requests – to a web server which return with a value containing the HTTP content. However HTTP is a text-based protocol which is relayed over the byte-oriented TCP connection.

The RPC Packet

Instead of using a lower layer byte-oriented protocol to convey ASO RPC calls, we could use packet structures that are directly based on RPC calls. [52] uses an RPC based protocol to implement application partitioning over an infrared link, and [51] proposes something similar for wireless devices. Figure 8.1 shows an example of a calling packet and return packet implementing an RPC call.

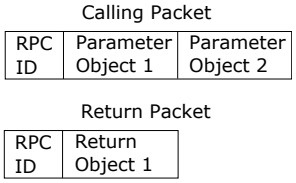

Figure 8.1: RPC Based Packets

All RPC calls of a software application are generated when the application is partitioned. The RPC ID is a field denoting a specific RPC and can be assigned when the application is loaded. An RPC ID can uniquely define the packet structure and no length fields are necessary if the definition of the RPC packets are present on both sides when the software application loads.

The RPC parameters and return objects represent the payload of the RPC packets. The type of object – or class in the parlance of object oriented programming – indicates the application. Thus the ASO API would include a standardized library of classes such as video, music, voice, text, and binary data. And the objects of an RPC packet would be coded based on their application classes.

ASO Hardware Architecture

Figure 8.2 shows the hardware architecture of an ASO interface which may be at the end terminal or access network.

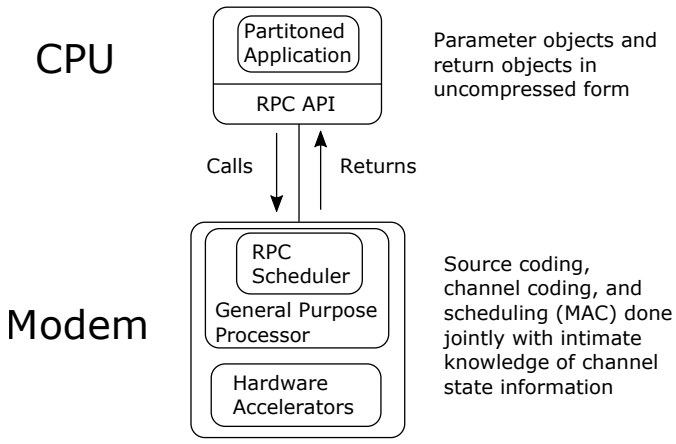

Figure 8.2: Hardware Architecture

Some objects could be passed to the modem uncompressed, and the source coding, channel coding, and scheduling are done by the modem with intimate knowledge of the channel conditions. In the case of recorded video, it is possible to avoid transcoding on the fly by passing pointers of different bitrate representations to the modem and have it obtain the appropriate representation via direct memory access based on the channel conditions. CRC checksums are usually set fields in protocol data units, however the error detection mechanism could be delegated to each individual object. For example channel codes such as LDPC have inherent error detection capabilities, and objects using an LDPC channel code could simply invoke this property. Thus error detection could be handled in an ASO way.

The RPC packets are the fundamental units of our ASO transmission scheme. For streaming applications, certain RPC packets need to be scheduled at a regular rate. Thus provisions for stream scheduling need to be present in the ASO API.

ASO RPC Scheduler

The ASO RPC scheduler functions as an application aware media access control. Not only is possible to know the application, but also the expected duration of the transmission which can be used to optimize the scheduling. Control of scheduling can be centralized by the access network, which turns out provide performance benefits when we analyze ASO for ADSL (see Section 9). The scheduler can provide fairness between software applications and different users as well as quality of service.

Software Architecture

In Section 7.3 we showed how ASO could be provided for legacy applications by rewriting the operating system libraries to use an ASO API. However new applications should directly use the ASO API which provides more information to the processes of dynamic software application partitioning and linking. Usage of ASO API objects and classes will clearly identify the application and allow the automatic software partitioning to better optimize the partitioning to reduce traffic over the access network and improve performance.

8.2 Software Defined Protocols

In Section 7 several ways to circumvent TCP/IP in the access network were discussed. These methods provide backward compatibility for applications running on the terminal that were originally written for TCP/IP while at the same time provide ASO over the access network which is not limited by the layering imposed by TCP/IP. Although these are possible ways to circumvent TCP/IP, in order for such solutions to be commercially viable we need to consider how access network infrastructure and end user terminals can be switched over from TCP/IP to support new cross layer protocols that supports ASO.

Background

Active Networks

At first glance the shift away from TCP/IP over the access network may seem contrary to the packet convergence movement [22] where infrastructure equipment in the access and core networks are transitioning from being circuit switched to TCP/IP packet switched. However if we consider several other trends in telecommunications research and practice, the time may be ripe for the software defined protocols as envisioned in active network research from the 90s [16]. While the vision of active networks encompassed the core network, perhaps ASO in the access network will provide the use case to drive the development of network platforms with software defined protocols.

Software Defined Networks

Feamster in [16] describes the evolution of ideas that lead to the Software Defined Networks (SDN) which is a recent trend in networking. In traditional IP networks, the routing tables are setup and maintained by routing protocols that are used between routers. These protocols were not able to support the automatic configuration of virtual networks in data centers where virtual machines are distributed dynamically among processor nodes spanning the intranet. The adoption of SDN was driven by this need [24]. The SDN solves this problem by establishing APIs that allow the necessary network functionality to be implemented by software.

While the idea of network functions that could be implemented via software on open network elements were areas of research even as the Internet was being commercialized, the pressing need for such functionality was not there for these ideas to gain traction. Instead TCP/IP with its layered design and end-to-end principle filled the immediate need of reliable data transmission across disparate link layer networking technologies.

However with the advent of cloud computing and the proliferation of data centers, decade old ideas originating with active network research finally found a problem big enough to drive its commercialization and adoption. While SDN today is mainly concerned with software defined network functions above the link layer, we believe that this trend will continue and eventually touch the lower layers.

Network Function Virtualization and Data Center Technology

Often concomitant with SDN in the literature is Network Function Virtualization (NFV). In NFV certain network functions traditionally implemented by standalone network devices called middleboxes are virtualized and run on general purpose hardware. The idea is that general purpose computing hardware is now fast enough to implement these functions and therefore it is cheaper to implement and maintain such functionality in software than on specialized hardware. Furthermore these functions can be run on virtual machines in a data center architecture.

While currently NFV is primarily associated the virtualization of middlebox functionality, there is activity in the industry and research in virtualizing the functionality of access network infrastructure. Specifically there has been a spate of recent activity in the C-RAN (see Section The Cloud Radio Access Network (C-RAN))

Network Functions on General Purpose Computing Hardware

There are several advantages of using a cloud data center architecture for access network infrastructure. The advantages here are similar to the advantages that cloud computing lends to IT infrastructure.

In cloud computing, economies of scale are leveraged for computing resources. Large centralized server farms can host infrastructure, platforms, and applications more economically than if these resources were handled privately by organizations that need them. Therefore instead of purchasing the infrastructure to support their computing needs, organizations can more economically purchase service agreement contracts with cloud computing providers such as Amazon and Google.

