Performance Enhancement of xDSL Systems Using LDPC Coding with DMT Modulation

by

THOMAS S H ZHENG

B.Sc., Peking University, 1986

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical & Computer Engineering)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

November 2003

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical & Computer Engineering

The University of British Columbia
Vancouver, Canada

Date November 26, 2003

Abstract

Channel coding is playing an pivotal role for improving the performance of xDSL systems in terms of higher data rate and long loop reach. The current channel coding scheme for standard DSL concatenates *Reed-Solomon* (R-S) codes and multi-dimensional *Trellis coded modulation* (TCM) codes. Although RS-TCM coding provides a significant coding gain for DSL systems, this gain still far away from the channel capacity.

Recently rediscovered low-density parity-check (LDPC) codes, which were invented by Gallager in 1962, have become the best-known practically implentmentable codes for closely approaching the channel capacity. The probabilistic decoding for LDPC codes is an effective and applicable technique based on the message-passing algorithm which employs probabilistic message-passing on a normal graph generated from the parity-check matrix of LDPC codes. Our empirical results showed that for flat AWGN channels and various QAM modulation schemes more than 2.0 dB performance improvement of DSL systems was obtained with LDPC coding instead of the RS-TCM scheme.

This thesis emphasizes the study of the performance of LDPC codes applied in xDSL systems operating on frequency-selective multilevel AWGN channels. Two different frequency-selective channel models were used in our simulation studies, where *bit-error rate* (BER) was measured (via computer simulation) vs. received *signal-to-noise ratio* (SNR). LDPC codes were employed together with multilevel QAM in DMT-xDSL systems. Coding gains greater than 2.8 dB were obtained at 10^{-6} BER when compared with RS-TCM coding.

Table of Contents

List of Tables

List of Figures

Dedication

To my wife Alice, my daughter Isabel, and my son Timothy

Acknowledgments

I would like to express my sincerest gratitude to my supervisor Professor Robert W. Donaldson, for his guidance, encouragement, and support throughout this thesis. I am extremely grateful to him for his assistance without exception and for his extensive suggestions for revision of this thesis. Thank you Dr. Donaldson for being such a nice mentor.

I wish to thank Professor Robert Schober for his insightful ideas and his precious time spent with me discussing some topics related my research. His help motivated and inspired me in many ways.

I would also like to thank the members of our communications group for their help and friendship, including Wujun Cao, Jian Chen, Zhibing Chen, Victor Fong, Cyril Iskander, Ying Luo, Li Ma, Xinrong Wang, Zhipeng Wang, George Wai Wong, Lijuan Wu, Kaiduan Xie, Zhanping Yin, Fei Yu, Chu Zhang, Jie Zhao, and Juan Zi.

Finally, I would like to express special thanks to my wife Alice for her consistent support during my studies at UBC. Her deep love and encouragement made this thesis possible. I am especially grateful to her for taking such a good care of our children.

Chapter 1 Introduction

In the beginning, the telephone access network was originally constructed at a large scale for carrying 4kHz voiceband telephony. This *plain old telephone system* (POTS) consisted of single twisted-pair copper wires which were reliable and could be laid out to a large and widespread population at relatively low cost. Later, as data transmission needs grew and electronic hardware improved, voiceband modems became available. POTS wires could then be used to carry other types of point-to-point communications such as telegraph, fax, and computer data. More recently, as computer networks and internet traffic grew rapidly the limited capacity of POTS links became problematic. It was believed that POTS wires had reached their limits. In the mid 1980s the *digital subscribe line* (DSL) modems were invented, thereby enabling spectrum usage on the twisted-pair copper wires to increase to tens of megahertz.

Some of the noteworthy DSL developments are as follows:

- First, the *integrated services digital network* (ISDN) was introduced as a service in 1986. The *basic rate interface* (BRI) of ISDN provides a total of 144 kb/s of symmetric digital information (including two 64 kb/s B channels and one 16 kb/s D channel) over loops of length up to about 5.5 km.
- In 1992, the first *high bit-rate digital subscribe line* (HDSL) went into service offering 1.544 or 2.048 Mb/s over loops of up to about 3.7 km using 0.5 mm (24 AWG) twisted pair without an intermediate repeater.
- The first prototype modems for *asymmetric digital subscriber line* (ADSL) appeared in 1992 following developments at Stanford University and AT&T Bell Labs. ADSL offers bit rates of up to about 6 Mb/s downstream (towards customer) and up to about 640Kb/s upstream (towards central office) by using the spectrum up to 1.1 MHz. The reach of ADSL is about 4 km and the old POTS service is maintained.
- *Very high bit-rate digital subscriber line* (VDSL) is an extension of ADSL technology to higher rates, up to 52 Mb/s downstream. At such high bit rates, the copper loops must be so short that optical fiber would be used for all but the last few thousand feet.

1.1 Motivation for DSL Study

Given a noisy communication channel, Claude E. Shannon [28] proved that there exists codes that make it possible to transmit information over the channel with an arbitrarily low probability of error if the information rate in bits per channel use is less than the *channel capacity.* Knowledge of this limit on information transmission rate is useful. However, Shannon's theorems are non-constructive and don't describe means to find codes which approach the capacity limit. More importantly, even if an oracle gave us sequences of such codes, it would still be necessary to know how to encode and decode such codes efficiently.

Today, suppliers of DSL technology are moving to optimize their plant usage and to operate their equipment closer to performance limits. It is evident that channel coding may play a pivotal role for improving the performance of DSL systems in terms of higher data rate and long loop reach. The current channel coding scheme for standard DSL concatenates *Reed-Solomon* (RS) codes and multi-dimensional *Trellis coded modulation* (TCM) codes. Although RS-TCM coding provides a significant coding gain for DSL systems, this gain still far away from reaching the channel capacity.

Recently rediscovered *low-density parity-check* (LDPC) codes have become the best-known practically implementable codes for closely approaching the capacity of various channels [7, 21]. Specifically, LDPC coding is already under consideration for use in DSL systems. This coding scheme for multilevel transmission should enhance the performance of the existing ADSL standards as well as those of the evolving VDSL.

In this thesis we apply LDPC coding techniques to DSL systems which operate over two types of channels, namely *additive white Gaussian noise* (AWGN) channels and frequency-selective multilevel AWGN channels. To study the mechanism of LDPC codes a *message-passing* algorithm on a *normal graph* exploited in the iterative probabilistic decoder is proposed. A generic scheme for implementing LDPC-coded modulation system is then introduced. Simulations are employed to compare this new coding scheme with the one currently used in the DSL RS-TCM standard. Our performance analysis and results show that significantly improved coding gains are obtained in DSL systems over both types of channels.

1.2 Basic Concepts

DSL is a local loop technology that uses existing standard twisted-pair copper wire to support voice and data communications simultaneously over a single wire pair without diminishing either the voice or data capacity.

1.2.1 ADSL

In ADSL systems, bandwidth provided one direction is different from that in the other. To simplify, more data can be delivered downstream (toward the user) than on the return upstream channel. A POTS splitter separates the voice channel from the data channel through highpass and lowpass filters to provide coexisting service. ADSL increases transmission rates from T1 (1.544 Mbps) and E1 (2.048 Mbps) to 6 Mbps downstream and from 64 Kbps to 640 Kbps upstream. ADSL is a single pair implementation and includes POTS service.

The initial application for ADSL was *video on demand* (VOD). ADSL has since become popular for access to the Internet, corporate networks, and interactive multimedia applications such as multi-player gaming, and for online education.

Additionally, ADSL operates at speeds of up to 6 Mbps so that it converts a POTS line into a high-speed digital line that provides simultaneous services of phone, fax and ultra-fast Internet access, as shown in Figure 1.1. Other applications might include broadcast entertainment video, distance learning, CD-quality music on demand, home shopping, and remote LAN access.

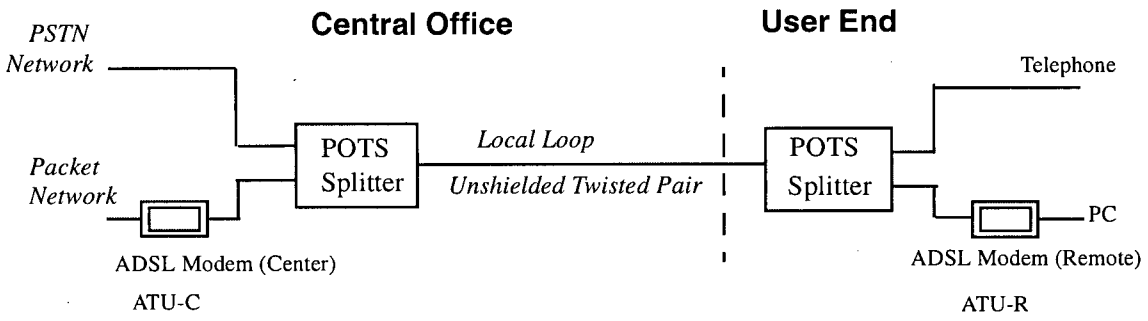


Figure 1.1 ADSL reference model.

