NOTE TO USERS

Université d'Ottawa • University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

BESSENS, Gilles

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

DSL Wideband Crosstalk Cancellation for Mixed DSL Technologies using Common Mode Information

Martin Bouchard

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

C. D'Amours

R. Dansereau

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

DSL Wideband Crosstalk Cancellation for Mixed DSL Technologies using Common Mode Information

Gilles Bessens, B. Eng.

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Applied Science

in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

March 2004

Your file Votre référence
ISBN: 0-494-01417-2
Our file Notre référence
ISBN: 0-494-01417-2

Abstract

Data services delivered on telephone twisted pair wiring can potentially suffer from crosstalk due to data services on other twisted pairs in proximity, limiting the data rate, the maximum loop length and the quantity and variety of data services that a bundle can support. The crosstalk from other data services couples to the twisted pair in common mode and leaks to differential mode due to imperfect line balance. This thesis uses the common mode signal as a reference to an adaptive wideband crosstalk canceller, in an attempt to remove or at least lower the effect of crosstalk on the useful differential signal.

Simulation results show the potential benefits of using this technique to reduce the level of crosstalk on an ADSL differential signal with crosstalk coming from data services commonly found in the telephone bundles. This technique is not limited to ADSL but can be applied to any technology using twisted pair wiring as its transmission medium.

Acknowledgments

I am very grateful for the advice and support of my supervisor Dr. Martin Bouchard for his continuous feedback on my research. I am indebted to my proofreaders, Marie-Claire and Gérard Bessens, whose comments have greatly improved this thesis. I would like to thank the University of Ottawa staff and administration for their help throughout my studies. I am particularly thankful for the valuable input and feedback of Dave Fenton on my first conference paper and article. I am also especially thankful to my colleague and friend Homayoun Kamkar-Parsi for the help with the simulation and the many interesting and informative conversations.

Gilles Bessens

Contents

List of Figures

List of Tables

List of Acronyms

2B1Q	Two-Binary One-Quaternary
4B3Q	Four-Binary Three-Quaternary
ADSL	Asymmetric Digital Subscriber Line
AWG	American Wire Gauge
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BRI	Basic Rate ISDN
CAP	Carrierless Amplitude Modulation
CLEC	Competitive Local Exchange Carrier
CO	Central Office
CRC	Cyclic Redundancy Check
CSA	Carrier Serving Area
dB	Decibel
DFT	Discrete Fourier Transform
DMT	Discrete Multitone
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DSM	Dynamic Spectrum Management
FAP	Fast Affine Projection
FDD	Frequency Division Duplexing
FDD-ADSL	Frequency Division Duplexed ADSL
FDI	Fibre Distribution Interface
FEQ	Frequency Equalizer
FEXT	Far End Crosstalk
FFT	Fast Fourier Transform
FTF	Fast Transversal Filtering
HAM	Amateur Radio
HDSL	High Bit Rate DSL
HDSL2	Second Generation High Bit Rate DSL

HDSL4	Extended Reach DSL1 transmission OR 2 pair SHDSL
HDTV	High Definition Television
Hz	Hertz (frequency unit)
IDFT	Inverse Discrete Fourier Transform
ILEC	Incumbent Local Exchange Carrier
IFFT	Inverse Fast Fourier Transform
ISDN	Integrated Services Digital Network
ISDN-BRA	Integrated Services Digital Network Basic Rate
ISDN-PRA	Integrated Services Digital Network Primary Rate
ISI	Inter Symbol Interference
kb/s	Kilobits per second
LMS	Least Mean Square
Mb/s	Megabits per second
NEXT	Near End Crosstalk
ONU	Optical Network Unit
PAM	Pulse Amplitude Modulation
PRI	Primary Rate ISDN
PSD	Power Spectral Density
QAM	Quadrature Amplitude Modulation
RFI	Radio Frequency Interference
RLS	Recursive Least Squares
RMS	Root Mean Square
RRD	Revised Resistance Design
RS	Reed-Solomon
SDSL	Single line DSL
SFEXT	Self-NEXT
SHDSL	Symmetric High Speed Digital Subscriber Line
SNEXT	Self-FEXT
SNR	Signal to Noise Ratio
TEQ	Time Equalizer
UTP	Unshielded Twisted Pair
VDSL	Very-high-bit-rate DSL
XDSL	Any kind of DSL

Chapter 1

Introduction

1.1 Problem Definition

Data transmission services used on telephone twisted pair wiring can suffer from crosstalk from data services on other twisted pairs in proximity, usually in the same bundle. This crosstalk can be a limiting factor for the data rate, the maximum loop length and the quantity and variety of data services that a bundle can support. The crosstalk from other data services couples to the twisted pair in common mode, then leaks to differential mode due to imperfect line balance. The key problem addressed in this thesis is the reduction of the crosstalk level in the useful differential signal.

1.2 The Current State of the Art

1.2.1 Static Spectrum Management

When other companies (Competitive Local Exchange Carriers or CLECs) were given the right to compete with the local phone company (Incumbent Local Exchange Carriers or ILECs) to offer data services (called "unbundling"), they were allowed to deploy their own equipment at the ILEC premises. The competing companies were not necessarily offering data services compatible with each other or with the ILEC. This problem

mandated some form of spectrum management that evolved into the standard T1.417-2001 [SM2001]. The standard is essentially a series of deployment guidelines to minimize the problems associated with crosstalk, placing restrictions on the variety and quantity of data services in the same or neighboring bundles, as well as line lengths, bit rates and transmitted power of each data service that the standard encompasses. Spectrum management is a predetermined deployment plan that limits the effect of disruptive crosstalk that data services generate on each other. The standard also defines several guidelines that new technologies have to adhere to. The deployment plan is based on measurements and calculations of the interactions between the different data services, but presently covers only cases where service is offered from the Central Office (CO). [KER02] states that each data transmission technology is designed to work at its specified rate under worst case crosstalk, and the T1.417-2001 standard is drafted considering the worst case crosstalk scenario for each technology. This leads to an overly pessimistic and restrictive deployment in terms of data rate, distance and data service density.

1.2.2 Dynamic Spectrum Management

Since the static spectrum management standard can be overly restrictive, it will needlessly hinder the advancement of higher speed data services. Methods that will dynamically minimize the transmitted power and spectrum to maintain the desired target data rate are in development.

One of the methods can be as simple as letting the bit loading algorithm place its bits at frequencies where the Signal to Noise Ratio (SNR) is best, and only transmit the minimum power required to achieve the level of quality of service [SON02] [MEL95], iterating until stable. This type of bit loading algorithm is already implemented in many Digital Subscriber Line (DSL) modems, including this simulation, but the minimization of transmitted power is not, which negates the benefit. Unfortunately, to work properly, this method requires that all modems implement this solution, so that they might eventually converge to a good solution.

Most other Dynamic Spectrum Management (DSM) methods rely on an extensive knowledge of the transmission line, the data services they provide, including their transmitted power and bit rate. A central entity would then compute the best operating point by controlling each modem's output power and operating parameters to achieve optimal performance. Some methods require the knowledge of the transfer functions between each twisted pair line in the bundle offering data services. [ZEN01] presents a method to evaluate these transfer functions, but requires that all the modems return a part of their incoming stream to the central entity. Some other methods include vectored Discrete Multitone (DMT) [SON02], which requires synchronization between the modems (even between providers to be most effective).

DSM is still the subject of very interesting long term research. Most of these methods require communication between the modems and a central entity. Unbundling has permitted CLECs to place their own equipment (namely DSL Access Multiplexers (DSLAMs)) at the sites, which will complicate or slow the adoption of any standardization of communication and feedback to a central entity.

Other interesting techniques aim at cancelling Near-End crosstalk (NEXT) only [ZEN02], as it is the main impairment when all the twisted pair lines are of the same length. Frequency Division Duplexed (FDD) technologies suffer much less from NEXT interference, as is the case for ADSL and VDSL. Additionally, the Far-End crosstalk (FEXT) can sometimes have a level that is not negligible, mostly in short lines

Using the common mode as a reference for crosstalk cancelling does not require that all modems implement a standard feedback, but it is a solution that could be implemented individually by manufacturers, which makes the investigation of this technique appealing. Another appealing feature is that, if it can be proven that this technique cancels crosstalk, it can potentially be applied to suppress Radio Frequency Interference (RFI) and the impulse noise.

1.3 Previous Work in the Area

[COM98] presents an analog circuit placed in series with the twisted pair line to null out the common mode signal. The theory is that nulling out the common mode signal (interference) would prevent it from leaking into the differential mode through lack of line balance. This can give good results, but requires the deployment of field personnel for the installation. The circuit has to be placed after each major imbalance point, which is usually considered where a bridged tap connection is made (or at every location where the line is untwisted to make a connection). Locating all of these imbalance points is a major inconvenience, although only those close to the CO and the customer are critical, since the only damaging FEXT is actually the one that is coupled closest to either receiver.

[LEF00] and [YEA60] present a narrowband adaptive noise canceller using the common mode as a reference to cancel radio frequency interference from Very-high-bit-rate DSL (VDSL) signals. [YEA03] extends this slightly to RFI noise of slightly larger bandwidth. This concept is similar to the one developed in this thesis, although the application (RFI vs. Crosstalk), noise models (narrowband radio vs. wideband multiple DSL source) and simulations are very different, since these references use Quadrature Amplitude Modulation (QAM) as the modulations technique and this thesis uses DMT combined with Reed-Solomon forward error correction.

[FEN99] examines the feasibility of the physical hardware implementation of a common mode canceller. This analysis includes common mode signal sampling finite precision effects, common mode signal acquisition analog front end requirements, design and implementation as well as the number of filter taps requirements. [FEN99] provides simulation results for narrowband RFI cancellation in a VDSL QAM modulated signal using the common mode.

1.4 Goal of Thesis

The purpose of this research is to expand on the idea of using the common mode signal as a reference to an adaptive crosstalk canceller in order to remove the crosstalk in the differential signal. This simulation will abstract from the hardware implementation (without ignoring it) and will implement the entire Asymmetric Digital Subscriber Line (ADSL) protocol stack using DMT modulation and Reed-Solomon forward error correction to examine the usefulness of the technique in an environment where crosstalk is generated by multiple DSL technologies in one bundle. A novel crosstalk model will be developed to combine the crosstalk from different sources originating from the CO. The simulation will analyze the common mode canceller's performance based on the improvement (or degradation) of the Bit Error Rate (BER). Any improvement (lowering) of the BER will translate in the possibility to increase the bit rate, the reach of DSL service, or the number of services that a bundle can support.

1.5 Thesis Structure

Chapter 2 introduces the various DSL data transmission technologies available on twisted pair wiring that can cause crosstalk to other DSL data services. The overview of these DSL technologies is done from the perspective of using the information to generate a realistic crosstalk signal and simulation scenarios. The knowledge sought is primarily the maximum distances, the corresponding Power Spectral Density (PSD) spectrums for different configurations of each technology and in which these technologies are likely to be found.

Chapter 3 introduces ADSL more formally than Chapter 2 and describes ADSL's protocol stack top layers, highlighting the forward error correction layer, which will be essential in evaluating the crosstalk canceller's performance. The lower layers are covered in Chapters 4.

Chapter 4 describes the DMT modulation, and the adaptation and modification of a bit loading algorithm, followed by the actual modulation and remaining details.

Chapter 5 discusses the differential twisted pair channel model and derives a model for the common mode channel that will be used in Chapter 6 to generate the crosstalk. Important additional loop impairments are covered at the end of the chapter.

Chapter 6 discusses the different types of crosstalk and introduces the novel crosstalk model of multiple crosstalk sources in the same bundle, its advantages and shortcomings. This model generates the differential and common mode crosstalk signals.

Chapter 7 starts by detailing the simulation and the crosstalk canceller, then describes the performance measurements that were taken on the crosstalk canceller and the measure of its effectiveness in several scenarios through simulations. The chapter concludes with an analysis of the results.

Chapter 8 summarizes the key conclusions of the simulations performed in Chapter 7 and discusses any possible future work in the field.

Original work is scattered throughout the thesis in order to maintain the flow of ideas but is mostly contained in the second half, where the novel model of common mode crosstalk is presented.

Chapter 2

Technologies Causing Crosstalk

This chapter enumerates and describes the different types of technologies that can be found in a bundle. This is important because of the necessity to have a basic understanding of the conditions in which the crosstalk canceller will have to operate to achieve realistic situations. Should the conditions in which the crosstalk canceller has to operate be exaggerated, they will still be kept realistic. Nevertheless, the most important element required for this simulation from this section is the transmitted Power Spectral Density (PSD) of each type of technology, which will be used to generate the Near-End crosstalk (NEXT) and Far-End crosstalk (FEXT) signals using equations 6-2 and 6-4 of Chapter 6.

2.1 T1 (DS1)

In North America, T1 links were originally designed to connect Central Offices (CO) at a rate of 1.544 Mb/s (Mega bits per second) in each direction using Alternate Mark Inversion (AMI). Europe's equivalent is E1, which transports data at a rate of 2.048 Mb/s using High Density Bipolar Order 3 (HDB3) modulation.

A T1 link requires 2 twisted pairs in simplex operation, which means one pair for each direction. A T1 line requires repeaters to operate on a lengthy line and is limited to lines with few or no bridged taps. The distance limitations of Figure 2-1 are approximate because they were specified at a time when the T1 lines were between two CO and had 19 or 22 AWG (American Wire Gauge), but now the T1 is more commonly deployed on 26 or 24 AWG. The true inter-repeater distance limitation is the attenuation of the line, which is shown in Figure 2-1.

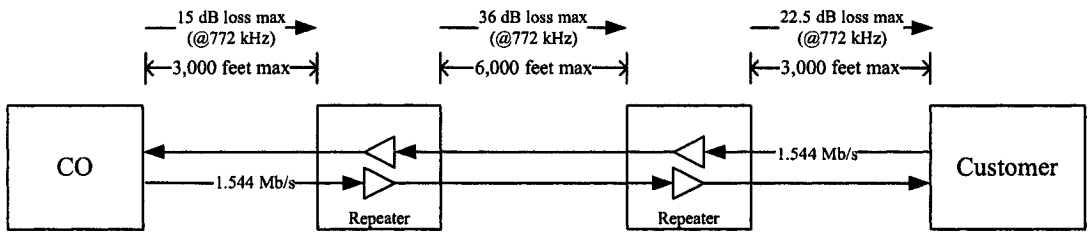


Figure 2-1: T1 link with its repeaters and attenuation limitations

The problem with repeaters is that they cause severe crosstalk when they are near a pair transporting another data service (Downstream ADSL in Figure 2-2). The crosstalk can somewhat be reduced if the two twisted pair lines delivering T1 are placed in separate bundles.

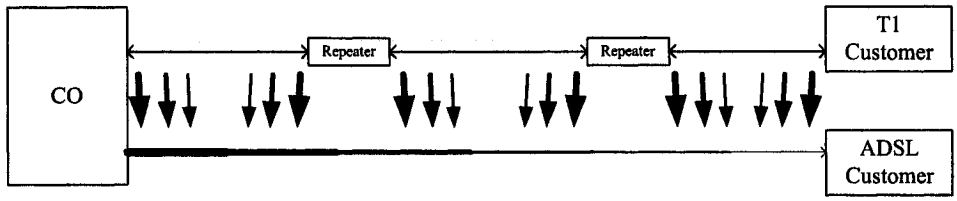


Figure 2-2: T1 causing severe crosstalk onto other DSL service in same bundle

The T1 transmit PSD used for this simulation is defined by equation 2-1 [IEC03] and plotted in Figure 2-3 for $f = 0$ to 1.104 MHz.

$$PSD_{T1} = \frac{V_p^2}{R_L} \cdot \frac{2}{f_0} \cdot \left[\frac{\sin\left(\frac{\pi f}{f_0}\right)}{\left(\frac{\pi f}{f_0}\right)} \right]^2 \cdot \sin^2\left(\frac{\pi f}{2 \cdot f_0}\right) \cdot \frac{1}{1 + \left(\frac{f}{f_{3dB-Shaping}}\right)^6} \cdot \frac{f^2}{f^2 + f_{3dB-Xfmr}^2} \qquad (2\text{-}1)$$

with $Vp = 3.6$ V, $R_L = 100\ \Omega$, $f_0 = 1.544$ MHz, $f_{3dB\text{-}Shaping} = 3$ MHz and $f_{3dB\text{-}Xfmr} = 40$ kHz.

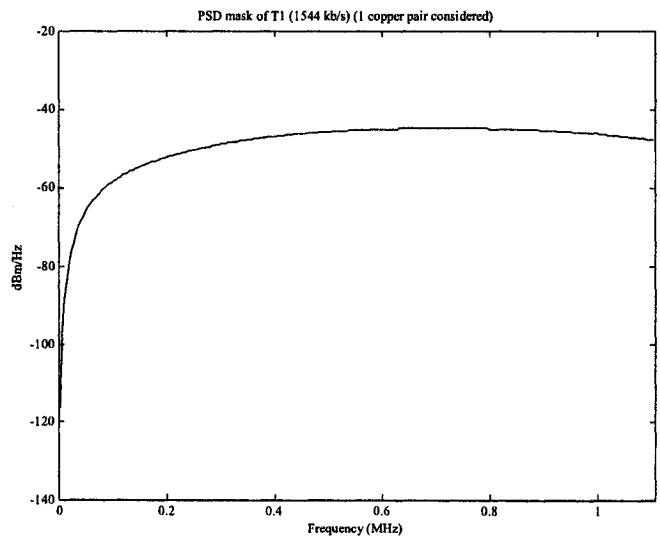


Figure 2-3: AMI T1 transmitter PSD

Since ADSL's spectrum is restricted to below 1.104 MHz, the spectrum of interest of the T1 line stops at 1.104 MHz, but T1's extends beyond 2 MHz.

2.2 ISDN

Integrated Services Digital Network (ISDN) was developed after T1 and was intended to deliver digital services directly to the home encompassing video, telephone, and data. The resulting standard adopted in the mid-eighties was for Two-Binary One-Quaternary (2B1Q) transmission in North America at a symmetrical data rate of 160 kb/s (including overhead, 128 kb/s + 16 kb/s available) on one twisted pair using echo cancellation. The maximum length of the twisted pair line is approximately 18,000 feet. This standard is referred to as Basic Rate ISDN (BRI or ISDN-BRA). The ISDN-BRA transmit PSD is defined by equation 2-2 [VOD02] and is plotted in Figure 2-4.

$$PSD_{ISND-BRA}(f) = \frac{5 \cdot V_p^2}{9 \cdot R_L} \cdot \frac{2}{f_0} \cdot \frac{\sin^2\left(\frac{\pi f}{f_0}\right)}{\left(\frac{\pi f}{f_0}\right)^2} \cdot \frac{1}{1+\left(\frac{f}{f_{3dB}}\right)^4} \tag{2-2}$$

with f_{3dB} = 40 kHz, Vp = 2.5 Volts, R_L = 135 Ω and f_0 = 80 kHz.

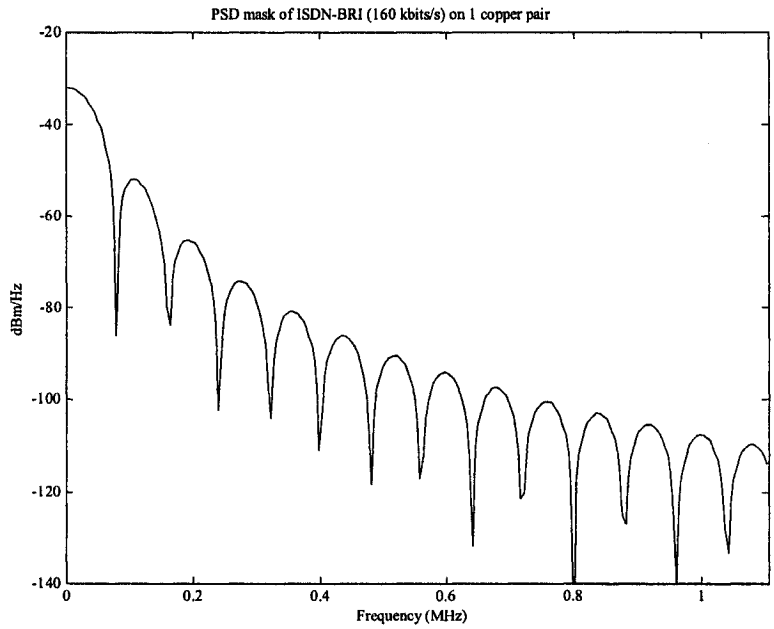


Figure 2-4: Transmit PSD for Basic Rate ISDN from 0 to 1.104 MHz

Primary Rate ISDN (PRI or ISDN-PRA) delivers a symmetrical 2.048 Mb/s link on 2 twisted pairs using HDB3 transmission. Each twisted pair supports one transmit direction at a distance that is equivalent to the T1 distance limitations unless repeaters are used. The transmit PSD is defined by equation 2-3 [VOD02] and is plotted in Figure 2-5.

$$PSD_{ISND-PRA}(f) = \frac{V_p^2}{R_L} \cdot \frac{2}{f_0} \cdot \frac{\sin^2\left(\frac{\pi f}{f_0}\right)}{\left(\frac{\pi f}{f_0}\right)^2} \cdot \sin^2\left(\frac{\pi f}{f_0}\right) \cdot \frac{1}{1+\left(\frac{f}{f_{3dB}}\right)^6} \cdot \frac{f^2}{f^2 + f_{3dB}^2} \tag{2-3}$$

where f_{3dB} = 1.2 MHz, Vp = 3 Volts, R_L = 120 Ω and f_0 = 2.048 MHz.

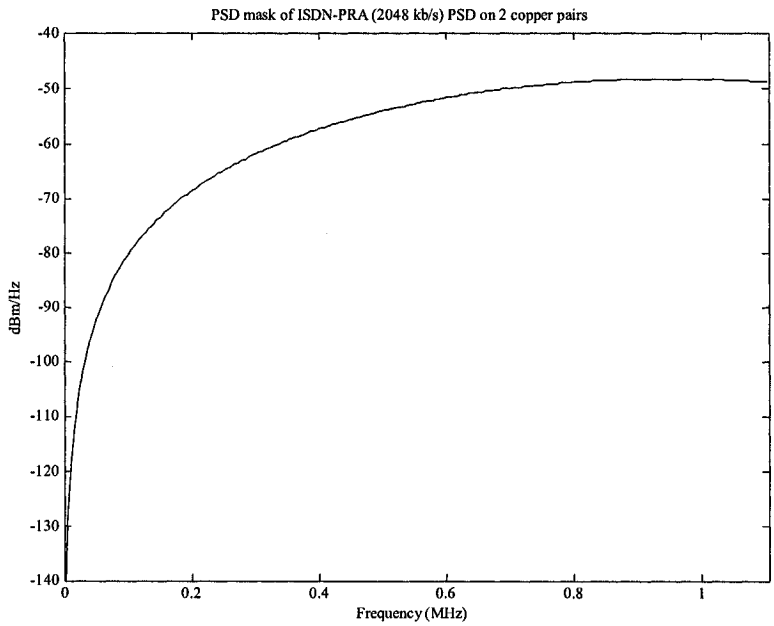


Figure 2-5: Transmit PSD for Primary Rate ISDN from 0 to 1.104 MHz

2.3 HDSL

High Bit Rate DSL (HDSL) was introduced to replace T1 (DS1) because DS1 circuits required many resources to be deployed and maintained. The DS1 lines necessitate the installation of repeaters every 6,000 feet that create a lot of crosstalk on adjacent lines. The installation of a DS1 line with all its repeaters can take up to 3 months and requires much manual work. HDSL can be up and running in less than 3 days and can be deployed on a line of 9,000 feet maximum at 26 gauge or 12,000 feet maximum at 24 gauge without using a repeater. HDSL is meant to be a drop-in replacement for the T1 link.

The first HDSL was never standardized, although some technical specifications were published. In Europe, a technical specification for HDSL was published that described how to have the E1 signal (2.048 Mb/s, Europe's equivalent to North America's T1/DS1 1.548 Mb/s) over HDSL using 1, 2 or 3 twisted pairs, each in full-duplex operation using echo cancellation. In Europe, with 3 twisted pairs, the data rate is 2.304 Mb/s in both directions, with each twisted pair at 784 kb/s. In the European 2 twisted pair HDSL link, each pair has a data rate of 1.168 Mb/s in full duplex operation.

In North America, there is no three-pair HDSL, therefore a T1 1.548 Mb/s link would have 784 kb/s data rate on each twisted pair line in both directions, for a total of 1.568 Mb/s in both directions and is displayed in Figure 2-6:

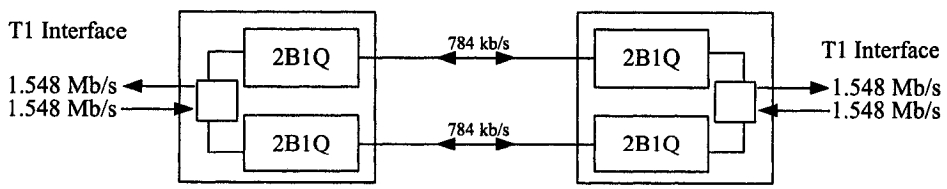


Figure 2-6: Twisted pair HDSL for interface to North America's T1 (DS1)

Each 2B1Q transmitter has a transmit PSD spectrum modelled by the following equation [STA02]:

$$PSD_{HDSL}(f) = K_{HDSL} \cdot \left(\frac{2}{f_0}\right) \cdot \frac{\left[\sin\left(\frac{\pi f}{f_0}\right)\right]^2}{\left(\frac{\pi f}{f_0}\right)} \cdot \frac{1}{1+\left(\frac{f}{f_{3db}}\right)^8} \tag{2-4}$$

where $K_{HDSL} = \frac{5}{9} \cdot \frac{V_p^2}{R}, V_p = 270\,V, R = 135\,\Omega$ and:

f_0: 392 kHz (for HDSL using 2 pairs in North America or 3 pairs in Europe)

f_{3dB}: 196 kHz (for HDSL using 2 pairs in North America or 3 pairs in Europe)

f_0: 584 kHz (for HDSL using 2 pairs in Europe)

f_{3dB}: 292 kHz (for HDSL using 2 pairs in Europe)

f_0: 984 kHz (for Single Line DSL (SDSL) using 1 twisted pair in Europe)

f_{3dB}: 492 kHz (for SDSL using 1 twisted pair in Europe)

Figure 2-7 displays the spectrums from equation 2-4 for 2 twisted pairs in North America. The European HDSL with 3 twisted pairs will not be used in this simulation because the spectrum of one transmitter is the same as its North American counterpart with two twisted pairs, with the addition of a transmitter per link. Increasing the number of 2-pair HDSL links in the bundle will show the effect of the crosstalk on ADSL.

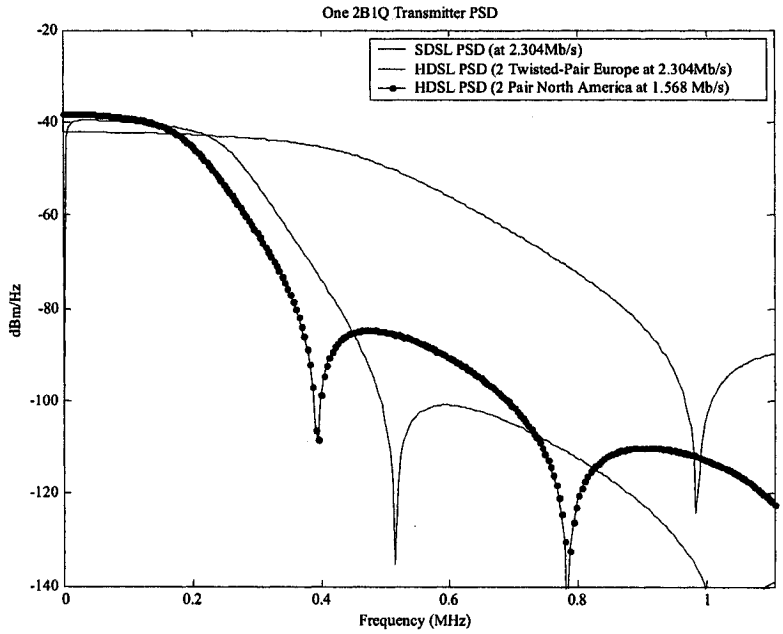


Figure 2-7: HDSL transmit PSD for three different types

HDSL Version 2 (HDSL2) will not be covered in this simulation because it was specifically designed to cause less interference to other services than normal HDSL. HDSL2 is a single twisted pair transmission protocol that uses Trellis modulation and has the same goal as HDSL. It should not be confused with SHDSL (Symmetric HDSL) or SDSL (Single Line DSL or Single-Pair DSL) that both have the same goal (to replace T1 without repeaters) but have different modulation schemes or PSD masks. Full Speed SDSL is limited to around 6,000 feet, lower speed at 13,000 feet, and HDSL2 to 12,000 feet 24 gauge. HDSL4 is a new standard to replace T1 that was developed to extend the reach to 11,000 feet with 26 gauge instead of 9,000 feet, but uses two twisted pairs instead of one. This simulation will consider these 3 types of HDSL:

- SDSL at 2.304 Mb/s (Single Pair HDSL using 2B1Q Modulation)
- HDSL at 1.544 Mb/s (North American 2 twisted pair using 2B1Q Modulation)
- SHDSL at various data rates (Symmetric HDSL using Trellis-Coded PAM)

2.4 SHDSL

The purpose of Symmetric HDSL (SHDSL) was, like HDSL, to replace T1 (DS1) in North America and E1 in Europe, but using only one twisted pair, much like SDSL and HDSL2. The main difference between SHDSL and SDSL is that SHDSL uses a Trellis Coded 16 Level Pulse Amplitude Modulation (PAM) technique, which permits SHDSL to deliver a slightly higher bit rate than 2B1Q SDSL is capable. The main difference between SHDSL and HDSL2 is that SHDSL permits the replacement of E1 2.048 Mb/s links in Europe or the North American T1 DS1 1.544 Mb/s link, whereas HDSL2 was intended to replace T1 only and therefore has a fixed bit rate. The reach of the single pair SHDSL is 9,000 feet with a 26 gauge wire, or 11,000 feet with a 24 gauge wire (Carrier Serving Area (CSA) rules) for the full 2.304 Mb/s and 15,000-20,000 feet at 192-256 kb/s (2.304 Mb/s is available at 15,000 feet when using 2 pairs (optional)).