In addition to economies of scale in purchasing, installing, and maintaining data centers, as opposed to the aggregate cost of many private facilities providing the same functionality, there is also the effect of statistical multiplexing: when more users share a resource with random access patterns, the average cost of providing a given level of service is reduced.

In the same way access networks stand to benefit from the economies of scale and statistical multiplexing by leveraging a cloud architecture. Furthermore, instead of distributed specialized hardware, the data center architecture consists of centralized general purpose computing hardware running

the network functions in software. [55] recognizes that despite the power of modern general purpose processors used in data centers, accelerator banks with specialized hardware may be needed to accommodate the baseband signal processing.

The point of mentioning these trends is that access network infrastructure is moving towards an open cloud based data center architecture where much more of the functionality is implemented in software. This also means that the protocols supported by the access network can be modified by changing the software. In other words this architecture can support software defined protocols that can evolve with the malleability of software. Combined with one of the methods in Section 7, a platform that can support ASO with backward compatibility to TCP/IP may soon become commercially available. And this development is in line with the recent trends of cloud computing, SDNs, and network function virtualization that have already gained a full head of steam.

End-Terminal Architecture

We have shown that a platform for access network infrastructure may be available in the near future to support software defined protocols. However the question remains whether custom programmed end terminals that depart from the established standards can be made economically. The importance of standards lie in creating a competitive marketplace where interoperable equipment and their components can reach economies of scale and pass on these savings to the end customer.

The advent of open source software such as Linux, and open source hardware such as the Arduino and BeagleBoard may portend the availability of open devices that are at the same time mass manufactured and yet fully custom programmable. This brings up the exciting possibility of defining interoperability not as the conformance to a strictly defined standard, but rather whether two devices can be *programmed* to be interoperable. This may usher in a bold new age of telecommunications where standards are not rigidly defined, but loosely suggested with broad consensus on the necessary custom hardware to accelerate computationally intensive signal processing, with an open programmable platform that allows protocols to evolve with the malleability of software.

It is interesting that in most modern modem hardware designs of end terminal equipment use specialized hardware to accelerate baseband signal

processing while the media access control (MAC) and higher layer network functionality run on a general purpose processor [32]. Although a general purpose processor is used for this, the computing environment of these subsystems are generally not open and cannot be programmed except by the manufacturer. Therefore the hardware architecture for software defined protocols – where at least the network processing from the MAC layer and above can be software defined – is already commonplace. There is only the need for the closed general purpose processing environment to be opened and standardized.

A Dichotomy Instead of Layers

A possible approach to software defined protocols may be a dichotomy of signal processing and network processing in lieu of the layers of the traditional protocol stack. The signal processing may need to be strictly standardized to accommodate the specialized hardware needed to support it. These include processing traditionally associated with the physical layer such as channel coding and modulation.

However the functionality traditionally provided by the media access control and even network layer could remain unspecified, or exist simply as a reference design. This functionality can be provided by software which can be modified by the network operator to support ASO and to evolve as demand for new applications and classes of service arise.

The General Purpose Network Processor

Instead of specifying the exact operation of these networking functions, the performance of the computing environment of the general purpose network processor executing these functions could be specified.

Today's protocol standards seek to enable interoperability by strictly defining all aspects of the communications protocol. In the spirit of active networks we propose a more flexible approach where interoperability is enabled by standardizing the signal processing and its performance requirements, but leaving the network processing software defined. Instead of defining the network processing as part of the standard, we standardize the performance requirements and programming interface of the network processor. We envision a standardized API and development environment for the network processor in much the same way the *Portable Operating*

System Interface (POSIX) has made UNIX code portable across different platforms and processors.

We believe that such a specification will still provide the interoperability and ultimately the economies of scale that traditional telecommunications standards provide. But at the same time this allows for the flexibility to optimize the network for specific applications and even the evolution of the network to accommodate demand for new applications.

A Confluence of Trends

In conclusion to this section, we believe there is a confluence of trends, technologies, and architectures that will facilitate software defined protocols and ASO in access networks. We were surprised at the lack of data on how much can be gained by ASO over access networks that are currently based on TCP/IP. In the next section we attempt to quantify the performance gain of ASO when applied to the ADSL access network.

Chapter 9

ASO for DSL

In this section we make a cursory investigation of the potential of ASO over DSL, and compare this with other optimization techniques that only involve the physical layer. This serves not only to illustrate the potential impact of ASO for DSL, but may also portend significant performance gains for ASO over other communication channels. In the following subsection we review the channel characteristics of DSL which are pertinent to our ASO design for DSL.

9.1 DSL

DSL provides digital data services over the venerable telephone network that was originally designed for analog voice signals over a century ago. Each line consists of a pair of copper wires, known as an *unshielded twisted pair* (UTP). An information bearing signal is embodied as a differential voltage across the pair.

There are an entire family of standards that implement DSL:

Integraded Services Digital Network	(ISDN)
High Bit Rate DSL	(HDSL)
Symmetrical DSL	(SDSL)
Asymmetrical DSL	(ADSL)
Very High Bit Rate DSL	(VDSL)

Here we study the specific case of ASO for ADSL. ADSL is the most widely deployed DSL service, and is used for providing internet access for individual

consumers rather than businesses.

The DSL line is for the most part an interference limited communications channel. In addition to the thermal and shot noise that originate form the electronics of the receiver, the DSL line suffers from crosstalk from the other pairs it is bundled together with. This crosstalk tends to dominate the other impairments on the line.

Figure 9.1 shows the *far end crosstalk* (FEXT) and *near end crosstalk* (NEXT) originating from *Pair 2* and affecting the B terminal. FEXT seen at terminal B originates from the signal generated at C intended for D but couples from *Pair 2* to *Pair 1* and interferes with the signal originating from A intended for B. NEXT seen at B is generated from the near end terminal D and couples onto *Pair 1*, also interfering with the intended signal from A.

There is likewise FEXT and NEXT affecting the other terminals but not shown in the figure. In a typical DSL deployment there are more than two pairs that affect one another with crosstalk. Groups of pairs that are bundled together are a *binder group* which in many cases contain 25 pairs.

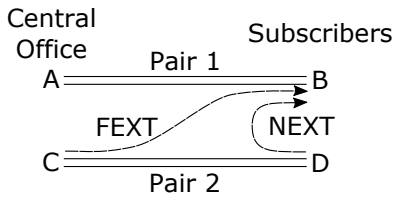

Figure 9.1: FEXT and NEXT on a DSL Line

In a typical ADSL implementation, NEXT is mitigated by frequency division duplexing (FDD), where upstream and downstream traffic are sent on separate frequency bands. In this case, the NEXT generated by terminal D in Figure 9.1 does not affect the downstream traffic from terminal A since they are on separate frequency bands, and NEXT can be eliminated by filtering. FEXT however is present in typical ADSL deployments and is factored into the ADSL link budget.