1.2.2 VDSL

VDSL is a short-distance, high-bandwidth solution proposed to complement optical fiber as it becomes widely deployed. The VDSL bandwidth is considerably wider than that of conventional DSL, but the technology is still in its infancy. There are currently few products supporting VDSL technology.

One drawback for VDSL is the *crosstalk* (XTalk) among adjacent wire pairs, which act as antennae in the short wave range thereby interfering with radio communications. VSDL will be used primarily for loops fed from an *optical network unit* (ONU), which is typically located less than a kilometer from the customer. Few VDSL loops will be served directly from a *central office* (CO). Optical fiber connects the ONU to the CO. VDSL transmission over a twisted wire pair is used for the few thousand feet (the "last mile") from the ONU to the customer premises.

1.3 DSL Channel

In 1881, Alexander Graham Bell invented the *unshielded twisted pair* (UTP). With a sufficiently short space between twists, the electromagnetic coupling of energy over a small segment of wire is canceled by the out-of-phase energy coupled on the next segment of wire. Modern telephone cables are designed with slightly different twist rates for each wire pair to assure minimal crosstalk. Copper conductors are used to minimize signal attenuation due to electrical resistance.

1.3.1 Attenuation

The conductor and insulation are the major contributions to UTP cable attenuation. Attenuation is both frequency dependent and cable length dependent. In general, the higher the signal frequency and/or the greater length of cable, the greater the attenuation. Figure 1.2 shows a typical qualitative example.

The two dips in the figure are created by bridged taps. In spite of the large attenuation, it is still possible to use the twisted pair over a large frequency range (up to a few MHz) and to achieve high transmission rates [31].

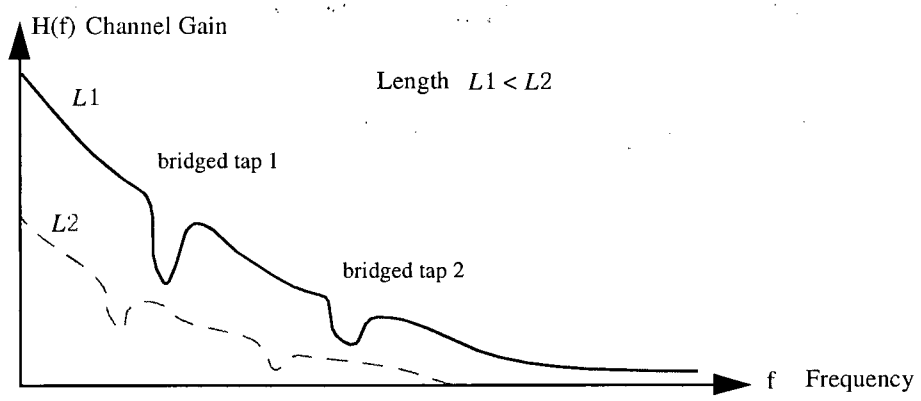


Figure 1.2 Channel gains in a twisted pair copper line with two bridged taps

Several types of DSL services are based on exploiting the bandwidth inherent in twisted wire pairs. Originally intended for transmission of speech (4 kHz voiceband) more than 100 years ago, the twisted pair copper wire has become increasingly valuable in terms of bandwidth utilization and commercial application. This has given rise to the popular saying that "DSL technology turns copper into gold".

1.3.2 Impairments

UTP cables were originally constructed to support services only in the voiceband spectrum. Several impairments become apparent when subscriber loops are used in the higher frequencies by DSL modems.

A. Crosstalk

Crosstalk noise in DSLs arises because the individual wires in the cable of twisted pairs radiate electromagnetically. The electric and magnetic field thus created induce currents in neighboring twisted pairs, leading to undesired crosstalk signals on those other pairs.

Generally there is a distinction between two different types of crosstalk, *near-end crosstalk* (NEXT) and *far-end crosstalk* (FEXT) (as shown in Figure 1.3). NEXT is the leakage from one wire pair into another at the same end of the cable whereas FEXT is the leakage between wire pairs on different ends of the cable. Normally, NEXT is most severe at the CO where an attenuated upstream signal can be drowned by strong NEXT interference. At the customer end the

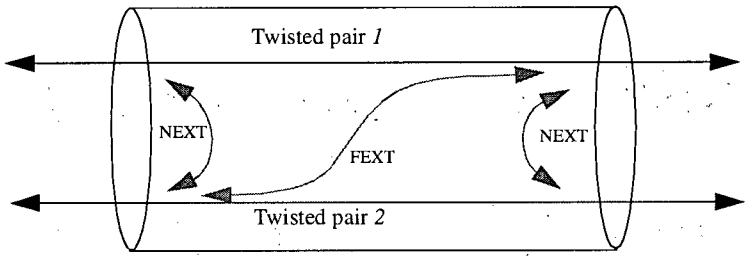


Figure 1.3 Crosstalk between twisted pair lines.

individual wires are spliced from the binder of cable thereby reducing NEXT levels.

The *power spectral density* (PSD) of NEXT and FEXT have been derived using transmission line theory [3]. The NEXT PSD from a source with PSD $S(f)$ is modeled by

$$S_{next}(f) = k_n f^{1.5} S(f), \text{ where } k_n = 10^{-13}\left(\frac{N}{49}\right)^{0.6}. \tag{1.1}$$

In (1.1), k_n is a constant determined from empirical studies [3] with N being the number of pairs in the same binder that carries the DSL service. The FEXT PSD is similarly modeled as follows:

$$S_{fext}(f) = k_f f^2 d\,|H(f,d)|^2\, S(f), \text{ where } k_f = 3.27 \times 10^{-19}\left(\frac{N}{49}\right)^{0.6} \tag{1.2}$$

In (1.2), $H(f, d)$ is the channel transfer characteristic with d denoting the transmission distance.

B. Radio frequency interference

Radio frequency interference (RFI) is the result of imbalance in the DSL pair. The imperfectly balanced UTP might emit radiation (egress) and receive outside radio-frequency signals (ingress).

For DSL loops, RFI is ingress noise introduced from radio transmitters in the vicinity of

UTP wires. Typical examples of RFI include amateur radio transmitters operating in the so called HAM radio band, and AM radios [3]. Although the transmit antennas of both type of radios are normally located far from UTP cable, some ingress into the wires of xDSL loops is always possible.

C. Impulse noise

There are many potential sources of nonstationary interference caused by transient electromagnetic fields in the region of UTP cables. Such burst noise is usually referred to as "impulse noise". Such noise arises from electronic components in the home such as refrigerator motor switching, phones ringing, microwave starting, electric drill motors, and other electrical appliances.

1.3.3 Composite channel

It is typical to have 50 twisted pairs bundled into one cable for several kilometers. As a result. the most dominant noise is the NEXT and FEXT interference created by services from

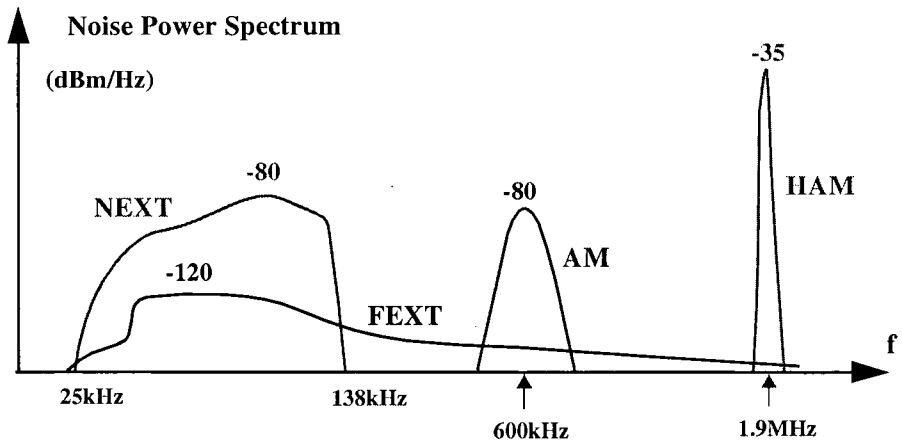


Figure 1.4 Noise spectrum in the ADSL loop combining various impairments

other cables. The statistics of these noises have been studied for many years both theoretically and by extensive measurements [29, 3]. Figure 1.4 shows typical power spectra [31] combining all impairment factors described previously in a 50-pair cable. The main point of this discussion is

that the total noise spectrum is quite complicated and is far from being a constant or a monotone decreasing function of frequency.

1.4 Simulink

This thesis involves a study of the various systems of DSL transceivers with the assistance of a Simulink environment. Simulink is a software package for modeling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, where different system parts are sampled or updated at different rates.

For modeling, Simulink provides a *graphical user interface* (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this interface, one can draw the models just one would with pencil and paper (or as most textbooks depict them). This approach has substantial advantages over those of previous simulation packages that require formulation of differential equations and difference equations in a language or program. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. One can also customize and create one's own blocks.

In the last few years, Simulink has become the most widely used software package in academia and industry for modeling and simulating dynamic systems. Simulink encourages one to explore various alternatives.