SHDSL can transport data at various data rates and the PSD of the transmitter is defined in Equation 2-5 [STA02]. Depending on the implementation, it can define a noise floor, which makes a flatter PSD at higher frequencies.

$$PSD_{SHDSL}(f) = \begin{cases} \dfrac{K_{SHDSL}}{R \cdot f_{Sym}} \cdot \dfrac{\sin^2\left(\dfrac{\pi f}{f_{Sym}}\right)}{\left(\dfrac{\pi f}{f_{Sym}}\right)^2} \cdot \dfrac{1}{1+\left(\dfrac{f}{f_{3db}}\right)^{2\text{-}Order}} \cdot \dfrac{f^2}{f^2+f_C^2}, & f < f_{Intercept} \\ 0.5683 \cdot 10^{-4} \cdot f^{-1.5}, & f \geq f_{Intercept} \end{cases} \quad (2\text{-}5)$$

Where R = 135 Ω, $Order$ = 6, f_{Sym}= $Bit\ Rate/3$ f_c = 500 kHz and K_{SHDSL} and f_{3dB} are determined in the following table:

Table 2-1: Parameters for equation depending on bit rate

Bit Rate *BR*	K_{SHDSL}	f_{3dB}
BR > 1.536 Mb/s	7.86	$f_{Sym}/2$
BR = 1.536 Mb/s or 1.544 Mb/s	8.32	$0.45 \cdot f_{Sym}$
1.544 Mb/s < *BR* < 2.048 Mb/s	7.86	$f_{Sym}/2$
BR ≥ 2.048 Mb/s	9.90	$f_{Sym}/2$

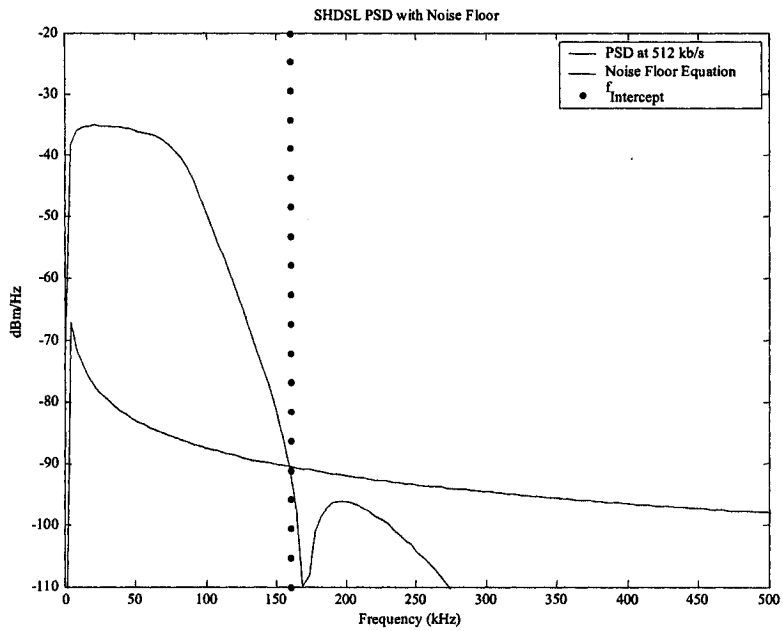


Figure 2-8: Intercept frequency location

The frequency at which both plotted functions intersect in Figure 2-8 is named $f_{Intercept}$, and PSD_{SHDSL} is set to the noise floor past this frequency. The resulting SHDSL transmit PSD with noise floor is shown in Figure 2-9 for various bit rates.

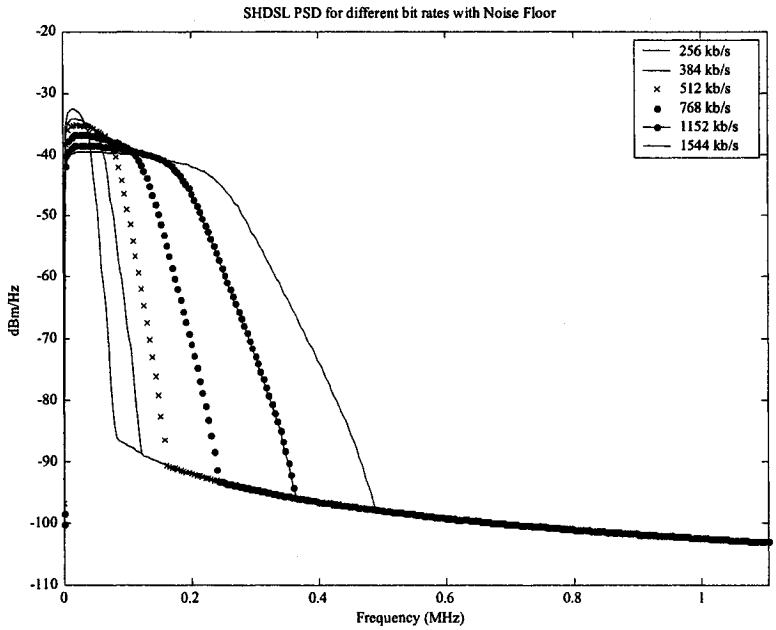


Figure 2-9: SHDSL transmit PSD for different bit rates

SHDSL has superseded the SDSL standard. SHDSL can be used for many applications using symmetric data transmission, including linking the CO to wireless base stations.

2.5 Asymmetric Digital Subscriber Line

Asymmetric Digital Subscriber Line (ADSL) can cause crosstalk to other ADSL systems, when the same technology is the crosstalker, it is called Self crosstalk, or Self-NEXT (SNEXT) and Self-FEXT (SFEXT). ADSL's spectrum can range from 0 Hz to 1.104 MHz with speeds up to 6 Mb/s (this speed can even be extended). At lowers speeds, ADSL loops can have a maximum length of 18,000 feet (on CSA loops or better). Equation 2-6 represents the ADSL PSD masks that were used in the simulation, where the parameters are changed to specify the upstream or downstream mask.

$$PSD_{ADSL}(f) = K_{ADSL} \cdot \left(\frac{2}{f_0}\right) \cdot \frac{\sin\left(\frac{\pi f}{f_0}\right)^2}{\left(\frac{\pi f}{f_0}\right)^2} \cdot \frac{1}{\left(1+\left(\frac{f}{f_{LP3db}}\right)^K\right)} \cdot \frac{1}{\left(1+\left(\frac{f_{HP3db}}{f}\right)^N\right)} \tag{2-6}$$

with the parameters for up and downstream in the following table:

Table 2-2: Parameters for equation 2-6 depending on PSD mask data direction

Parameter	Upstream	Downstream
K_{ADSL}	0.02187 Watts	0.1104 Watts
f_{HP3dB}	25.875 kHz	307 kHz
f_0	276 kHz	2.208 MHz
f_{LP3dB}	138 kHz	$f_0/2$
K	16	12
N	8	16

The result of equation 2-6 for the upstream (dashed line) and downstream (full line) directions is plotted in Figure 2-10. The choice of using the formula instead of the linear interpolation of the given PSD masks in the ADSL standard G992.1 [G9921] was determined by the fact that the Yule-Walker function needed for the *PSD Shaping* blocks (Figure 6-4 of Chapter 6) could more easily generate a transfer function that followed the

equation rather than the mask of the standard to generate the NEXT and FEXT time samples.

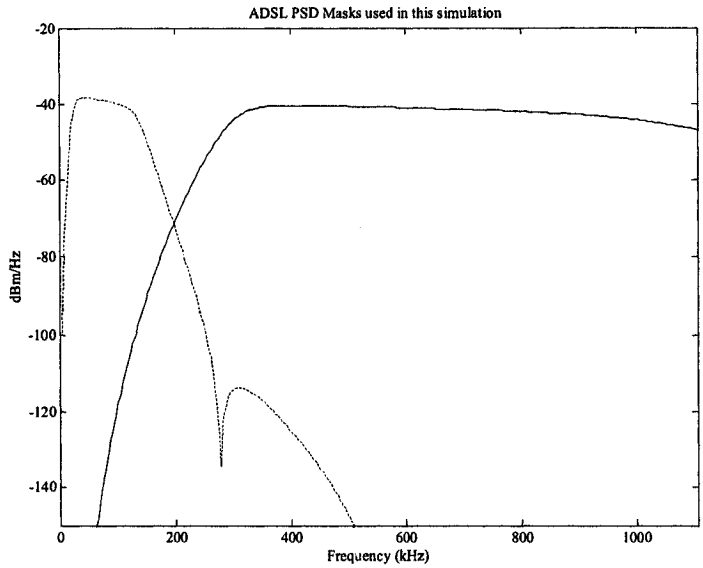


Figure 2-10: ADSL PSD masks used in this simulation

In fact, the spectrum that an ADSL signal occupies varies with the data rate and channel characteristics (bit loading). In this simulation, the entire bandwidth of each of the masks is used to create all the self-crosstalk cases. This will create more crosstalk than necessary, at frequencies that are not necessarily used by the considered ADSL at slower data rates. A crosstalker using that bandwidth for ADSL would probably be delivering data rates of 6 Mb/sec to a customer and most ADSL deployed will have similar spectral characteristics in the same bundle. The crosstalk that is damaging to the considered ADSL line is the crosstalk overlapping in frequency with the useful signal. The additional crosstalk at a higher frequency will not cause more damage than if it had not been there, therefore only one mask is used for simplicity and because it would make little difference. However, the entire ADSL spectrum is not used for the considered line; tones that are not used are set to zero power. However, if the tone is used, the power density is approximately –40 dBm/Hz.

2.6 VDSL Technology

The considered VDSL (Very-high-bit-rate DSL) technology as an interferer to ADSL in this simulation is the asymmetric version delivered from the CO with the data delivery speeds in Table 2-3.

Table 2-3: VDSL data rate vs reach [BIN00]

Range (feet)	Downstream (Mb/s)	Upstream (Mb/s)
1,000	52	6.4
3,000	26	3.2
4,500	13	1.6
6,000	6.5	1.6

Since VDSL uses a wider bandwidth than all other DSL technologies, its spectrum management plan 998 for deployment in North America, shown in Figure 2-11, is complex as it overlaps many other services in frequency (including HAM radio).

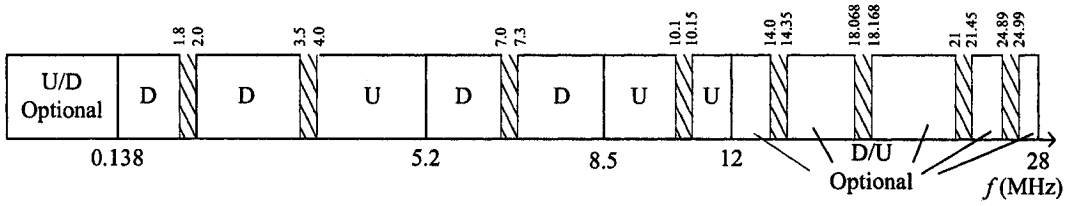


Figure 2-11: VDSL Plan 998 spectrum management [STA02]

Depending on the VDSL data rates and the channel characteristics, the amount of bandwidth and power that will be used by VDSL to send its data will vary, but will almost certainly use the spectrum from 0-1.104 MHz at close to full power. The rest of the power will vary at higher frequencies, but this is not interesting for this simulation, so the full PSD mask from 0-1.104 MHz is used.

The VDSL PSD masks were taken from the VDSL DMT Metallic Interface T1E1.4/2003-210R2 VDSL standard [VDM03]. At the frequencies considered for disturbance with ADSL, the PSD masks for the VDSL DMT and the VDSL QAM draft standards [VQA03] are identical at the time of writing, therefore no distinction is made between the two in this document and they are both referred to as VDSL. These masks are still under study. The VDSL specifications define PSD templates and the masks can be derived from these templates using the relation: $PSD_{Mask} = PSD_{Template} + 3.5$ dB. The mask will be used instead of the template to keep it identical to the other technologies, even though the template is a more accurate approximation of the actual average transmitted PSD. Power back off was not considered to build the interference model as it is still under study for the downstream at the time of writing. It relies on the insertion loss to calculate the minimal power that VDSL can send its data and still be received with a certain amount of accuracy to minimize crosstalk. This simulation considers the maximal mask to generate VDSL crosstalk for a worst case scenario.

The VDSL PSD masks frequency values used for the simulation are given in the following table, linear interpolation is used to calculate the masks:

Table 2-4: VDSL PSD masks for upstream and downstream if deployed from the central office and the optional band is given to the upstream

Frequency (kHz)	PSD_{Mask} (dBm/Hz)	
	Upstream	Downstream
0-4	-97.5	-97.5
25	-34.5	-97.5
138	-34.5	-36.5
307	-86.5	-36.5
482	-96.5	-36.5
1104	-97.5	-36.5

The masks for the upstream and downstream are plotted in Figure 2-12.

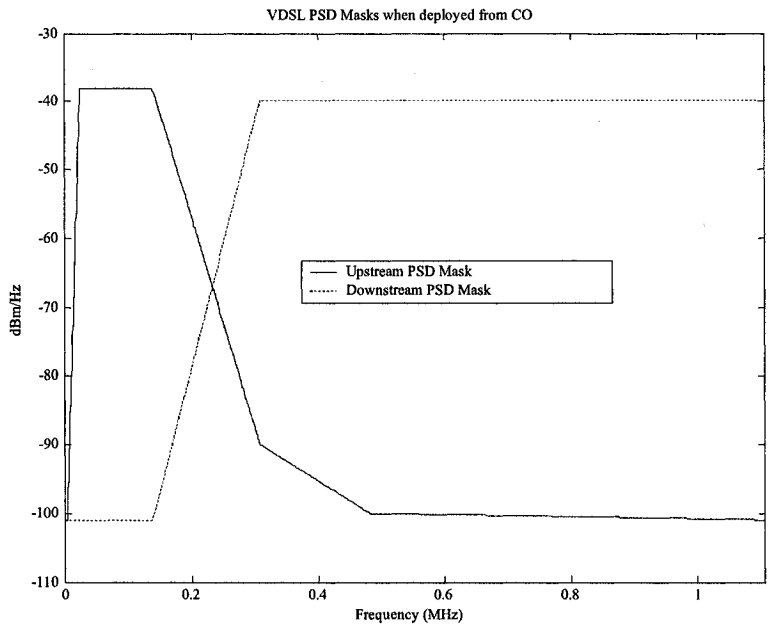


Figure 2-12: VDSL PSD masks when deployed from the CO (when using the optional band for upstream)

The upstream and downstream masks can both use the optional band between 4 kHz and 138 kHz. Actually, since FDD is used, only one of them can use this band. For the simulation, the optional band is given to the upstream VDSL. The reason is that, if VDSL were to be deployed from the CO, the ADSL services already in place in the bundle should not be affected. Therefore, VDSL would probably be deployed with the optional band allotted to the upstream direction.

2.7 Summary

Table 2-5 gives a summary of the comparison between different transmission technologies for use on twisted pairs. The numbers quoted are approximate; the majority of technologies are stated without the use of repeaters and the numbers are mostly for North American 26-gauge loops.

Table 2-5: Transmission technologies comparison for this simulation

Technology	Speed (bits/second)	Loop Reach (x1000 feet)	# of pairs
T1 (DS1)	1.5 Mb/s	3 to repeater 6 between repeaters	2
ISDN-BRA	128 kb/s (160 kb/s)	15 - 18	1
ISDN-PRA	2M	Same as T1	2
ADSL	192 k–8 Mb/s downstream 64 k–900 kb/s upstream	15 at 256 kb/s	1
HDSL	1.5 Mb/s	9	2
HDSL2	1.5 Mb/s	9	1
HDSL4	1.5 Mb/s	11	2
VDSL	6-52 Mb/s downstream 1.5-6.5 Mb/s upstream	1 – 6	1
SHDSL	192kb/s –2.3 Mb/s	9 at 2.3 Mb/s single pair 15 at 192 kb/s single pair 15 at 2.3 Mb/s dual pair	1 or 2
SDSL	256kb/s –2.3 Mb/s	16 at 256 kb/s 9 at 1.5 Mb/s	1

Chapter 3

ADSL and its Protocol Stack

Any digital subscriber line technology could have been implemented to measure the efficiency of the crosstalk canceller. The choice to use Asymmetric Digital Subscriber Line (ADSL) as the reference technology for the simulation was the availability of documentation on the technology, the maturity of the standard at the start of the project combined with the fact that ADSL is asymmetrical, which gave it some interesting properties. It also presented an option of Frequency Division Duplexing (FDD), which made it easier to justify the use of only one direction at a time in the simulation.

The ADSL is a technology developed to serve the residential customer. It was designed to operate on a single twisted pair line in conjunction with the regular Plain Old Telephone Service (POTS) unlike many of the other DSL technologies. ADSL operates at frequencies above the voice band of POTS and can be combined in several ways to the POTS service using splitters or filters. An approximate spectrum of full rate ADSL combining POTS is illustrated in Figure 3-1.

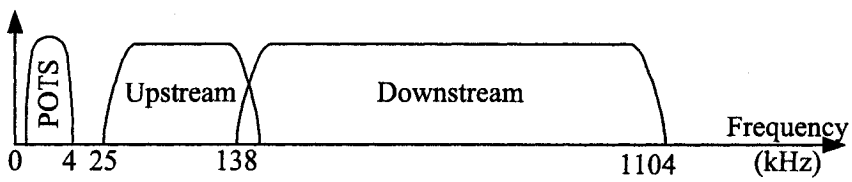


Figure 3-1: POTS and FDD-ADSL, upstream and downstream

The ratio of downstream data rate to upstream data rate is approximately 8 to 1 to reflect and exploit the asymmetrical nature of typical Internet traffic. In North America, typical data rates range from 1.5 Mb/s to 6 Mb/s downstream and 384 kb/s to 784 kb/s upstream, and are determined by the service provider (lower and higher data rates are available). A chopped-down version of ADSL, called ADSL lite, is available. It uses approximately half the bandwidth and is meant for customers with a lower budget or with a lesser need for high data rates.

The ADSL preliminary draft standard started by considering Carrierless Amplitude and Phase (CAP) modulation but finally adopted Discrete Multitone (DMT) modulation for the current standard. DMT allows a simple management of the frequency spectrum. ADSL allocates the frequency range from 4 kHz to 138 kHz to the upstream direction and from 138 kHz to 1.104 MHz to the downstream direction. The channel is a twisted pair line that allows ADSL operation to have a maximum length of 18,000 feet at lower data rates and, unlike some of the other DSL technologies, can tolerate some bridged taps.

This chapter presents the protocol stack and defines certain terms used to designate the different signals, including the position of the proposed crosstalk canceller in the structure. Cyclic Redundancy Check (CRC) coding is then presented, followed by an analysis of the performance of the Reed-Solomon (RS) coding used in ADSL. This is important since it allows justifying whether or not the proposed crosstalk canceller is effective. In fact, the crosstalk canceller does not have to correct all errors, but it aims to lower the Bit Error Rate (BER) to a level that the Reed-Solomon decoder can correct. The last section briefly talks about the Trellis encoding. The rest of the protocol stack and the model of the twisted pair channel are presented in the following chapters.

3.1 ADSL Protocol Stack

Figure 3-2 presents the ADSL protocol stack with the channel and crosstalk canceller block diagrams. The signals introduced in the diagram will be used throughout.

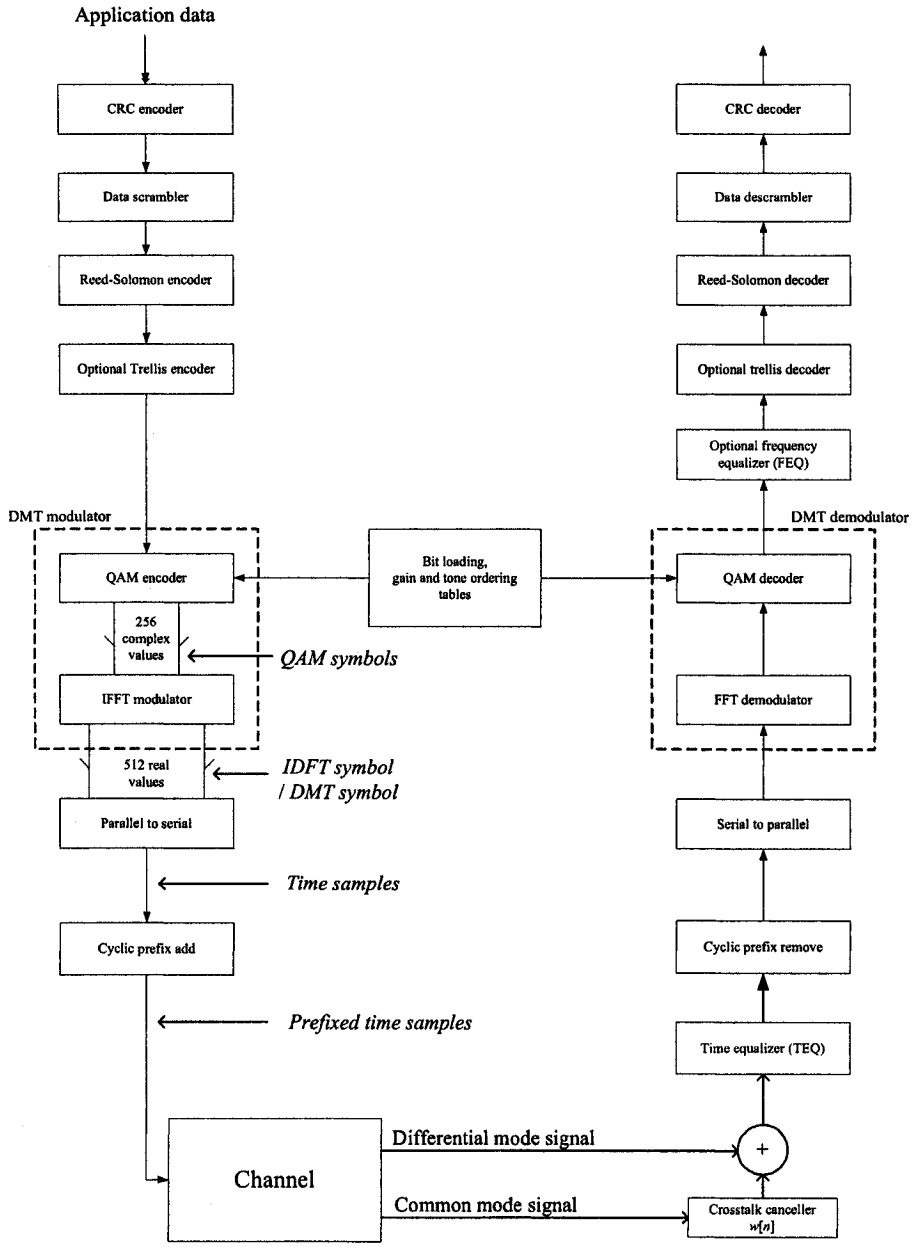


Figure 3-2: Protocol Stack with the channel and crosstalk canceller block diagram

3.2 CRC Coding

The Cyclic Redundancy Check (CRC) layer is used to detect errors in the incoming data stream. ADSL specifies that some data that are sent with CRC coding and others without. It also specifies how to include bits from the direct and interleaved buffer, but this simulation has no interleaved buffer. This simulation includes all bytes for CRC coding since the goal of this thesis is to get an indication of BER on all bits. The encoding/decoding polynomial $G(D)$ used for the CRC check is [G9921]:

$$G(D) = D^8 + D^4 + D^3 + D^2 + 1$$

The overhead created is approximately 3.13% as it adds 8 bytes to each application data of 247 bytes, yielding a coded word length of 255 bytes.

3.3 Data Scrambling

The data scrambler tries to randomize the data to avoid repetitive binary patterns, such as any upper layer protocol headers and synchronization sequences. Repetitive patterns cause a concentration of energy in the output spectrum at certain frequencies. Frequency concentration causes more severe interference and crosstalk problems for others. Randomizing the data bits causes the output spectrum to be more continuous and easier to control, as it is less dependent on the format of the application data being transmitted. The data scrambler operates on bits and the input is a series of bytes. Like all conversions in ADSL, the bytes are converted to bits, with the least significant bit first, then scrambled and reconverted to parallel byte format. The data scrambler implemented in the simulation was the one imposed in the ADSL standard 992.1 [G9921], section 7.5, which implements the equation: $d'_n = d_n \oplus d'_{n-18} \oplus d'_{n-23}$ where d_n is the n^{th} input and d'_n is the n^{th} output.

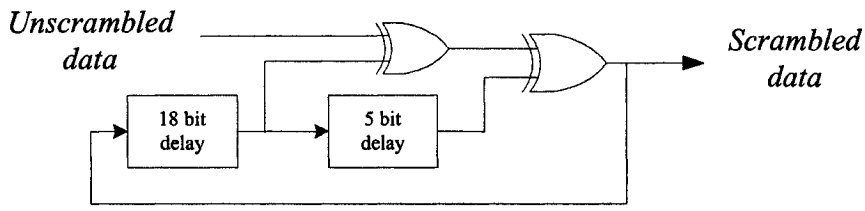


Figure 3-3: ADSL data scrambler at the transmitter

The descrambler does the opposite of the scrambler to recover the original unscrambled data and also operates on the serial bit data. Since the standard does not specify the decoding, the descrambler's structure was derived from [FER97]:

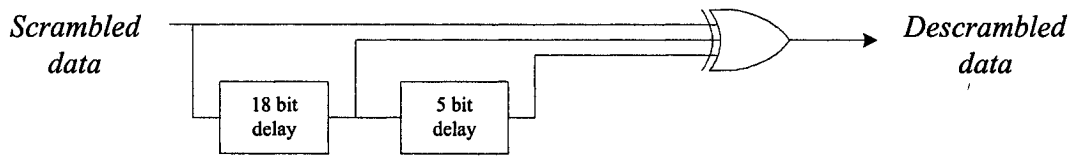


Figure 3-4: ADSL data descrambler at the receiver

The *Descrambled data* and the original *Unscrambled data* should be identical.

3.4 Reed-Solomon Coding

Reed-Solomon coding enables correction of slightly corrupted received data without the need for retransmission. At the transmitter, the Reed-Solomon coder adds a certain fixed amount of overhead to the message to permit this correction. The exact amount of overhead to add is discussed in the 992.1 standard [G9921] and greatly depends on the situation, including upstream/downstream, bit rates and other considerations. Reed-Solomon coding is widely employed and, in general, 5-15% of overhead to the message is added to enable an adequate correction (10-15% is added in High Definition Television (HDTV) [KUN95]). In this document, the percentage of overhead is defined as the amount of overhead bytes in the Reed-Solomon codeword (22 bytes of overhead in a 255 byte codeword corresponds to 8.6% of overhead). The more overhead is used, the more errors can be corrected.

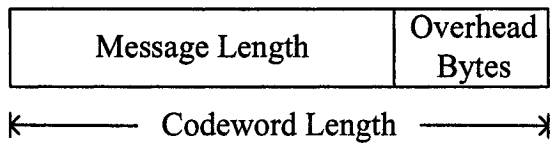


Figure 3-5: Reed-Solomon codeword

The Reed-Solomon codeword length depends primarily on the bit rate. Using an example of an output bit rate of 1.5 Mb/s (assuming framing is not used, which lowers the actual available bit rate), and knowing that there are 4058 DMT symbols / second (assuming no Sync word is used, as will be explained later), yields:

Number of Bits per DMT Symbol = 1.5e6/4058 = 369 bits → 46 bytes/*DMT Symbol*

The ADSL Reed-Solomon codeword can have a maximum length of 255 bytes and can contain 1, 2, 4, 8 or 16 DMT symbols for the message length of the codeword (the maximum is always chosen because it increases the Reed-Solomon code's efficiency). Here, choosing 4 DMT symbols per codeword would yield a message length of 184 bytes; 4 is the highest number of DMT symbols that can be chosen without going

over the limit. Finally, the overhead bytes are added, generally 16 bytes, yielding a total length of 200 bytes. This 200 bytes codeword length is the length that cannot exceed 255. So here, the percentage of overhead is 8%.

The standard specifies that the codeword length and amount of overhead can vary and both depend on many factors; unfortunately, Matlab™ only permits codeword lengths of (2^M-1) where M is an integer larger than 2. This simulation has a fixed amount of overhead at 22 bytes for every 233 bytes of message, yielding a codeword of 255 bytes (approximately 8.6% overhead) for all situations. This chosen amount of overhead is reasonable and a good average compromise for downstream ADSL. "In most cases, the maximum error-correcting protection against Gaussian noise when the system operates near capacity is obtained when the parity overhead is approximately 6 to 10% [STA02 P62]"

A small experiment was performed to judge the performance of the Reed-Solomon coding scheme with respect to the chosen overhead of 22 bytes for every 233 bytes of data. The bit errors were inserted at equally spaced intervals in time by inverting the value of the bit to create an error.

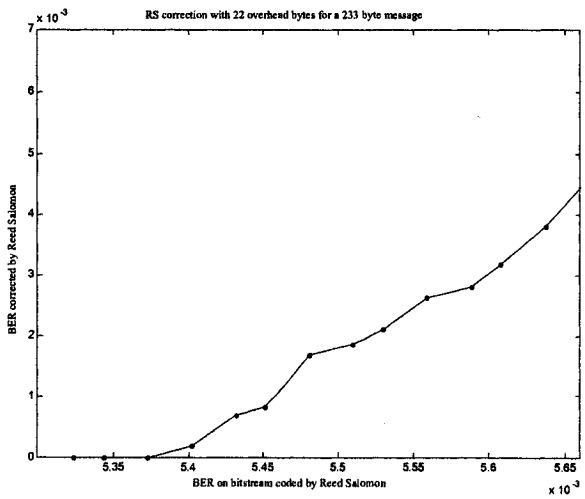


Figure 3-6: Reed-Solomon correction capability with 22 overhead bytes for a 233 byte message

Figure 3-6 displays the result of the experiment with 100,000 bits; 22 bytes of overhead corrected a BER of 5.37e-3 (with bit errors equally spaced in time and only 1 bit in error per byte). Because of the small number of data bits used, this BER should not be taken as an absolute value, but it gives an idea of the correcting ability of the algorithm.

The obtained maximum BER of 5.37e-3 also conforms to the theory that says that the number of errors that Reed-Solomon can correct in the codeword is half the amount of its number of overhead bytes in the codeword [WAN02]. In this case, 22 bytes of overhead mean that the Reed-Solomon algorithms can correct a maximum of 11 byte errors in the codeword, which is of length 255. If a byte error is caused by only 1 bit being in error in the byte (as was done in this experiment), then the following formula calculates the maximum BER that the Reed-Solomon can correct in this case:

$$BER_{MAX} = \frac{11\ Allowed\ Bit\ Errors}{255\ \text{bytes} \cdot 8\ \text{bits/byte}} = \frac{11\ Allowed\ Bit\ Errors}{2040\ \text{bits}} = 5.39e-3$$

To see how much Reed-Solomon overhead is required, another small experiment shows the BER Reed-Solomon coding can correct versus the percentage of overhead used. The BER value retained for a given percentage of overhead was the last BER on the coded stream that was corrected by the Reed-Solomon decoder. As in the previous experiment, the bit errors were inserted at equally spaced intervals in time and only one bit per erroneous byte is in error. This experiment cannot be considered very accurate as the length of the data used was limited due to lack of processing power, but it does indicate a certain tendency, and again does give an idea of its magnitude. The results are shown in Figure 3-7.

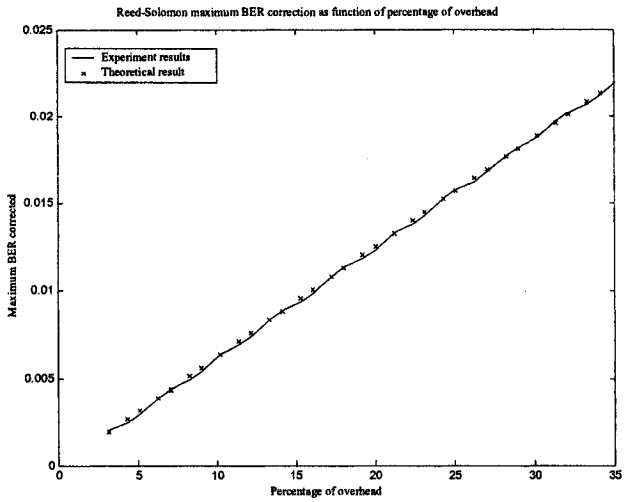


Figure 3-7: Reed-Solomon maximum BER correction as function of percentage of overhead

The theoretical result was calculated using the following reasoning:

$$BER_{MAX} = \frac{\frac{Percentage\ Overhead\ /100}{2} \cdot 255 \text{ bytes}}{255 \text{ bytes} \cdot 8 \text{ bits/byte}}$$

The ½ multiplying the *Percentage Overhead* is because the Reed-Solomon can correct half of the errors of the amount of overhead. Therefore, one can see from the figure that the experiment conforms to the theoretical results.

The ADSL standard specifies that, after all the error correction is included, ADSL must operate at a maximal BER of 10^{-7}. This BER constraint is one of the factors that limit the range and speed of ADSL. The ADSL standard permits byte interleaving, and if interleaving is used, a burst error would be spread over many Reed Solomon codewords after the de-interleaving. Consequently, having bit errors placed at equally spaced intervals is an adequate test when this interleaving is used. However, like most commercial ADSL modems, this simulation does not incorporate the interleaved path.

This is why a higher Reed-Solomon overhead percentage is used to compensate since more errors will occur in a given codeword. The experiments performed in this section are not a true measure of the error correcting abilities of the Reed-Solomon algorithm. A derivation of the actual performance of the algorithm for ADSL can be found in [STA02].