Figure 9.2 shows the average FEXT power expected in one pair of a 25-pair binder group for different lengths of 26 AWG[1] UTP. These calculations were based on the 1% worst case FEXT parameters from Table 2 (300kHz

[1]American wire gauge

– 40 MHz) of [50]. The power spectral density of FEXT is

$$P_{FEXT}(f) = P_{TX}(f)\ |S_{21}(f)|\ 10^{-5/10}\ 2.62 \times 10^{-19}\ L\ f^{1.85}\ 3.28$$

where $P_{TX}(f)$ is the ADSL transmit PSD which is nominally -40 dBm, $S_{21}(f)$ is the UTP transfer function expressed as a scattering matrix coefficient as a function of frequency, L is the UTP length in kilometers, and f is the frequency in Hz. A 5dB offset from the 1% worst case FEXT was obtained by finding the median of the plots in Figure 5, which corresponds to the average FEXT value [7]. These average values are important for the calculation of bit rates in Section 9.2.

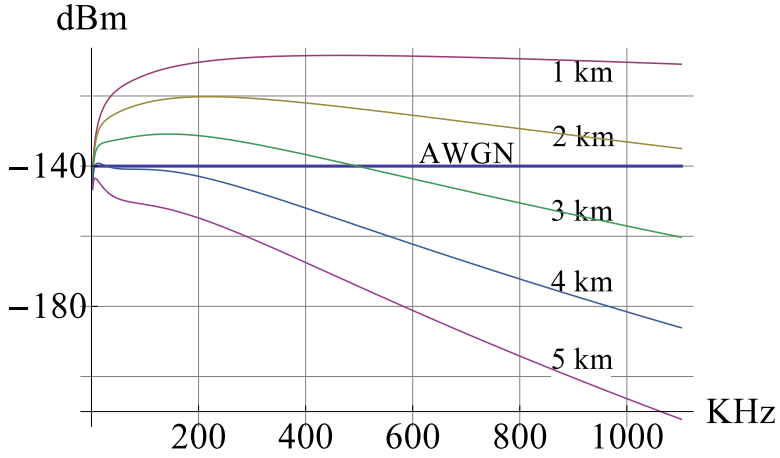

Figure 9.2: Average FEXT on 26 AWG UTP in 25 Pair Binder Group

9.2 Optimizing ADSL

Next we consider various optimization techniques to increase the performance of ADSL. *Channel coding* and *vectoring* are two physical layer techniques that can be used to increase ADSL performance. We compare these techniques to ASO. Figure 9.3 compares the bit rates of standard ADSL with these different optimization schemes. The following subsections describe these optimization schemes and the method used to calculate their respective bit rates.

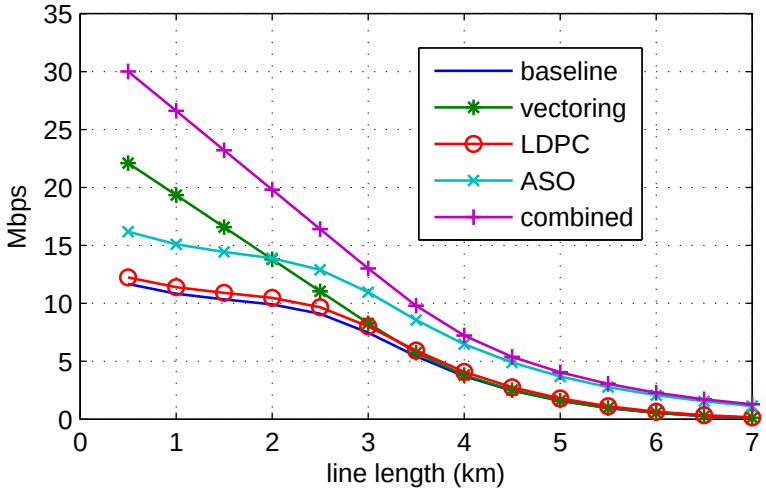

Figure 9.3: Bit Rates of DSL Optimization Schemes

ADSL Baseline

The baseline bit rate represents the downlink performance of a standards compliant ADSL system. An estimate of the bitrate is given by

$$\left(1 - \frac{OH}{100\%}\right) \int_{f_0}^{f_1} \log_2 \left(1 + \frac{SNR(f)}{\Gamma}\right) \mathrm{d}f \tag{9.1}$$

where f_0 and f_1 are the bounds on the frequency band, $SNR(f)$ is the *signal to noise ratio* (SNR) of the channel, and Γ is the gap to capacity of the coded system. This equation is just Shannon's formulation for channel capacity for an *additive white Gaussian noise* (AWGN) channel subject to the SNR constraint $SNR(f)$ and discounted by a gap to capacity Γ, to account for realistic coding and modulation. For the baseline case we used the following parameters:

[24]This stands for IP over Ethernet over ATM Adaption Layer 5 using an ATM virtual circuit

Table 9.1: Baseline Parameters

Parameter	Value	Notes
OH	13%	The overhead given in [46] for IP/Eth/AAL5 VC[2]in *Table 2* This is a typical configuration in ADSL deployment and represents the protocol overhead.
f_0	140.156 kHz	Beginning of downlink ADSL bandwidth
f_1	1.10184 MHz	End of downlink ADSL bandwidth
Γ	10.5dB	=9.5dB-5dB+6dB 9.5dB = uncoded gap to capacity 5dB = coding gain of standard ADSL 6dB = typical ADSL link budget margin

$SNR(f)$ was calculated as

$$SNR(f) = \frac{P_{RX}(f)}{P_{AWGN}(f) + P_{FEXT}(f)} \tag{9.2}$$

$$P_{RX}(f) = P_{TX}(f)\ |S_{21}(f)|^2$$

$$P_{TX}(f) = -40 \text{ dBm/Hz, the transmit PSD}$$

$$S_{21}(f) = \text{transfer function of the 26 AWG UTP line}$$

$$P_{AWGN} = -140 \text{ dBm/Hz, the assumed receiver white noise PSD} \tag{9.3}$$

$S_{21}(f)$ is the port 1 to port 2 transmission coefficient of the S-parameter scattering matrix representing the twisted pair transmission line as a function of frequency given by the so called *British Telecom Models* [19].

The upper frequency limit of 1.1 MHz in our calculation matches that of standard ADSL. There is a provision in the ADSL2+ standard to double this limit to 2.2 MHz, and VDSL pushes the frequency to as high as 30 MHz. In these cases the bit rates would be much higher for short line lengths because of the expanded bandwidth. However for longer line lengths such as 4 km and beyond, these additional frequencies will not make a difference since the transmitted signal will be attenuated below the noise floor of the receiver. While this significantly boosts the bit rates for short line lengths, the effect should not be substantial on our comparison of different optimization techniques since the baseline would include this performance boost.

Figure 9.4 shows the bit rate improvement factor of the different optimization techniques over the baseline.