1.5 Outline of Thesis

This thesis proposes methods for applying LDPC codes in DSL systems over frequency-selective multilevel AWGN channels. The iterative probabilistic decoding approach utilizes a message-passing algorithm on a normal graph that is constructed by the parity-check matrix. The obvious improvement of performance is obtained by using this coding scheme in *discrete multitone* (DMT) systems over both flat AWGN and distorted AWGN channels.

In Chapter 2, the framework of current DSL systems is presented. In Section 2.1 we first

describe the specifications and implementation of DMT QAM modulation. Then, we demonstrate a generic blockflow of the DSL systems with DMT modulation. After that we show processes of RS TCM encoding and Viterbi decoding of the current ADSL standard.

In Chapter 3, we briefly introduce the properties of regular LDPC codes invented by Gallager and present effective means of decoding for such codes. In Section 3.3, we focus on the topic of probabilistic iterative decoding algorithms for linear binary block codes, particularly for LDPC codes. The mechanism of message-passing on a normal graph is introduced. We apply this message-passing algorithm for LDPC codes and further obtain the representations of posterior probabilities and log-likelihood posterior probabilities for corresponding messages. A simplified application for the algorithm is proposed by utilizing "min" approximations.

In Chapter 4, we construct appropriate LDPC codes for DSL systems. In Section 4.1, we use a trianglized systematic sparse parity-check matrix for LDPC codes to simplify the procedures of encoding and decoding. In Section 4.2, we choose a structure that is compatible with that of DMT DSL standards. In Section 4.3, to simplify the computation for decoding we employ M-ary double Gray-coded *pulse amplitude modulation* (PAM). We propose schemes which using frequency-domain zero-forcing equalization in the channel detector for demodulation and to facilitate the decoding of LDPC codes over various frequency-selective AWGN channels.

In Chapter 5, we show the results of our simulation. These results demonstrate performance improvement in the form of coding gain of around 2.5 dB at 10^{-6} bit error rates over existing DSL system designs. These results indicate that received power-to-noise levels may be reduced by 44%, either by transmitting at reduced power levels (thereby also reducing NEXT and FEXT on adjacent DSL lines), or by increasing DSL line length between repeaters, or by some combination of these approaches.

Finally, in Chapter 6 we provide a summary for this thesis, articulate important conclusions and propose future research related this topic.

Chapter 2 DSL Systems

Discrete multitone (DMT) modulation was introduced by Kalet [17] in 1989. DMT is based on the methods proposed by Peled and Ruiz [24] in 1980 to take advantage of digital signal processors and the *discrete Fourier transform* (DFT). Use of DMT for ADSL was first proposed by John M. Cioffi in 1991. In March 1993 the DMT system was chosen to be the basis of the ANSI standard for ADSL.

2.1 DMT Modulation

The DMT is a form of *multicarrier modulation* (MCM), the basic principle of which is shown in Figure 2.1. A block of b bits is transmitted over a set of parallel passband channels covering disjoint frequency bands. The transmitter sums N independent waveforms and the receiver separates them with frequency selective filters.

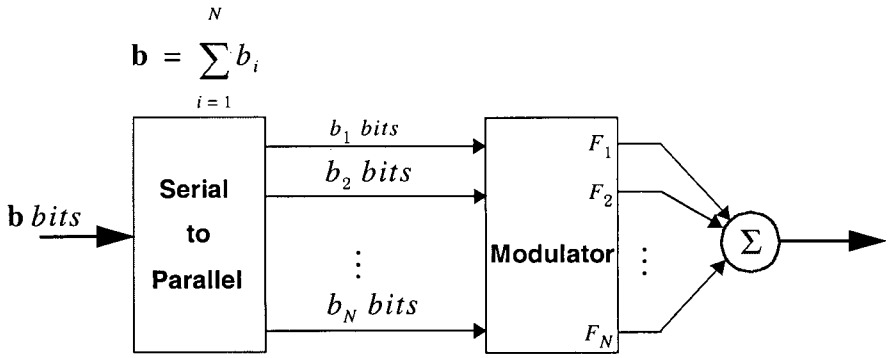


Figure 2.1 Basic multicarrier transmitter.

2.1.1 Adaptive multichannel transmission

In data communication systems, a common and difficult problem is that of data transmission over a wideband channel with severe *intersymbol interference* (ISI). The basic concept is illustrated in Figure 2.2, where two DSL transmission line characteristics are depicted; each

would have severe ISI if a single wideband signal were transmitted. An efficient method to solve this problem is DMT, whose essence is to divide a wideband channel into a large number of parallel narrowband AWGN subchannels. These subchannels usually correspond to contiguous disjoint frequency bands. Data transmission over a wideband channel is transformed into the parallel transmission over the subchannels. The overall data rate is the sum of the individual data rates of the subchannels.

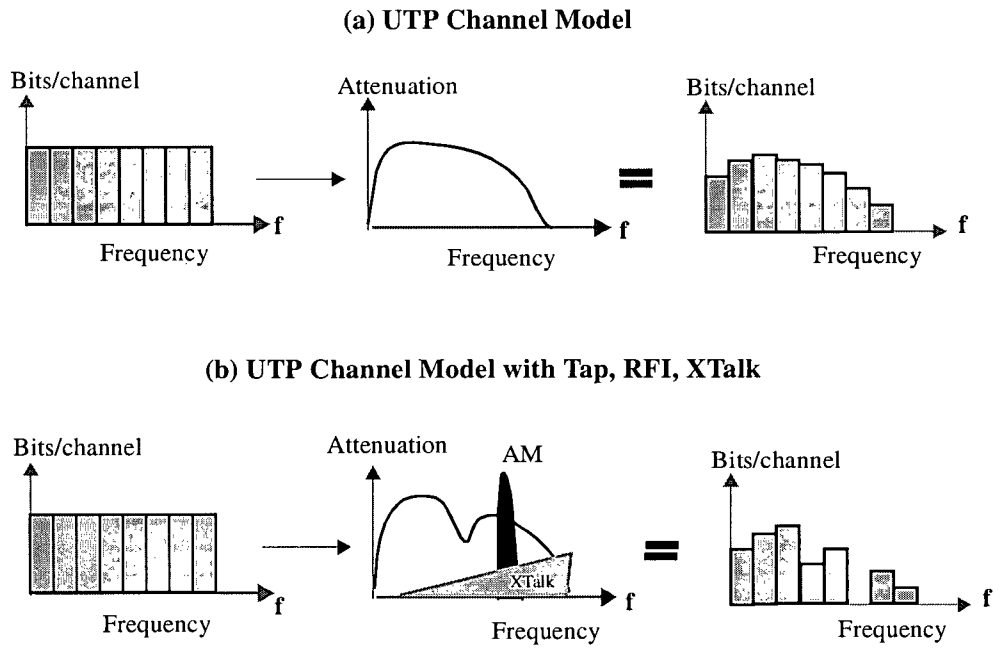


Figure 2.2 Basic concept of adaptive multichannel (multitone) transmission

If each of such subchannels has sufficiently narrow bandwidth, then, with appropriate signal design each subchannel has little or no ISI [16, 29], and each subchannel independently approximates an AWGN channel.

In a DMT system, the transmission channel is partitioned into equal bandwidth subchannels [17]. The purpose of DMT in DSL is to effectively utilize the frequency-dependent channel characteristics. The multiple subchannels are modulated by *quadrature amplitude modulation* (QAM). The high-rate bit stream is thus divided into tens or hundreds of low-rate streams. Each QAM modulator works in a different frequency band and transmits bit streams at different rates.

An example of the frequency spectrum and modulation strategy of an ADSL system is shown in Figure 2.3. Additionally, a specific example of DMT is *orthogonal frequency division multiplexing* (OFDM)[1], which is using in many wireless transmission applications.

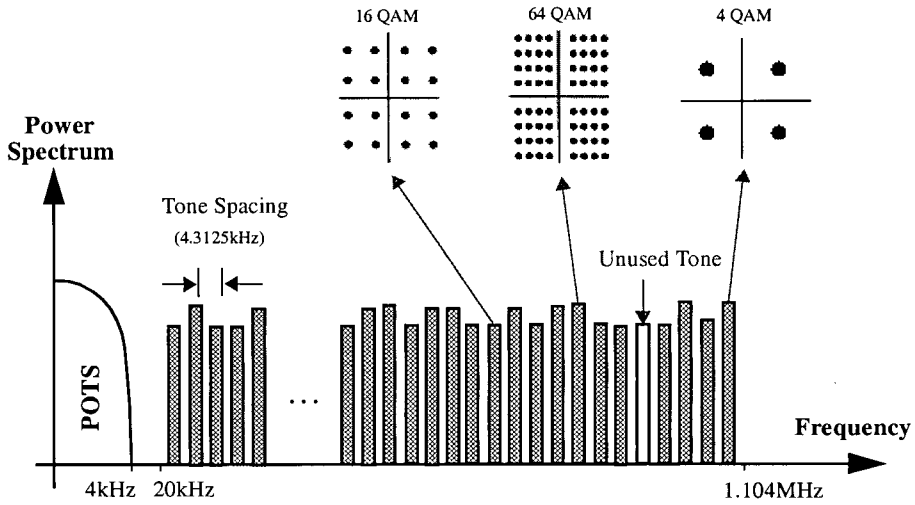


Figure 2.3 ADSL frequency spectrum.