It should be noted that, in ADSL, the Reed-Solomon algorithm does not operate on bits but rather on symbols of 8 bits (byte). This is because if single bits are chosen as the symbols for Reed-Solomon codes, the efficiency of the code is almost zero. The byte is a choice of symbol that makes sense in the computing world. It should be noted that, if more than one bit is in error in any given byte, the Reed-Solomon code treats it as one error since the entire symbol is erroneous. Even knowing this, all graphs were done using the Bit Error Rate because that is what is ultimately interesting to know for the purpose of error correction on the bit stream.

3.5 Trellis Coding

The optional Trellis Coding increases the coding gain, which in turn allows higher bit loads at the same SNRs compared to not using Trellis Coding. The coding gain for the ADSL standard's [G9921] version of Wei's 4D-16 state Trellis encoding is approximately 3.3 – 4.5 dB [WAN02]. A Viterbi decoder is used at the receiver end. Again, Trellis Coding is offered for the fast buffer (direct path) or the interleaved buffer. Here, the interleaved path is not implemented, so only the direct path is considered. The interleaved path is not implemented because it is understood to deal with impulse noise. Since this simulation is built to analyze the effects of crosstalk, which is considered to have stationary properties, it does not include impulse noise. It will be seen later that the Trellis encoding/decoding is not used in this simulation for the tested cases.

Chapter 4

Discrete Multitone Modulation

This chapter presents the Discrete Multitone (DMT) modulation in use for ADSL. DMT is a modulation technique that divides a channel's available bandwidth (1.104 MHz for ADSL) into a certain number of equally spaced sub-channels called "tones". This modulation technique facilitates the inclusion or exclusion of certain frequency bands in the transmitted signal [MEL95] (like the POTS band, HAM (Amateur) radio bands for VDSL, or even perform complex Frequency Division Multiplexing (FDM) with the upstream/downstream). Quadrature Amplitude Modulation (QAM) is used to modulate data on each sub-channel separately. The incoming data stream's bits are distributed over the tones according to their respective Signal to Noise Ratio (SNR). The higher the SNR for a given tone, the more bits will be placed in that tone since it can support a higher order QAM constellation, as long as it respects certain criteria like power constraints. This chapter starts by explaining how the SNR is calculated in this simulation, followed by the calculation of the maximum capacity of bits on each tone. The following section explains the algorithm determining how many bits to place in each tone (bit loading algorithm) that was used and how it was modified. The next section of this chapter covers tone ordering. The following sections cover the generation of QAM symbols, the gain for each tone, IDFT modulation (Inverse Discrete Fourier Transform modulation) with some mathematical analysis, and finally the cyclic prefix.

4.1 Signal to Noise Ratio Calculation

The SNR for each tone is required to calculate the bit loading table. Since each tone (or frequency bin) is a narrow 4.3125 kHz, the Power Spectral Density (PSD) values are considered to be constant throughout the entire tone. Therefore, the SNR for each tone is calculated using the ratio of the PSD value of the useful signal over the PSD value of the noise at the centre frequency of each tone. The ratio will yield 256 SNR values contained in the vector $SNR_{Max}[n]$, where n = 0,..,255. Referring to Figure 4-1, the formula for SNR_{Max} at a given tone n is taken to be:

$$SNR_{Max}[n] = \frac{PSD_{RX_{SIGNAL}}[n]}{PSD_{Noise_{SIGNAL}}[n]} = \frac{PSD_{TX_{SIGNAL}}[n] \cdot |H_d[n]|^2}{PSD_{Noise_{SIGNAL}}[n]} \qquad n = 0,..,255 \qquad (4\text{-}1)$$

where:

$SNR_{Max}[n]$: SNR value for tone n

$PSD_{RX_{SIGNAL}}[n]$: PSD of the output of the differential channel at centre frequency of tone n

$PSD_{TX_{SIGNAL}}[n]$: PSD of the input to the differential channel at centre frequency of tone n

$PSD_{Noise_{SIGNAL}}[n]$:PSD of the crosstalk and noise at centre frequency of tone n

$H_d[n]$: Differential channel transfer function at centre frequency of tone n

All the PSDs and the transfer function for equation 4-1 are in linear scale.

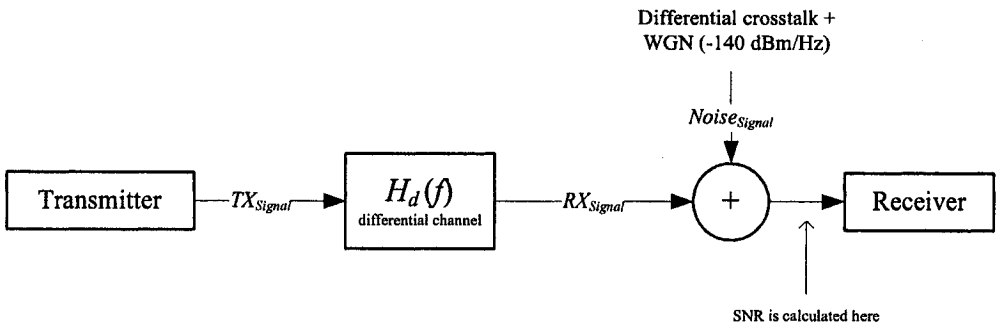


Figure 4-1: Location of SNR calculation

Since the TX_{Signal} and RX_{Signal} are not available until the bit loading is done, which requires the calculation of the SNR, $PSD_{TX_{SIGNAL}}$ is taken to be the PSD mask of ADSL for the considered direction (upstream/downstream). To show the procedure to calculate $SNR_{Max}[n]$ with actual data, an example is given using the downstream signal on a 5,000 feet 24 gauge line with 49 FDD-ADSL self cross-talkers. Figure 4-2 shows the downstream PSD mask for FDD-ADSL. To get $PSD_{RX_{SIGNAL}}$, the PSD mask is multiplied by the squared magnitude of the differential channel $H_d[n]$. $|H_d[n]|^2$ is shown in Figure 4-3 and was calculated using the model of Chapter 5. Figure 4-4 shows $PSD_{Noise_{SIGNAL}}$ for the example line, where the crosstalk levels were found using the equations in Chapter 6, and finally Figure 4-5 displays the resulting SNR calculation using equation 4-1. The 256 SNR values of Figure 4-5 will be used in the next section to calculate an example maximum bit loading capacity table.

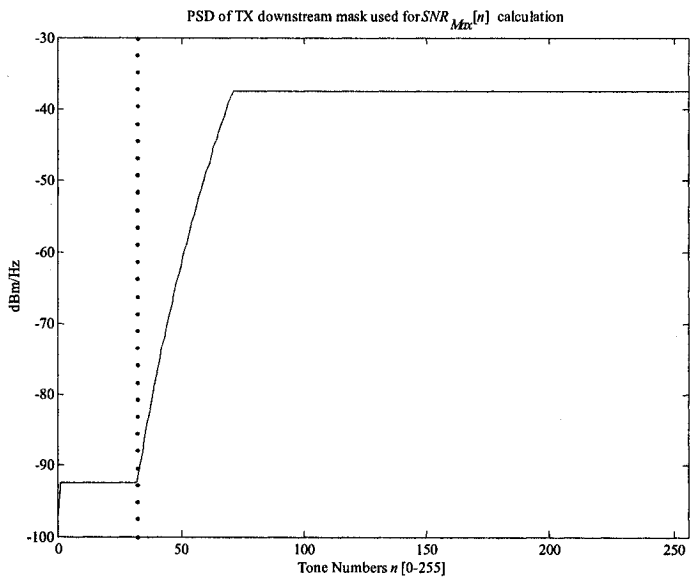


Figure 4-2: PSD mask of the FDD downstream ADSL

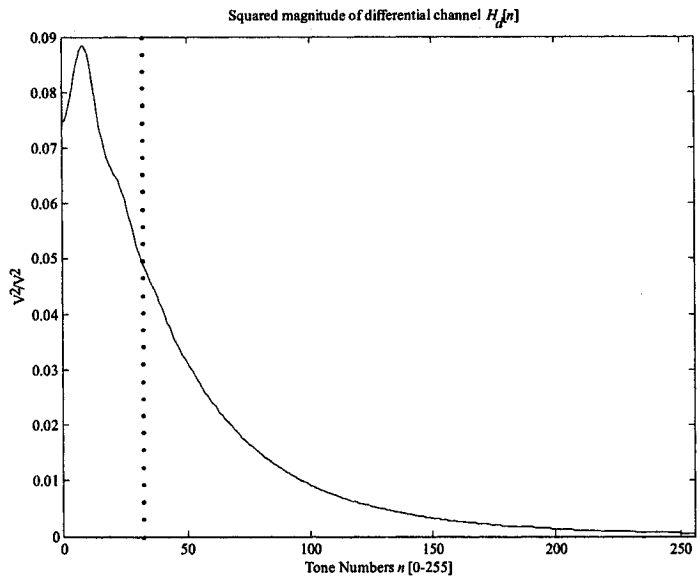


Figure 4-3: Squared magnitude of differential channel of a 5,000 feet 24 gauge line

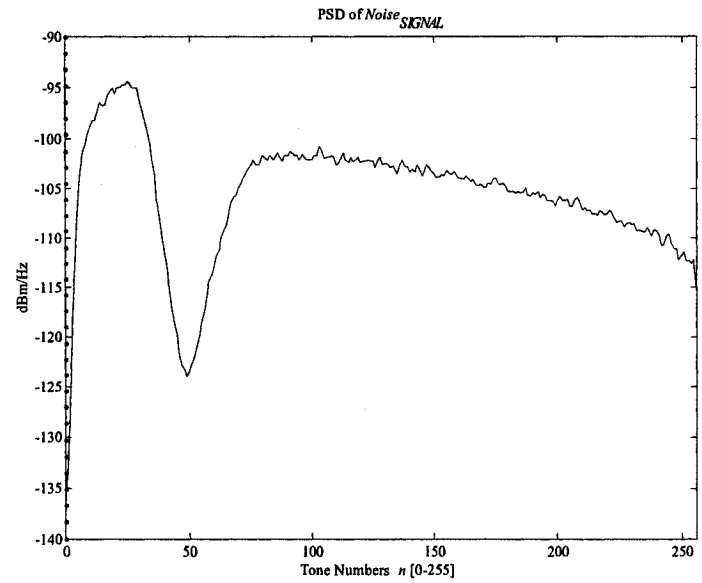


Figure 4-4: Resulting $PSD_{Noise_{SIGNAL}}$ including FEXT and NEXT crosstalk with AWGN at –140 dBm/Hz for the example line

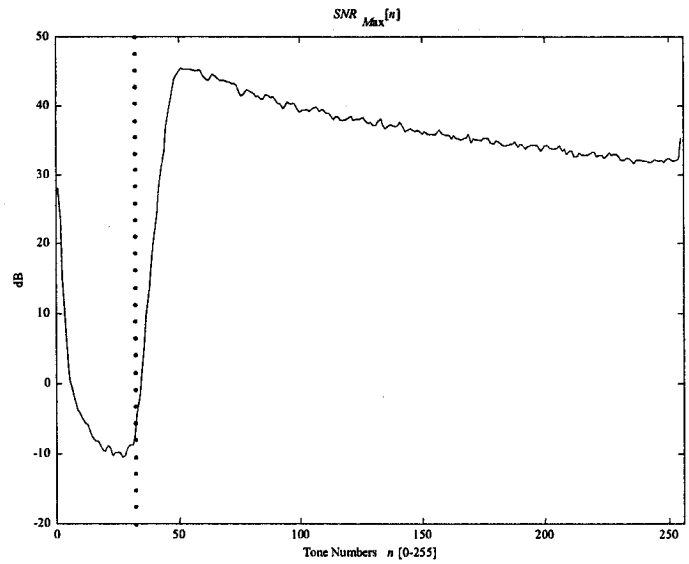


Figure 4-5: Resulting $SNR_{Max}[n]$ calculation for a 5,000 feet 24 gauge line with 49 FDD-ADSL self crosstalkers for the downstream direction

In reality, the SNR cannot be calculated this way since the $PSD_{RX_{SIGNAL}}$ and $PSD_{Noise_{SIGNAL}}$ are not available separately. A technique to estimate the SNR in practice is to transmit a pseudo-random training sequence known to both the transmitter and the receiver. By measuring the probability of error on each tone in the received training sequence, the SNR for each tone can be estimated from this error rate as described in [KOU99]. This technique was not used in this simulation because using the real data gives a much better estimate.

4.2 Maximum Bit Allocation per Tone using Shannon's Law

In ADSL, the 1.104 MHz frequency spectrum is split into 256 4.3125 kHz tones. In each of these tones is placed a QAM encoded symbol. The number of bits encoded in each QAM encoded symbol on each tone is determined by the available SNR for that tone, which is considered constant throughout the tone. Other factors affect the placement of the bits in the bit loading table, such as considerations for the regular POTS service (or

even ISDN-BRA on the same line), the total desired bit rate and whether the bit loading is being done for the upstream or downstream directions.

Before performing the bit loading algorithm, it is required to estimate the maximum number of bits that can be placed in any given tone using the SNR. Shannon's formula for the channel capacity is as follows:

$$Cap = BW \cdot \log_2(1 + S/N) \tag{4-2}$$

where Cap : Channel capacity in bits per second

BW : Total single-sided bandwidth available for the channel in Hertz

S : Signal strength

N : Noise strength

In DMT, the interest is to have the bit capacity per tone, therefore considering each tone as a channel, Shannon's law can be repeated for each tone, with BW equal to the bandwidth of a single tone, and the signal and noise strengths can be replaced by the SNR for that particular tone. A factor Γ is introduced to reduce the SNR to compensate for the fact that Shannon's rule is theoretical and cannot be reached in practice. It considers the modulation technique (QAM), coding (Trellis), the desired error probability (BER<10^{-7}) and a noise margin. This yields the equation 4-3.

$$C[n] = BW \cdot \log_2\left(1 + \frac{SNR_{Max}[n]}{\Gamma}\right) = 4.3125\ \text{kHz} \cdot \log_2\left(1 + \frac{SNR_{Max}[n]}{\Gamma}\right) \tag{4-3}$$

where

$C[n]$: Capacity of tone n in bits/second, $n = 0,..,255$

BW: Bandwidth available for the tone (4.3125 kHz for ADSL)

Γ: Compensation factor in linear scale

$SNR_{Max}[n]$: SNR in tone n (linear scale) (considered constant throughout the tone)

The above equation yields the capacity of a given tone n in bits per second. However, what is required is the number of bits that can be placed in one *QAM symbol* (Refer to Figure 3-2). By knowing how many *QAM symbols* are generated in one second can help derive the capacity in bits for a given tone. The number of *DMT/QAM symbols* that must be generated in one second can be calculated by knowing that the ADSL sampling rate of the output is 2.208 MHz and that there are 544 time samples per *DMT symbol* (this will be explained later).

of *DMT symbols* per second = 2.208 MHz / 544 time samples per *IDFT symbol*
= 4058 *IDFT symbols* /second (=*DMT symbols*/second)

In other words, every tone will have to generate 4058 *DMT symbols* per second, meaning that 4058 *QAM symbols* will also have to be generated for each tone. Knowing this enables to intuitively derive a theoretical maximum capacity for a given tone n in bits per *QAM symbol* (yielding the vector *Max Bit Capacity*[n], where n = 0,..,255) using equation 4-3:

$$\begin{aligned} \textit{Max Bit Capacity}[n] &= \frac{C[n] \ \text{bits/second}}{4058 \ \textit{QAM symbols/second}} \\ &= \frac{\left(4.3125 \ \text{kHz} \cdot \log_2\left(1+\frac{SNR_{Max}[n]}{\Gamma}\right)\right)}{4058} \\ &\cong \log_2\left(1+\frac{SNR_{Max}[n]}{\Gamma}\right) \end{aligned}$$

Since the bit capacity must be an integer, the next lowest integer is taken:

$$\textit{Max Bit Capacity}[n] \cong \left\lfloor \log_2\left(1+\frac{SNR_{Max}[n]}{\Gamma}\right)\right\rfloor \qquad (4\text{-}4)$$

Equation 4-4 is proven more rigorously in [BIN00]. The compensation factor Γ is calculated as follows for ADSL [WAN02]:

Γ = *Shannon gap* + *margin gain* – *coding gain*

The following values are used in ADSL:

Shannon gap = 9.8 dB (used to get a BER $\cong 10^{-7}$)

margin gain = 6 dB (extra margin for unforeseen impairments)

coding gain = 3.3 dB if Trellis coding is used [WAN02], $\cong$ 3.5 dB if Reed-Solomon is used, 0 dB otherwise. Coding allows a higher number of bits in a certain tone at a lower SNR.

It should be noted that Γ is calculated in dB, but must be converted to linear scale in the formulas as well as SNR_{Max}. The maximum bits per tone that the ADSL standard allows is 15, therefore if the bit capacity of a given tone exceeds 15, it is brought back down to 15. On the other hand, the early ADSL standard specifies that 1 bit constellations are not available; therefore if the bit capacity of a given tone is 1 bit, it is brought back to 0. This is done for the purpose of this simulation, but in reality, the revision was added because allowing 1 bit constellation can increase the maximum bit rate slightly.

Using the SNR values in Figure 4-5 that were calculated for a 5,000 feet 24 gauge line with 49 FDD-ADSL self cross-talkers, the maximum bit loading allocation per tone is calculated using equation 4-4 with Reed-Solomon modulation (*coding gain* = 3.5 dB) and is displayed in the following figure for the downstream direction.

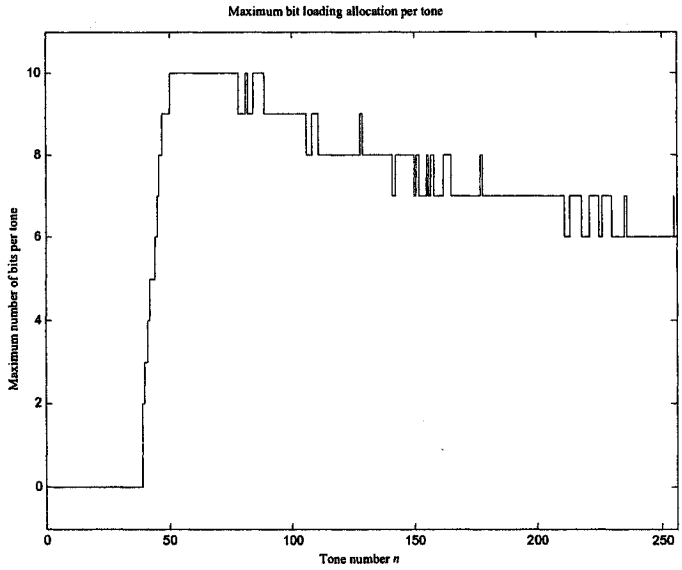


Figure 4-6: Maximum bit loading calculation for a 5,000 feet 24 gauge line with 49 FDD-ADSL self cross-talkers for the downstream direction

By summing all the bits in the bit loading table (in the example of Figure 4-6 1679 bits), the resulting bit rate can be calculated knowing how many DMT symbols are output per second (4058). The maximum bit rate available on the line in the example above is:

$$\textit{Maximum bit rate} = 1679 \frac{\text{bits}}{\textit{DMT Symbol}} \cdot 4058 \frac{\textit{DMT Symbol}}{\text{second}}$$

$$= 6.81 \text{ Mb / second} \quad \text{MAX}$$

As can be seen in Figure 4-6, tones 0-39 are not used; this is because the strong NEXT coming from the upstream ADSL interferers makes the SNR very low at these frequencies (tones).

4.3 Bit Loading Algorithm

The ADSL standard G992.1 [G9921] does not specify which algorithm to use to calculate the bit loading, and the choice is therefore left to the implementer. The algorithm used in this simulation was found in the paper *Optimum bit allocation algorithm for DMT-based systems under minimum transmitted power constraints* [KOU99]. The title suggests that this algorithm minimizes the transmitted power to achieve its desired bit rate. Actually, this is not a concern with the standard as it stands currently since the power transmitted varies very slightly and can be considered almost fixed to keep a predictable PSD on all tones.

This algorithm would be perfectly suited in an environment where the transmitted PSD can change and adapt to changing line and noise conditions as in the water-filling techniques presented in [SON02]. Nevertheless, this algorithm is very straightforward and is still a good choice for this simulation. It was slightly adapted for the specific ADSL DMT standard. Although subsequently revised, the DMT ADSL standard specified that 1 bit QAM constellations could not be used. The algorithm was modified to avoid the 1 bit constellations and not to exceed 15 bit constellations since ADSL limits the number of bits in a tone to 15. Zero bit constellations are permitted, in which case no power is transmitted on this tone in this simulation. This implies that the transmitted PSD is not always the same and depends largely on the bit loading algorithm. The algorithm used is general and can be adapted for any DMT system, but is used here for 256 tones ($N = 256$).

This algorithm starts by calculating the relative power per tone required to transmit one bit in each tone. This procedure is performed for each tone n and is contained in the vector $P_{Threshold}[n]$. $P_{Threshold}[n]$ is calculated using the following formula, modified from [KOU99]:

$$P_{Threshold}[n] = \frac{1}{SNR_{Max}[n]} \qquad n = 0,..,255 \tag{4-5}$$

The equation above was adapted from Equation (8) in [KOU99] because the original was:

$$P_{Threshold}[n] = \frac{P_{\max}[n] \cdot \Gamma}{SNR_{Max}[n]} \tag{4-6}$$

but since $SNR_{Max}[n]$ is calculated here using equation 4-1, by plugging equation 4-1 in equation 4-6 and noting that $P_{Max}[n]$ corresponds to $PSD_{TX_{SIGNAL}}[n]$ yields:

$$P_{Threshold}[n] = \frac{PSD_{TX_{SIGNAL}}[n] \cdot \Gamma}{1} \cdot \frac{PSD_{Noise_{SIGNAL}}[n]}{PSD_{TX_{SIGNAL}}[n] \cdot |H_d[n]|^2} = \frac{PSD_{Noise_{SIGNAL}}[n] \cdot \Gamma}{|H_d[n]|^2}$$

Therefore, the formula is basically the noise PSD over the channel power. This does not include the PSD mask for ADSL and so cannot be used, because it will place bits at frequencies that may not be acceptable by the standard. This statement holds true only if the *Max Bit Capacity*[n] is not used as a constraint (as will be described in the Simulation chapter). If *Max Bit Capacity*[n] is used, it will limit the use of the unallowable frequency bands to the bit loading algorithm. Equation 4-5 was therefore developed to include the SNR only, which already considers the PSD transmit mask. The Γ constant was removed as it only shifts the entire $P_{Threshold}$ but changes nothing to the relative values of each tone. Therefore the result of the algorithm is not affected. Figure 4-7 displays $P_{Threshold}[n]$ for the $SNR_{Max}[n]$ values presented in Figure 4-5.

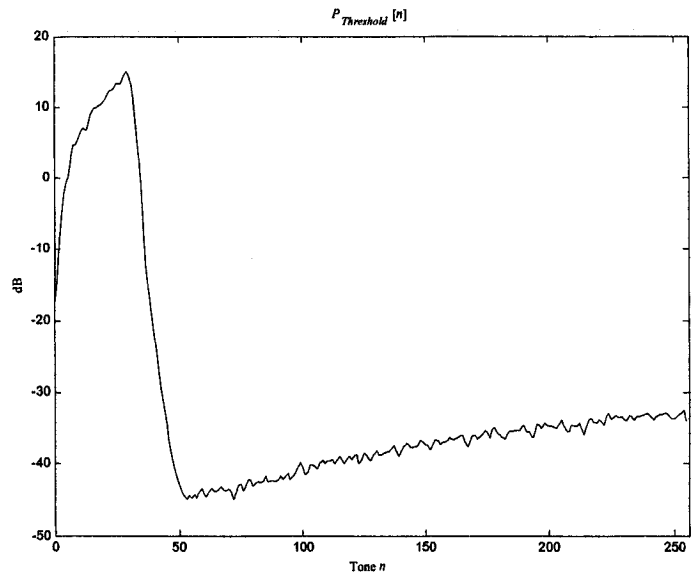


Figure 4-7: $P_{Threshold}[n]$ values for a 5,000 feet 24 gauge line with 49 FDD-ADSL self cross-talkers for the downstream direction.

$P_{Threshold}[n]$ will be used in the first iteration of the algorithm to place the first bit in the tone with the minimum value of the relative powers required to transmit one bit in each tone. Then an adjusted $P_{Threshold}[n]$ is calculated, named P', to include the fact that one bit has already been placed in a given tone and that placing an additional bit in that tone will require more power. For each iteration, the algorithm recalculates P' and places one bit in the tone with the minimum required relative power found in P'.

The number of bits that must be placed can be calculated by knowing the desired target bit rate and that there are 4058 *DMT symbols* / second. If the desired target bit rate is 1.5 Mb/s, the number of bits that the bit loading algorithm has to try to place is:

$$Number\ of\ bits\ to\ place = \frac{1.5\ \text{Mb/s}}{4058\ DMT\ symbols / \text{second}} = 369\ \text{bits} / DMT\ symbol$$

The algorithm pseudo-code follows, in which the bit loading table, named $C_{Ass}[n]$, is the output of the algorithm, which contains the number of bits that will be assigned to each tone n.

Table 4-1: Bit loading algorithm presented in [KOU99]

```
i = 0
Repeat while (i < Number of bits to place) {

    Repeat for n = 1 to N {
        P'[n] = P_Threshold[n] · (2^(C_Ass[n]+1) − 1)
    }
    Repeat for n=1 to N {
        difference[n] = Max Bit Capacity[n] − C_Ass[n]
    }
    minimum = ∞
    Repeat for n = 1 to N {
        if difference[n] > 0 {
            if minimum > P'[n] {
                minimum = P'[n]
                p = n
            }
        }
    }
    C_ass[p] = C_ass[p] + 1
    i = i + 1
}
```

With the algorithm of Table 4-1 in its current form, the bit loading table $C_{Ass}[n]$ will contain tones in which only one bit was assigned. It must be modified to deal with this problem as mentioned above. After the algorithm is executed, the bit loading table $C_{Ass}[n]$ is passed through another algorithm which verifies its validity and changes the values to avoid the one bit per tone problem. The verification and reallocation algorithm pseudo-code can be found in Appendix A.

The post processing done by the adapted algorithm of Appendix A starts by removing all the single bits and keeping the amount of bits removed in *Number of bits to place.* It will disallow the use of tone t that was assigned a single bit having the highest $P_{Threshold}$ of all the tones that were assigned a single bit. To simplify the explanation, the algorithm then re-executes the original bit loading algorithm with the tone t removed. Prior to this process, it verifies that there is enough room to place all the single bits it has removed.

The result of the complete bit loading algorithm including the verification and re-allocation phase for the example 5,000 feet 24 gauge line with 49 FDD-ADSL self cross-talkers for a 1.5 Mb/s downstream direction, is displayed in the following figure:

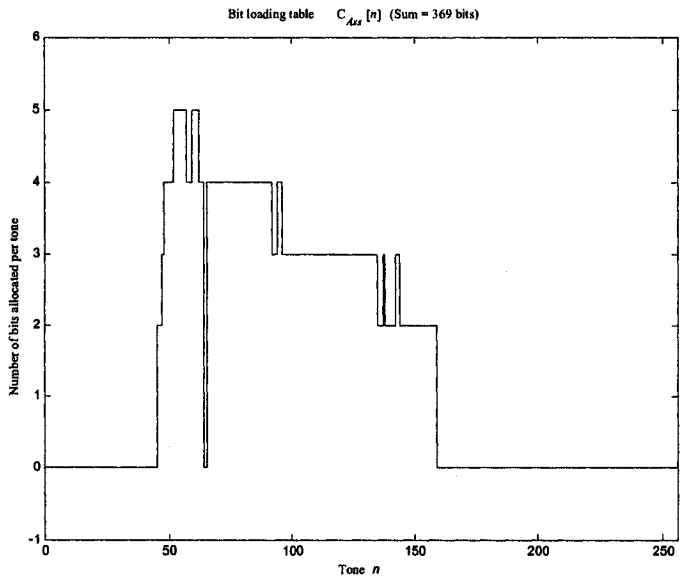


Figure 4-8: Bit loading table $C_{Ass}[n]$ for a 1.5 Mb/s downstream direction using $P_{Threshold}[n]$ of Figure 4-7

Notice that, in the above figure, the title indicates the sum of assigned bits. It represents the number of bits in the bit loading table. Tones 0 through 6 are not used for reasons of compatibility with the POTS; furthermore, tone 64, being a pilot tone, cannot be used for data, and is shown in Figure 4-8 as a dip in the bit loading graph at tone 64.

To show a different example, one that displays the upstream bit loading, the result of the complete bit loading algorithm, ncluding the verification and re-allocation phase for a 5,000 feet 24 gauge line with 49 FDD-ADSL self cross-talkers for a 360 kb/s upstream direction,is displayed in the following figure for comparison. The figure uses only tones 0 to 31 for the upstream case.

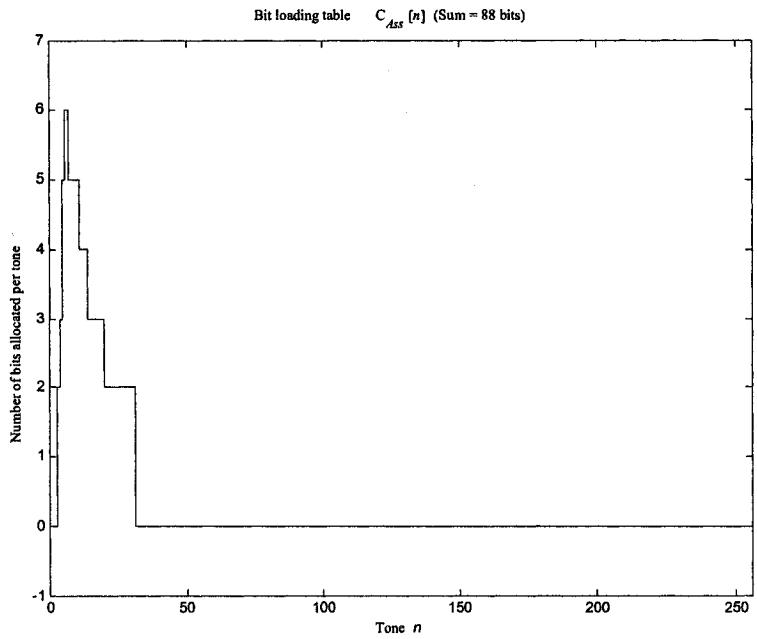


Figure 4-9: Bit loading table $C_{Ass}[n]$ for a 360 kb/s upstream direction

In practice, the bit loading table can be recalculated in real-time during the connection to adjust for changing line conditions, but this was not implemented here as line conditions including the noise and crosstalk levels are fixed (i.e. stationary).

4.4 Bit Assignment using Tone Ordering

The tone ordering procedure, implemented as in the standard [G9921], takes the bit loading table and sorts it according to the *Number of bits assigned* column. See Table 4-2 for an example of an 8 tone bit loading table and its corresponding tone ordered table.

Table 4-2: a) 8 tone bit loading table b) its corresponding tone ordered table

a) Bit loading table $C_{Ass}[n]$	
Tone n	# of bits assigned
0	0
1	3
2	2
3	3
4	7
5	6
6	3
7	0

b) Tone ordered table	
Tone n	# of bits assigned
0	0
7	0
2	2
1	3
3	3
6	3
5	4
4	5

The assignment of the bits of the input data stream will be determined by the tone ordered table. An example of how this tone ordered bit loading table is used follows. The total number of bits that will be encoded in one QAM symbol can be easily calculated by summing all the bits in either the bit loading table or the tone ordered table. In the example above, the total number of bits is 20 per symbol. In the following example, only 20 bits of the input buffer are indicated, but they are assumed infinite, the next 20 bits will be assigned identically, but to the next QAM symbol.