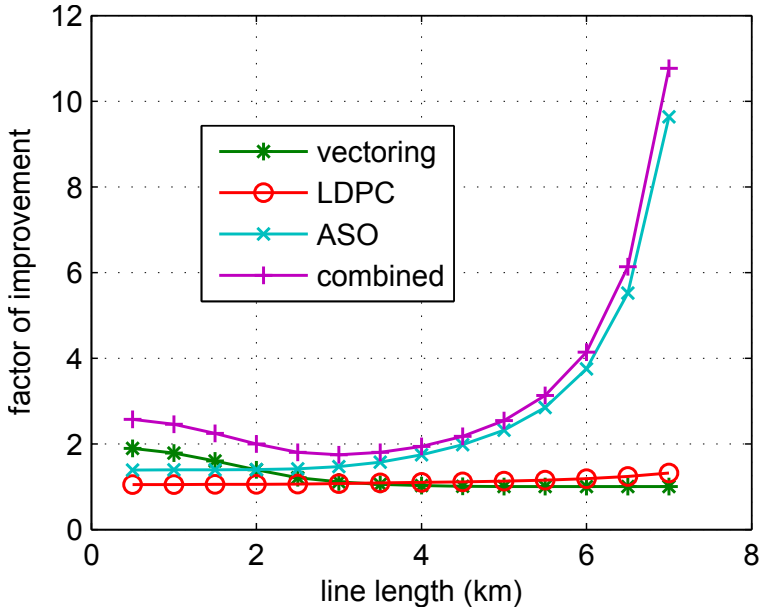

Figure 9.4: Factor of Improvement for DSL Optimization Schemes

Channel Coding

The ADSL standard mandates an inner *trellis coded modulation* (TCM) code with an outer *Reed-Solomon* code. This concatenated code gives a coding gain of about 5 dB at a target bit error rate of 10^{-7} [59]. Modern capacity approaching codes such as *low density parity check* (LDPC) codes and *turbo codes* can give greater coding gain of around 7 dB, which is within 1dB of the capacity without shaping [37]. Thus the improvement of using an LDPC code is about 2dB. The marginal effect on bit rates of upgrading the channel coding to LDPC over the current standard baseline can be seen in Figures 9.3 and 9.4. The parameters used to calculate the LDPC bitrates are the same as in Table 9.1, except $\Gamma = 8.5$ dB.

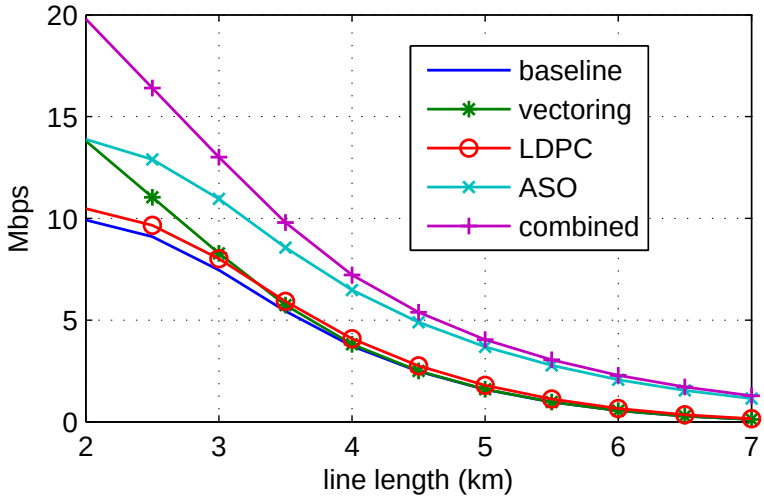

Figure 9.5: 2 km+ Bit Rates of DSL Optimization Schemes

Vectoring

Vectoring, also known as vectored *multiple input multiple output* (MIMO), is a physical layer technique that can mitigate crosstalk. In a typical point-to-multipoint[3] DSL deployment, vectoring can mitigate FEXT and NEXT at the central office, as well as FEXT at the subscriber side [9]. This is because the vectored MIMO device is at the central office (CO), while the NEXT generated at the subscriber side as shown in Figure 9.1 cannot be canceled since since the subscribers are not collocated and do not have the necessary side information.

The potential bit rate performance of a vectored MIMO system can be calculated by removing the FEXT from the SNR which is then calculated as

$$SNR(f) = \frac{P_{RX}(f)}{P_{AWGN}(f)}$$

In practical systems it may not be possible to remove all the FEXT[4], so this serves as an upper bound on vectoring bitrates.

[3]Central office to multiple subscribers

[4]For example if there is significant FEXT originating from a line or source that is not part of the vectored MIMO system

Using this expression for the SNR, the integral in (9.2) with the same parameters from Table 9.1 gives the *vectoring* bit rates in Figure 9.3 and factor of improvement in Figure 9.4. Notice that while vectoring provides the largest gains for short line lengths up to 2km, it provides hardly any benefit beyond 3km. The reason is that there is hardly any FEXT present in these longer lines at the outset (see Figure 9.2), and hence no performance gain from FEXT cancellation.

ASO

Here we consider the ASO of ADSL for video streaming, which is the predominant form of consumer Internet usage. In 2014 consumer internet video traffic was 64% of all consumer internet traffic, and is projected to increase to 80% in 2019 [10]. Our optimization involves maximizing the downlink bandwidth while recognizing that there will be the occasional need to deliver moderate to high rate uplink traffic in the case of uploading files or two-way video conferencing. This optimization also applies to web surfing where the traffic is disproportionately in the downlink direction.

ADSL stands for *asymmetric* digital subscriber line, and the asymmetry in uplink and downlink bandwidths was designed to accommodate video [45]. However because the lower layers[5] do not have real-time knowledge of the application, a tradeoff was made to permanently demarcate the uplink and downlink frequencies. More bandwidth was allocated to the downlink in anticipation of the demand for video, but enough uplink was reserved for the occasional applications that demand bulk uploads. To complicate matters further, the upstream and downstream bitrates are also a function of the length and geometry of the UTP line. Thus the bandwidth allocation that may serve one line well may not be so optimal for another line. In fact the largest performance gain for ASO happens over longer lines which implies that the standard ADSL bandwidth allocation is not well suited for longer lines.

Our simple approach to ASO is to assume that the lower layers have real-time knowledge of the application and can dynamically adjust the uplink and downlink bandwidths to accommodate the application. The optimal allocation for the application of video streaming would be to give the entire downlink to the video. In practice it would make sense to reserve a small

[5]Namely the physical and link layers

amount of uplink bandwidth for messaging[6], but we calculate the bit rate of the entire bandwidth as an upper bound on ASO bitrates and improvement. Thus for the calculation of the ASO bit rate we change make the following changes to the baseline parameters in 9.1: $f_0 = 2156.25Hz$ which is half the tone spacing of the ADSL DMT modulation[7], and $OH = 0\%$. We assume that the ASO protocol does not include any of the overhead associated with traditional layered protocols, and any protocol overhead would be insignificant for large packet sizes.

The permanent division between upstream and downstream bandwidths in standard ADSL is like a divided highway where although the traffic in either direction is a function of the time of day, the highway configuration remains static regardless of the asymmetric demands of the rush hour. Our ASO design allows the divider to be dynamically reconfigured based on the collective demand for different kinds of applications.