2.1.2 Bit-loading algorithm

Since the frequency response of the channel in DSL systems can be considered relatively fixed, the *signal to noise ratio* (SNR) on each subchannel can be determined with good accuracy. Bit-loading means that different numbers of bits are assigned to each subcarrier depending on the associated subchannel SNR. Better subchannels (with high SNR) carry more information, while poor subchannels (with low SNR) carry little or no information (see Figures 2.2 and 2.3). For an AWGN channel (assume all subchannels are narrow enough that they can be taken as AWGN channels), the average number of bits on each subchannel is defined [17, 29] by

$$\overline{b_k} = \frac{1}{2} log_2\left(1 + \frac{SNR_k}{\Gamma}\right)(bits)/dimension \quad 0 \le k \le N-1 \tag{2.1}$$

[1] The difference between DMT and OFDM is that DMT uses optimal bit-loading algorithm while OFDM puts a fixed and equal number of bits on all subchannels.

where Γ is a gap that quantifies the effective loss in SNR *with respect to* (*w.r.t.*) the channel capacity.

The water-filling (or water-pouring) energy distribution algorithm for subchannel loading can be approximated by a flat distribution on virtually all DSL subchannels with minuscule loss in performance, as long as the correct transmission band is used, as shown by P. Chow [6].

2.1.3 DMT application

The DMT technique is used as the standard method for ADSL [1] and VDSL [8]. It is a method by which to approximate complex filters needed to accomodate to the channel filter characteristic by simpler DFT operations. These DFT operations are applied to implement linear matrices transformation for the entire modulation/demodulation process [3, 29].

In DMT, an extended guard interval contains a cyclic prefix. The v samples in the prefix repeat those at the last of the symbol, i.e.,

$$x_{-i} = x_{N-i} \text{ for } i = 1, 2, \ldots, v. \tag{2.2}$$

With the existence of the cyclic prefix, much processing simplification occurs. In such case, the matrix description **H** of the DSL channel with frequency transfer characteristic $H(f)$ is an $N \times N$ equivalent "circulant" matrix of the following form [29, 16]:

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{v-1} & h_v & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{v-2} & h_{v-1} & h_v & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_v \\ h_v & 0 & 0 & \cdots & 0 & 0 & h_0 & \cdots & h_{v-1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ h_1 & h_2 & h_3 & \cdots & h_v & 0 & 0 & \cdots & h_0 \end{bmatrix} \tag{2.3}$$

If the discrete-time representation of the channel is depicted as shown in Figure 2.4, then the relation of variables can be described in the compact matrix form as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{2.4}$$

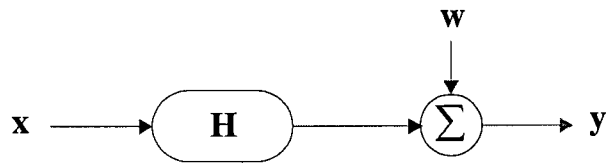


Figure 2.4 Discrete-time representation of multichannel data transmission system

In (2.4), **x**, **w** and **y** denote, respectively, the transmitted symbol vector[2], the channel noise vector and the received symbol vector. They are defined as follows: $\mathbf{x} = \left[x_{N-1}\ x_{N-2}\ \ldots\ x_0\right]^{\mathrm{T}}$, $\mathbf{w} = \left[w_{N-1}\ w_{N-2}\ \ldots\ w_0\right]^{\mathrm{T}}$ and $\mathbf{y} = \left[y_{N-1}\ y_{N-2}\ \ldots\ y_0\right]^{\mathrm{T}}$.

An important property of the circular matrix **H** in equation (2.3) is that it can be spectrally decomposed as follows [16]:

$$\mathbf{H} = \mathbf{Q}^{\dagger}\Lambda\mathbf{Q} \tag{2.5}$$

In (2.5), the superscript † denotes *Hermitian transposition,* **Q** is a matrix corresponding to DFT and Λ is a diagonal matrix containing the N Fourier transform values denoted by $\lambda_{N-1}, \ldots, \lambda_1, \lambda_0$ for the sequence $h_0, h_1, \ldots, h_v$ characterizing the channel. The kl-th element of $N \times N$ matrix **Q** starting from the bottom right at $k = 0$ and $l = 0$ and counting up and leftward is

$$Q_{kl} = \frac{1}{\sqrt{N}}\exp\left(-j2\pi\frac{kl}{N}\right) \text{ for } k, l = 0, 1, \ldots, N-1. \tag{2.6}$$

Thus one sees that the matrix Q is an orthonormal matrix with a property that

$$\mathbf{Q}^{\dagger}\mathbf{Q} = \mathbf{I} \tag{2.7}$$

---

[2] When a cyclic prefix is used, the last v output samples are excluded from **x**, **w** and **y**.

where **I** is the identity matrix. Therefore, $\mathbf{Q}^{-1} = \mathbf{Q}^{\dagger}$.

Assume that $\mathbf{x} = \mathbf{Q}^{\dagger}\mathbf{X}$ where **X** is the frequency-domain vector of channel input sequence. The first $N/2$ elements of the vector **X** can be taken as $N/2$ parallel complex-valued QAM constellation points. Set the left $N/2$ elements as $X_i = X_{N-i}^*$ for $i = N/2, \ldots, N-1$ [3], then the symbol **x** must be real.

Define the frequency-domain vector representation of the channel output vector **y** as $\mathbf{Y} = \mathbf{Qy}$, then

$$\mathbf{Y} = \mathbf{Q}(\mathbf{Hx} + \mathbf{w}) = \mathbf{Q}(\mathbf{Q}^{\dagger}\Lambda\mathbf{Q}\mathbf{Q}^{\dagger}\mathbf{X} + \mathbf{w}) = \Lambda\mathbf{X} + \mathbf{W} \tag{2.8}$$

where $\mathbf{W} = \mathbf{Qw}$. From equation (2.8), we obtain

$$Y_i = \lambda_i X_i + W_i \text{ for } i = 0, 1, \ldots, N-1. \tag{2.9}$$

Equation (2.9) indicates that N parallel independent operations can be implemented. For the AWGN channel with knowledge of the measured frequency response λ_i is available for $i = 0, 1, \ldots, N-1$. By computing a maximum likelihood estimate of each received element Y_i the frequency-domain value of each transmitted element X_i is determined.

It should be mentioned that the price for the computation simplification is the "wasted energy" in the cyclic prefix. The effective power rate is thereby lowered to $\frac{N}{N+v}$.

2.2 DSL Systems

DMT modulation technology is a part of standards of both ADSL [1] and VDSL [8].

---

[3] In DSL standards [1, 8] the elements $X_i = X_{N-i}^*$ are defined for $i = N/2 + 1, \ldots, N-1$, there is not data transmitted in the subcarriers 0 and $N/2$.

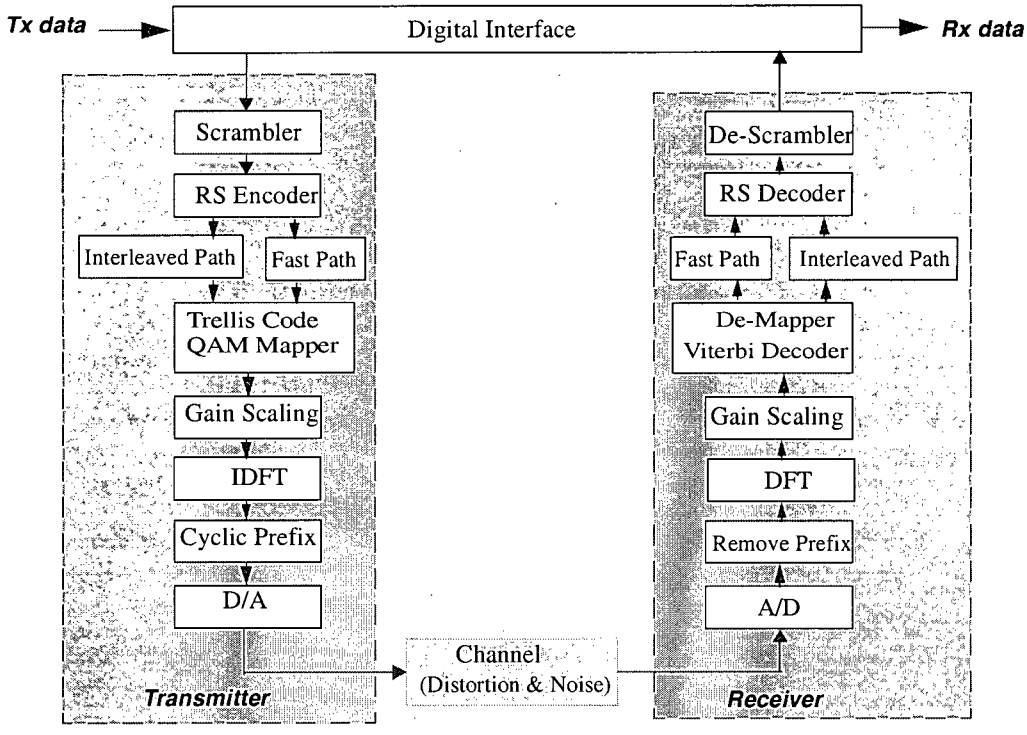


Figure 2.5 Block Diagram of a DSL system with DMT modulation.