Given the input buffer of Table 4-3:

Table 4-3: Example input buffer

...	b_{20}	b_{19}	b_{18}	b_{17}	b_{16}	b_{15}	b_{14}	...	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1

with the tone ordered Table in Table 4-2 b), the bit assignment follows in Table 4-4.

Table 4-4: Example tone ordered loading of bits using the previous ordered table

Tone	Bit numbers
0	None
1	b_3 b_4 b_5
2	b_1 b_2
3	b_6 b_7 b_8
4	b_{16} b_{17} b_{18} b_{19} b_{20}
5	b_{12} b_{13} b_{14} b_{15}
6	b_9 b_{10} b_{11}
7	None

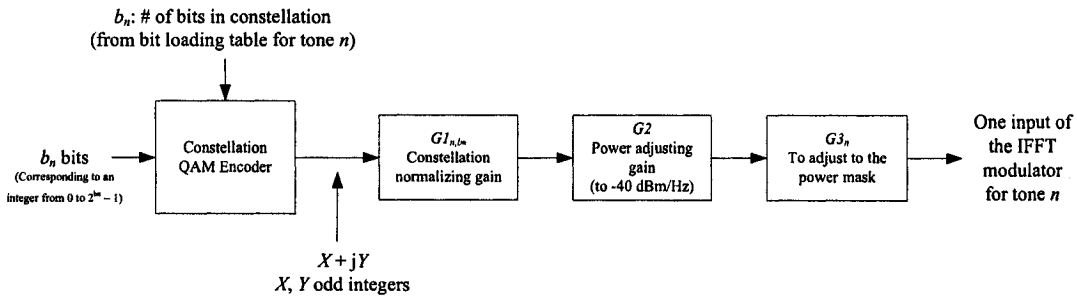
The purpose of tone ordering, when used in combination with interleaving, is to reduce the effect of impulse noise encountered on the line [BIN00]. Since impulse noise usually affects all the tones (impulse in time causes a white frequency spectrum), the tones supporting the highest constellation orders will be affected the most. Tone ordering organizes the bit loading table to place the bits from the incoming buffer to the tones with the lower order constellations first, then the bits in the tones with the highest number of bits last. When interleaving is used, the bits will be loaded from the fast buffer first (the non-interleaved ones) and then the bits from the interleaved buffer will be loaded. This will result in loading the interleaved bits into the higher order constellations. When an impulse noise affects the higher order constellations, the error bits are spread across several Reed-Solomon codewords due the interleaving, improving the chance of recovery using the FEC.

This simulation does not use interleaving, therefore the tone ordering procedure is not necessary, but was implemented because of the possible inclusion of interleaving in future work.

4.5 QAM Encoding

Once the tone ordering has taken place and the bits from the input buffer have been assigned to their respective tones, the QAM Encoding is performed in each tone using the bit loading table, allowing to know how many bits must be encoded in each specific tone. Therefore, the size of the QAM constellation depends on the amount of bits that have to be encoded, which can differ from tone to tone. The number of bits to be encoded in a given tone n, will be named b_n.

The following figure displays the steps that have to be performed in the QAM encoder for each of the 256 tones. The 4 steps will be broken down and explained individually. The gains are not an intrinsic part of QAM encoding, but both were combined, as it was a better choice than to have them dissociated. The combination of the 3 gains can be stored in a Gain Table, which will also be explained in this section.

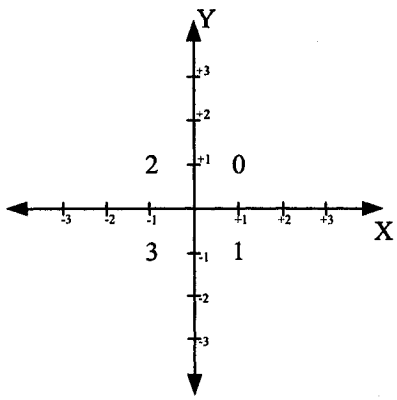


$G1_{n,b_n}$: Constellation normalizing gain for tone n with corresponding b_n
$G2$: Power adjusting gain to -40 dBm/Hz
$G3_n$: Power mask adjustment gain for tone n

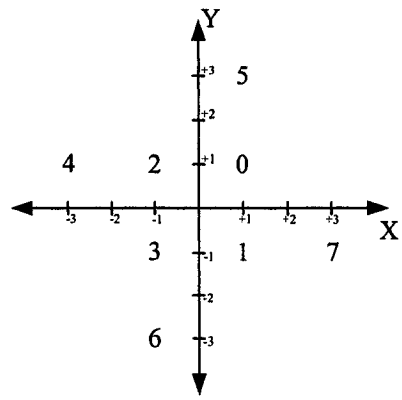
Figure 4-10: QAM encoder and gain adjustments for any given tone *n*

4.5.1 Constellation QAM Encoder

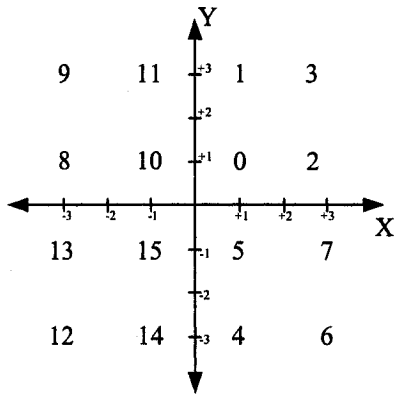
The quantity of bits that can be encoded in a tone are 0, or 2 to 15 bits, as defined by the ADSL standard G992.1 [G9921]. The 1 bit constellation was not used. Figure 4-11 displays the QAM constellations used in ADSL G992.1 for 2 to 5 bits.



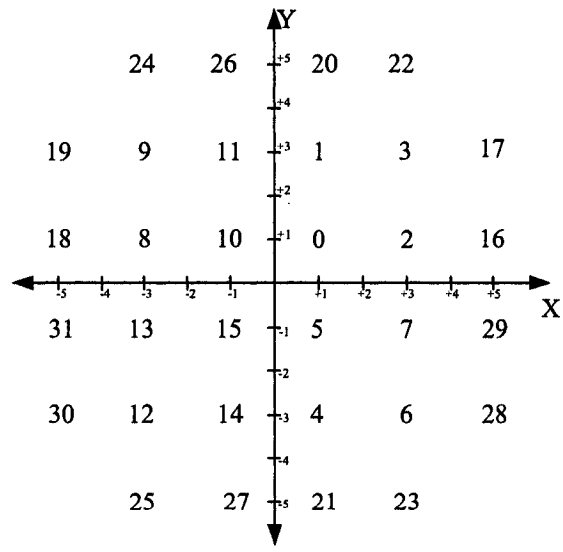
a) b=2 bit 4-QAM constellation



b) b=3 bit 8-QAM constellation



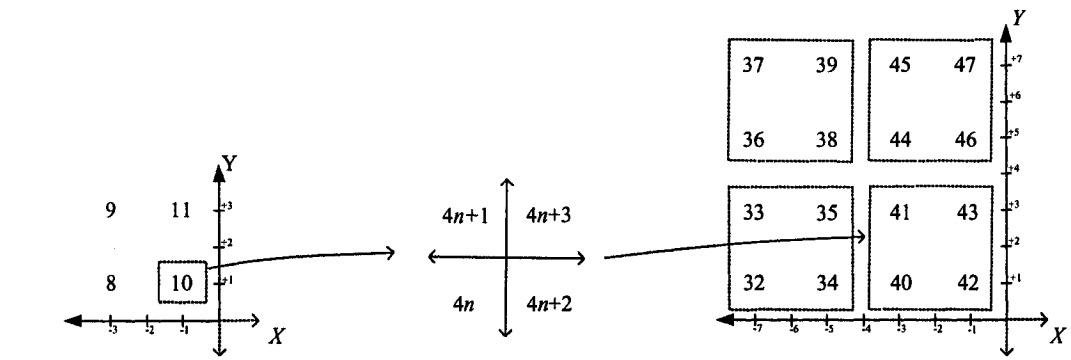
c) b=4 bit 16-QAM constellation



d) b=5 bit 32-QAM constellation

Figure 4-11: QAM constellations in ADSL for a) 2 bits b) 3 bits c) 4 bits d) 5 bits

All the indices of the coordinates are odd numbered; the even coordinates are not used. The higher bit order constellations are derived from these constellations; the technique depends on whether b_n is even or odd, but both rely on constellation expansion. One value from the constellation is expanded into 4. Therefore expanding a 4 bit 16-QAM constellation will yield a 6 bit 64-QAM; in the same way as a 5 bit constellation must be expanded to generate the 7 bit constellation. An example of an expansion is shown in Figure 4-12.



a) b = 4 bit 16-QAM constellation top left quadrant

b) b = 6 bit 64-QAM constellation top left quadrant

Figure 4-12: Example of an expansion from a 4 bit constellation to a 6 bit constellation with only the top left quadrant shown

In the example of Figure 4-12, where only the top left quadrant is shown to save space, the value *10* is expanded into the *40*, *41*, *42* and *43* values using the expansion graph shown in the middle that contains $4n \rightarrow 4n+3$. All even and odd constellations are expanded in this way, starting with the 6 bit 64-QAM constellation. It can be noticed that none of the even coordinates are used, meaning that when one value is expanded, all the others are "shifted" to the next odd index as in the example above.

In summary, the QAM constellation encoder is used on each tone. The bit loading table establishes how many bits will be in the constellation. The bits to encode are fed from the input buffer and are determined from the tone ordered bit loading table.

4.5.2 Constellation Normalizing Gain *G1*

$G1_{n,b_n}$ is the gain factor that is applied to the output of the QAM encoder for tone n. Its purpose is to normalize the average energy (per symbol) of the different sizes of QAM constellations to the same values of normalized average energy (per symbol); the size of the constellation of tone n is 2^{b_n}, where b_n is the number of bits for tone n, and can have a value of 0 or 2 through 15. The larger higher order constellations have a higher average energy than the smaller ones. The average energy of a constellation with 2^b constellation points ($E_{AV,b}$) is calculated by dividing the total energy of the constellation (E_{Total}) by the number of points in the constellation. This is done considering that all constellation points are in reference to the same basic distance d, which is the smallest distance between 2 points. The actual value of this distance is unimportant and can be omitted, as it was in the constellation graphs axis units, if it is deemed identical for all constellations.

$G1_{n,b_n}$ normalizes the energy coming from all constellation sizes to the same reference average energy (power). The reference can be any value, but can be chosen to be the average energy of the smallest constellation, here $E_{AV,2}$, eliminating the need to define d as it will be eliminated when taking the ratio. Since $G1_{n,b_n}$ must multiply data (complex magnitudes) coming from the QAM constellation encoder, the square root of the ratio between average energies is taken to get a magnitude gain as follows:

$$G1_{n,b_n} = \sqrt{\frac{E_{AV,2}}{E_{AV,b_n}}}$$

where b_n must range from 2 to 15. When b_n is 0, $G1_{n,0}$ is defined as 0. The next table summarized the gain factor $G1_{n,b_n}$ for values of b = 0,2,..,15

Table 4-5: Gain factor $G1_{n,b_n}$ for values of b = 0, 2,..,15

b	0	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$G1_{n,b_n}$	0	1	$\frac{1}{3}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{21}$	$\frac{1}{41}$	$\frac{1}{85}$	$\frac{1}{165}$	$\frac{1}{341}$	$\frac{1}{661}$	$\frac{1}{1365}$	$\frac{1}{2645}$	$\frac{1}{5461}$	$\frac{1}{10581}$

4.5.3 Average Power Adjusting Gain

The power adjusting gain *G2* is the gain that must be applied to the output of every tone to bring the normalized power of all the tones to a power density of –40 dBm/Hz. A -40 dBm/Hz power density value is given in the continuous time domain and must be converted to the discrete time domain since all the processing is done in discrete time, and is done as follows:

$$\textit{Discrete power density} = \frac{1}{2}\cdot F_S \cdot \left(\frac{10^{-40\ \text{dBm/Hz}/10}}{1000}\right)\left[\frac{\text{V}^2}{\text{Cycle/Sample}}\right]$$

where F_S is the sampling frequency. The division factor 1000 is to convert from mW/Hz to W/Hz. The ½ is included to account for the fact that this value was specified for a single sided power density. *G2* is calculated by dividing the power density by the reference average energy (reference power). The reference average energy used was that of the 2 bit 4-QAM constellation, which was $E_{AV,2} = 2$. For the same reason as for $G1_{n,b_n}$, the square root is taken. *G2* is then given by:

$$G2 = \sqrt{\frac{\textit{Power density}}{E_{AV,2}}} = \sqrt{\frac{\frac{1}{2}\cdot F_S \cdot \left(\frac{10^{-40/10}}{1000}\right)}{E_{AV,2}}} = \sqrt{\frac{\frac{1}{2}\cdot 2.208\cdot 10^6 \cdot \left(\frac{10^{-40/10}}{1000}\right)}{2}} \approx 0.235$$

This can only work if the IDFT modulator does not introduce a factor that would change the energy of the signal, which will be shown later by using the normalized IDFT.

4.5.4 Power Mask Adjustment Gain

The gain $G3_n$ is to adjust the power density of each tone n to its respective maximum as defined by the power density mask imposed by the ADSL standard [G9921]. After *G2*, every tone is at a power density of –40 dBm/Hz, which can be adjusted between -2.5 dB to +2.5 dB around the –40 dBm/Hz (or even set to 0), as long as the resulting power density of the tone is equal to or less than the power density mask. In any case, the power density of the tone must be reduced to the power density mask if it exceeds it, therefore some tones will be greatly attenuated, notably the tones overlapping the POTS frequencies. The power density mask depends on the data stream direction (Upstream/Downstream). The masks are partially reproduced here, starting with the upstream power density mask, and are taken from the standard [G9921].

Table 4-6: Partial upstream power mask table

Frequency band f (kHz)	Equation for line (dBm/Hz)
$0 \le f < 4$	-97.5
$4 \le f < 25.875$	$-92.5 + 21.5 \cdot \log_2(f/4)$
$25.875 \le f < 138$	-34.5
$138 \le f < 307$	$-34.5 - 48 \cdot \log_2(f/138)$
$307 \le f < 1104$	-90

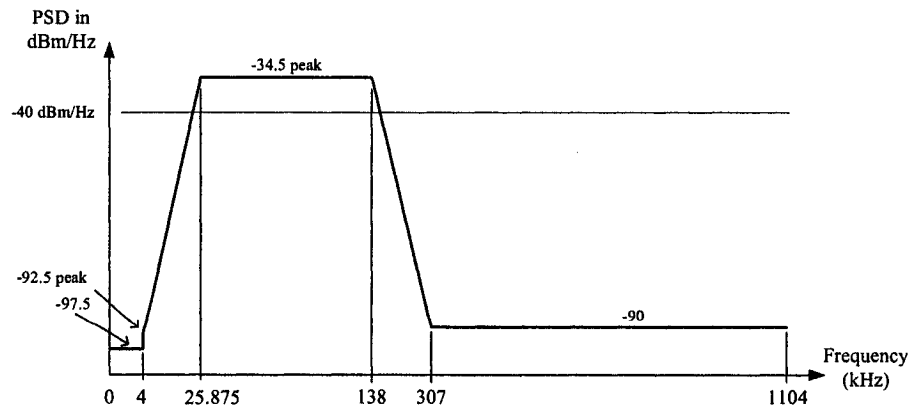


Figure 4-13: Partial upstream power mask graph

The word "partial" in the description of Figure 4-13, Figure 4-14, Table 4-6 and Table 4-7 means that the masks in [G9921] are defined to frequencies above the maximum frequencies considered for ADSL. The extra requirements specified with the power density mask (specification of frequency resolution of the measurements, sliding window maximum power measurements, etc.) are not considered in this thesis.

Table 4-7: Partial downstream power density mask for FDD-ADSL table

Frequency band f (kHz)	Equation for line (dBm/Hz)
$0 \le f < 4$	-97.5
$4 \le f < 25.875$	$-92.5 + 21.5 \cdot \log_2(f/4)$
$25.875 \le f < 138$	-34.5
$138 \le f < 307$	$-34.5 - 48 \cdot \log_2(f/138)$
$307 \le f < 1104$	-90

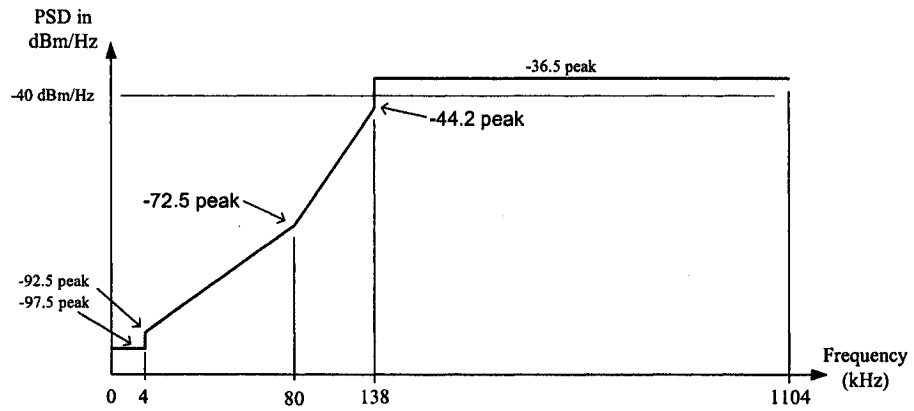


Figure 4-14: Partial downstream power density mask for FDD-ADSL graph

From Figure 4-13 and Figure 4-14, it can be seen that the upstream and downstream directions PSD masks overlap somewhat. Even though the specified tones for each direction are separate in the case of FDD-ADSL, the windowing effect of the IFFT modulation (which will be covered later) will cause spectral leakage in the adjacent tones. The first step to calculate $G3_n$ is to verify if the resulting power spectral density set by *G2* (–40 dBm/Hz) is higher than the specified power mask for tone n. If it is, and the tone is

actually used (constellation is not 0), the following power ratio is taken (in linear scale). Since a magnitude gain is sought, the square root of the power ratio is as follows:

$$G3_n = \sqrt{10^{\left(\frac{PSD\,Mask[n]}{10}\right)} \Big/ 10^{\left(\frac{-40\ \text{dBm/Hz}}{10}\right)}}$$

where *PSD Mask*[*n*] is the power density of the transmit mask at the frequency corresponding to tone *n*. This will set the transmitted PSD for tone *n* to the PSD mask.

For the tones where the –40 dBm/Hz is lower than the PSD mask, the minimum required power is transmitted while still adhering to the constraint of +/-2.5 dB around the –40 dBm/Hz. This minimum power density for tone *n* is [KOU99]:

$$P_{Min}[n] = \frac{P_{Mask}[n]}{SNR_{Max}[n]} \cdot \Gamma \cdot \left(2^{b_n} - 1\right)$$

where b_n is the number of bits assigned to tone *n* by the bit loading algorithm, $P_{mask}[n]$ is the power spectral density at tone *n*, *SNR*[*n*] is the SNR at tone *n*, and Γ is the compensation factor used in equation 4-4.

If P_{Min} at tone *n* yields a value below –40 dBm/Hz - 2.5 dB (-42.5 dBm/Hz), then:

$$G3_n = \sqrt{10^{\left(\frac{-42.5\,\text{dBm/Hz}}{10}\right)} \Big/ 10^{\left(\frac{-40\ \text{dBm/Hz}}{10}\right)}}$$

If P_{Min} at tone *n* yields a value above the –40 dBm/Hz + 2.5 dB (-37.5 dBm/Hz), then:

$$G3_n = \sqrt{10^{\left(\frac{-37.5\ \text{dBm/Hz}}{10}\right)} \Big/ 10^{\left(\frac{-40\ \text{dBm/Hz}}{10}\right)}}$$

In the remaining situation, $G3_n$ = 1 to maintain the power at –40 dBm/Hz. To summarize, $G3_n$ allows the adjustment of the transmitted power by calculating the minimum required power to transmit b_n bits in tone n. $G3_n$ adjusts the power to send more (+2.5 dB), less (–2.5 dB), or to bring the transmit power of the tone to the maximum specified by the mask, whichever is lower.

4.5.5 Resulting Gain Table

The gains are generally combined into one global gain for tone n, to yield a gain vector:

$$G_n = G1_{n,b_n} \cdot G2 \cdot G3_n$$

where G_n is the gain for tone n, calculated using the appropriate b_n. This vector, named "Gain table", is attached to the bit loading table like in the 8 tone example below.

Table 4-8: Bit loading and gain table example

Bit loading C_{Ass} and Gain table G_n		
Tone	# of bits assigned	G_n
0	0	0
1	3	0
2	2	0.8
3	3	1
4	7	0.9
5	6	0.2
6	3	0.1
7	0	0

Since the SNR for each tone is measured and calculated at the receiver, the receiver also calculates the bit loading table, and therefore the gain table. The 2 tables are sent back to the transmitter. In practice, the bit loading and gain tables can change during run time to adapt to the changing line conditions, so the gain table can be updated.

4.6 IDFT Modulation

The modulation technique that is used in DMT systems is the IDFT modulation. The underlying reason is that each tone must be modulated to a different carrier frequency. In essence, the operation to be performed is to have each complex value contained in each tone modulate its respective carrier frequency (determined by the tone number) and sum the result as can be seen in the following equation:

$$Output = \sum_{k=0}^{255} 2 \cdot \left| QAM_k \right| \cdot \cos\left(2\pi \cdot F_k \cdot t + \angle QAM_k\right) \qquad F_k = \left(\frac{k}{256}\right) \cdot F_S$$

where k represents the index of the tone, QAM_k is the QAM encoded value of tone k, and F_S is the sampling frequency.

This represents the modulation of several narrowband channels. Doing this in the analog domain presents many problems and is not a viable option, nor a desirable one. The processing is much easier in the discrete domain. Thus, the complex exponentials will be modulated by complex values and therefore need their complex conjugate counterpart to be modulated at the same but negative frequency to yield a real-valued time signal. The modulation is performed in phase and amplitude. In the discrete domain, where the DFT is periodic of 2π, these values can be placed above π and do not have to be negative. This is better represented in Figure 4-15:

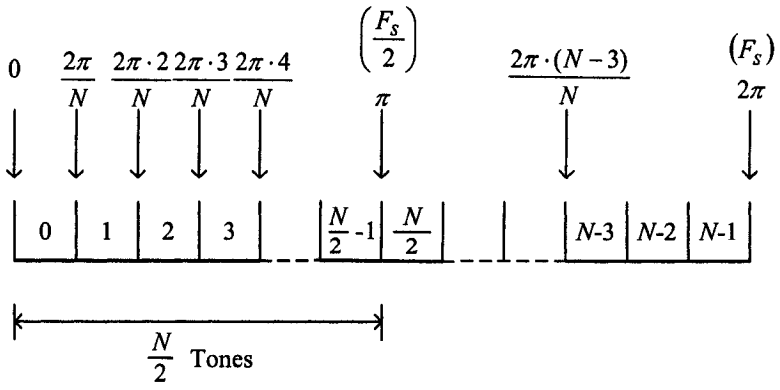


Figure 4-15: Tone complex conjugate symmetry

The tone 0 is not used since it will be modulated at frequency 0 radians/second, which corresponds to the POTS band. Tone 0 also represents DC, which does not propagate well due to capacitor coupling. It is real valued, so not suitable for QAM. Tone *N*/2 is not used since it would be modulated at frequency π and would need to be real valued, and again, not suitable for QAM. To get the values from *N*/2+1 to *N*-1, the Hermitian complex conjugate symmetry is used.

The next step is to perform the normalized IDFT (to maintain power) on this vector of complex values. The normalized IDFT is defined as follows:

$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) \cdot e^{j\left(\frac{2\pi}{N} nk\right)}$ where $X(k)$ corresponds to the complex input vector.

If $X(k)$ has Hermitian symmetry, $X(k) = X^*(N-k)$ for $k = 0,..,\frac{N}{2}-1$, then the IDFT operation will perform the modulation.

In the frequency domain, if the 3 tones in the next example have the same input, the adjacent tones cross each other at half power ($0.5^{½}$ the amplitude if each tone is normalized), defining the tone boundaries. Tone 0 is a special case, but it is never used. Figure 4-16 displays the normalized frequency spectrum of the 3 tones individually.

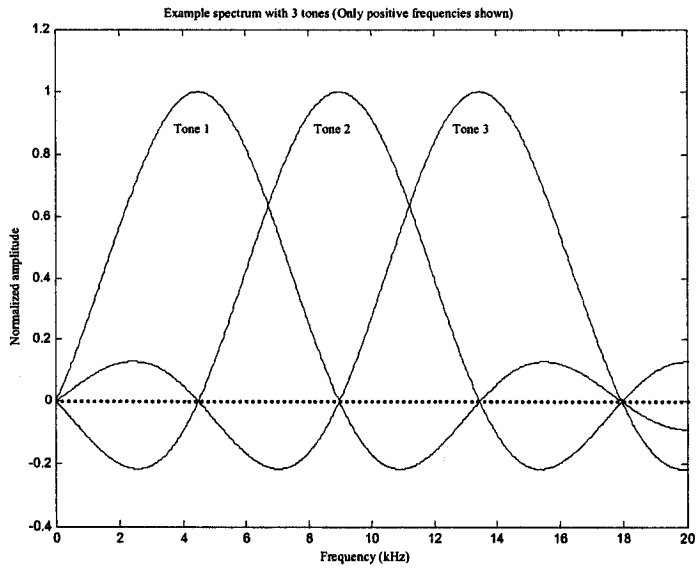


Figure 4-16: Three individual tone spectrums

Since the direct computation of the IDFT using the equation is proportional to N^2, the IFFT is used to implement the IDFT instead, making for a very efficient modulator. Figure 4-17 shows a diagram of the entire modulator.

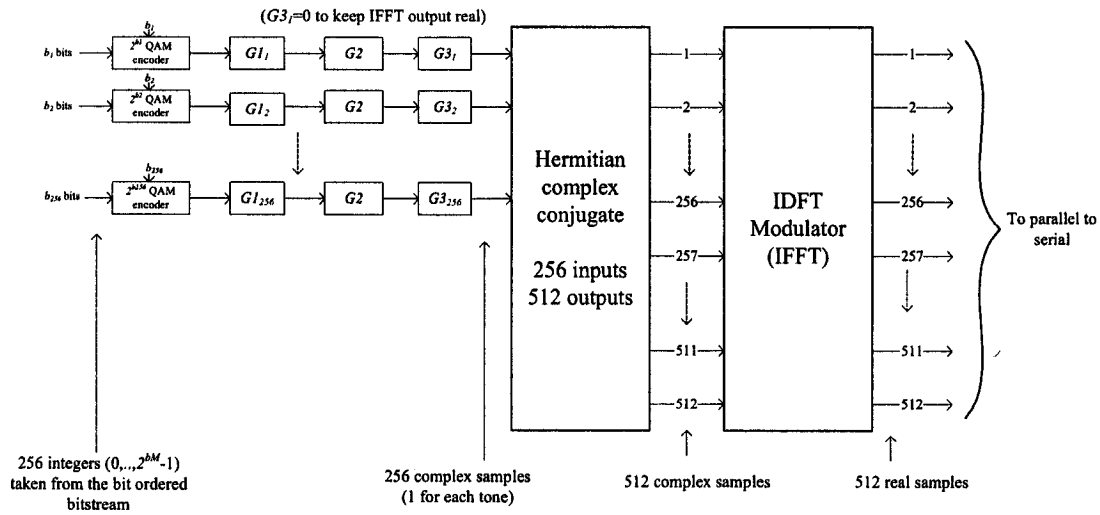


Figure 4-17: IFFT modulator including the QAM constellation and Gain

The output of the Hermitian complex conjugate block has 512 complex samples, of which the first one (labelled 1 in Figure 4-17) and the complex sample labelled 257 are set to 0. The value 257 is set to 0 because the input 257 of the IDFT must be a real value (frequency at $\omega = \pi$), but any value could have been used. Sometimes the average between the value at 256 and 258 is used to avoid a phase discontinuity.

The ADSL type considered in this simulation is FDD-ADSL, therefore the frequency spectrums of both data directions do not overlap. In this simulation, restricting the use of tones 5 to 32 to upstream and 36 to 255 to downstream does the frequency separation. The upstream still uses the 256 tone model presented above but has tones 0-4 and 33-255 set to 0. Performing a 512 point IFFT and calculating the required gains for each tone even though only a fraction are used would uselessly increase complexity and output sampling frequency. Therefore, in practice, the IFFT size is 64 and the sampling rate is 276 kHz for the upstream direction; both cases should yield the same output time signal. The downstream direction still uses a 512 point IFFT but has its inputs 0-31 set to 0. Tones 32-36 are not used in this simulation because of the concerns with overlapping frequency bands, but are sometimes used in practice.

4.7 Cyclic Prefix

In ADSL, a 32 byte cyclic prefix is added at the beginning of each block of 512 samples of real data samples coming from the IFFT modulator for the downstream direction. This is done to try to mitigate for the Inter Symbol Interference (ISI). Here a symbol is considered a block of 512 real data samples, corresponding to a DMT symbol. As ISI will occur at the beginning (and at the end) of the block, it will occur in the cyclic prefix, which will get corrupted. The cyclic prefix is removed at the receiver side, thereby removing much of the corrupted data. It is actually a copy of the last 32 bytes of the real data samples, which are pre-pended to the beginning of the block.

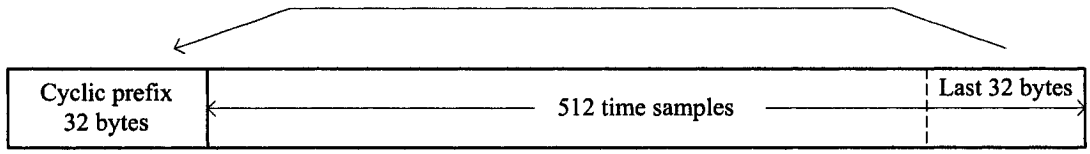


Figure 4-18: Time samples after cyclic prefix (*Prefixed time samples*)

The cyclic prefix could have been 40 samples [G9921], since the symbol rate is 4 kHz, the IFFT size is 512 and the frequency spacing between each of the 256 tones is 4.3125 kHz:

$$\left(512 \text{ Useful bytes / } DMT\ symbol + CyclicPrefix_{Length}\right) \cdot 4 \text{ kHz} = 512 \cdot 4.3125 \text{ kHz}$$
$$= 2.208 \text{ MHz}$$

$$\therefore CyclicPrefix_{Length} = 40$$

But $CyclicPrefix_{length}$ was reduced to 32, saving 8 bytes per symbol so that, for each 68th data symbol, a 544 byte synchronization symbol is inserted. This symbol permits to recover the Sync for small interruptions without retraining. The resulting total number of generated time data samples remains the same over time (but with less useful data):

$$(512+32)\cdot 69 \;=\; (512+40)\cdot 68 \;=\; 37536 \quad \text{Data symbols}$$

Using this small calculation, the number of DMT symbols can also be derived, knowing that the output DAC sampling rate is $F_S = 2.208$ MHz :

$$DMT\ symbols/\text{second} \;=\; \frac{2.208\ \text{MHz}}{544\ Time\ samples/DMT\ symbol}$$

$$=\; 4058.82\ DMT\ symbols/\text{second}$$

This number is the amount that would be generated if all the data were useful, but the synchronization symbol introduced every 68th symbol reduces this to:

$$Useful\ DMT\ symbols/\text{second} = \frac{2.208\ \text{MHz}}{544\ \ Time\ samples/DMT\ symbol}\cdot\frac{68}{69}$$

$$= 4000\ \ Useful\ \ DMT\ symbols/\text{second}$$

This simulation does not include the synchronization word. Therefore, its throughput is considered 4058.8 DMT symbols/second, since the $CyclicPrefix_{length}$ was kept to 32 bytes. In addition, the standard specifies that the length of the cyclic prefix is different for the upstream case and the downstream case, but the downstream length was kept the same for both cases to facilitate the programming of the simulation.