In a lightly-loaded ASO system, users could enjoy access to the full spectrum for both uplink and downlink at the ASO rate most of the time as the system dynamically configures the spectrum allocation per user and per application. Actually this scenario is not so far fetched if we consider all the users in a binder group. Most of the time a user will not be generating traffic[8]. When more than one user is active, chances are that there will not be a conflict as the users will tend to be engaging in applications that demand downlink bandwidth while only generating sporadic uplink activity.

However as the ADSL access network becomes more heavily loaded, there will eventually be a conflict in application where one user will require some kind of bulk upload requiring more significant uplink bandwidth while most of the other users are engaging in downlink applications such as watching videos or surfing the web. An obvious approach to this problem would

[6]Such as uplink RPC calls

[7]We do not include frequencies starting from 0 Hz since the ADSL signal excludes DC

[8]This is actually quite different from standard ADSL where a signal is constantly being transmitted regardless of whether there is meaningful data to be sent. This is done so that the network can converge to a steady state where FEXT is factored into the bit allocations at each tone. If each line is constantly turning on and off depending on the traffic conditions, then the distributed channel sounding and bit allocation mechanism would not converge properly to a steady state. Therefore for ASO to work properly in this case, a centralized channel sounding and bit allocation mechanism needs to implemented. This is already a necessity for vectored MIMO and would be a natural design if ASO were to be merged with vectoring, as explored in Section 9.2

be to divide the uplink and downlink bandwidths as a compromise between the users during the times of conflict. However if we combine vectored MIMO with ASO we can do much better.

Combined Performance Gains

The potential performance gains of vectored MIMO and ASO are quite complimentary. Vectoring provides the highest gains at the shorter line lengths, while ASO provides its greatest improvements for longer lines. Furthermore if we can dream of a next generation DSL deployment based on data center technology and network function virtualization, we would expect to include vectoring, capacity approaching channel coding, and ASO. The performance of such a system is shown as the *combined* plots in both Figures 9.3 and 9.4. The parameters for this system is shown in Table 9.2.

Table 9.2: Combined Parameters

Parameter	Value	Notes
OH	0%	Protocol overhead for ASO is assumed to be negligible, especially for high speed transfers
f_0	2156.25 Hz	Half of the tone spacing
f_1	1.10184 MHz	End of downlink ADSL bandwidth
Γ	10.5dB	= 9.5dB-7dB+6dB 9.5dB = uncoded gap to capacity 7dB = coding gain of LDPC at \$10ˆ{-7} BER 6dB = typical ADSL link budget margin

If we combine the merits of vectoring with ASO, our calculations show that we can simultaneously allow a significant uplink rate for one user in a binder group with little effects in the maximum downlink rates of the rest of the users.

Vectored MIMO can cancel NEXT at the CO. This allows reception of NEXT-free upstream traffic while delivering downlink traffic to the other users at the full bandwidth. Without this feature, the upstream traffic rate would be degraded by NEXT due to downlink transmission. Figure 9.6 shows the uplink rate of the combined design using an uplink bandwidth of 130 kHz, which is the same uplink bandwidth of standard ADSL. Also

shown is the full downlink or uplink rate when the system allocates the full bandwidth to either uplink or downlink direction. The *1% worst case* and *pair-to-pair* are the plots of degradation to all downlink users enjoying the full downlink bandwidth while one user is performing an upload across the traditional 130 kHz uplink bandwidth. Figure 9.7 shows the same calculation in terms of degradation to the downlink rate.

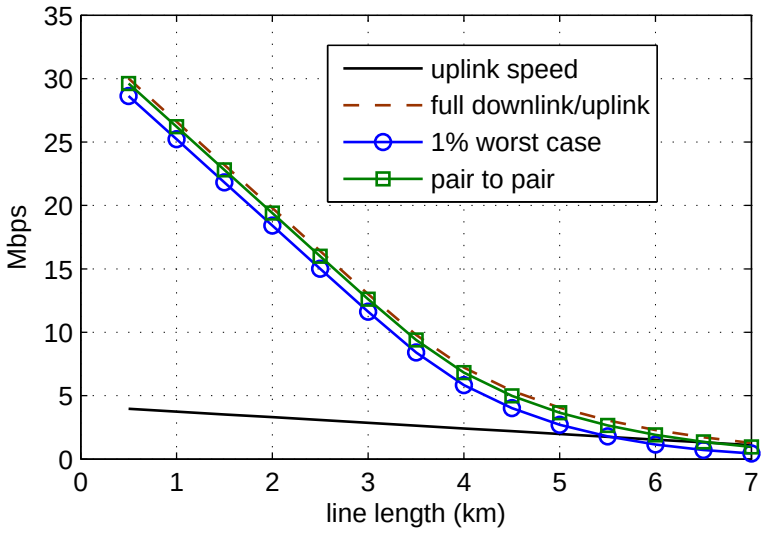

Figure 9.6: Uplink Rates for One User and Downlink Rates of Full Bandwidth Users

The phenomena of NEXT between two UTP lines across the frequency band is inherently a Rayleigh distribution in magnitude. As a result, on the power scale this distribution is exponential [13]. However the parameter of this exponential distribution is random across different UTPs. Thus the pair-to-pair NEXT power transfer function is a mixture of exponential distributions which has been empirically characterized as a log-normal distribution[9].

In our calculations we used the *1% worst-case* NEXT model from the ADSL2 standard [23] with one interfering user operating in the traditional uplink frequency band. The 1% worst case NEXT was formulated by em-

[9]However [27] argues that the log-gamma distribution is a better fit

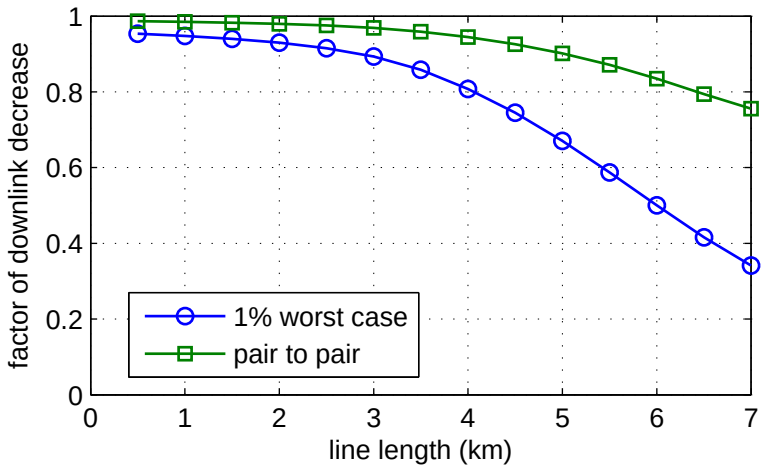

Figure 9.7: Downlink Decrease of Full Bandwidth Users with One Simultaneous Uplink User

pirical studies where the 1% worst case tail is calculated from estimated parameters of a normal distribution in the decibel domain.