Figure 2.5 is a general block diagram of a DSL system with DMT modulation. In the transmitter the data from the digital interface is scrambled, RS encoded, and interleaved or not (in the fast path). Data from interleaved and fast paths are then mapped onto a sequence of constellation points, each of which is for respective used tone in a DMT symbol. (In an ADSL system, the mapping function may include a Wei's 16-state *4-dimensional* (4-D) trellis [32] encoder.) After that, each used tone is multiplied by a gain-scaling coefficient to control the power of the subchannel. The scaled frequency domain symbol is then transformed into the time domain by using an *inverse discrete Fourier transform* (IDFT). Before entering the channel following D/A conversion the time domain symbol is extended by a cyclic prefix.

Following A/D conversion at the receiver and removal of the cyclic extension, the time domain symbols are converted into the frequency domain by using a DFT. The de-mapper function makes decision on the noisy constellation points. The corresponding Viterbi decoding [12, 32] is employed in this stage. Following that, the output byte stream is de-interleaved, the RS codewords are decoded, and the data is then de-scrambled and sent to the digital interface.

2.2.1 TCM encoding in ADSL

The simplest *forward error correction* (FEC) scheme provided in DSL systems is the RS code [1, 8], which is a cyclic code. RS code arithmetic executes in *Galois Field 256* (GF256) and allows up to 16 erroneous bytes in a codeword of up to 255 bytes to be corrected. A further option used in the ADSL ANSI T1.413 standard is to concatenate the RS code with TCM [29, 1, 3]. The RS code is here the outer code while the Trellis code is the inner code.

In ADSL, TCM uses Wei's 16-state 4-D convolutional encoder [32] to produce convolutional codes [1]. The encoding circuit is shown in Figure 2.6. The trellis encoder which uses a set

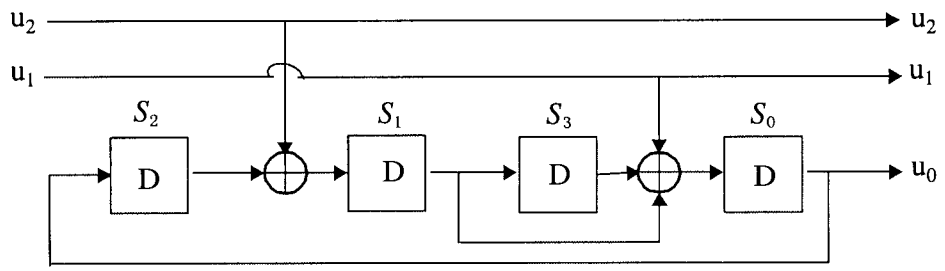


Figure 2.6 Wei's 16-state 4-D convolutional encoder.

of bits $\mathbf{u} = [u_1, u_2, u_3, \ldots, u_{x+y-1}]$ as its input vector is shown in Figure 2.7. Using a 4-D convolutional encoder, each codeword **u** is encoded into two binary words $\mathbf{v} = [v_0, v_1, \ldots, v_{x-1}]$ and $\mathbf{w} = [w_0, w_1, \ldots, w_{y-1}]$ which are modulated into two constellations of points for the successive subchannels. Some specifications of this encoding scheme are as follows:

- $u_0 = S_0$, and (u_0, u_1, u_2) is used to select one of the eight 4-D subsets.
- The subset chosen is mapped to two indices that determine the *least significant bits* (LSBs) of **v** and **w**. The mapping scheme is as following:

$$\begin{cases} v_0 = u_3 & w_0 = u_2 \oplus u_3 \\ v_1 = u_1 \oplus u_3 & w_1 = u_0 \oplus u_1 \oplus u_2 \oplus u_3 \end{cases} \tag{2.10}$$

- The remaining bits of **u** are mapped to the *most significant bits* (MSBs) of **v** and **w**.

In a trellis code modulation system, the expanded constellation is labeled and partitioned

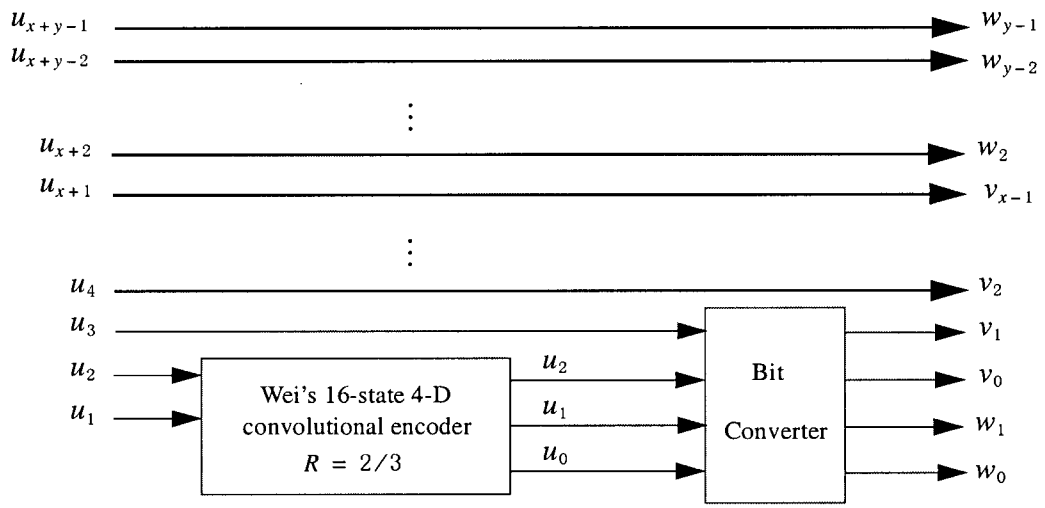


Figure 2.7 TCM Encoder in ADSL.

into subsets ("cosets") using a technique called mapping by set-partitioning. The eight 4-D cosets in Wei's code are composed of concatenation of a pair of *2-dimensional* (2-D) cosets. For example, $C_4^{\,5} = (C_2^{\,0} \times C_2^{\,1}) \cup (C_2^{\,3} \times C_2^{\,2})$. The four constituent 2-D cosets, denoted by $C_2^{\,0}$, $C_2^{\,1}$, $C_2^{\,2}$, $C_2^{\,3}$, are shown in Figure 2.8.

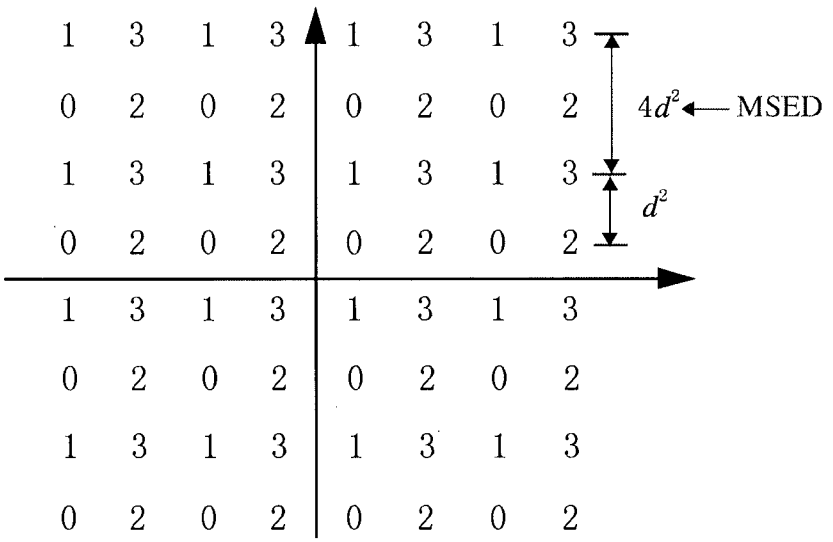


Figure 2.8 Constituent 2-D cosets for Wei's code.

The encoding algorithm ensure that the 2 LSB bits of a constellation point comprise the index k of the 2-D coset C_2^k where the constellation point locating is specified for $k = 0, 1, 2, 3$. The bits (v_1,v_0) and (w_1,w_0) are actually binary representations of the index k. The three bits (u_2, u_1, u_0) determine which cosets to be chosen from the 8 4-D cosets. The 8 cosets are labeled C_4^n where n is the integer with binary representation (u_2, u_1, u_0). The bit u_3 is used to determine which one of the two Cartesian products of 2-D cosets in the 4-D coset is chosen (see Figure 2.7). The 2 bits (v_1,v_0) and (w_1,w_0) are obtained from (u_3, u_2, u_1, u_0) using the linear equations (2.10). All relationships are shown in Table 2.1.

Table 2.1 Relation between 4-D and 2-D cosets.