Chapter 5

Twisted Pair Channel Modelling

The cable type generally used for the last mile connection to the customer is a Category-3 Unshielded Twisted Pair (UTP) with an American Wire Gauge (AWG) of 22 (≈0.6 mm diameter), 24 (≈0.5 mm) or 26 (≈0.4 mm) having approximately 2 twists every 2 to 6 inches (Note that the bundle is shielded but not the twisted pairs inside the bundle). The two wires forming the twisted pair are named Tip and Ring. In this document, all transmission lines are considered non loaded (no coil) but they may have bridged taps and various wire gauge sizes.

This chapter will present the models for the differential mode and the common mode transmissions over the twisted pair line, from which the channel frequency and impulse responses will be derived. The last two sections of the chapter discuss the balance of the twisted pair line and the distribution cable. Since ADSL signalling is performed in differential mode, the differential mode transmission model will be covered first and will be followed by the common mode model, derived from the differential mode model.

5.1 Differential Channel

The ADSL signal is transmitted on the telephone line twisted pair using differential signalling, which means that the voltage amplitude applied to the Tip wire is inverted and applied to the Ring wire. This transmission technique, in combination with wire twisting, should eliminate the external interference applied to the line. The twisting will, in theory, allow the external interference to be coupled equally (common mode) and with the same amplitude (magnitude and polarity) on both wires. At the receiver, the difference between the voltages on the Tip and Ring wires is taken, thereby cancelling the external noise. In practice, the cancelling of the external noise is not perfect because of the lack of perfect line balance, but this aspect will be covered in the third section of this chapter.

To deal with the various line wire gauges, the model will break down the transmission line into its different sections. The lengths of line with different wire gauges or bridged taps will be considered independently and represented by a two port model, then the different constituents (lengths of line two port models) of the transmission line will be regrouped into one overall two port model. The transfer function and impulse response of the differential channel will be derived from that model.

5.1.1 Line Section Two Port System Modelling

Many models have been developed for twisted pair transmission lines. To model these transmission lines, the two port system derived from the wave propagation equations was used in combination with four primary parameters of the line which are the resistance R, the inductance L, the conductance G and the capacitance C. These frequency dependent parameters can be used up to 30 MHz and the transmission line section characteristics are considered uniform throughout the section (capacitance, resistance, inductance and conductance). This simulation only considers frequencies from 0 to 1.104 MHz.

Any given line section can be modelled using a two port system. Given a transmission line section:

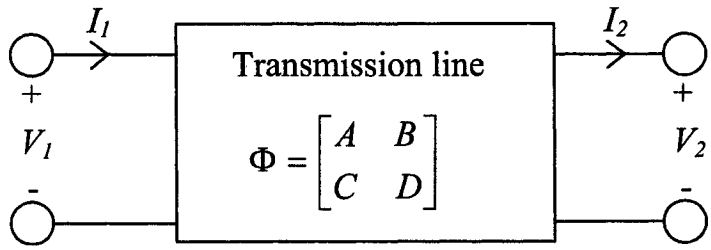


Figure 5-1: Two port system representing a section of transmission line

the two port system can be expressed as follows:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \Phi \cdot \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad \text{yielding the equations:} \quad \begin{aligned} V_1 &= A \cdot V_2 + B \cdot I_2 \\ I_1 &= C \cdot V_2 + D \cdot I_2 \end{aligned}$$

The values of the A, B, C and D parameters in the matrix have been determined by the plane wave propagation theory, and the resulting matrix follows [STA98]:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} \cosh(\gamma d) & Z_0 \cdot \sinh(\gamma d) \\ \frac{1}{Z_0} \cdot \sinh(\gamma d) & \cosh(\gamma d) \end{bmatrix} \cdot \begin{bmatrix} V_2 \\ I_2 \end{bmatrix}$$

where d is the length of the section of line considered, Z_0 is the characteristic impedance of the transmission line section and γ is the propagation constant. The transmission line can be modelled with a cascade of the following circuit:

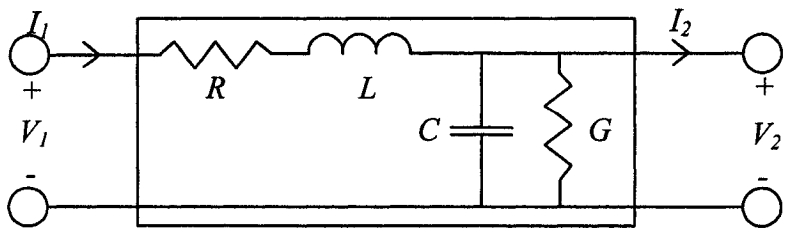


Figure 5-2: Circuit representing an incremental length of transmission line

By using the previous circuit to model an incremental length of transmission line, γ and Z_0 can be calculated using the primary R, L, G and C parameters [JOH97] [STA98]:

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)\cdot(G + j\omega C)} \qquad Z_0 = \sqrt{\frac{(R + j\omega L)}{(G + j\omega C)}}$$

The frequency dependent R, L, G and C parameters, required to obtain Z_0 and γ (note that Z_0 and γ are also frequency dependent), are calculated using the following equations with the constants found in Table 5-1 through Table 5-4, which depend on the wire gauge:

$$R(f) = \frac{1}{\frac{1}{\sqrt[4]{r_{OC}^4 + a_C \cdot f^2}} + \frac{1}{\sqrt[4]{r_{OS}^4 + a_S \cdot f^2}}} \qquad L(f) = \frac{l_0 + l_\infty \left(\frac{f}{f_m}\right)^b}{1 + \left(\frac{f}{f_m}\right)^b}$$

$$C(f) = c_\infty + c_0 \cdot f^{-c_e} \qquad G(f) = g_0 \cdot f^{+g_e}$$

Table 5-1: Parameters used to calculate *R(f)*

Wire gauge	r_{0c}	r_{0s}	a_c	a_s*
24	174.55888 Ω/km	∞ Ω/km	0.053073481	0.0
26	286.17578 Ω/km	∞ Ω/km	0.14769620	0.0

Table 5-2: Parameters used to calculate *L(f)*

Wire gauge	l_o	l_∞	B	f_m
24	617.29539 µH/km	78.97099 µH/km	1.1529766	533.760
26	675.36888 µH/km	488.95186 µH/km	0.92930728	806.33863

Table 5-3: Parameters used to calculate *C(f)*

Wire gauge	c_∞	c_0*	c_e*
24	50 nF/km	0.0 nF/km	0.0
26	49 nF/km	0.0 nF/km	0.0

Table 5-4: Parameters used to calculate *G(f)*

Wire gauge	g_0	g_e
24	234.87476 fS/km	1.38
26 .	43 nS/km	0.70

(*Note that some parameters are equal to 0. This can change for other types of lines)

5.1.2 Cascade of Transmission Line Sections

To model a cascade of transmission lines comprised of sections having different characteristics (varying wire gauges or bridged taps in the case of this simulation), the Φ matrices are simply multiplied to give a global Φ matrix, as in the following example with 2 line sections:

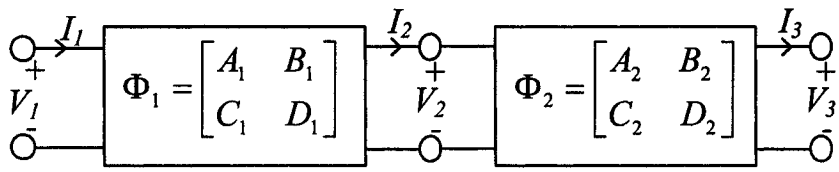


Figure 5-3: Cascaded system block diagram

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \cdot \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \cdot \begin{bmatrix} V_3 \\ I_3 \end{bmatrix} = \Phi_1 \cdot \Phi_2 \cdot \begin{bmatrix} V_3 \\ I_3 \end{bmatrix}$$

Multiplying the matrices in the order of their physical appearance combines the matrices. Since the telephone line is bi-directional, this can be done in either direction. In the present simulation, the matrices are multiplied in the order that they would appear physically in the downstream direction. The parameters A, B, C and D are determined for each section, with its own propagation constant γ and length d, and the appropriate parameters for the wire gauge in order to calculate R, L, G and C for that section.

5.1.3 Bridged Taps

Bridged taps cause periodic notches in the frequency response. A shorter length bridged tap will cause a deeper notch. The notches are commonly 10 to 20 dB deep. A notch in the frequency domain will appear when the length of the bridged tap is a multiple of a

quarter of the wavelength at the corresponding frequency. The bridged taps considered in this simulation are open circuits and are modelled by incorporating a Φ_{bt} two port matrix [STA98]:

$$\Phi_{bt} = \begin{bmatrix} 1 & 0 \\ 1/Z_{bt} & 1 \end{bmatrix} \quad \text{where} \quad Z_{bt} = Z_0 \cdot \frac{\cosh(\gamma \cdot d_{bt})}{\sinh(\gamma \cdot d_{bt})}$$

and γ and Z_0 are calculated exactly in the same way as for a normal line section, using the appropriate parameters for the wire gauge of the section to calculate R, L, G and C for that bridged tap section. The bridged tap is actually a 3 port system simplified to a two port system. For example, in the following circuit line:

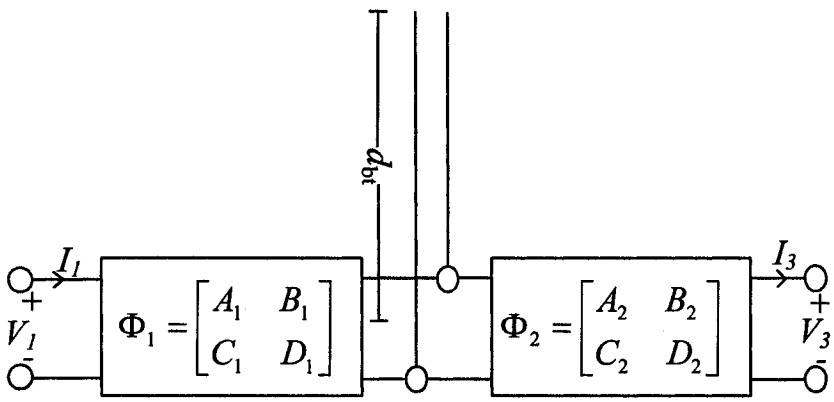


Figure 5-4: Bridged tap section diagram

The matrix cascade corresponding to the transmission line above is:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1/Z_{bt} & 1 \end{bmatrix} \cdot \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \cdot \begin{bmatrix} V_3 \\ I_3 \end{bmatrix} = \Phi_1 \cdot \Phi_{bt} \cdot \Phi_2 \cdot \begin{bmatrix} V_3 \\ I_3 \end{bmatrix}$$

A matrix Φ_{bt} is added for every bridged tap encountered in the order of appearance on the physical line.

5.1.4 The Global Transfer Function

After multiplying all the two port matrices $\begin{bmatrix} A_K & B_k \\ C_K & D_K \end{bmatrix}$ for each section of the transmission line and bridged taps together to yield an overall two port matrix, it is possible to find the overall transfer function of the transmission line in the frequency domain. With the following reference diagram:

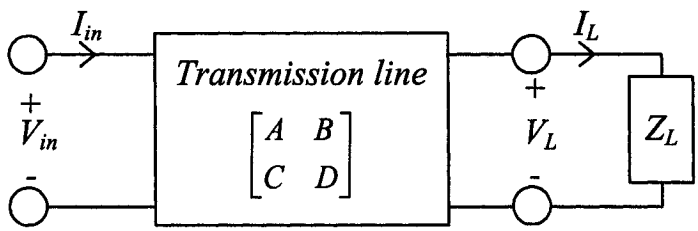


Figure 5-5: Global two port system

the corresponding global two port system is:

$$\begin{bmatrix} V_{in} \\ I_{in} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} V_L \\ I_L \end{bmatrix} \qquad \text{or equivalently:} \qquad \begin{aligned} V_{in} &= A \cdot V_L + B \cdot I_L \\ I_{in} &= C \cdot V_L + D \cdot I_L \end{aligned}$$

Taking only the voltage relation for V_{in}, and dividing by V_L:

$$\frac{V_{in}}{V_L} = A + B \cdot \frac{I_L}{V_L} \quad \Rightarrow \quad \frac{V_{in}}{V_L} = A + B \cdot \frac{1}{Z_L}$$

The derivation of the differential transmission line transfer function $H_d(f)$, the ratio of V_L and V_{in}:

$$H_d(f) = \frac{V_L}{V_{in}} = \frac{1}{A + B \cdot \frac{1}{Z_L}} = \frac{Z_L}{A \cdot Z_L + B}$$

The impulse response is calculated with the inverse Fourier Transform. Z_L is the impedance of the end equipment in differential mode and will be discussed below. Note that the source impedance is not considered. This is because the PSD masks, any power and signals to be sent on the transmission line are calculated as though they were directly applied to the transmission line, without considering the source impedance.

The load impedance Z_L, however, is matched as closely as possible to the transmission line Z_0 to maximize the power transfer (actually $Z_L = Z_0^*$, but Z_0 is approximated to a real value for frequencies above 150 kHz [JOH97]). Z_0 is approximated by a real value because the real part of Z_0 is much larger than the imaginary part. Instead of taking the real part of Z_0 only, the absolute value was taken to minimize the loss of precision. Z_0 depends on the line characteristics and is frequency dependent. The absolute value of Z_0 is plotted here for 24 and 26 gauge 1 km lines from 0 to 1.104 MHz. Frequencies below 25 kHz have no interest since tones 0-5 are not used.

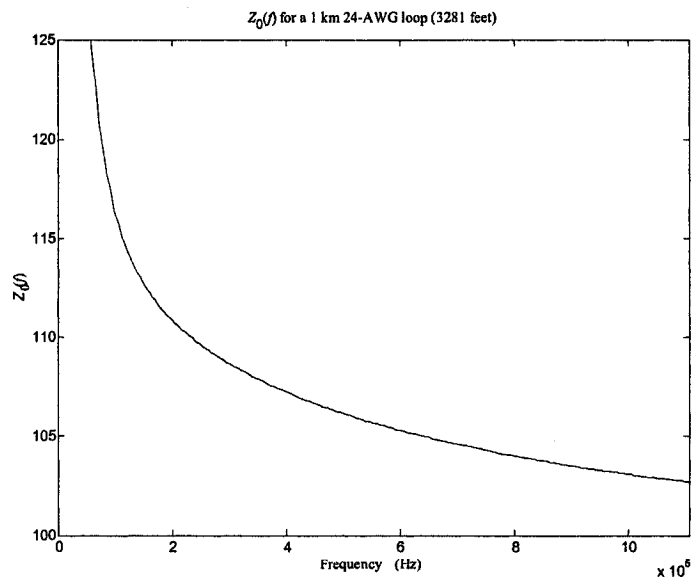


Figure 5-6: Absolute value of differential mode line impedance $Z_0(f)$ of a 24 gauge twisted pair

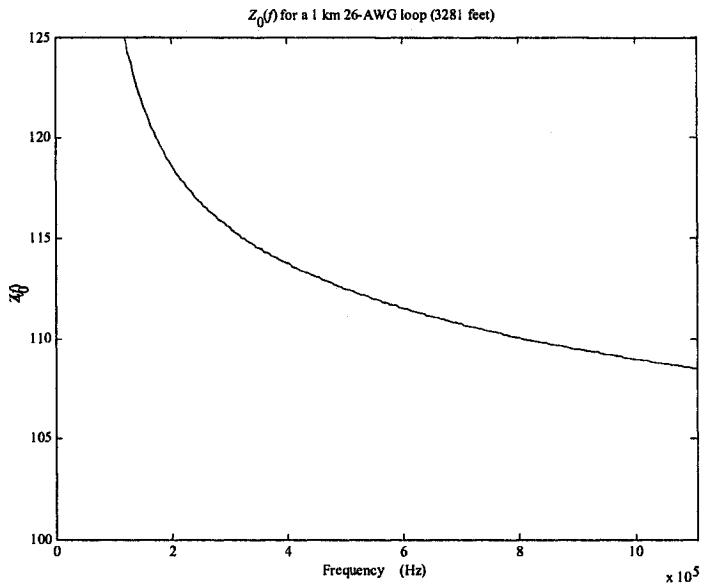


Figure 5-7: Absolute value of differential mode line impedance $Z_0(f)$ of a 26 gauge twisted pair

Figure 5-6 and Figure 5-7 show that any value between 100 and 110 Ω would yield a good approximation. [JOH97] suggests the differential load should be Z_L = 110 Ω for a good match, but makes the assumption of G = 0, which does not reflect reality if differential Category 3 twisted pair is used. For this simulation, Z_L was chosen to be 100 Ω for the differential mode because it is a widely adopted value [LAO02] [STA98], and it reflects the actual hybrid circuit impedance of the physical implementations of many ADSL transceivers and actual cables at high frequencies. Figure 5-8 is the resulting frequency response $H_d(f)$ of a 26 gauge 10,000 feet twisted pair line used in differential mode with the two port model above.

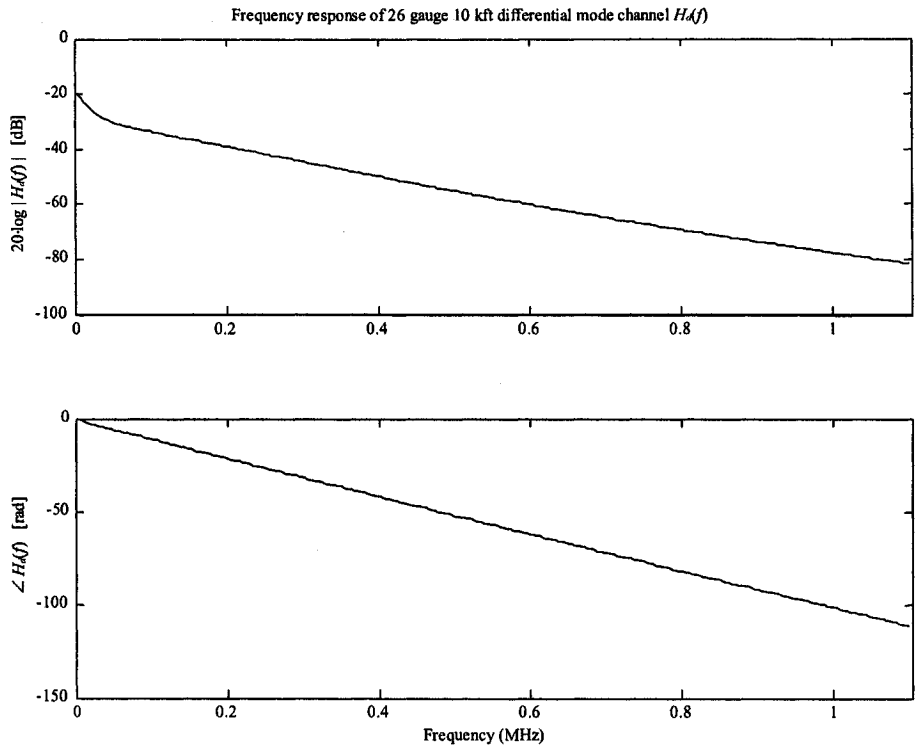


Figure 5-8: Frequency response of 26 gauge 10,000 ft twisted pair in differential

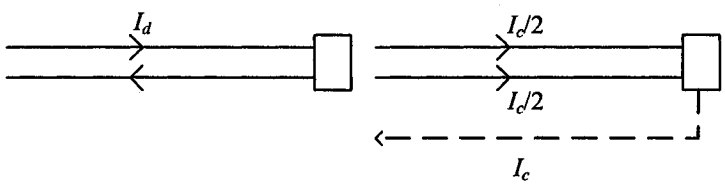
5.2 Common Mode Signalling

The common mode signal is taken to be the sum of the voltages on the Tip and Ring wires. In theory, since the differential signal has equal but inverted amplitudes on the Tip and Ring wires, the sum of the voltages on these wires should be null (if the transmitted signal only is considered). Since the external interference is coupled to both wires with equal amplitudes (same magnitude and polarity) on both Tip and Ring [SHE03], the sum of the voltages should yield the interference signal (with twice the amplitude). Again, this is not quite exact, but will be covered when the loop imbalance is explained.

Since there is very little data on the common mode behavior for twisted pairs at high frequencies, this common mode model was derived from the differential model, mainly

using information found in [COM98] and [FEN99], where the four frequency dependent parameters R, L, G and C were adapted to conform to the known behavior of common mode signals and differential mode signals. Everything else remains the same as for the differential mode. In this section, the frequency dependent differential R, L, G, C, Z_0, Z_L and γ parameters of the previous section are renamed R_d, L_d, G_d, C_d, Z_{0d}, Z_{Ld} and γ_d respectively.

R_c: The common mode resistance R_c is the easiest to derive. When looking at the simplified signal current diagram of the following figure:



a) Differential current I_d b) Common mode current I_c

Figure 5-9: Current return comparison

It can be seen that the differential current I_d must go through the line in both directions, thereby opposed by the resistance of the twisted pair twice. The common mode current I_c is shared between both wires of the twisted pair, and the return current goes through the shield of the bundle (assumed to have a much lower resistance). In Figure 5-10, the differential mode resistance $R_d(f)$ is plotted and it can be seen that the lowest value is around 290 Ω/km (considered the worst case, meaning the return path shielding has more importance). If the bundle shield is considered to have a resistance less than 15 Ω/km, then it is reasonable to assume that the resistance of the differential mode is a little less than twice the common mode resistance.

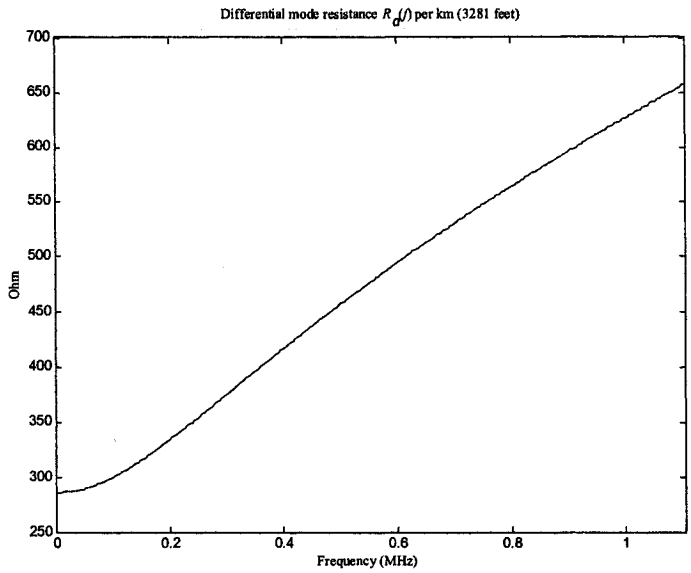


Figure 5-10: Differential mode resistance $R_d(f)$ per km for a 26 gauge twisted pair

For the simulation, the common mode resistance R_c was taken to be 0.55 of the differential resistance R_d for the 26 gauge wire (R_c = 0.55 R_d) and 0.6 for the 24 gauge wire, since the proportionality of the shielding resistance to the loop resistance increases when the loop resistance drops.

C_c: From [COM98] and [FEN99], the common mode capacitance used in the model will differ very little from the C_d. For the simulation, C_c is taken to be: $C_c = 0.9\ C_d$.

G_c: The conductance is a measure of the inverse of the resistance between the conductor and the current return path, and decreases inversely with the conductor diameter. The reason for the decrease in $G_d(f)$ with respect to the diameter is that the larger diameter conductors are generally farther apart from each other, and have a higher resistance, therefore the inverse value is smaller. For this simulation, the value of $G_c(f) = 0.1 \cdot G_d(f)$ since [COM98] neglects the effect of $G(f)$ entirely.

L_c: Having derived the three other parameters helps to derive L_c. [COM98] claims that a good approximation for the common mode impedance is 430 Ω per conductor, yielding a common mode impedance of approximately Z_{0c} = 215 Ω if two conductors are considered together. By looking at the equation for Z_0 and evaluating the Z and Y components at different frequencies for the differential case, it can be seen that, at higher frequencies, the real terms of Z and Y become small compared to the imaginary terms. Thus, eliminating the real terms yields a good approximation at higher frequencies (>200 kHz).

$$Z_0 = \sqrt{\frac{(R + j\omega L)}{(G + j\omega C)}} = \sqrt{\frac{Z}{Y}} \Rightarrow Z_0 = \sqrt{\frac{(j\omega L)}{(j\omega C)}} = \sqrt{\frac{L}{C}}$$

To relate the differential and common mode:

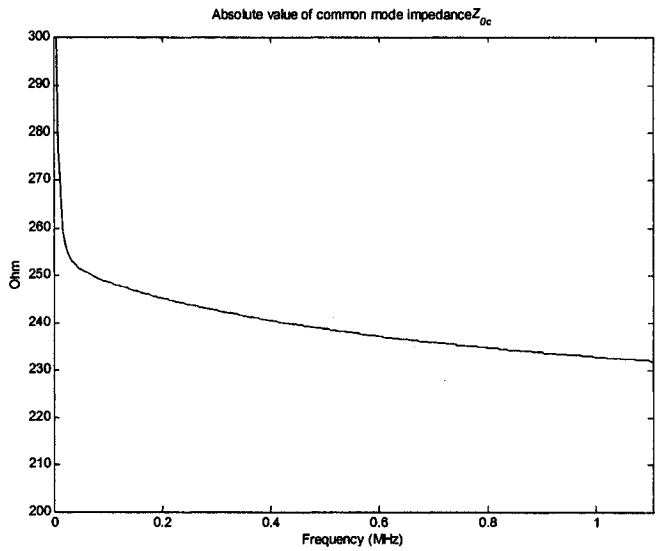
$$Z_{0C} = \sqrt{\frac{L_C}{C_C}} = \left(\frac{215}{100} Z_{0d}\right) = \sqrt{\frac{x \cdot L_d}{0.9 \cdot C_d}} \quad \Rightarrow \quad Z_{0d} = \frac{100}{215}\sqrt{\frac{x}{0.9}} \cdot \sqrt{\frac{L_d}{C_d}}$$

The equation above is valid if the factor $\frac{100}{215}\sqrt{\frac{x}{0.95}}$ is equal to unity, in which case $x = 4.16$. Therefore, $L_C = 4.16 \cdot L_d$.

To obtain the frequency and impulse responses for the common mode, the same technique is used as in the differential mode, which combines all the line and bridged tap sections. After the global two port matrix has been established for the common mode channel, the following relationship can be used to find the common mode frequency response $H_c(f)$, and consequently the impulse response:

$$H_c(f) = \frac{Z_{Lc}}{A_c \cdot Z_{Lc} + B_c}$$

where A_c and B_c are the parameters of the global common mode two port matrix (which are calculated using the four common mode primary constants R_c, L_c, G_c and C_c with the same relationships as in the differential section). Z_{Lc} is the common mode load impedance and differs from the differential mode load impedance. Since both the differential mode and the common mode need to be terminated by different load impedances, Z_{Ld} and Z_{Lc} respectively, a special circuit must be designed, and is discussed in [COO93]. To find the proper impedance value to terminate the common mode line, the absolute value of the common mode impedance Z_{0c} of the resulting model is displayed in Figure 5-11 for a 24 gauge wire.



*

Figure 5-11: Absolute value of common mode impedance Z_{0c} for a 26 gauge wire

Figure 5-11 shows that the resulting impedance is approximately 230 - 240 Ω for a 26 gauge wire. As mentioned previously, the impedance value was measured in [COM98] at approximately 430 Ω at higher frequencies for a single 26 gauge wire, taken to be 215 Ω for a two conductor model. The slight impedance difference is considered acceptable for this simulation, therefore the parameters are assumed to be correct. The common mode impedance Z_{0c} was calculated without using any approximations (even though the R term has much lesser effect that C and L in the formula, it still bears an

impact on the value of Z_{0c}). The resulting model has a common mode impedance between 220 and 235 Ω for a 24 gauge wire and between 230 and 240 Ω for a 26 gauge wire, therefore the load impedance seen in common mode was chosen to be $Z_{Lc} = 230\ \Omega$.

5.2.1 Common Mode Parameter Choices Verification

Obviously, the validity of the derivation of the common mode parameters relies heavily on a few assumptions. The exactitude of each parameter is not primordial since the common mode impulse response will not be used to transmit a signal, as will be seen later, but to evaluate the phase difference between the differential and common modes.

Still, the validity of the choice of two of the factors used to adapt the differential mode model parameters into the common mode parameters can be verified by using the group velocity. The group velocity of a differential mode signal is commonly taken as $v_d = 0.677 \cdot c$ [LAO02], where c is the speed of light ($c = 3.8 \cdot 10^8$ meters/second). Using the relationship for the propagation constant γ in [JOH97] and expanding. For higher frequencies, only the term with ω^2 is kept, as all others will be much smaller:

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)\cdot(G + j\omega C)} = \sqrt{RG + j\omega LG + j\omega RC + j^2\omega^2 LC}$$

$$\cong \sqrt{-\omega^2 LC} = j\omega\sqrt{LC}$$

Yielding: $\beta \cong \omega\sqrt{LC}$. Knowing that the group velocity is $v = \frac{\omega}{\beta}$, then $v = \frac{1}{\sqrt{LC}}$.

Therefore the relationship between the differential mode group velocity v_d and the common mode group velocity v_c is:

$$v_c = \frac{1}{\sqrt{L_C \cdot C_C}} = \frac{1}{\sqrt{(4.17 \cdot L_d)\cdot(0.9 \cdot C_d)}} = \frac{1}{\sqrt{4.17 \cdot 0.9}\sqrt{\cdot L_d \cdot C_d}}$$

$$= \frac{0.517}{\sqrt{L_d \cdot C_d}} = 0.5176 \cdot v_d = 0.517 \cdot (0.677c)$$

$$= 0.35 \cdot c$$

According to measurements in [COM98], the common mode group velocity is $0.34 \cdot c$, or approximately half the velocity of the differential mode propagation. Therefore, the approximations given in this chapter for the common mode L_C and C_C parameters can be considered valid for the purposes of this simulation. The most important element is the common mode propagation delay in comparison to the differential mode propagation delay. The attenuation will not be explicitly considered since only the phase difference (delay) is needed, as will be seen in the next chapter. Therefore, the values of R_C and G_C are less critical, but can still be partially verified by looking at the common mode magnitude transfer function. [COM98] claims that, at 1 MHz, the common mode attenuation is approximately 6.4 dB/km when both wires forming the pair are considered together and the line is properly terminated for the considered mode. The following figure displays the magnitude of the common mode transfer function based on the model derived in this chapter to compare it with the actual measurement.

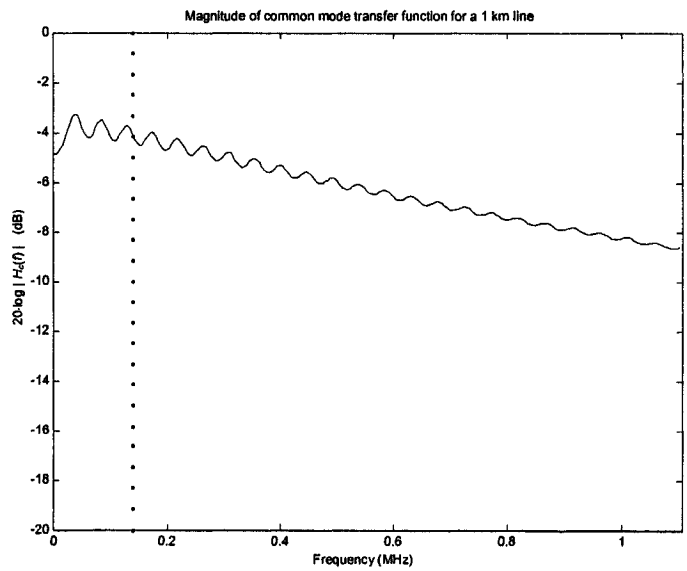


Figure 5-12: Magnitude of common mode transfer function for a 1 km line

From Figure 5-12, it can be seen that the common mode model will present an attenuation of 8.25 dB/km at 1 MHz, which is relatively close to the desired attenuation. This verification does not permit to determine directly that R_c and G_c are correct since the transfer function depends on all four parameters R_c, L_c, G_c and C_c. Conversely, since the transfer function does depend on all four parameters, it gives a general idea that they could all be considered roughly accurate.