The *1% worst-case worst case* NEXT allows downlink users to establish an SNR estimate with knowledge of the channel and the fact that one other user is transmitting in the uplink. Typical ADSL operation estimates the SNR independently of the activity of other lines. However with ASO, the network knows the application as well as an estimate of the duration of the usage. With this knowledge the network can notify subscriber terminals so that they can back off the SNR estimate and establish the correct channel coding and modulation scheme to account for the expected NEXT for the duration of the uplink usage.

However it is possible to do even better if the NEXT pair-to-pair power transfer functions are estimated by a centralized training procedure as is done with vectoring. Each downlink user can know the pair-to-pair NEXT characteristics at each subcarrier for each potential interferer. The bit rates of this design is based on the NEXT pair-to-pair log-normal loss distribution given in [27]. The advantage of this design, like the ASO performance gains in Section 9.2, is more noticeable at longer line lengths as seen in Figures 9.7 and 9.8.

Thus with this combined ASO scheme all users can operate at the full ASO downlink bandwidth without too much disturbance when only one

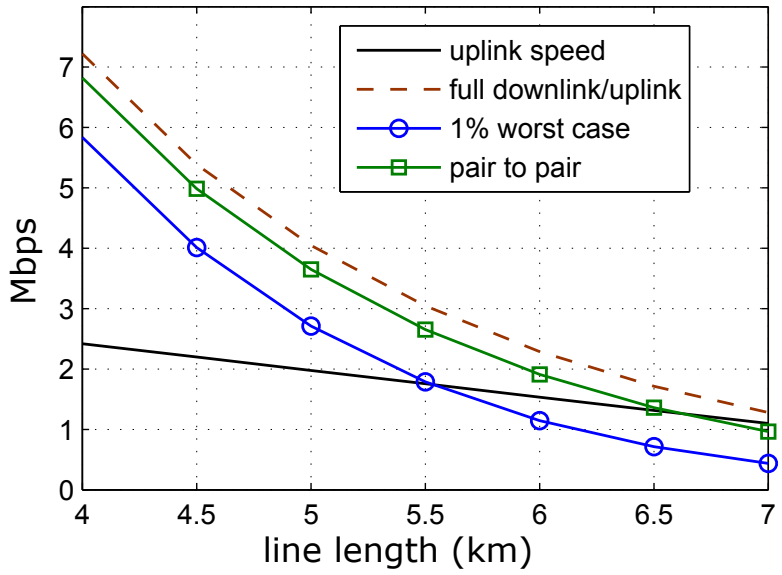

Figure 9.8: Uplink Rates for One User and Downlink Rates of Full Bandwidth Users

user in the binder group is using an application that demands high uplink bandwidth. Since these occasions are rare in relation to high downlink bandwidth applications, it may be possible to support these high ASO downlink rates with little interruption. And at times of excess capacity the full bandwidth could be switched to the uplink direction for to facilitate file uploads that have been queued, such as in sync operations. Furthermore since the subscriber device has been partitioned between the access network, most of the storage is maintained in the data center cloudlet, obviating the need for much of the uplink traffic through the access network. This would alleviate much of the file sharing traffic over the access network.

9.3 Conclusion

ASO over ADSL provides significant performance advantages, especially for 26 AWG lines over 2 km long. Lines over 2 km actually represent 50% of all telephone lines in the United States [45]. For a 26 AWG line at 2 km, ASO provides the same improvement in performance as vectoring. Beyond this length, ASO provides performance increases that far surpasses the other

purely physical layer optimizations considered here. It is notable that the change from the standard channel coding used in ADSL to a capacity approaching code such as LDPC offers a rather marginal improvement. The invention of practical capacity approaching codes and their associated iterative decoding techniques is considered a monumental achievement in communications engineering. However in terms of optimizing DSL, the practical impact is not as impressive. This supports our suspicion that efforts to improve performance at the physical layer alone come with a high cost in complexity and are already approaching a fundamental limit. MIMO technology continues to provide promise of further increase of DSL performance, as it also does in the wireless communications. However the ASO techniques examined here require less signal processing complexity and these schemes could be implemented with changes to the MAC functionality on a general purpose processor. And it turns out that for ADSL, vectored MIMO and these ASO techniques are very complementary, offer their respective advantages over different line lengths, and can be combined to support nearly full downlink ASO speeds concurrent with high bandwidth uplink use.

Our proposed ASO architecture provides not only raw performance gains in bitrate, but also a mechanism for quality of service and quality of experience. In the case of video streaming, the application aware network can attempt to schedule a constant bitrate for a good video experience while packing data that is less sensitive to timing around the video traffic.

ASO also allows the source coding can be adjusted with intimate knowledge of the channel conditions, and joint source channel coding techniques where the source and channel coding and can be jointly optimized to match the application. For the video applications, this means that the source coding can be operate with intimate knowledge of the channel conditions, and the decoding process can take advantage not only of the channel coding redundancy but also the residual redundancy in the source coding. This is an interesting area of research that has not received its due attention since it is assumed that there is no easy way around the layering construct of protocols [14] – an assumption that our ASO design challenges. This can provide additional performance gains that was not analyzed in our example of ASO over ADSL.

ASO design allows the physical layer and media access control functionality to be optimized based on intimate knowledge of the application. As exemplified with our study of ASO over ADSL, these changes can achieve significant performance gains with relatively low complexity costs. ASO

may provide the greatest performance gains at the least marginal complexity costs. This seems to be the case with ADSL, and we suspect this to be true with other communications channels.

Chapter 10

Conclusion and Future Work

This dissertation on optimizing communications systems *seeks to find the greatest performance gain at the least marginal cost* in signal processing complexity. By targeting well known inefficiencies due to protocol layering, significant performance gains at low complexity costs are possible simply by allowing application visibility into the functions of the lower layers. Though its benefits are well known, ASO is assumed to be difficult due to the entrenched TCP/IP protocol. We have proposed *Subscriber and Application Partitioning* along with an *RPC-Based Software Defined Access Network Protocol* as an architecture which leverages current trends to provide ASO over the access network while still maintaining a TCP/IP stack in the subscriber for Internet connectivity.

Future work may include:

- *Calculating a better estimate of Subscriber Partitioning based ASO performance benefit by using statistical models of Internet traffic.* Pertinent topics are the amount of traffic over the access network can be eliminated by Subscriber Partitioning, and how much ASO improves the efficiency of access network traffic across all significant applications.

- *Determining the advantages of Subscriber Partitioning based ASO for access networks other than ADSL.* such as radio access networks or coaxial cable networks.

- *Determining the advantage of JSCC and JSCD for different applications .* JSCC in particular has not received adequate attention due

to assumptions about entrenched layered protocols. However iterative receivers, which are popular due to capacity approaching codes, can be extended to incorporate side information due to residual redundancy left over from source coding in JSCD. Furthermore JSCC which jointly designs the source coding and channel coding may provide even greater benefits in performance and complexity reduction.