4-D Coset	$u_3\ u_2\ u_1\ u_0$	$v_1\ v_0$	$w_1\ w_0$	**2-D Cosets**
C_4^0	0 0 0 0 1 1 1 1	0 0 1 1	0 0 1 1	$C_2^0 \times C_2^0$ $C_2^3 \times C_2^3$
C_4^4	0 1 0 0 1 1 0 0	0 0 1 1	1 1 0 0	$C_2^0 \times C_2^3$ $C_2^3 \times C_2^0$
C_4^2	0 0 1 0 1 0 1 0	1 0 0 1	1 0 0 1	$C_2^2 \times C_2^2$ $C_2^1 \times C_2^1$
C_4^6	0 1 1 0 1 1 1 0	1 0 0 1	0 1 1 0	$C_2^2 \times C_2^1$ $C_2^1 \times C_2^2$
C_4^1	0 0 0 1 1 0 0 1	0 0 1 1	1 0 0 1	$C_2^0 \times C_2^2$ $C_2^3 \times C_2^1$
C_4^5	0 1 0 1 1 1 0 1	0 0 1 1	0 1 1 0	$C_2^0 \times C_2^1$ $C_2^3 \times C_2^2$
C_4^3	0 0 1 1 1 0 1 1	1 0 0 1	0 0 1 1	$C_2^2 \times C_2^0$ $C_2^1 \times C_2^3$
C_4^7	0 1 1 1 1 1 1 1	1 0 0 1	1 1 0 0	$C_2^2 \times C_2^3$ $C_2^1 \times C_2^0$

The trellis diagram relates to the finite state in Figure 2.6 as shown in Figure 2.9. In Figure 2.9, $\mathbf{S} = (S_3, S_2, S_1, S_0)$ represents source state, while $\mathbf{T} = (T_3, T_2, T_1, T_0)$ represents target state in the finite state machine. Each state **S**, represented by its hex value in bold, is connected to four

states **T**, similarly represented by four branches determined by the values of u_2 and u_1. Each branch is labeled with the 4-D coset specified by the value of u_2, u_1 (and $u_0 = S_0$, see Figure 2.6).

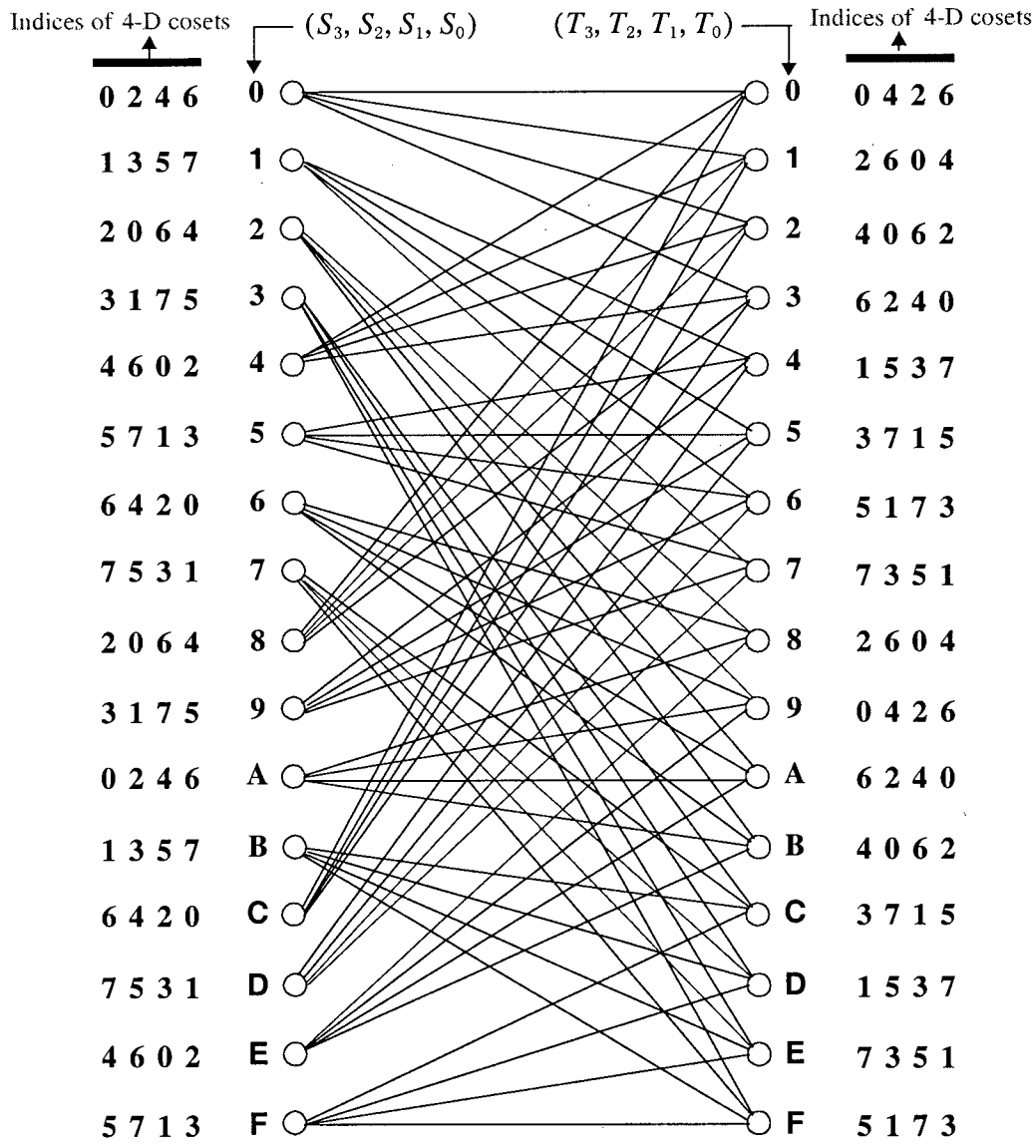


Figure 2.9 One stage of trellis diagram of the Wei's code.

The branches connect each source state to those target states that can be reached from that particular source state. The trellis flow is from left to right and each branch represents a 4-D coset. The eight possible indices of 4-D cosets are numbered from 0 to 7. The index numbers are ordered such that the leftmost index corresponds to the topmost branch. The indices on the left side

indicate the cosets numbers of the outgoing branches from the that source state. The indices on the right side indicate the coset numbers corresponding to the incoming branch to that target state.

2.2.2 TCM decoding in ADSL

The TCM decoder implements the Viterbi algorithm [12, 32] that performs *maximum likelihood* (ML) decoding. The Viterbi algorithm finds the path through a trellis that is most likely to have resulted in the sequence of received symbols. As a preliminary step, the algorithm must determine that point in each of the multidimensional subsets which is closest to the received point, and compute its associated *metric* (the squared Euclidean distance between the two points). Since the 4-D constellation is partitioned into eight 4-D subsets ("2-D cosets") with $4d^2$ as the *minimum squared Euclidean distance* (MSED) (see Figure 2.8), each received 4-D point can be divided into a pair of 2-D points. The closest point in each 4-D subset and its associated metric can be found based on the point in each of the 2-D subsets which is closest to the corresponding received 2-D point and its associate metric [32]. Figure 2.10 shows the process of the Viterbi decoding.

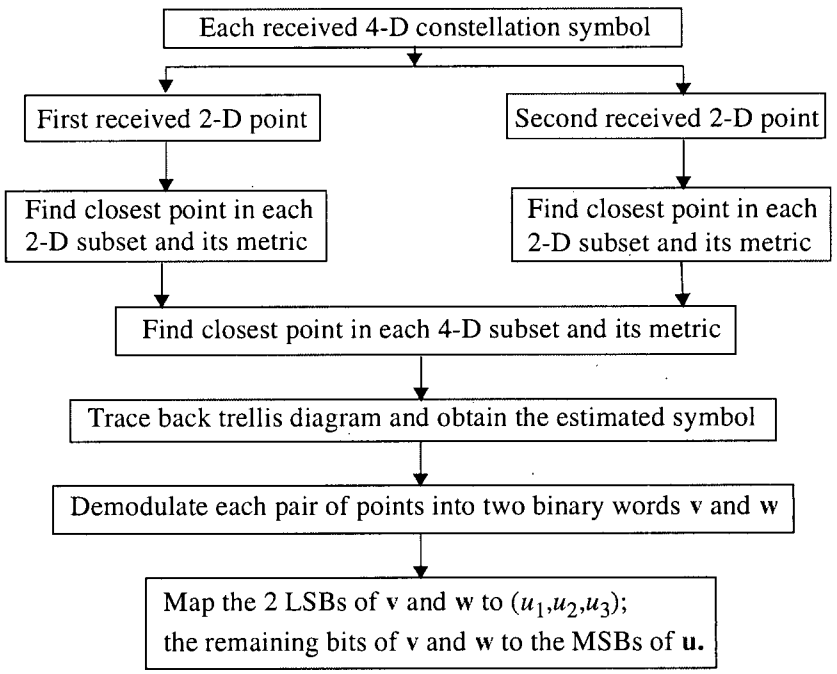


Figure 2.10 Diagram for decoding Wei's 16-state 4-D code.