5.3 Line Balance

The balance is a measure of the ability of the twisted pair transmission line to maintain the balance between the two signals (equal magnitude, inverted polarity) on each wire of the pair that comprises the differential signal. The line's ability to maintain the balance is affected by many factors such as the number of twists per unit distance of the line, transmitter and receiver design and the uniformity of the characteristics of the line [BRO03]. As the line balance becomes lower, more differential signal leaks into the common mode signal, and vice-versa. Since the external interference couples to the line in common mode, if the balance is low, it will be partially transferred into differential mode, thus causing noise in the useful signal. The resulting magnitude of the balance function is defined by equation 5-1 for a Category 3 twisted pair (taken from [STA98]) and is plotted in Figure 5-13:

$$B(f) = \begin{cases} \sqrt{10^5} & 0 \le f < 150\,\text{kHz} \\ \sqrt{10^5 \cdot \left(\frac{150 \cdot 10^3}{f}\right)^{1.5}} & 150\,\text{kHz} \le f < 30\,\text{MHz} \end{cases} \qquad (5\text{-}1)$$

This magnitude of the line balance is a function of the frequency; the line balance gets lower as the frequency increases, implying that more signals will leak from the differential to common mode and vice-versa at higher frequencies. Even though there are imbalance points across the entire transmission line (bridged taps), this magnitude

transfer function specifies the resulting balance of the entire transmission line (the average resulting balance) and is applied at the end of the transmission line receiver.

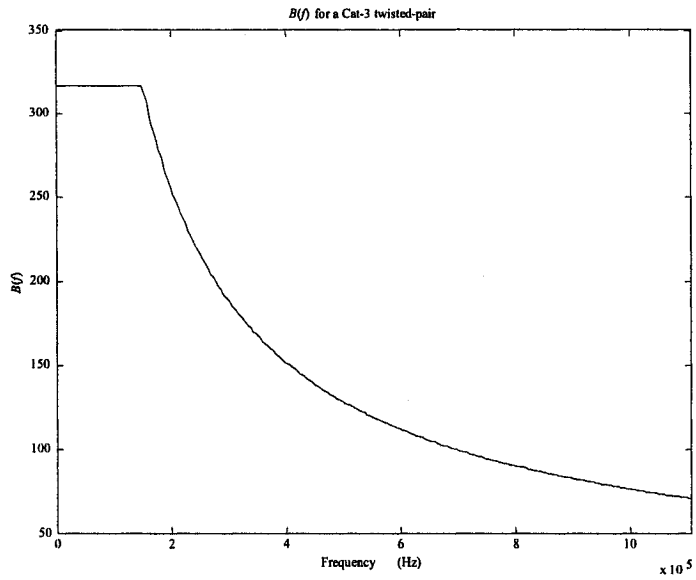


Figure 5-13: Magnitude of the balance *B(f)* of a Category 3 twisted pair line

The magnitude balance *B(f)* is used to attenuate the signal when it is transferred from one mode to the other. The useful differential signal will partially transfer to the common mode by being attenuated by this balance magnitude transfer function *B(f)*, although this is considered negligible since the common mode signal is typically larger in practice. Conversely, the common mode signal (interference including crosstalk) will transfer to differential mode by being attenuated by *B(f)*.

The balance function has only its magnitude specified but no phase. The phase can be considered zero, or the phase information can be created, as will be done in this simulation for the FEXT and is described in the next chapter.

5.4 Simulated Bundle

There are many possible configurations for the telephone loop plant. The one considered is a particular case of the structure shown in Figure 5-14, which is a slightly modified variant taken from [BIN00]. The information for the following configuration is mostly taken from [BIN00].

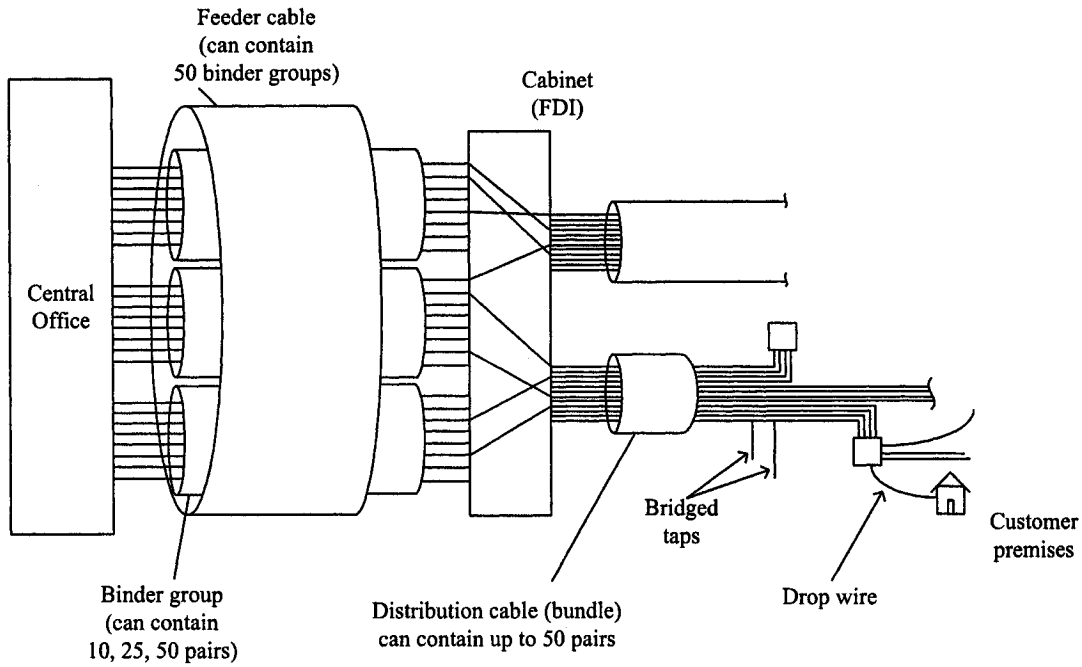


Figure 5-14 : One type of configuration of the loop plant

In the general case, from a Central Office (CO), feeder cables can contain up to 50 binder groups, where each binder group can contain 10, 25 or 50 twisted pairs. Usually a cabinet is located closer to the customers, where cross-connections are done between the pairs contained in the binder groups and the pairs in the distribution cables. The most common telephone distribution cables (sometimes referred to as bundles in this document) generally contain 50 Category 3 unshielded twisted pairs with wire gauges ranging from 26 to 19 AWG. Another type of cable, called a drop wire, is used to extend

the loop from the distribution cable to the customer premises. The drop wire is usually of poor quality or even a flat untwisted cable, but fortunately is only a few feet long (100').

The wire gauge in any given binder is the same for each pair. The wire diameter of the twisted pairs in the feeder cable is generally smaller than that of the twisted pairs in the distribution cables, because there is a limit on the total loop resistance (1500 Ω for Revised Resistance Design, or RRD) and the distance between the CO and customer premises varies. To save on copper, the wire gauge can be kept small unless it exceeds the maximal resistance; therefore, the diameter of the part of the loop from the CO to the Cabinet/FDI (Feeder Distribution Interface) is kept small. If a greater distance is required, the wire diameter is increased for the remaining distance to avoid exceeding the maximal set resistance.

The inclusion of bridged taps is common in the North American loop plants for a number of reasons, including repair work, shared lines (old practice), expansion and flexibility. Bridged taps can be of any length (but generally shorter than 2,000 feet), of any gauge from 26 to 19 AWG and anywhere along the distribution cable. There can be multiple bridged taps on one twisted pair line in a distribution cable or none at all, and each line can have a different configuration of bridged taps.

The considered loop plant for the simulation in this document contains binder groups with 50 twisted pairs; the distribution cables also have 50 twisted pairs and all the pairs in any given binder group are directly connected to all the pairs in one distribution cable. It is also assumed that the twisted pairs are unshielded, but that the bundle is shielded. The bridged tap configuration of the simulated distribution cable is considered identical for all twisted pair loops in the distribution cable; this permits to simplify certain formulas for the crosstalk (explained later). The short drop wires are disregarded in this simulation because they do not pick up crosstalk from other sources. However, it is important to note that they are not well balanced, sometimes suspended in the air unshielded and act as an antenna for RFI.

Chapter 6

Crosstalk Model

This chapter presents the global channel model including the crosstalk interference. The first section covers the Near-End Crosstalk (NEXT), and then explains how the time samples for the NEXT interference in the differential mode and in the common mode are generated in this simulation. The Far-End Crosstalk (FEXT) will then be covered, explaining how the time samples are generated, and how the delay along the line is modelled. Finally, the model that combines NEXT and FEXT from several different types of DSL technologies causing crosstalk in the bundle will be explained.

6.1 Near-End Crosstalk

NEXT is the interference from outgoing transmitters close to the considered incoming receiver in the same bundle. NEXT becomes problematic when the interference is in an overlapping frequency band with the considered signal. An excellent representation of NEXT is reproduced here from [BIN00] in Figure 6-1:

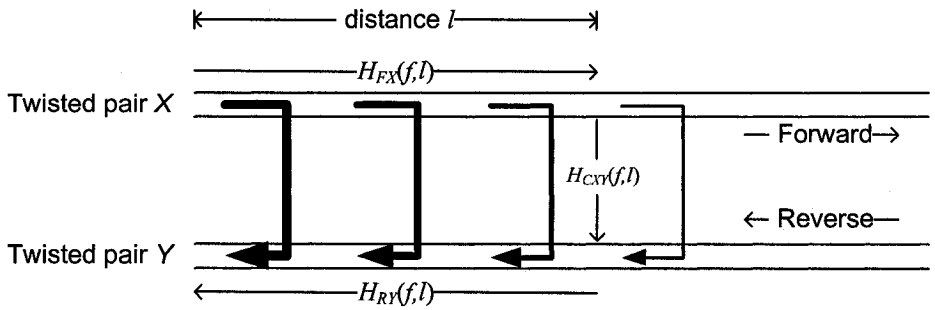


Figure 6-1: NEXT inter-pair coupling

In Figure 6-1, the signal from a transmitter in proximity on twisted pair X is transmitting and the interference is coupled onto twisted pair Y and is received (assumed to be in the same bundle). The thickness of the lines represents the intensity of the NEXT coupling. The crosstalk on pair Y at the receiver $R_Y(f)$ is expressed with (found in [BIN00]):

$$R_Y(f) = O_X(f) \cdot \int_0^L H_{FX}(f,l) \cdot H_{CXY}(f,l) \cdot H_{RY}(f,l)\, dl \qquad (6\text{-}1)$$

where:

$R_Y(f)$: Interference received at the receiver of twisted pair Y

$O_X(f)$: Output of the transmitter on twisted pair X

$H_{FX}(f,l)$: Forward direction transfer function of the twisted pair X from the transmitter Y up to distance l

$H_{RY}(f,l)$: Reverse direction transfer function of twisted pair Y from distance l to the receiver

$H_{CXY}(f,l)$: Coupling from twisted pair X to twisted pair Y at distance l from the transmitter

The integration in equation 6-1 is performed from distance 0 up to distance L, which can be considered as the length of the line, or infinite if the line is relatively long. Equation 6-1 is hard to evaluate because the coupling transfer function between twisted pair X and twisted pair Y, $H_{CXY}(f,l)$, must be known at all points of l, which is not the case. This integration yields the NEXT caused by only one twisted pair X. If the NEXT crosstalk

from all the lines is desired, the coupling transfer functions at all points of l must be known for all twisted pairs in the bundle to pair Y, which is presently impractical.

Fortunately, many simplifications to equation 6-1 can be done to produce a model of the NEXT crosstalk. The first simplification is that, since both twisted pairs are in the same bundle, there is a good probability that they both have similar transfer functions, in the reverse and forward directions, so $H_{FX}(f,l) = H_{RY}(f,l)$, and can both be approximated with a decreasing exponential. This assumes that both twisted pairs have the same configuration of bridged taps, which is rarely the case in reality, but as the length of the line is not really a factor for NEXT, the bridged tap configuration and line arrangement are ignored in this model.

Additionally, by accepting the loss of phase information, $O_X(f)$ can be replaced by its power spectral density (yielding another PSD). Another improvement would be to have an equation that includes the number of disturbers of the same type N (having the same power spectral density) in the bundle. Simply adding a term N is not realistic as it is overly pessimistic to add all the respective PSDs. Based on measurements in [STA98], a better approximation is to have N to the power 0.6.

All these simplifications can lead to the following equation ([STA98]), which is based on many measurements and experiments to yield an average crosstalk PSD equation:

$$PSD_{NEXT} = PSD_{Disturber} \cdot \left(\frac{N}{49}\right)^{0.6} \cdot 10^{-13} \cdot f^{1.5} \qquad (6\text{-}2)$$

with:

PSD_{NEXT}: The PSD of the crosstalk received by receiver on twisted pair Y in the differential mode

$PSD_{Disturber}$: PSD of an adjacent transmitter in differential mode.

N: Number of crosstalkers with the same considered $PSD_{Disturber}$ in the bundle

f: Frequency (Hz)

This equation considers that the bundle has 50 twisted pairs and the crosstalkers are all of the same type. Thus, all the transmitters causing NEXT on the considered line have the same transmit PSD, which implies they are all of the same technology and they operate at approximately the same speed. The situation where more than one type of technology is used in the bundle will be covered later. Obviously, the transmitters causing the disturbance are not at the same location as the receiver they are causing disturbance to, as shown in the following diagram [BIN00]:

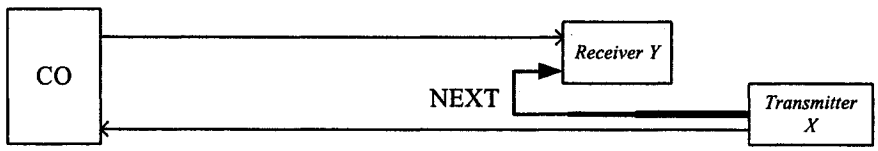


Figure 6-2: Transmitter and receiver at different distances from CO

In the situation of Figure 6-2, the NEXT is attenuated before causing problems to the receiver. This attenuation is negligible enough to consider that the transmitter and receiver are at the same distance from the CO if they are in the same bundle (mostly for ADSL, as the frequency band used for upstream is not heavily attenuated). Their physical distance will be fairly close, so equation 6-2 is still a good approximation.

6.1.1 NEXT Time Samples

This sub-section describes how the time domain samples representing the NEXT interference at the receiver are generated in the common and differential modes for one type of crosstalker.

6.1.1.1 Differential Time Samples

Equation 6-2 gives the differential mode PSD of the NEXT at the receiver. To get differential time domain interference samples from the desired NEXT PSD, White Gaussian Noise (WGN) is injected into a system, named *NEXT Differential PSD Shaping*, $S_{Nd}(f)$, with magnitude $|S_{Nd}(f)|$ and phase $\angle S_{Nd}(f)$, which will shape the WGN to the desired NEXT PSD. WGN is an acceptable source because the amplitudes (in time) of crosstalk samples are usually well approximated by a Gaussian probability distribution [KER93] (and match the frequency characteristics after shaping).

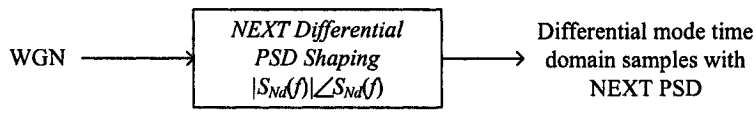


Figure 6-3: Shaping from WGN to generate the output with the desired PSD

The Yule-Walker method was used with the desired magnitude $\sqrt{PSD_{NEXT}}$ where equation 6-2 gives PSD_{NEXT}. This works because, in a given system $S_{Nd}(f)$, the input and output PSD have the following relation:

$$PSD_{Output} = |S_{Nd}(f)|^2 \cdot PSD_{Input}$$

$$PSD_{NEXT} = |S_{Nd}(f)|^2 \cdot PSD_{WGN} = |S_{Nd}(f)|^2 \cdot \sigma^2 = |S_{Nd}(f)|^2 \cdot 1$$

$$\therefore |S_{Nd}(f)| = \sqrt{PSD_{NEXT}}$$

The Yule-Walker method takes only the desired magnitude function as a parameter, yields a system with the desired magnitude and adds a "random" phase element. For verification purposes, the PSD of the resulting signal generated with this technique was compared to the PSD of actual crosstalk measurements in [STA98] for different DSL technologies and were found to concord.

6.1.1.2 Common Mode Time Samples

The assumption is that the crosstalk interference was originally coupled to the common mode but, because of an imperfect line balance, was slightly transferred ('leaked') to differential mode. The NEXT PSD yielded by equation 6-2 is the resulting PSD at the receiver in differential mode after the leakage, but the common mode PSD is not available (and usually not significant information). Therefore, using the assumption of leakage to differential mode through imperfect line balance, the common mode interference time domain samples can be derived (or found in reverse) from the differential interference time domain samples combined with the magnitude scaling function $B(f)$ of equation 5-1, which represents the leakage. This leakage magnitude function $B(f)$ is the amount of attenuation that a signal gets when going from the common mode to the differential mode; since the reverse (differential to common mode) is desired, the frequency response of the differential time domain samples must be multiplied by this "attenuation" (instead of divided), which will yield a larger amplitude signal for the common mode. The theoretical task to perform is displayed in Figure 6-4, which follows:

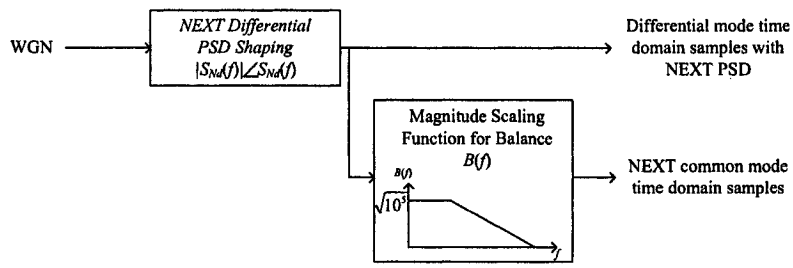


Figure 6-4: Common mode time samples generation

Figure 6-4 can be rearranged and presented as follows:

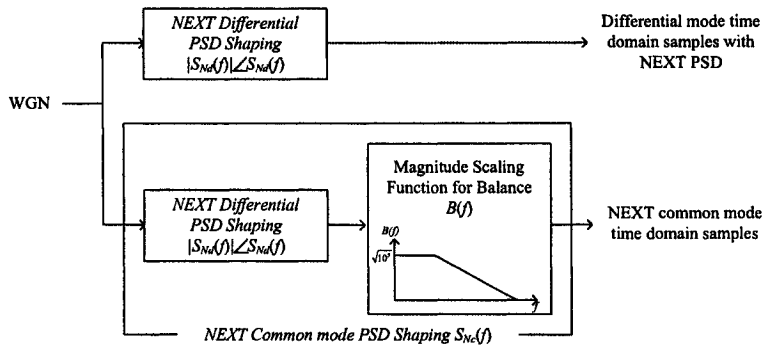


Figure 6-5: Common mode NEXT time samples generation

$B(f)$ is a magnitude scaling function and, since no phase information is specified, $B(f)$ is assumed to be zero-phase (not to introduce any delay). Instead of trying to program the zero-phase *Magnitude Scaling Function for Balance*, $B(f)$, of Figure 6-5, the *NEXT Common PSD Shaping* transfer function $S_{Nc}(f)$ was generated using the *NEXT Differential PSD Shaping* phase $\angle S_{Nd}(f)$, combined with the magnitude $|S_{Nd}(f)|$ multiplied by the $B(f)$ magnitude scaling function. This procedure is better represented in Figure 6-6, which follows:

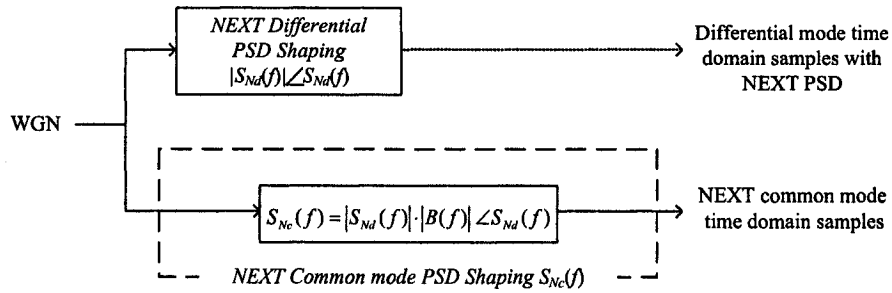


Figure 6-6: The actual *FEXT common mode PSD shaping* function $S_F(f)$

To summarize, since Yule-Walker was used to generate the *NEXT Differential PSD Shaping*, the phase information $\angle S_{Nd}(f)$ was generated by the algorithm, and therefore the technique of Figure 6-6 maintains the same phase for the *NEXT Differential PSD Shaping*

and *NEXT Common mode PSD Shaping*, and adjusts the magnitude of the common mode generator by $B(f)$ accordingly.

The WGN for both the *NEXT Differential PSD Shaping* transfer function and the *NEXT Common mode PSD Shaping* is the same source, to maintain correlation. The *NEXT Differential PSD Shaping* transfer function depends on the crosstalker PSD and the number of crosstalkers of that type present in the bundle. Therefore, the block diagram of Figure 6-6 is only valid to generate the NEXT for only one type of crosstalker.

The situation where more than one type of technology operates in the bundle is covered later, but basically, a block diagram like in Figure 6-7 is necessary to generate the NEXT for each type of crosstalk technology present in the bundle. The representation of the *NEXT Common PSD Shaping* in Figure 6-7 returns to the block diagram of Figure 6-5 to abstract from the inner workings of the way it was programmed.

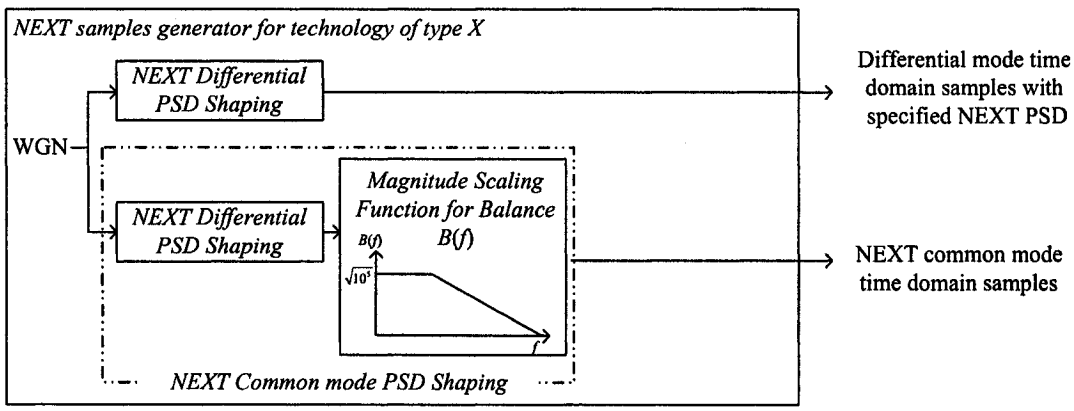


Figure 6-7: NEXT time domain samples generation for crosstalker of type *X*

The output samples generated in Figure 6-7 represent the total NEXT that would have been picked up on the considered twisted pair transmission line from all the other twisted pair lines in the bundle operating the DSL technology of type *X*. They are therefore added at the receiver in both the common mode and the differential mode (separately).

6.2 Far-End Crosstalk

Far-End Crosstalk (FEXT) is the interference from outgoing transmitters at the far end of the considered incoming receiver in the same bundle. Like NEXT, FEXT becomes problematic when the interference is in an overlapping frequency band with the considered signal. A graphical representation of FEXT is reproduced here from [BIN00] in the following figure:

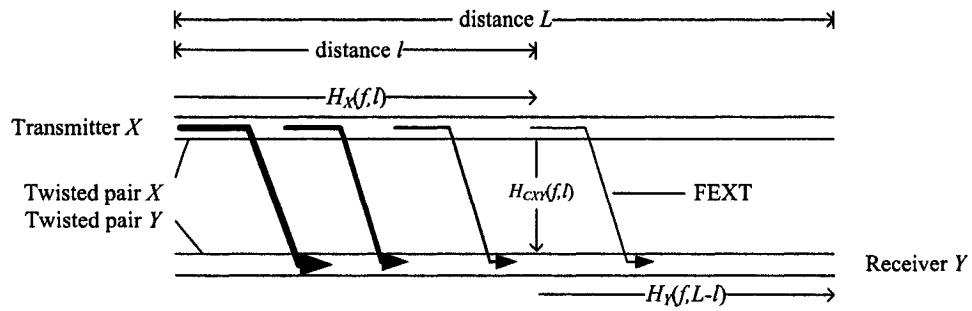


Figure 6-8: FEXT inter-pair coupling

In Figure 6-8, the signal from a transmitter X on twisted pair X is transmitting and the interference is coupled onto twisted pair Y and being received by Y as they both are in the same bundle. The line thickness represents the intensity of the FEXT coupling. The FEXT at the receiver Y, $R_Y(f)$, is expressed in the following equation [BIN00]:

$$R_Y(f) = O_X(f) \cdot \int_0^L H_X(f,l) \cdot H_{CXY}(f,l) \cdot H_Y(f,L-l)\,dl \qquad (6\text{-}3)$$

with:

$R_Y(f)$: Interference received at the receiver of twisted pair Y

$O_X(f)$: Output of the transmitter on twisted pair X

$H_X(f,l)$: T. F. of the twisted pair X from the transmitter Y up to distance l

$H_Y(f,l)$: T. F. of the twisted pair Y from distance l to the receiver Y at distance L

$H_{CXY}(f,l)$: Coupling from twisted pair X to twisted pair Y at distance l from the transmitter

The integration in equation 6-3 is performed from distance 0 up to distance L, which can be considered as the length of the line. As in the NEXT case, equation 6-3 is hard to evaluate because the coupling transfer function between twisted pair X and twisted pair Y, $H_{CXY}(f,l)$, must be known for all distances l, and the equation only gives the result for one twisted pair causing crosstalk. Using many of the same arguments as for the NEXT case, and taking account of the FEXT's heavy dependency on the line characteristics and length, the assumptions can lead to the following equation taken from [STA98]:

$$PSD_{FEXT} = PSD_{Disturber} \cdot |H(f)|^2 \cdot \left(\frac{N}{49}\right)^{0.6} \cdot 9 \cdot 10^{-20} \cdot d \cdot f^2 \qquad (6\text{-}4)$$

with:

PSD_{FEXT}:	The PSD of the crosstalk received by receiver Y
$PSD_{Disturber}$:	PSD of a remote transmitter for FEXT in differential mode
$H(f)$:	Differential channel transfer function for the considered line
N:	Number of crosstalkers that have the considered $PSD_{Disturber}$
d:	Loop length (in feet)
f:	Frequency (Hz)

Just like the NEXT equation, equation 6-4 considers a bundle of 50 twisted pairs with all the crosstalkers of the same type and is based on many measurements and experiments that were performed to find an equation that would yield an average PSD model for FEXT crosstalk. The situation with more than one type of technology in the bundle is covered later. [BIN00] describes many good examples of different possible locations of the transmitters causing FEXT, which are displayed in Figure 6-9:

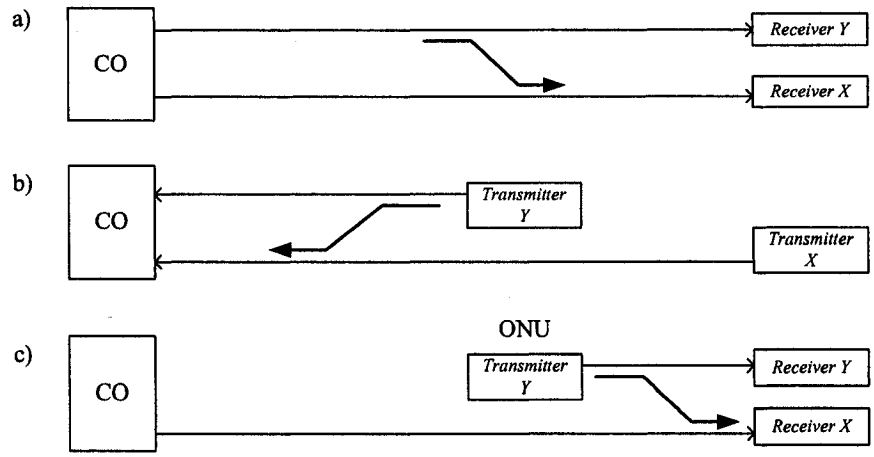


Figure 6-9: Different configurations causing varying degrees of FEXT

Only case a) of Figure 6-9, where the line lengths are the same, will be considered for this simulation, but for both the upstream and downstream cases, even though cases b) and c) would incur more severe FEXT interference. Case b) is frequent the *CO* is not usually of great concern since the line is usually much longer than the distance difference.

6.2.1 FEXT Time Samples

This section describes how the time domain samples representing the FEXT interference at the receiver are generated in both the common and the differential modes for one type of crosstalker. The technique used to generate the FEXT time samples for one crosstalker starts by using the same process as to generate the NEXT time samples for one crosstalker, and then requires one additional step to delay the common mode time domain samples to simulate the propagation along the twisted pair line, which propagates at a slower speed than the differential mode signal.

The differential mode time domain samples and non delayed common mode time domain samples are generated by a block diagram with the same structure as the NEXT generator, and are displayed in the following figure:

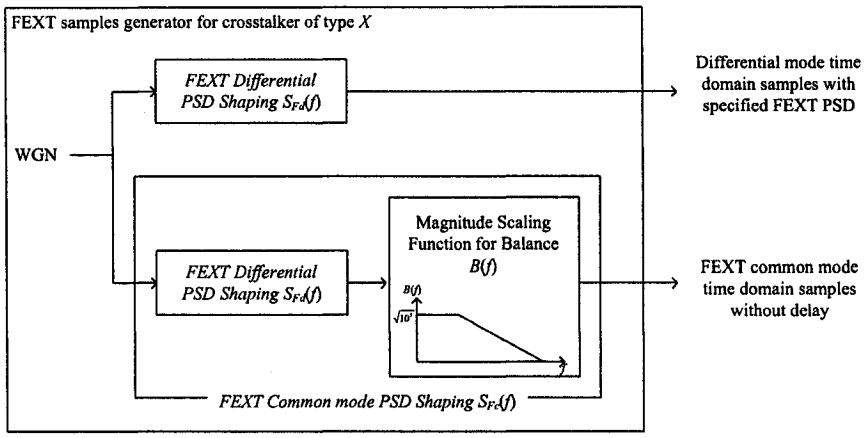


Figure 6-10: FEXT non delayed time samples generator for a crosstalker of type *N*

The *FEXT Differential PSD Shaping* filter $S_{Fd}(f)$ is designed using the Yule-Walker technique with equation 6-4 determining the desired magnitude of:

$$|S_{Fd}(f)| = \sqrt{PSD_{FEXT}},$$

and the *FEXT Common mode PSD Shaping* filter's magnitude $|S_{Fc}(f)|$ is:

$$|S_{Fc}(f)| = B(f) \cdot \sqrt{PSD_{FEXT}}$$

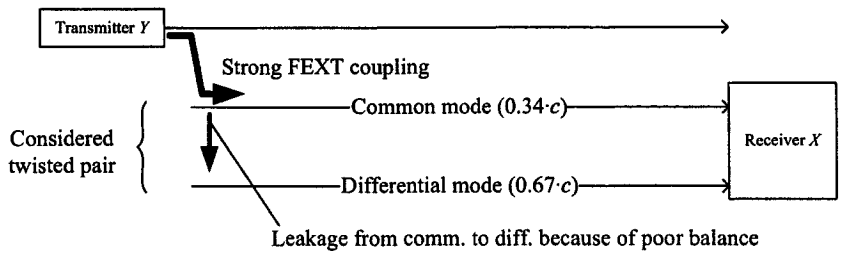
and is constructed as in the NEXT case. Independent WGN sources are used for each crosstalk type (technology) in the bundle as well as for the NEXT and FEXT.