- *Determining what to standardize when protocols can be software defined.* The ASO APIs and the network processor should probably be standardized. The performance of certain signal processing routines should also probably be standardized since application specific integrated circuit (ASIC) hardware acceleration is necessary for power constrained devices.

Bibliography

[1] JB Anderson, F Rusek, and V Owall. "Faster-than-Nyquist signaling". In: *Proceedings of the IEEE* (2013). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6479673.

[2] John B. Anderson and Adnan Prlja. "Turbo equalization and an M-BCJR algorithm for strongly narrowband intersymbol interference". In: *2010 International Symposium On Information Theory & Its Applications*. IEEE, 2010, pp. 261–266. ISBN: 978-1-4244-6016-8. DOI: 10.1109/ISITA.2010.5648949. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5648949.

[3] Dieter M Arnold et al. "Simulation-based computation of information rates for channels with memory". In: *IEEE Transactions on Information Theory* 52.8 (2006), pp. 3498–3508.

[4] F Aune. "Cross-layer design tutorial". In: *Norwegian University of Science and Technology, Dept. of Electronics and Telecommunications* (2004). URL: http://www.iet.ntnu.no/projects/cuban/archive/1812041.pdf.

[5] L. Bahl et al. "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)" English. In: *IEEE Transactions on Information Theory* 20.2 (1974), pp. 284–287. ISSN: 0018-9448. DOI: 10.1109/TIT.1974.1055186. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1055186.

[6] S Benedetto and E Biglieri. *Principles of Digital Transmission: With Wireless Applications*. Information Technology: Transmission, Processing and Storage. Springer US, 2006. ISBN: 9780306469619. URL: https://books.google.com/books?id=cBrjBwAAQBAJ.

[7] JAC Bingham. *ADSL, VDSL, and multicarrier modulation*. 2000. URL: http://share.laogu.com/download/adsl.pdf.

[8] J Border. *Performance Enhancing Proxies Intended to Mitigate Link Related Degradations*. 2001. URL: http://tools.ietf.org/html/rfc3135.

[9] JM Cioffi and S Jagannathan. "CuPON: the Copper Alternative to PON 100 GB/s DSL Networks". In: *Communications Magazine, IEEE* (2007). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4251082.

[10] Cisco. "Cisco Visual Networking Index : Forecast and Methodology , 2014 – 2019". In: June (2015). URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white{_}paper{_}c11-481360.pdf.

[11] I. Cosovic, S. Brandes, and M. Schnell. "Subcarrier weighting: a method for sidelobe suppression in OFDM systems". English. In: *IEEE Communications Letters* 10.6 (2006), pp. 444–446. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2006.1638610. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1638610.

[12] T M Cover and J A Thomas. *Elements of Information Theory*. Wiley, 2012. ISBN: 9781118585771. URL: https://books.google.com/books?id=VWq5GG6ycxMC.

[13] H Cravis and TV Crater. "Engineering of T1 carrier system repeatered lines". In: *Bell System Technical Journal* (1963). URL: http://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1963.tb00508.x/abstract.

[14] Pierre Duhamel and Michel Kieffer. *Joint source-channel decoding: A cross-layer perspective with applications in video broadcasting*. Academic Press, 2009. ISBN: 0080922449. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=OK9mEpdMtGAC{\&}oi=fnd{\&}pg=PP2{\&}dq=related:SSoMBZQn84oJ:scholar.google.com/{\&}ots=k8pDVv0WKD{\&}sig=dbaQkRA8aUaT1E6mS351aAV3XYA.

[15] Kevin R. Fall and W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Vol. 8. Addison-Wesley, 2011, p. 850. ISBN: 0132808188. URL: http://books.google.com/books?hl=en{\&}lr={\&}id=a23OAn5i8ROC{\&}pgis=1.

[16] N Feamster, J Rexford, and E Zegura. "The road to SDN". In: *Queue* (2013). URL: http://dl.acm.org/citation.cfm?id=2560327.

[17] Thomas R M Fischer. "Some Remarks on the Role of Inaccuracy in Shannon's Theory of Information Transfer". In: *Proc. 8th Prague Conf. Information Theory* A (1978), pp. 211–226.

[18] Anand Ganti, Amos Lapidoth, and L Emre Telatar. "Mismatched decoding revisited: general alphabets, channels with memory, and the wide-band limit". In: *IEEE Transactions on Information Theory* 46.7 (2000), pp. 2315–2328.

[19] P Golden, H Dedieu, and KS Jacobsen. *Fundamentals of DSL technology*. 2005. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=O2B6AgAAQBAJ{\&}oi=fnd{\&}pg=PP1{\&}dq=related:z5dSb2{_}pFuEJ:scholar.google.com/{\&}ots=ZsNztCpLwa{\&}sig=gWevC4b25LNjsPOe6zpMKEYLldo.

[20] P Golden, H Dedieu, and KS Jacobsen. *Implementation and applications of DSL technology*. 2007. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=Jjkd74jY47oC{\&}oi=fnd{\&}pg=PP1{\&}dq=fundamentals+of+dsl{\&}ots=FLLRsO6w5r{\&}sig=Vpvgic9yp7pKV57EiMFWbn3pLn4.

[21] S S Haykin. *Communication systems*. Wiley, 2001. ISBN: 9780471178699. URL: https://books.google.com/books?id=ieZSAAAAMAAJ.

[22] Eugenio Iannone. *Telecommunication Networks*. CRC Press, 2011, p. 918. ISBN: 1439846367. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=5qMYY4NkPXIC{\&}pgis=1.

[23] ITU. *ITU-T 992.3 Asymmetric digital subscriber line transceivers 2 (ADSL2)*. 2009.

[24] R Jain and S Paul. "Network virtualization and software defined networking for cloud computing: a survey". In: *Communications Magazine, IEEE* (2013). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6658648.

[25] Atta Ur Rehman Khan et al. "A Survey of Mobile Cloud Computing Application Models". In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), pp. 393–413. ISSN: 1553-877X. DOI: 10.1109/SURV.2013.062613.00160. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6553297.

[26] S Larsen and P Sarangam. "Architectural breakdown of end-to-end latency in a TCP/IP network". In: *International Journal of Parallel Programming* (2009). URL: http://link.springer.com/article/10.1007/s10766-009-0109-6.

[27] S. H. Lin. "Statistical Behavior of Multipair Crosstalk". In: *Bell System Technical Journal* 59.6 (1980), pp. 955–974. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1980.tb03041.x. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6771711.

[28] David J C MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[29] JE Mazo. "Faster-than-Nyquist signaling". In: *Bell System Technical Journal, The* (1975). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6772210.

[30] H Nam and BH Kim. "A mobile video traffic analysis: Badly designed video clients can waste network bandwidth". In: *... Workshops (GC Wkshps) ...* (2013). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6825038.

[31] H Nam et al. "Mobile video is inefficient: A traffic analysis". In: *Columbia University Academic Commons* (2013). URL: http://hdl.handle.net/10022/AC:P:21762.