Chapter 3 LDPC Coding and Decoding

Low-density parity-check (LDPC) codes were introduced along with an iterative probabilistic decoding algorithm by Gallager [14, 15] in the early 1960's. These codes were constructed using sparse random parity check matrices. However, they were not fully utilized until the advent of turbo codes [2], when they were rediscovered and generalized by MacKay and Neal [21], who showed that they perform almost as close to capacity as turbo codes. This reality makes LDPC codes not only attractive from a theoretical point of view, but also ideal for practical applications. Recently, it has been recognized that the various message passing decoding algorithms have provided good decoding performances for LDPC codes [5, 18, 23, 33]. In this chapter we give a brief overview of the origins of LDPC codes and the methods used for their coding and decoding.

3.1 Gallager LDPC Coding

LDPC codes, invented by Gallager [15] in his Ph.D. thesis, are defined by a very sparse parity check matrix **H** with n columns and m rows. The sparsity of **H** is key property that allows for the algorithmic efficiency of LDPC codes.

In Gallager's original construction [14], a (n, ρ, γ) LDPC code is a binary linear block code of length n, with matrix **H** having a fixed column weight ρ and a fixed row weight γ, where $\rho \geq 3$ and $\gamma > \rho$. The integers ρ and γ should be small *w.r.t.* n in order for **H** to be sparse. Therefore, m is the total number of parity equations of the code and there are a total of 2^k codewords, where the message length $k = n - m$. The code rate $R = \frac{k}{n} = 1 - \frac{m}{n} = 1 - \frac{\rho}{\gamma}$.

A specific example is a 15×20 matrix **H** of the regular $(n, \rho, \gamma) = (20,3,4)$ binary LDPC code constructed by Gallager is shown in Figure 3.1. Obviously, $R = 1/4$. The parity-check matrix is divided horizontally into $\rho = 3$ equal size sub-matrices, each of which has a single '1' in every column. We define $\beta_{overlap}$ as the maximum number of ones in the same row position in

$$\mathbf{H} = \left[\begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

$n = 20$
$m = 15$
$\rho = 3$
$\gamma = 4$

Figure 3.1 Gallager's construction of **H** matrix of a regular (20, 3, 4) LDPC code

any two columns. In Figure 3.1, one sees that $\beta_{overlap} = 1$ and the code defined by this parity check matrix is a linear block code with the minimum distance $d_{min} = 6$. Therefore, the error correcting capability [34] of the code is $t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor = 2$.

If we denote the rows of **H** matrix as $\mathbf{h}_j = (h_{j,1}, h_{j,2}, \ldots, h_{j,n})$, and the codeword vector as $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, we obtain the parity check constraint equations as the inner product of $\mathbf{h}_j$ and **c** as follows:

$$\mathbf{h}_j \cdot \mathbf{c} = 0 \qquad j = 1, 2, \ldots, m \tag{3.1}$$

or equivalently:

$$\sum_{i=1}^{n} c_i h_{j,i} = 0 \qquad j = 1, 2, \ldots, m \tag{3.2}$$

From (3.2), one sees that every bit c_i of codeword **c** is checked by exactly ρ parity check

equations.

If there is a single bit error in the received vector **v** at position i $(1 \le i \le n)$, then the ρ parity check equations for checking bit v_i will be violated, i.e.,

$$\mathbf{v} \cdot \mathbf{h}_j = 1 \qquad j \in J_i = \{ j \mid h_{j,i} = 1 \}, 1 \le j \le m \text{ with } |J_i| = \rho \tag{3.3}$$

where $|J_i| = \rho$ means there are ρ '1's in column i of parity matrix **H**. For example, for the code given in Figure 3.1 if there is a single error bit in the received vector **v** at position $i = 3$, then $J_3 = \{1, 8, 13\}$. Thus the 1st, 8th and 13th parity check equations will fail.

If there are two error-bits in the received vector **v**, since $\beta_{overlap} \ll \rho$, the total number of parity check failures will increase (due to $2\rho - \beta_{overlap} > \rho$), and for each erroneous bit position the number of failed parity-check equations will be greater than $\lfloor \rho/2 \rfloor$. In the example of the code given in Figure 3.1, assume two errors occur at position 1 and 6. Thus the 1st, 2nd, 6-th and 7-th parity-check equations fail. The error in position 1 is contained in the 1st and 6-th failed parity-check equations, and the error in position 6 is contained in the 2nd and 7-th failed parity-check equations. If the errors occur at positions 1 and 11, then there are 6 parity-check failures, namely the 1st, 3rd, 6th, 9th, 11th, and 12th parity-check equations. Each erroneous bit position causes 3 failed parity-check equations.

3.2 Gallager LDPC Decoding

LDPC codes are linear block codes with a specific construction. In theory they can be decoded utilizing conventional schemes such as standard array and syndrome table decoding [34]. For a binary linear (n, k) block code, the standard array decoder requires a 2^n size of memory for each possible binary n tuple. Syndrome table decoding needs to store each of the 2^{n-k} error patterns of length n. These schemes have huge storage requirements as n and $n-k$ are usually large values for LDPC codes, typically with both n and $n-k$ being on of the order of hundreds or thousands.

As parity-check matrices of LDPC codes are sparse matrices with very low density of ones, iterative techniques used to decode LDPC codes become feasible. Based on specified parity-check matrices **H** for LDPC codes, two decoding schemes devised by Gallager were the bit-flipping decoding algorithm and the probabilistic decoding algorithm.

The bit-flipping decoding algorithm is particularly simple but is usable only for the *binary symmetric channel* (BSC) at the rates far below channel capacity. It is an iterative decoding scheme based on a hard decision of the received bits [14, 15]. We do not consider this topic in this thesis.

The probabilistic decoding algorithm utilizes the structure created by the parity-check equations more systematically than does the bit flipping algorithm. The probabilistic decoder is based on the message-passing algorithm [26], which involves passing probabilistic messages based on a graph or a network structure [18, 33] generated from the parity-check matrix **H**. Probabilistic decoding algorithms have become a general class of decoding algorithms for LDPC codes. The algorithm decodes directly from a *posteriori* probabilities at the channel output and exhibits much better performance than the bit-flipping algorithm. We will expand discussion of this algorithm in the next section.

3.3 Probabilistic Decoding for LDPC Codes

Gallager provided a probabilistic decoding algorithm that works directly on the nodes and edges of the *bipartite graph* representing the code [15]. In 1981, Tanner [30] introduced a bipartite graphical model ("*Tanner Graph*") for LDPC codes and generalized the parity check constraints of LDPC to general linear block code constraints. Tanner proved the optimality of the *sum-product* and *min-sum* algorithms for decoding codes defined on *cycle-free* graphs. The belief propagation [22, 33] algorithm can be used to derive a number of iterative decoding algorithms for error control systems, including for LDPC codes, turbo codes, and others. Based on that the iterative decoding of compound codes is a instance of probability propagation algorithms that operate on a graphical model of the codes [33], the *factor graphs* were introduced by Kschischang, et al. [18]. They were shown to subsume many graphical models such as *Bayesian networks*, *Markov random fields* and the *Tanner graph*. The *sum-product* algorithm operates on

factor graphs to compute various marginal functions by distributed message passing in the graphs. In 2001, Forney [11] proposed the normal realizations that introduced a *normal graph*, which has some advantage over the *factor graph*. There are no topological restrictions and a clean functional separation is obtained between operations in the message-passing algorithm.

In following sections, the *normal graph* and message-passing algorithm are presented. Then we derive LDPC decoding algorithm based on them.

3.3.1 Normal graph

A *normal graph* Φ is an undirected graph, consisting of nodes and edges, as introduced by Forney [11]. Each node can be connected to any number of edges, and each edge can be connected to one or two nodes. An edge connected to two nodes is called an *internal edge*, while an edge connected to only one node is called an *external edge* (a.k.a., "half-edge" or "leaf-edge").

Each edge is usually associated to a variable x_i that takes value over a finite, discrete alphabet denoted by A_i. The compound term "*edge-variable*" is used where we discuss the combination of the edge and its associated variable. The variables associated with the external edges are known as the "*symbol variables*" and represent the coupling of the system with its environment, while the internal edge-variables are known as the "*state variables*" and represent internal states of the system.

To each node is associated a *local constraint*, which expresses a relationship between the variables associated with the edges connected to that node. As shown in Figure 3.2, node N connects to $k+1$ edges that are associated with the variables $x_0, x_1, \ldots, x_k$, and it has a constraint set S_N that combines the local constraints of $k+1$ variables. There is one external edge associated with the symbol variable x_0 and there are k internal edges associated with the state variables $x_1, \ldots, x_k$.