6.2.2 FEXT Common Mode Propagation Delay

As shown in the previous chapter, the common mode signal travels slower (about half the speed) than the differential mode signal. The consequence is that, at the receiver, the corresponding common mode FEXT samples will be delayed in comparison to the differential mode FEXT samples. It should be noted that the FEXT is coupled along the entire line, therefore not all the components of the FEXT interference at the receiver will have incurred the entire delay of the line. However, from Figure 6-11, the strong FEXT occurs at the beginning of the line, travels along the entire line and so incurs the full delay of the line. There is a smaller amount of attenuation in the common mode than in the differential mode; the strongly coupled component of the FEXT interference will get to the receiver at a greater strength than the weaker coupled component farther down the line.

Therefore, to get a good model, on the basis of the argument that the prevalent FEXT component is the one that was coupled at the beginning of the line, the delay to be applied to the common mode FEXT samples should be the delta of time it takes a signal to travel the line in the differential mode and in the common mode. This is not a perfect simulation of the delay even though the FEXT from other sources mostly gets coupled to common mode at the beginning of the line where its strength is greater, the common mode signal FEXT transfers to differential mode all along the line due to imperfect imbalance. This delay implies that the transfer of the FEXT from common mode to differential mode occurs at one point (at the beginning of the line) and then both signals travel along the twisted pair in common mode and in differential mode at their respective speeds ($0.33 \cdot c$ and $0.67 \cdot c$).

a) What is being simulated:



b) Reality:

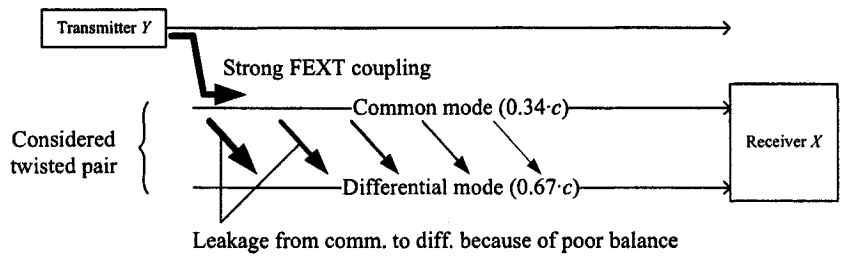


Figure 6-11: Difference between reality and simulation for the delay consideration

In Figure 6-11 b), it can be seen that each component of the leakage to differential mode will have a different delay. Unfortunately, there is no phase information available to model this properly for the purpose of this simulation, so the delay was approximated for the entire line, as most of the leakage is assumed to be at the beginning of the line, where the FEXT is stronger. If the delay was more accurately modelled, it would cause a little dispersion in the differential mode signal.

The term 'delay' used in the discussion above refers to the velocity differences between the propagation of the differential mode signal and the common mode signal, but this delay is in fact frequency dependent, and the propagation velocities are approximations. To model this frequency dependent delay, the phase of the individual frequency responses of the differential mode $H_d(f)$ and common mode $H_c(f)$ models were subtracted: $PhaseDifference(f) = \angle H_c(f) - \angle H_d(f)$.

To make a filter out of this delay, a unity magnitude is given to the *PhaseDifference(f)* and then transformed into an impulse response filter. The frequency response $H_d(f)$ and $H_c(f)$ are displayed in the next figure for a 5,000 foot line containing one 2,000 foot bridged tap.

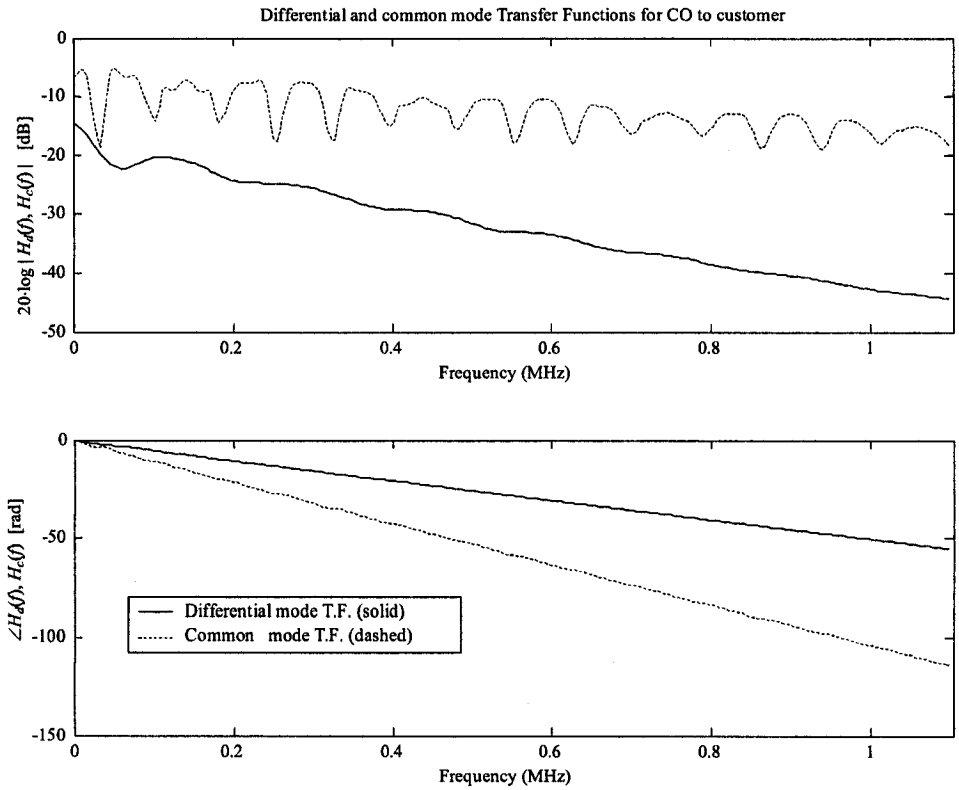


Figure 6-12: Frequency responses of the differential and common modes

Figure 6-12 is a good illustration of the impact of a bridged tap on the magnitude frequency response in the common mode and in the differential mode. By taking the difference in phases represented in Figure 6-12, and applying a unity magnitude and converting to the time domain, the impulse response of the filter representing the frequency dependent delay is obtained and is displayed in Figure 6-13:

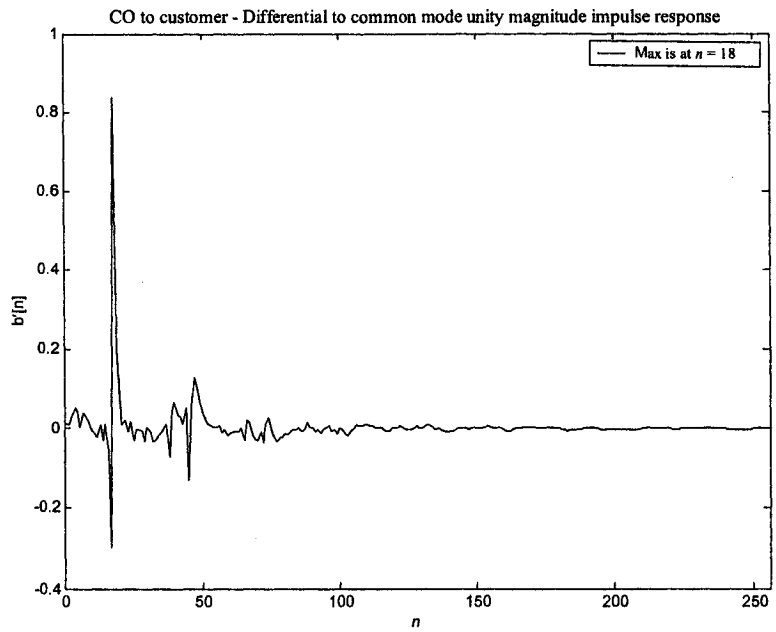


Figure 6-13: Impulse response of "delay" filter

The resulting filter is named the *Phase Transfer Function to Common mode*, as can be seen in Figure 6-14 of the resulting total FEXT time sample generator for the differential and the common modes:

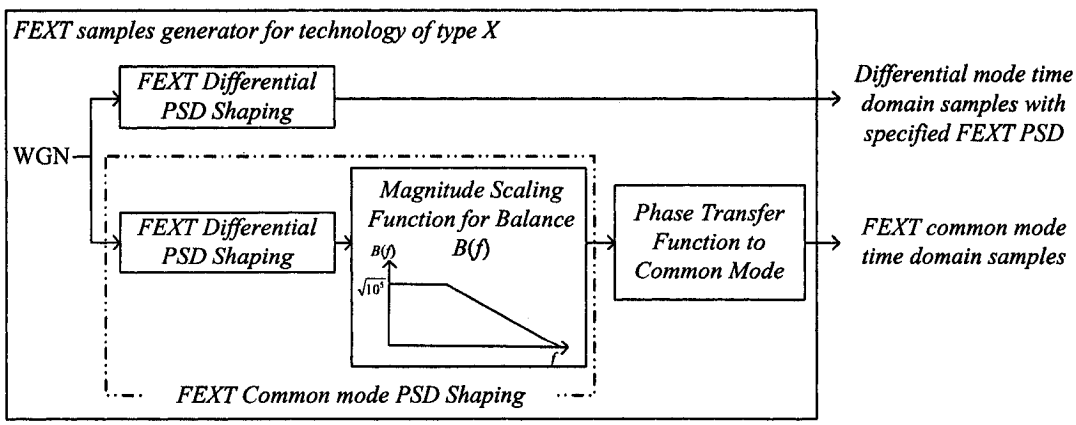


Figure 6-14: FEXT samples generator for any given technology

The output samples generated in Figure 6-14 represent the total FEXT that would have been picked up on the twisted pair transmission line and are therefore added at the receiver in both the common mode and the differential mode.

6.3 Global Channel Model

In this section, the entire channel model will be presented, with the inclusion of the crosstalk canceller. To include the FEXT and the NEXT for only one type of technology (crosstalker) in the bundle, the NEXT and FEXT sample generating block diagrams are included in the following figure:

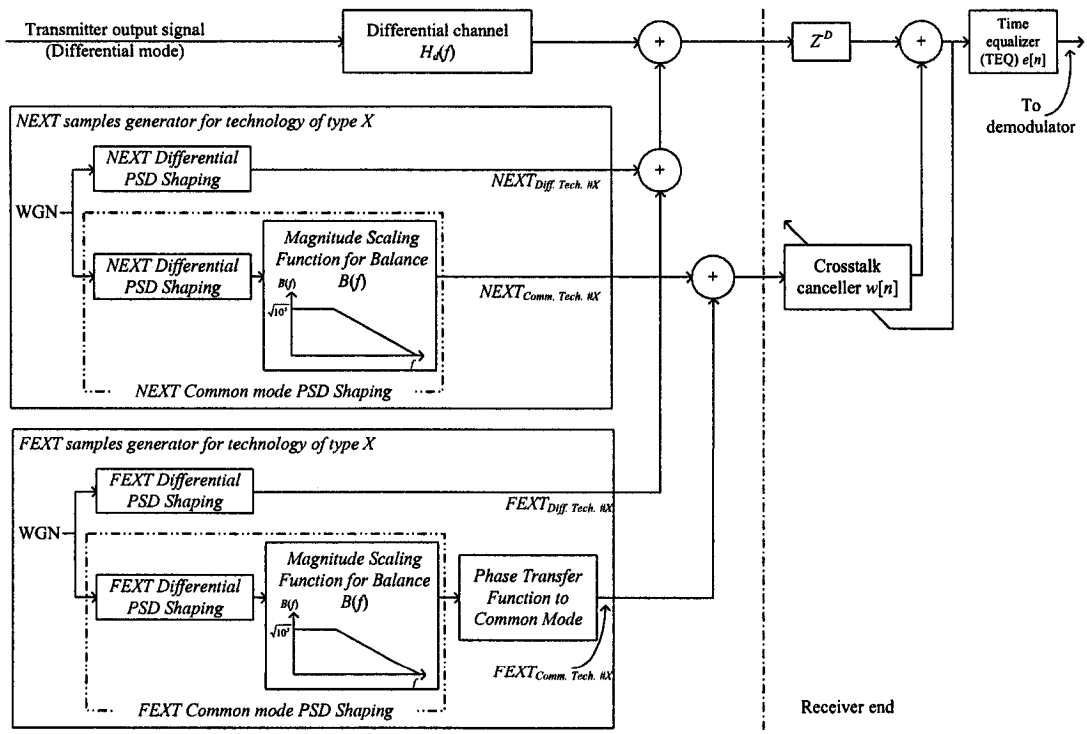


Figure 6-15: Channel model for a single technology (of type *X*) causing crosstalk

The useful signal also leaks into the common mode signal, but is attenuated by the balance function $B(f)$. It can be considered negligible compared to the strong common mode NEXT and FEXT coupling [RAU96], and therefore it is not considered in this model. The differential NEXT and FEXT time domain signals are both summed (in the time domain) with the output of the differential channel. The common mode signal is the sum of the common mode NEXT and FEXT signals and will be used as the reference signal to the *Crosstalk canceller* $w[n]$. The delay Z^D is to force the crosstalk canceller $w[n]$ solution to be causal, it is required because of the compensation of the *Phase Transfer Function to Common mode* filter.

There are many ways to consider crosstalk coming from various sources at the receiver. Great discussions of the available techniques are presented in [GAL02] and [KER02B]. These two articles also present a novel way to combine the effect of multiple types of crosstalk sources to generate the resulting PSD. Most of the methods presented in these articles yield a resulting overall PSD for the crosstalk, but this thesis requires a time domain signal where all the different types of crosstalk have distinct noise sources to minimize correlation. To generate a realistic overall crosstalk signal, each noise source should be independent. In this simulation, each type of crosstalk has its own WGN source sample generator, and all the spectrally shaped sources are then summed together in the time domain.

To include more than one type of technology in the bundle, the block diagram is generalized in the following figure:

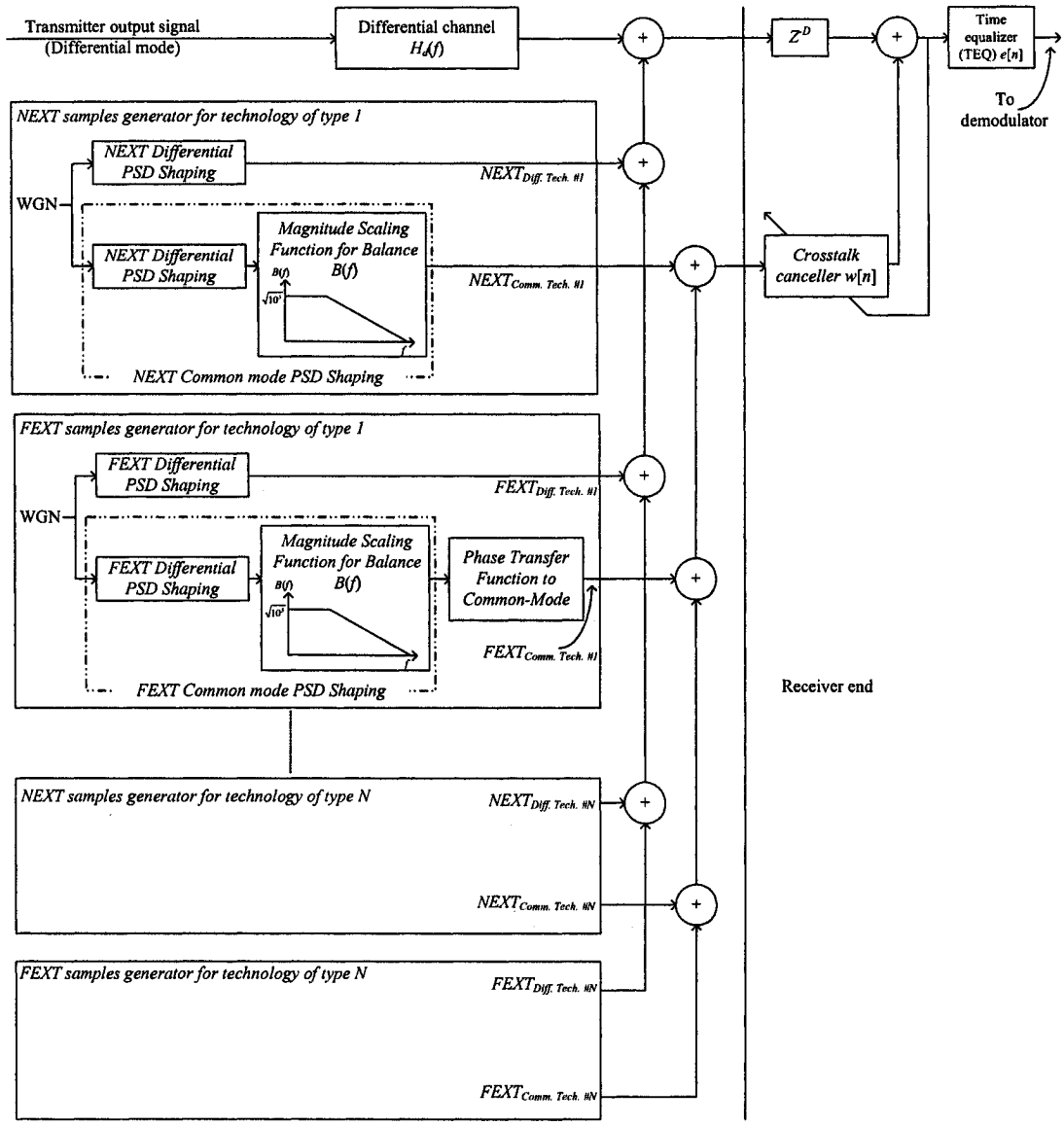


Figure 6-16: Generalized diagram with *N* technologies causing crosstalk

It should be noted that any number of types of technology could be included in this model, as long as their respective transmit PSD spectrums are known.

Chapter 7

Simulation

The first two sections of this chapter discuss the important details of the simulation not covered in the previous chapters, namely the Time Equalizer (TEQ) and the crosstalk canceller. The third section shows simulation results for scenarios where the line is 15,000, 9,000 and 3,000 feet long for the upstream and the downstream cases. The fourth section covers the case where bridged taps are involved, and the last section evaluates the performance of the crosstalk canceller when the common mode reference signal is noisy.

The Matlab™ simulation implements the entire protocol stack presented in Chapter 3, the channel and the crosstalk model (total of 14391 lines of code). It was implemented using many software engineering paradigms, with good modularity, to allow easy extension for future work. The simulation, as it stands now, only considers the signal for one transmission direction at a time. In other words, if the downstream signal is simulated, there is no upstream signal present on the line (and more precisely at the receiver) and vice-versa. This assumes a perfect hybrid circuit that cancels out the transmitted signal in the received signal. Although this is a strong assumption, the implemented ADSL protocol uses Frequency Division Duplexing (FDD), so the overlapping in the frequency domain is minimal. If the hybrid is not perfect or there is echo on the line, an echo canceller can be used to eliminate the remaining transmitted signal from the received signal. The simulation can consider an upstream signal or a downstream signal at any given data rate, and the effect of FEXT and NEXT individually or together.

7.1 Differential Channel Equalizer

The differential channel equalizer (or time domain equalizer, TEQ) of Figure 6-16 was designed to be non adaptive because no adaptation is necessary in the simulation since the channel characteristics do not change over time. It is a modified frequency inverse as shown in the following equation [OLE98]:

$$E(f) = \frac{H_d^*(f)}{|H_d(f)|^2 + \beta} \tag{7-1}$$

where β is a small valued parameter which helps to compensate when the magnitude of the differential channel $H_d(f)$ is near the noise floor. On longer loop lengths, at higher frequencies, the attenuation can be so high ($|H_d(f)|$ is small) that the output signal would be comparable to the noise floor, and therefore inverting $|H_d(f)|$ would be a pointless exercise. At those higher frequencies, the inverse $|E(f)|$ would have a very high gain, which would in turn amplify unwanted noise. The effect of β can be better seen when equation 7-1 is rearranged as follows:

$$E(f) = \frac{1}{|H_d(f)| + \frac{\beta}{|H_d(f)|}} \angle H_d^*(f)$$

In the equation above, if $|H_d(f)|$ is small at a given frequency f_0, and in the same magnitude range as β, then the denominator will be in the range of unity, and consequently $|E(f)|$ would remain approximately at unity at frequency f_0. Therefore, at frequency f_0, no magnitude equalization would be performed. The parameter β of the filter $E(f)$ must be changed for each type of line. In addition to this modified inverse, the magnitude of $E(f)$ at the frequencies not being used by the receiver are sometimes zeroed out in $|E(f)|$ to remove even more unwanted noise.

The example presented here is for a 24 gauge 15,000 feet loop, equalized for downstream reception (and β = 7e-9). The dotted line in Figure 7-1 is the frequency response of the inverse filter. The amplitude of the frequencies below 138 kHz are purposely nulled out, and the high frequencies above approximately 700 kHz are attenuated.

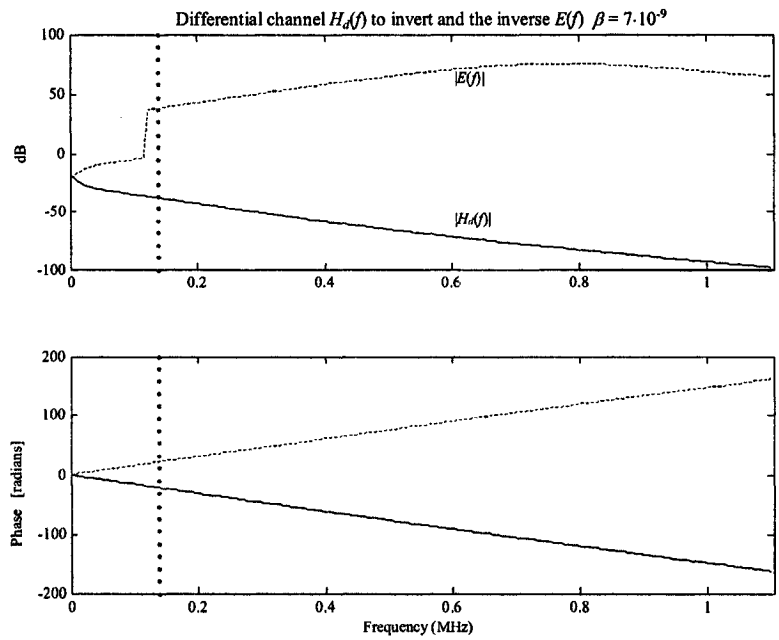


Figure 7-1: Frequency response of differential channel $H_d(f)$ and the inverse $E(f)$

The performance of the TEQ is based on what was defined as the channel SNR. The channel SNR has no reference with the actual signal power and the noise power, but it is a performance measure of the equalizer. Theoretically, the convolution of a channel $h[n]$ and its equalizer $e[n]$ would yield a Dirac function $\delta[n]$, and in the frequency domain, should yield a unity magnitude. Therefore, the performance is based on the frequency response of a Dirac function compared to the frequency response of the convolution of the differential channel $h_d[n]$ and the equalizer $e[n]$. The performance equation used follows (*F.T.* is the Fourier transform).

$$Performance = 10 \cdot \log_{10}\left(\left|F.T.(\delta[n]) - F.T.(h_d[n] * e[n])\right|^2\right)$$

The actual equalizer $e[n]$ is a non causal filter, and therefore a certain delay (*Delay*) was added, so that the actual performance used (named *Channel SNR*) compared to a delayed Dirac impulse:

$Channel\ SNR = 20 \cdot \log_{10}\left[\left|F.T.(\delta[n - Delay]) - F.T.(h_d[n] * e[n])\right|\right]$ and is plotted in Figure 7-2 for the example above.

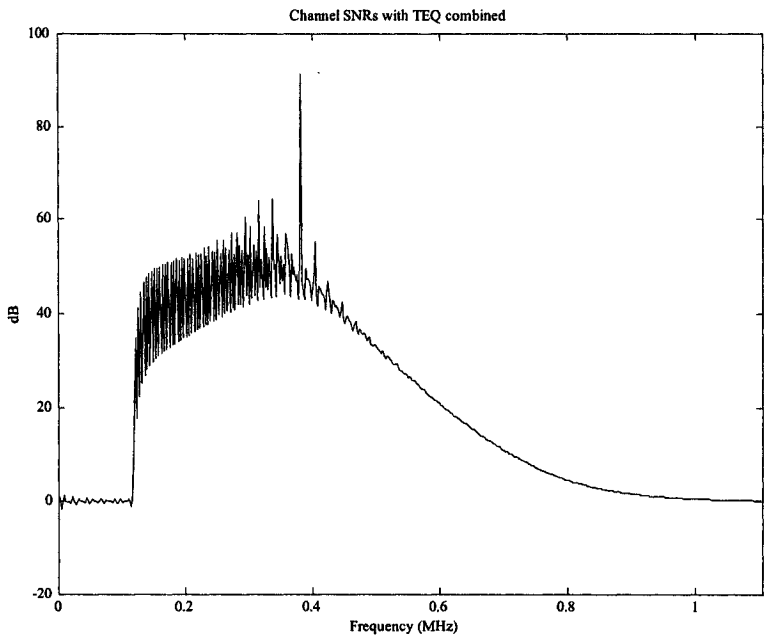


Figure 7-2: TEQ performance measure *Channel SNR*

The minimum desirable performance for the *Channel SNR* is approximately 25 dB for the frequencies that will support the QAM encoded symbols (the used bandwidth). A higher *Channel SNR* is required as the number of bits encoded per symbol increases. The frequency response of the convolution of the differential channel $h_d[n]$ and the equalizer

$e[n]$ is plotted in Figure 7-3. The gain should always be as close to 0 dB as possible at the frequencies used by the useful signal.

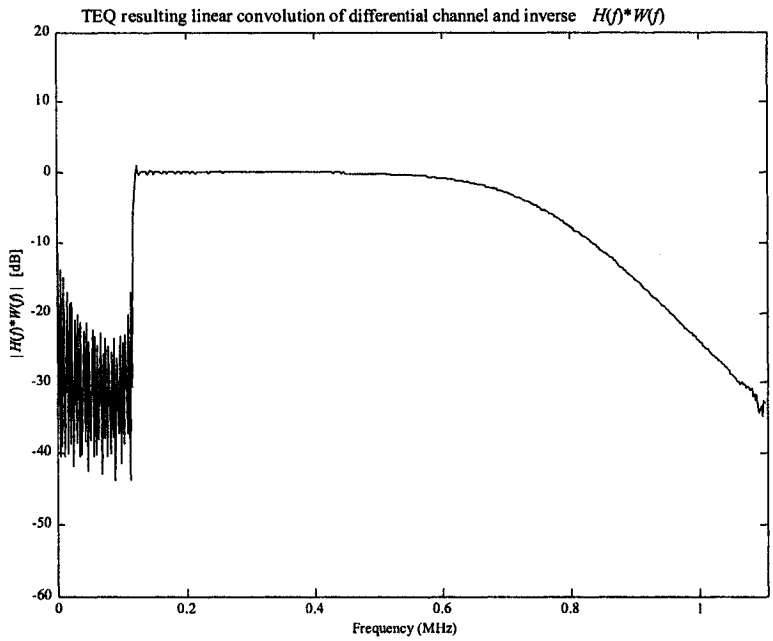


Figure 7-3: TEQ $e[n]$ and $h_d[n]$ linear convolution frequency response

The impulse response of the convolution of the channel $h_d[n]$ and the equalizer $e[n]$ usually has the general appearance of a delayed Dirac function but it is not a must as long as the frequencies being used are equalized properly. For each scenario presented in the following sections, the TEQ was adapted (β was adjusted) to get the maximum *Channel SNR* for the frequencies in use and a low *Channel SNR* for the tones not in use. In all cases, the TEQ has 512 taps, and a *Delay* of 256. The value of β is determined through trial and error by looking at the tones that are needed for data transmission. In other words, the gain of the frequency response of the convolution of $e[n]$ and $h_d[n]$ (represented in Figure 7-3) should yield a gain of 0 dB for the tones in use for data transmission and a low gain for the tones not in use.

7.2 The Crosstalk Canceller

The crosstalk canceller was implemented using the Fast Transversal Filters (FTF) algorithm, which permits a quick convergence to the desired solution. The FTF adaptive filtering algorithm has the same convergence properties as the RLS (Recursive Least squares) algorithm [HAY86]. The difference lies in its computational complexity, which increases linearly with the number of filter taps (O(N)), whereas the RLS algorithm's complexity increases with the square of the number of filter taps (O(N^2)).

In the ADSL protocol, at the beginning of the initialization and synchronization, there is time allotted for the adaptation of the channel equalizer. With the introduction of the crosstalk canceller, two filters would have to be adapted in that time (three if frequency equalization is used, or FEQ). Around four seconds is allotted for the adaptation at the beginning, meaning around 8 million samples. The crosstalk canceller could therefore be adapted during the initialization and its coefficients remain fixed for the duration of the connection. If the crosstalk environment changes to the point where the crosstalk canceller loses its usefulness, a break could occur in the connection to retrain the canceller. The crosstalk environment changes when the other data service modems in the bundle are switched on or off. The other possibility would be to train it continuously, but this technique's problem is that the stability of the crosstalk canceller solution (filter tap coefficients) must be ensured at all times, because instability would increase errors.

To evaluate the performance of the adaptive crosstalk canceller, it was trained until it reached a selected convergence level (e.g. the performance was evaluated at 15, 20, 25 and 30 dB of convergence). This was done to evaluate the required level of convergence and to verify if that level was reasonable. The convergence factor is defined here as the power ratio of the initial FEXT/NEXT levels, over the remaining FEXT/NEXT components not removed by the canceller. Once the training was performed and the crosstalk canceller had reached the selected level of convergence, the simulation was run with new data samples and the performance of the adaptive crosstalk canceller was

measured by comparing the bit error rate, first without and then with the crosstalk canceller. The simulation was performed with an amount of *Prefixed time samples* (defined in Figure 3-2 on page 26) that was proportional to the bit error rate value of that simulation. When the value in the bit error rate was low (few errors), the amount of *Prefixed time samples* used in the simulation was increased. The amount of *Prefixed time samples* that was generated ranged from $1 \cdot 10^5$ to $3 \cdot 10^6$. Looking at Figure 3-2, the *Prefixed time samples* are the actual samples transmitted on the twisted pair and should not be confused with the amount of application data of the upper layers.

Since the ADSL performance gain is better represented at the byte level at higher layers, the usefulness of the crosstalk canceller was measured by comparing how much it was able to reduce the byte error rate after the Reed-Solomon (RS) FEC. Its usefulness was maximum when the RS output (bytes) had a high error rate when the crosstalk canceller was disabled, and then reduced to the level required by the ADSL standard.

There were several ways to achieve the set goal of evaluating the performance of the crosstalk canceller. One way could have been to set a certain number of crosstalkers and let the bit loading determine the maximum bit rate available according to Chapter 4, then evaluate by how much the use of the crosstalk canceller was able to increase the throughput. It was decided not to use this technique because the resulting power level of the crosstalk signal generated by the model presented in Chapter 6 is based on a few assumptions discussed in that chapter. If these levels were inaccurate for any reason, the resulting achievable data rate would not be plausible. Therefore it seemed better to fix the target bit rate and force the bit loading algorithm to place all the required bits without considering the limit. Although this may seem illogical at first, it permits to compare the resulting bit error rates before the crosstalk canceller is used, and after, at several levels of convergence. It was felt that comparing these bit rates would yield a deeper understanding rather than comparing the maximum throughputs from the bit loading. For most scenarios, the forced target bit rates are usually set fairly high, to force some errors, and then try to remove them with the crosstalk canceller, showing its usefulness.