[32] G Nychis et al. "Enabling MAC Protocol Implementations on Software-Defined Radios." In: *NSDI* (2009). URL: https://www.usenix.org/event/nsdi09/tech/full{_}papers/nychis/nychis{_}html/.

[33] John Peng and Stephen G. Wilson. "Achievable rates for reduced-complexity receivers on nonlinear satellite channels with memory". In: *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 4143–4148. ISBN: 978-1-4799-2003-7. DOI: 10.1109/ICC.2014.6883970. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6883970.

[34] John Peng and Stephen G. Wilson. "Maximizing Capacity Subject to a Spectral Mask: Faster than Nyquist or OFDM?" In: *2015 IEEE Latin-America Conference on Communications*. Arequipa, Peru: IEEE, 2015.

[35] John Peng et al. "Two-way (Same-Frequency) Relaying: Information Rates and DVB-S2 Performance". In: *31st AIAA International Communications Satellite Systems Conference*. International Communications Satellite Systems Conferences (ICSSC). American Institute of Aeronautics and Astronautics, 2013. DOI: doi:10.2514/6.2013-5645. URL: http://dx.doi.org/10.2514/6.2013-5645.

[36] Henry D Pfister, Joseph B Soriaga, and Paul H Siegel. "On the achievable information rates of finite state ISI channels". In: *IEEE Global Telecommunicatins Conference GLOBECOM'01, November 25, 2001 - November 29*. Vol. 5. 2001, pp. 2992–2996.

[37] Scott Powell. *Shedding Some Light on Coding Gain*. 2004. URL: http://www.ieee802.org/3/10GBT/public/jan04/powell{_}1{_}0104.pdf.

[38] J G Proakis and M Salehi. *Digital Communications*. McGraw-Hill International Edition. McGraw-Hill, 2008. ISBN: 9780071263788. URL: https://books.google.com/books?id=ksh0GgAACAAJ.

[39] H Rohling. *OFDM: concepts for future communication systems*. 2011. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=r3oD87z4E7UC{\&}oi=fnd{\&}pg=PR4{\&}dq=related:dbZkzTYkjM4J:scholar.google.com/{\&}ots=f-axMX6qui{\&}sig=riAVBmU2G9xhrm-RJaZ7W1DE22E.

[40] F Rusek and JB Anderson. "Constrained capacities for faster-than-Nyquist signaling". In: *Information Theory, IEEE Transactions on* (2009). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4777625.

[41] Adel A M Saleh. "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers". In: *Communications, IEEE Transactions on* 29.11 (1981), pp. 1715–1720. ISSN: 0090-6778.

[42] Mahadev Satyanarayanan et al. "The case for VM-based cloudlets in mobile computing". In: *IEEE Pervasive Computing* 8.4 (2009), pp. 14–23. ISSN: 15361268. DOI: 10.1109/MPRV.2009.82.

[43] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: http://

ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024.

[44] T Starr. *DSL advances*. 2003. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=VWaMdIq2w8QC{\&}oi=fnd{\&}pg=PR13{\&}dq=dsl+advances{\&}ots=rSmWyry{_}PL{\&}sig=k5NmudBOXaPKWFMsztOOZer31SE.

[45] T Starr, JM Cioffi, and PJ Silverman. "Understanding digital subscriber line technology". In: (1999). URL: http://dl.acm.org/citation.cfm?id=SERIES9965.293816.

[46] The Broadband Forum. *TR-043 Protocols at the U Interface for Accessing Data Networks using ATM/DSL*. Tech. rep. The Broadband Forum, 2001.

[47] E Tilevich and YW Kwon. "Cloud-based execution to improve mobile application energy efficiency". In: *Computer* (2014). URL: http://cat.inist.fr/?aModele=afficheN{\&}cpsidt=28178285.

[48] Eli Tilevich and Yannis Smaragdakis. "J-Orchestra: Enhancing Java programs with distribution capabilities". In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 19.1 (2009), pp. 1–40. ISSN: 1049331X. DOI: 10.1145/1555392.1555394. URL: http://apps.webofknowledge.com/full{_}record.do?product=WOS{\&}search{_}mode=CitingArticles{\&}qid=7{\&}SID=4FPkcU6mCXZUQANIHIj{\&}page=1{\&}doc=10.

[49] Stefano Tomasin and N. Benvenuto. "Fractionally spaced non-linear equalization of faster than Nyquist signals". English. In: *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*. Lisbon: IEEE, 2014, pp. 1861–1865. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6952672.

[50] Craaig F. Valenti and Bellcore. *T1E1.4 Spectral Compatability: Cable Crosstalk Parameters and Models*. 1997.

[51] T. Watson. "Effective wireless communication through application partitioning". In: *Proceedings 5th Workshop on Hot Topics in Operating Systems (HotOS-V)*. IEEE Comput. Soc. Press, 1995, pp. 24–27. ISBN: 0-8186-7081-9. DOI: 10.1109/HOTOS.1995.513449. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=513449.

[52] T Watson. *Wit: An infrastructure for wireless palmtop computing*. Department of Computer Science and Engineering, University of Washington, 1994. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.5742{\&}rep=rep1{\&}type=pdf.

[53] Wikipedia. *Web 2.0*. URL: http://en.wikipedia.org/wiki/Web{_}2.0.

[54] S. Wilson, J. Peng, and T. Tie. "On Faster-than-Nyquist Transmission". 2015. URL: sgw@virginia.edu.

[55] J Wu et al. "Cloud radio access network (C-RAN): a primer". In: *Network, IEEE* (2015). URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=7018201.

[56] Chenguang Xu et al. "Achievable information rates for nonlinear satellite channels in unidirectional and bidirectional relaying". In: *2012 IEEE Latin-America Conference on Communications*. IEEE, 2012, pp. 1–6. ISBN: 978-1-4673-5080-8. DOI: 10.1109/LATINCOM.2012.6506009. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6506009.

[57] Chenguang Xu et al. "Improving Decoding Performance of DVB-S2 Transmission on a Nonlinear Channel". In: *31st AIAA International Communications Satellite Systems Conference*. International Communications Satellite Systems Conferences (ICSSC). American Institute of Aeronautics and Astronautics, 2013. DOI: doi:10.2514/6.2013-5644. URL: http://dx.doi.org/10.2514/6.2013-5644.

[58] K Zhang and D Li. *Electromagnetic theory for microwaves and optoelectronics*. 2013. URL: https://books.google.com/books?hl=en{\&}lr={\&}id=7DnvCAAAQBAJ{\&}oi=fnd{\&}pg=PA1{\&}dq=related:uf9J44q79oQJ:scholar.google.com/{\&}ots=lKxQ4ZvKlm{\&}sig=Rtpy{_}zOsOYEIji7nJ8TB-pblN0o.

[59] T.N. Zogakis, P.T. Tong, and J.M. Cioffi. *Performance Comparison of FEC/Interleave Choices with DMT for ADSL*. Tech. rep. Chicago: ANSI Contribution T1E1.4/93-091, 1993.