7.3 Scenarios with Different Loop Lengths

The various following scenarios were chosen because they illustrate well the performance of the crosstalk canceller under different conditions. The different line lengths will present the crosstalk canceller with different challenges. Longer line lengths present a very small level of FEXT, shorter lines a higher level. The different lengths also permit the use of different technologies. Some scenarios will consider NEXT and FEXT separately and then together, but all scenarios will have –140 dBm/Hz AWGN on the differential signal at the input of the receiver. The WGN noise power will be around –79.5 dBm for the ADSL bandwidth of 0 to 1.104 MHz. The output power of the transmitter for the downstream case will be approximately from –17 to –20 dBm depending on how many tones are used.

All settings and setups must meet the minimum performance requirement of withstanding 49 self-crosstalkers (FDD-ADSL) with AWGN without the use of the crosstalk canceller before any simulation is performed. Since the amount of data collected during a simulation is relatively large, only the resulting BER will be presented for most cases.

As explained in the previous chapter, the length of the loop will determine the amount of delay that FEXT incurs in the common mode. As a consequence, this requires that the delay Z^{D} of Figure 6-16 will have to be applied to the differential signal to make the solution of the crosstalk canceller $w[n]$ causal. By running the simulations repeatedly for many scenarios, it appeared that the average minimum delay value of D in Z^{D} required to compensate for FEXT was $D \geq 10$ for each 1,000 feet of channel. So a 15,000 foot line would require a delay D of 150 to compensate for FEXT. For some scenarios, the delay D was set much smaller than the observed minimum required to compensate for FEXT to show the effect of this reduction. The number of filter taps for the crosstalk canceller $w[n]$ was chosen equal to $2 \cdot D$ in all the cases presented.

7.3.1 Loop Length of 15,000 feet

The following simulations use a 24 gauge 15,000 foot loop (≈ 4.5 km). This scenario should present a low level of FEXT because of the strong attenuation of a long line, but it cannot be assumed that FEXT will not cause a problem because the useful signal will also be significantly attenuated on longer lines. The target downstream data rate is 3 Mb/s using tones 36 to 126, and the upstream data rate is 960 kb/sec using tones 5 to 31. The channel equalizer was $\beta = 0.7 \cdot 10^{-7}$ for the downstream case and $\beta = 5 \cdot 10^{-7}$ for the upstream case.

7.3.1.1 Downstream with FEXT only

The first case will present a situation where FEXT is the only type of crosstalk. Only one case is presented here. Although extreme and not plausible, it shows that FEXT at this distance is not a problem.

The format of the table containing the results is presented first and will apply to all the following sections. The first row of the table contains the number of filter taps of the crosstalk canceller and the value of D in Z^{D} of Figure 6-16 in Chapter 6 with which the crosstalk canceller was adapted. The left part of the table contains the resulting Bit Error Rate (BER) when the simulation is run without the crosstalk canceller (Off or Reference), then with the various convergence levels of the crosstalk canceller. A '-' in the BER column means that the specified level of convergence could not be reached. The *Byte Errors* column represents the percentage of errors remaining after the Reed-Solomon forward error correction (FEC); a "0 %" indicates that there were no errors remaining in the block of data that was simulated. It should be regarded as meaning that the remaining bit error rate is less than 10^{-7} after the Reed-Solomon FEC. The power levels of the various signals appear in the right side of the table for reference. To compensate for

FEXT, the value of D was fixed at 150 for this scenario to compensate for the propagation delay of a 15,000 foot line.

Table 7-1: Results for the downstream case at 3Mb/s at 15,000 feet with only FEXT from 9 ADSL, 5 ISDN-BRA, 5 ISDN-PRA, 5 SDSL, 5 SHDSL, 5 HDSL, 5T1 and 5 VDSL (not plausible scenario used here)

Number of crosstalk canceller filter taps: 300 Delay D: 150				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$2.0 \cdot 10^{-3}$	0 %	Diff. useful signal	-32.64
15 dB	$1.0 \cdot 10^{-3}$	0 %	NEXT diff.	$-\infty$
20 dB	$1.9 \cdot 10^{-4}$	0 %	NEXT comm.	$-\infty$
25 dB	$8.9 \cdot 10^{-4}$	0 %	FEXT diff.	-69.7
30 dB	-	-	FEXT comm.	-23.6

If the 49 other loops in the bundle are excited with any type of DSL technology (not using repeaters), the results will not be significantly different. This case is presented because it is the most severe (or extreme) case from the point of view of adaptation of the crosstalk canceller and requires a longer filter.

7.3.1.2 Downstream with NEXT only

This following case where only NEXT is considered does not require a delay Z^D of 150 since FEXT compensation is not required. The only technologies able to reach 15,000 feet being ISDN-BRA and ADSL, they are the only ones considered in the table.

Table 7-2: Results for the downstream case at 3Mb/s at 15,000 feet with only NEXT from 12 ADSL, 24 ISDN-BRA

Number of crosstalk canceller filter taps: 150 Delay *D*: 75				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$2.9 \cdot 10^{-3}$	1 %	Diff. useful signal	-33.9
20 dB	$2.7 \cdot 10^{-3}$	0 %	NEXT diff.	- 46.9
25 dB	$2.1 \cdot 10^{-3}$	0 %	NEXT comm.	0.53
30 dB	$2.0 \cdot 10^{-3}$	0 %	FEXT diff.	$-\infty$
35 dB	$1.8 \cdot 10^{-3}$	0 %	FEXT comm.	$-\infty$

The NEXT case is evaluated at the Customer premises (ATU-R). As very few DSL technologies can reach 15,000 feet without using repeaters, so only ISDN-BRA could be used realistically for this case. Supposing that ISDN-PRA could reach this distance without the use of repeaters, the yielded results would be:

Table 7-3: Results for the downstream case at 3Mb/s at 15,000 feet with only NEXT from 12 ADSL, 12 ISDN-BRA and 12 ISDN-PRA

Number of crosstalk canceller filter taps: 150 Delay *D*: 75				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$3.1 \cdot 10^{-1}$	99 %	Diff. useful signal	-31.9
20 dB	$6.1 \cdot 10^{-2}$	55 %	NEXT diff.	- 36.4
25 dB	$7.1 \cdot 10^{-3}$	4 %	NEXT comm.	1.85
30 dB	$1.5 \cdot 10^{-3}$	0 %	FEXT diff.	$-\infty$
35 dB	$7.9 \cdot 10^{-4}$	0 %	FEXT comm.	$-\infty$

This gives a clear indication that even without considering the extra crosstalk induced by the repeaters, the interference due to this technology causes severe problems. It is true of many other technologies whose bandwidth has a substantial overlap with ADSL's.

7.3.1.3 Downstream with FEXT and NEXT

This case presents a maximally loaded bundle with ISDN-BRA with delay *D* set to 150 to try to compensate for the FEXT at 15,000 feet, as well as NEXT. Only ISDN-BRA was used since it is the only DSL technology able to reach 15,000 feet without repeaters.

Table 7-4: Results for the downstream at 3Mb/s at 15,000 feet with FEXT and NEXT from 49 ISDN-BRA

Number of crosstalk canceller filter taps: 300 Delay *D*: 150				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$7.7 \cdot 10^{-3}$	5 %	Diff. useful signal	-32.9
15 dB	$3.8 \cdot 10^{-3}$	0 %	NEXT diff.	- 50
20 dB	$3.4 \cdot 10^{-3}$	0 %	NEXT comm.	-3.2
25 dB	-	-	FEXT diff.	- 75
30 dB	-	-	FEXT comm.	- 28.3

When both FEXT and NEXT are present, the crosstalk canceller adapts to the more powerful of the two (NEXT in this and most cases), because of the use of a single common mode reference signal for all crosstalkers. By a more detailed analysis of the results of the simulation and all its graphs (not shown here), the crosstalk canceller adapted to the NEXT only, which means that the number of filter taps and delay D could have been drastically reduced.

7.3.1.4 Upstream with FEXT only

This upstream case with only FEXT is presented here to be able to compare to the downstream case of Table 7-1 where only FEXT was considered. Although the DSL technologies present on the bundle are different, the results would not have been significantly different if all the technologies in the downstream case of Table 7-1 were present in the bundle. The technologies are different because those deployed in Table 7-1 are from the CO, and do not necessarily reach all the way to 15,000 feet (the customers could be much closer to the CO, but the FEXT would propagate in the common mode). In the following case, where the FEXT comes from customers at 15,000 feet, all the technologies of Table 7-1 could not have reached the customers at that distance.

Table 7-5: Results for the upstream case at 960 kb/s at 15,000 feet with only FEXT from 24 ADSL and 24 ISDN-BRA

Number of crosstalk canceller filter taps: 300 Delay *D*: 150				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$< 10^{-5}$	0 %	Diff. useful signal	-21
15 dB	$< 10^{-5}$	0 %	NEXT diff.	$-\infty$
20 dB	$< 10^{-5}$	0 %	NEXT comm.	$-\infty$
25 dB	$< 10^{-5}$	0 %	FEXT diff.	-73
30 dB	$< 10^{-5}$	0 %	FEXT comm.	-26

These results indicate that the upstream case is even more error free than the downstream case when only FEXT is considered. This is largely due to the lower attenuation of the upstream frequency band in comparison to the stornger attenuation of the downstream frequency band.

7.3.1.5 Upstream with NEXT only

This scenario is similar to the one of Table 7-1, but the data direction is not the same. It is presented here to vary the technology types (the interfering PSDs) in order to complicate the task of the crosstalk canceller and evaluate its performance. Since the adaptation to FEXT is not required, the value of D was reduced to 75, and could have been reduced more.

Table 7-6: Results for the upstream case at 960 kb/s at 15,000 feet with only NEXT from 9 ADSL, 5 ISDN-BRA, 5 ISDN-PRA, 5 SDSL, 5 SHDSL, 5 HDSL, 5 T1 and 5 VDSL

Number of crosstalk canceller filter taps: 150 Delay D: 75				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$8.9 \cdot 10^{-2}$	68 %	Diff. useful signal	-19.5
15 dB	$4.7 \cdot 10^{-3}$	1.2 %	NEXT diff.	-28
20 dB	$1.8 \cdot 10^{-3}$	0 %	NEXT comm.	10.4
25 dB	$4.9 \cdot 10^{-4}$	0 %	FEXT diff.	$-\infty$
30 dB	-	-	FEXT comm.	$-\infty$

In this case, the crosstalk canceller makes a great improvement. Obviously, it exceeds what services might be expected to be in a bundle.

7.3.1.6 Upstream with FEXT and NEXT

This case is presented here for NEXT and FEXT where the value of D in Z^D was reduced to 75 (should be 150 to compensate for FEXT). This case was done to show that since NEXT is the stronger of the two interferers in this situation, the crosstalk canceller still works since it would not have adapted to FEXT. SHDSL was added as a technology, it is a slight stretch of what could be actually deployed without the use of repeaters, but it would be possible to have this technology create NEXT for a CO located receiver.

Table 7-7: Results for the upstream case at 960 kb/s at 15,000 feet with FEXT and NEXT from 12 ADSL, 12 ISDN-BRA and 12 SHDSL

Number of crosstalk canceller filter taps: 150 Delay D: 75				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$3.4 \cdot 10^{-2}$	29 %	Diff. useful signal	-19.2
15 dB	$6.4 \cdot 10^{-3}$	2 %	NEXT diff.	-29.7
20 dB	$1.2 \cdot 10^{-3}$	0 %	NEXT comm.	7.9
25 dB	$2.9 \cdot 10^{-4}$	0 %	FEXT diff.	-74
30 dB	$5.2 \cdot 10^{-4}$	0 %	FEXT comm.	-27.2

Reducing the value of the delay Z^D from 150 required to compensate for FEXT to 75 did not impair the ability of the crosstalk canceller to adapt. It is therefore interesting to note that the number of filter taps can be reduced to save processing power in this case.

7.3.2 Loop Length of 9,000 feet

The following simulations are done with a 26 gauge 9,000 foot loop (≈ 2.75 km). The level of FEXT in this scenario will be slightly larger than in the 15,000 foot case. The cases with FEXT only are not presented to avoid redundancy with the 15,000 feet scenario, as well as the NEXT only case for the upstream, which yields similar results. The downstream cases were done at 6 Mb/sec and the equalizer was set at β = 0.7e-7. The upstream case was at 960 kb/s and the equalizer was set at β = 3.0e-6. For a 9,000 foot line, the value of D should be approximately 100 to be able to compensate for FEXT.

7.3.2.1 Downstream with NEXT only

This case with NEXT only differs from the 15,000 foot NEXT only case in Table 7-2 with the distance of the line from the transmitter, meaning that the level of the useful received signal will be greater than for the 15,000 foot line. With this shorter 9,000 foot line, other technologies can be deployed without the use of repeaters, so SDSL is used in this case because it is very harmful to ADSL.

Table 7-8: Results for the downstream case at 6Mb/s at 9,000 feet with only NEXT from 49 SDSL

Number of crosstalk canceller filter taps: 200 Delay D: 100				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$2.4 \cdot 10^{-1}$	82 %	Diff. useful signal	-18
15 dB	$1.9 \cdot 10^{-2}$	19 %	NEXT diff.	- 35.7
20 dB	$1.0 \cdot 10^{-3}$	0 %	NEXT comm.	7.0
25 dB	$5.6 \cdot 10^{-4}$	0 %	FEXT diff.	- ∞
30 dB	$5.0 \cdot 10^{-5}$	0 %	FEXT comm.	- ∞

SDSL is the strongest interferer of the DSL technologies tested in the simulation for the downstream NEXT case at 9,000 feet. SHDSL, HDSL2 and HDSL do not create a worse problem than SDSL, so they are not presented here, except when they are mixed.

7.3.2.2 Downstream with FEXT and NEXT

As with the results in Table 7-8 for NEXT only, the worst interferer was SDSL, and it was therefore included. HDSL and ADSL were included to vary the types of technologies in the bundle, in an attempt to create an adaptation problem for the crosstalk canceller.

Table 7-9: Results for the downstream at 6Mb/s at 9,000 feet with FEXT and NEXT from 24 SDSL, 24 HDSL and 1 ADSL

Number of crosstalk canceller filter taps: 200 Delay *D*: 100				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$2.2 \cdot 10^{-1}$	79 %	Diff. useful signal	-18.4
15 dB	$1.2 \cdot 10^{-2}$	11	NEXT diff.	-36.55
20 dB	$2.8 \cdot 10^{-5}$	0	NEXT comm.	7.2
25 dB	-	-	FEXT diff.	-67.5
30 dB	-	-	FEXT comm.	-21.6

The crosstalk canceller was very effective in this case. This was partially due to the SDSL, which is very destructive, but is easy to remove from the ADSL signal because of the wide frequency overlap with the ADSL spectrum. The level of NEXT in a 9,000 foot bundle is equal to the level in a 15,000 foot bundle, but the case with 9,000 feet presents the additional challenge that the ADSL signal can share a bundle with more technologies that can reach 9,000 feet.

7.3.2.3 Upstream with FEXT and NEXT

This upstream case including FEXT and NEXT was kept identical to the same case with the 15,000 foot line in Table 7-7, including the same technologies, data rate and Delay *D*. The intent was to permit a comparison of the results.

Table 7-10: Results for the upstream at 960 kb/s at 9,000 feet with FEXT and NEXT from 12 ADSL, 12 SHDSL and 12 ISDN-BRA

Number of crosstalk canceller filter taps: 150 Delay *D*: 75				
Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
Off (Reference)	$1.5 \cdot 10^{-2}$	16 %	Diff. useful signal	23.4
15 dB	$2.1 \cdot 10^{-3}$	0	NEXT diff.	-29.1
20 dB	$3.2 \cdot 10^{-4}$	0	NEXT comm.	8.22
25 dB	$3.3 \cdot 10^{-5}$	0	FEXT diff.	-69.62
30 dB	$< 10^{-5}$	0	FEXT comm.	-23.0

For the cases with 9,000 and 15,000 feet lines with NEXT and FEXT in Table 7-10 and Table 7-7 respectively, the crosstalk canceller adapts to roughly the same maximum convergence level. It can be seen that the BER is reduced more in the 9,000 foot case because the useful signal strength is greater. The level of NEXT is the same in both cases, and the level of FEXT, although slightly stronger in the 9,000 foot line than in the 15,000 foot line, does not affect the results because of its smaller power.

7.3.3 Loop Length of 3,000 feet

A 3,000 foot loop presents the interesting situation where the level of FEXT becomes more important. The increased level of FEXT does not necessarily affect the useful differential signal because it is also less attenuated and therefore is stronger. The crosstalk canceller has more difficulty to adapt.

Table 7-11: Results for the downstream at 6 Mb/s at 3,000 feet with FEXT and NEXT from 12 VDSL, 12 HDSL, 12 SHDSL and 12 ISDN-PRA

Number of crosstalk canceller filter taps: 150 Delay *D*: 75

Convergence (in dB)	Bit Error Rate (BER)	Byte Errors (after FEC)	Power at receiver (dBm)	
			Diff. useful signal	2.1
Off (Reference)	$3.7 \cdot 10^{-2}$	0.7 %	NEXT diff.	-34.7
14 dB	$1.9 \cdot 10^{-3}$	0	NEXT comm.	7
-	-	0	FEXT diff.	-49
-	-	0	FEXT comm.	-6.5

It is possible to realistically include VDSL as a technology in this test, but VDSL is not expected to cause any damage to the ADSL signal since its frequency spectrum was carefully planned to be identical to ADSL below 1.104 MHz, to specifically avoid problems on same length lines. The crosstalk canceller adapted to only 14 dB in this case, which roughly corresponds to the difference between the FEXT and NEXT common mode references (-49 - -34.7 = -15.3). This is only a heuristic approach at judging the maximum convergence of the crosstalk canceller but is valid in many cases.

The upstream was not shown here because it does not provide any more information than the downstream case. In most cases, when considering NEXT and FEXT together at 3,000 feet or less, the crosstalk canceller cannot converge to a level of more than 10-15 dB. This low level of convergence prevents the crosstalk canceller from removing crosstalk from the useful signal, but never aggravates the error. Bit it is comforting to know that, at this short distance, the ADSL signal remains strong enough to sustain high levels of crosstalk without too much degradation.

7.4 Effect of Bridged Taps

Bridged taps are an impairment that most DSL technologies must be able to deal with, even with a bundle containing no bridged taps. A customer's home wiring usually contains many unterminated phone jacks, which are bridged taps. Bridged taps should not affect the crosstalk canceller's performance, but the quality of the differential signal will be degraded by attenuation and reflections, therefore the bit error rate might increase. Compensating for the effect of bridged taps is the purpose of the channel equalizer (TEQ), but it is worthwhile to get a glimpse of this effect. This scenario has one 500 foot bridged tap at 5,000 feet from the CO downstream, followed by another 4,000 feet of line (all 26 gauge). This was compared in Table 7-12 to the downstream case for a case with FEXT and NEXT as in Table 7-9.

Table 7-12: Results for the downstream at 6Mb/s at 9,000 feet with FEXT and NEXT from 24 SDSL, 24 HDSL and 1 ADSL

Number of crosstalk canceller filter taps: 200 Delay *D*: 100				
Convergence (in dB)	Without bridged tap		With bridged tap	
	Bit Error Rate (BER)	Byte Errors (after FEC)	Bit Error Rate (BER)	Byte Errors (after FEC)
Off (Reference)	$2.2 \cdot 10^{-3}$	79 %	$2.8 \cdot 10^{-3}$	90 %
20 dB	$2.8 \cdot 10^{-5}$	0	$1.1 \cdot 10^{-4}$	0

It can be seen that the bridged tap slightly increases the BER before FEC, however the crosstalk canceller was still able to bring down the BER to a level where the FEC could compensate for the remaining errors.

7.5 Noisy Common Mode Reference

This section evaluates the crosstalk canceller's performance when the common mode reference signal is noisy. This test is performed because two assumptions that were made in this document had to be verified. The first is that the level of the differential signal that leaks to the common mode is considered negligible [RAU96], and the second (implied but never explicitly stated) is that the common mode signal noise is also negligible. If these assumptions are not verified, more noise will appear in the common mode signal, which may not be negligible, and it would be important to study the effect of noise on the reference for crosstalk cancelling.

Using Table 7-9 as a basis scenario, an independent source of WGN was added to the common mode reference signal to see the effect on the convergence level of this scenario.

Table 7-13: Results for the downstream at 6Mb/s at 9,000 feet with FEXT and NEXT from 24 SDSL, 24 HDSL and 1 ADSL

Number of crosstalk canceller filter taps: 200 Delay *D*: 100			
Level of common mode AWGN (dBm/Hz) (dBm)	Maximum attainable convergence (in dB)	Power at receiver (dBm)	
None (Reference)	22.6	NEXT diff.	-36.5
-140 (-79.5)	22.6	NEXT comm.	+7.2
-85 (-24.5)	21.8	FEXT diff.	-67.5
-60 (+0.3)	8.6	FEXT comm.	-21.6
-50 (+10)	2.5	Total comm. power	+7.3

An important amount of WGN (-60 dBm/Hz) is necessary to prevent the crosstalk canceller from converging. A reasonable value of common mode noise was used in [RAU96] at a level of –140 dBm/Hz. At this level of WGN, the effect is negligible. Therefore, the assumptions about the common mode noise are not unrealistic.

Chapter 8

Conclusion

8.1 Primary Conclusion

This thesis has presented a crosstalk canceller that uses the common mode signal as its reference to reduce the level of crosstalk in the received differential signal with simulation for an ADSL link. Since the literature is lacking in the area of common mode crosstalk in ADSL, a novel model of the common mode crosstalk signal had to be developed for this thesis. Simulations based on this model show that the technique is beneficial to the differential signal quality in many scenarios since the error rate is lowered (both before and after the FEC). Although this technique was simulated using the ADSL protocol, it could be applied to other DSL technologies.

8.2 Secondary Conclusions

When the crosstalk canceller is unable to converge to a good solution, it does not negatively affect the received signal, even when it has only a few decibels of convergence. Since the received signal will not be degraded by the use of the crosstalk canceller, it is assumed that it can be left in the signal path at all times.

When NEXT is the stronger interferer, the crosstalk canceller only requires a few filter taps to adapt to most NEXT interference and a very small delay D in Z^{-D}. This is the case in longer lines, where the level of FEXT is much smaller than the level of NEXT.

The adaptation of the crosstalk canceller is better when the frequency content of the common mode reference contains the entire bandwidth of the useful ADSL signal.

When FEXT is the stronger interferer, the crosstalk canceller requires more filter taps and delays. The required delay D is approximately linearly proportional to the length of the line when using the crosstalk model presented in Chapter 6. It requires a delay D of approximately 50 samples for each 5,000 feet of line.

The adaptive crosstalk canceller can usually converge in cases where it is most useful, as in the presence of strong NEXT in the same frequency band as the useful signal.

Bridged taps will affect the useful differential mode signal by attenuating it and cause reflections and phase distortion, most notably at certain frequencies (the frequencies depend on the bridged tap lengths). The crosstalk canceller cannot do much to compensate for these channel impairments, but if FEXT is the stronger interferer, the model considers these phase distortions for the crosstalk, and therefore the crosstalk canceller will adapt to the "distorted" FEXT. The model for NEXT does not depend on the channel characteristics, which is why the bridged taps are not considered for NEXT interference.

When WGN is present on the reference signal, heuristically, the crosstalk canceller's maximum convergence level will be limited to approximately the difference (in dB) between the common mode reference signal power and the noise power (this is only an approximation that seems to work for most cases).

8.3 Future Work

It would be important for any further investigation of the common mode crosstalk cancellation technique to validate the main assumptions on the common mode signal characteristics by taking the required common mode signal measurements on actual loops. This was part of the initial project, until the local telephone company withdrew its measuring equipment from the field due to a change in management midway through the project. The original setup for measurements enabled a dial-in access to the measuring equipment. The unit could measure the signal on each individual wire of an unused loop for a fraction of a second and report the results remotely. It could calculate the position and the approximate length of bridged taps on the line and make loop length estimations.

An interesting study would be the effect of repeaters on the level of induced crosstalk. It would be very valuable to measure the effect of one repeater and of multiple repeaters. Depending on the technology using repeaters, the level of FEXT would probably increase dramatically. Current static spectrum management deals with this problem by avoiding possible problematic situations. Crosstalk cancelling using the common mode signal might deal with this quite well, solving certain issues of quantity of data services that a bundle can support.

The delay in the common mode propagation is modelled as the entire length of the line but, in reality, the delay depends on the point at which the particular FEXT component was coupled, therefore a slightly more diffuse signal would be present at the receiver. A more accurate model for the FEXT delay could be devised.

Appendix A:

Adapted Loading Algorithm

Repeat while Solution Not Found {
Find Indexes i in $C_{Ass}[i]$ *where the assignment is* $C_{Ass}[i] = 1$
If No '1's are found in $C_{Ass}[i]$ *{*
Solution Found
}
Else {
Remove all '1's in $C_{Ass}[i]$ *(The sum of these '1's is the new Number of bits to place)*
Find the tone number t where a '1' was assigned that has the highest $P_{Threshold}$
Set the maximum Bit Capacity for that tone to 0 (Max Bit Capacity[t] = 0)
Tone t cannot be used anymore, so make sure there is enough room left to place bits:
If $\sum_{m=0}^{255} Max\ Bit\ Capacity[m] - \sum_{k=0}^{255} C_{Ass}[k] \leq Number\ Of\ Bits\ To\ Place$ *{*
Do Bit Loading Algorithm again starting with the new $C_{Ass}[n]$ *with '1's removed but keeping the other bits already assigned as is.*
}
Else {
$Number\ Of\ Bits\ To\ Place = \sum_{m=0}^{255} Max\ Bit\ Capacity[m] - \sum_{k=0}^{255} C_{Ass}[k]$
If Number Of Bits To Place = 0 {
Solution Found (Or at least the best that can be done)
}
Else {
Do bit loading Algorithm again starting with the new $C_{Ass}[n]$ *with '1's removed but keeping the other bits already assigned as is.*
}
}
}
}

Bibliography

[BIN00] John A. C. Bingham, *ADSL, VDSL, and Multicarrier Modulation,* Wiley Series in Telecommunications and Signal Processing, 2000.

[BRO03] Jim Brown and Bill Whitlock, "Common-Mode to Differential-Mode Conversion in Shielded Twisted pair Cables (Shield-Current-Induced Noise)", *Audio Engineering Society*, Convention Paper 5747, March 2003.

[CIO91] John M. Cioffi, "A Multicarrier Primer", ANSI Contribution TIE1.4/91-157, November 1991

[COM98] Richard Combellack, "Improving Range and Bandwidth of Telco Loop Plant", Proceedings of the *International Wire and Cable Symposium*, 1998.

[COO93] J. W. Cook, "Wideband Impulsive Noise Survey of the Access Network", *BT Technology Journal*, Vol. 11, No. 3, July 1993.

[FEN99] D. K. Fenton, "Digital Noise Cancellation for xDSL", Master's Thesis, Ottawa University, School of Information Technology and Engineering, August 1999.

[FER97] Dennis Ferguson, Ravi Cherukuri, "Self-Synchronous Scramblers for PPP Over Sonet/SDH: Some Analysis", *PPP Extensions Working Group*, November 1997.

[G9921] "ITU-T Recommendation G.992.1", ITU-T (ADSL metallic interface standard), June 1999.

[GAL02] Stefano Galli, Kenneth J. Kerpez, "Methods of Summing Crosstalk from Mixed Sources – Part I: Theoretical Analysis", *IEEE Transactions on Communications*, Vol. 50, No. 3, March 2002.

[HAY86] Simon Haykin, *Adaptive Filter Theory*, First Edition, Prentice-Hall, 1986.

[IEC03] "Spectral Compatibility of Digital Subscriber Line (DSL) Systems", *International Engineering Consortium.*

[JOH97] David A. Johns and Daniel Essig, "Integrated Circuits for Data Transmission over Twisted pair Channels", *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 3, March 1997.

[KER02] Kenneth J. Kerpez, "DSL Spectrum Management Standard", *IEEE Communications Magazine*, November 2002.

[KER02B] Kenneth J. Kerpez, Stefano Galli, "Methods of Summing Crosstalk from Mixed Sources – Part II: Performance Results", *IEEE Transactions on Communications*, Vol. 50, No. 4, April 2002.

[KER93] K. J. Kerpez, "Near End Crosstalk is Almost Gaussian", *IEEE Transactions in Communications*, Vol. 41, May 1993, pp. 670–672.

[KOU99] S. Kourtis, "Optimum bit allocation algorithm fro DMT-based systems under minimum transmitted power constraints", *Electronics Letters*, December 9th 1999, Vol. 35 No. 25.

[KUN95] Ling-Pei Kung, "Reed-Solomon Coding and Decoding", MIT Media Lab, 1995.

[LAO02] Richard LAO, "The Twisted pair Telephone Transmission Line", *High Frequency Electronics*, November 2002.

[LEF00] Pierre D. Lefebvre, "Adaptive Multiple Sub-band Common Mode RFI Suppression", Master's Thesis, Ottawa University, School of Information Technology and Engineering, April 2000.

[MEL95] Melbourne Barton, Michael L. Honig, "Optimization of Discrete Multitone to Maintain Spectrum Compatibility with Other Transmission Systems on Twisted Copper Pairs", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 9, December 1995.

[OLE98] Ole Kirkeby, Philip A. Nelson, Hareo Hamada and Felipe Orduna-Bustamante, "Fast Deconvolution of Multichannel Systems Using Regularization", *IEEE Transactions on Speech and Audio Processing*, Vol. 6, No. 2, March 1998.

[RAU96] Dennis J. Rauschmayer, "Crosstalk Reduction Using Common Mode Signal Correlation", T1E1.4/96-069, April 22, 1996.

[SHE03] Jim Sherwin, "Understanding common-mode signals", *Electronic Design News Magazine (EDN)*, April 17, 2003.

[SM2001] "Spectrum Management for Loop Transmission Systems", ANSI Standard T1.417-2001, January 2001.

[SON02] Kee Bong Song and Seong Taek Chung, George Ginis, John M. Cioffi, "Dynamic Spectrum Management for Next-Generation DSL Systems", *IEEE Communications Magazine*, October 2002.

[STA02] Thomas Starr, Massimo Sorbara, John M. Cioffi and Peter J. Silverman, *DSL Advances*, Prentice Hall, December 26, 2002.

[STA02 P62] Quote taken from: Thomas Starr, Massimo Sorbara, John M. Cioffi, and Peter J. Silverman, *DSL Advances*, Prentice Hall, December 2002, p.62.

[STA98] Thomas Starr, John M. Cioffi and Peter J Silverman, *Understanding Digital Subscriber Line Technology*, Prentice Hall, December 1998.

[VDM03] "Very-high-bit-rate Digital Subscriber Line (VDSL) Metallic Interface", T1E1.4/2003- 210R2, Montreal, Canada, August 22-26 2003.

[VOD02] Jiří Vodrážka, "Spectral characteristics of digital subscriber line systems", *Electronic Letters*, November 26th 2002.

[VQA03] "Very-high-bit-rate Digital Subscriber Line (VDSL) Metallic Interface", T1E1.4/2003- 215R2, Montreal, Canada, August 18-22 2003.

[WAN02] Jing Wang, "ADSL Coding Schemes and Discrete Multitone Modulation for Virtual Peripheral Engine", Doctoral Thesis, University of Tsukuba, February 2002.

[YEA03] T.H. Yeap, D.K. Fenton and P.D. Lefebvre, "A novel common mode noise cancellation technique for VDSL applications", *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, pp. 1325 -1334, August 2003.

[YEA60] T. H. Yeap and P. Lefebvre, "Adaptive Multiple-Subband Common-mode RFI Suppression", U.S. Patent Number 6052420.

[ZEN01] Chaohuang Zeng, Carlos Aldana, Atul A. Salvekar, and John M. Cioffi, "Crosstalk Identification in xDSL Systems", *IEEE Journal On Selected Areas In Communications*, Vol. 19, No. 8, August 2001.

[ZEN02] Chaohuang Zeng and John M. Cioffi, "Near-End Crosstalk Mitigation in ADSL Systems", *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 5, June 2002.