Let $(x_0, x_1, \ldots, x_k)$ is a set of variables with a configuration space $S = A_0 \times A_1 \times \cdots \times A_k$, and let S_N be a subset of S, then the elements of S_N are the valid configuration [18]. Suppose

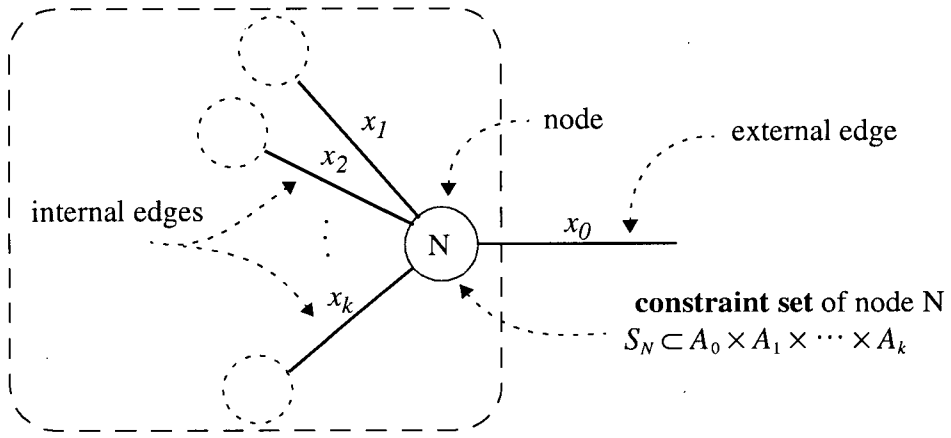


Figure 3.2 A example of normal graph with one node

$(x_0, x_1, \ldots, x_k)$ belongs to S_N, which is a constraint set ($S_N \subset S$). If the prior probabilities of $x_0, x_1, \ldots, x_k$ are available *w.r.t.* node N, and in addition, the variables are mutually independent (which is true for the graphs without cycles), then the joint prior probability $P_N(\{x_i = a_i\}|_{i=0}^{k})$ can be factored into the product of the individual prior probabilities, i.e.,

$$P_N(\{x_i = a_i\}|_{i=0}^{k}) = \prod_{i=0}^{k} P_N(x_i = a_i). \tag{3.4}$$

3.3.2 Basic definitions and simplified notations

In Figure 3.2, there are several types of probabilities that can be associated with the variables x_i for $i = 0, 1, \ldots, k$.

A. Basic definitions

- the *intrinsic probability* for x_i *w.r.t.* node N is the prior probability denoted by

$$P_N^{in}(x_i = a_i) = P(x_i = a_i); \tag{3.5}$$

- the *posterior probability* for x_i *w.r.t.* node N is the conditional probability based on node N denoted by

$$P_N^{post}(x_i = a_i) = P(x_i = a_i|N); \tag{3.6}$$

- the *extrinsic probability* for x_i *w.r.t.* node N is the new information for x_i that has been obtained from node N denoted by

$$P_N^{ex}(x_i = a_i) = C_{x_i} P(N|x_i = a_i), \tag{3.7}$$

where the normalization constant $C_{x_i} = \left(\sum_{a_i \in A_i} P(N|x_i = a_i) \right)^{-1}$ is used to meet the condition $\sum_{a_i \in A_i} P_N^{ex}(x_i = a_i) = 1$.

By using Bayes' rules, $P(x_i = a_i|N) = \frac{1}{P(N)} P(N|x_i = a_i) P(x_i = a_i)$, then

$$P_N^{post}(x_i = a_i) = C_{x_i}' P_N^{in}(x_i = a_i) P_N^{ex}(x_i = a_i), \tag{3.8}$$

where $C_{x_i}' = \left(\sum_{a_i \in A_i} P_N^{in}(x_i = a_i) P_N^{ex}(x_i = a_i) \right)^{-1} = (P(N) C_{x_i})^{-1}$ is also a normalization constant to meet the condition $\sum_{a_i \in A_i} P_N^{post}(x_i = a_i) = 1$.

In (3.7), $P_N^{ex}(x_i = a_i)$ is a probability that describes the new information for x_i that has been obtained from message-passing. The extrinsic message is an outgoing message from node N. For instance, an extrinsic probability $P_N^{ex}(x_0 = a_0)$ for the edge-variable x_0 can be computed at node N based on the intrinsic probabilities $P_N^{in}(x_i = a_i)$, $i = 1, \ldots, k$ for the other edge-variables connected to node N, as follows:

$$P_N^{ex}(x_0 = a_0) = C_{x_0} \sum_{\substack{a_1, \ldots, a_k \\ (a_0, a_1, \ldots, a_k) \in S_N}} \prod_{i=1}^{k} P_N^{in}(x_i = a_i) \tag{3.9}$$

In (3.9) C_{x_0} is a normalization constant, and summation is over all combinations of the variables $a_1, a_2, \ldots, a_k$ $(a_i \in A_i)$ such that the *configurations* $(a_0, a_1, \ldots, a_k) \in A_0 \times A_1 \times \cdots \times A_k$ satisfy

the constraints set for S_N.

B. Simplified notations

- any variables x_i, may take on a value that is omitted in the equations but can be inferred from context. For example, we use $P(x_i)$ to represent the probability $P(x_i = a_i)$, where a_i is some implicit value.
- Instead of indicating the variables being summed over, the "not-sum" or summary notation [18] indicates those variables not being summed over. For example, $\sim\{x_2\}$ is the notation of the not-sum over variable x_2, and h is a function of three variables x_1, x_2, and x_3, then the "not-sum" or "summary of h for x_2" is denoted as $\sum_{\sim\{x_2\}} h(x_1, x_2, x_3) = \sum_{x_1, x_3} h(x_1, x_2, x_3)$.

3.3.3 Message-passing algorithm

To study message-passing on graph with two nodes, the graph labeled Φ in Figure 3.3 has

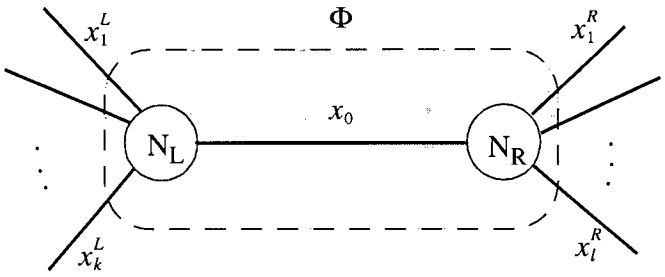


Figure 3.3 A normal graph with two nodes

two nodes N_L and N_R connected by an edge-variable x_0. Besides x_0, node N_L has k additional edges (labeled $x_1^L, x_2^L, \ldots, x_k^L$) and node N_R has l additional edges (labeled $x_1^R, x_2^R, \ldots, x_l^R$). Let the local constraint sets for nodes N_L and N_R be denoted by S_L and S_R, i.e. $(x_0, x_1^L, \ldots, x_k^L) \in S_L$ and $(x_0, x_1^R, \ldots, x_l^R) \in S_R$. Let the intrinsic probabilities *w.r.t.* N_L and N_R be given by

$\begin{cases} P_L^{in}(x_i^L) = P_\Phi^{in}(x_i^L) & i = 1, 2, \ldots, k \\ P_R^{in}(x_j^R) = P_\Phi^{in}(x_j^R) & j = 1, 2, \ldots, l \end{cases}$, and the intrinsic probability *w.r.t.* Φ for the internal edge x_0 be set to a uniform value, i.e., $P_\Phi^{in}(x_0) = \frac{1}{|A_0|}$ for all $x_0 \in A_0$.

Consider the extrinsic probability for x_i^R *w.r.t.* the graph Φ (for example $P_R^{ex}(x_1^R)$ in Figure 3.4). Assume the notation $\mu_{L \to R}(x_0)$ indicates the message from node N_L to node N_R along the edge associated with the variable x_0. The message carries the extrinsic probability of x_0 from node N_L to be used as intrinsic probability of x_0 at the node N_R, i.e.,

$$\mu_{L \to R}(x_0) = P_L^{ex}(x_0) = P_R^{in}(x_0). \tag{3.10}$$

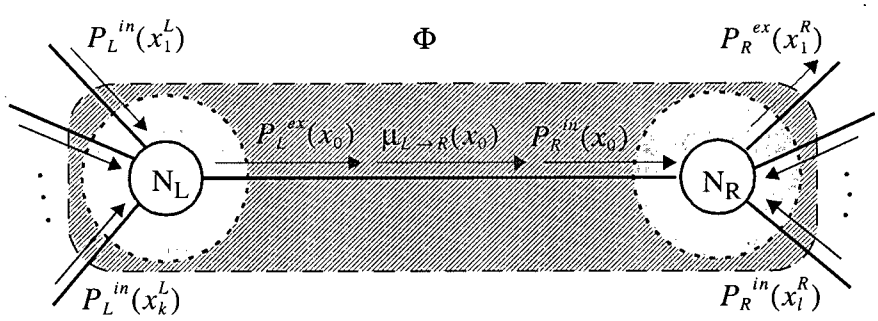


Figure 3.4 Message-passing for two nodes

By utilizing (3.9), $P_L^{ex}(x_0) = C_{x_0} \sum_{\substack{\sim\{x_0\} \\ (x_0, x_1^L, \ldots, x_k^L) \in S_L}} \prod_{j=1}^{k} P_L^{in}(x_j^L)$ and then $P_R^{ex}(x_i^R)$ for $i = 1, 2, \ldots, l$ can be computed as follows:

$$P_R^{ex}(x_i^R) = C_{x_i^R}' \cdot \sum_{\substack{\sim\{x_i^R\} \\ [(x_0, x_1^R, \ldots, x_l^R) \in S_R] \cap [(x_0, x_1^L, \ldots, x_k^L) \in S_L]}} \left(\prod_{j=1}^{k} P_L^{in}(x_j^L) \prod_{\substack{j=1 \\ j \neq i}}^{l} P_R^{in}(x_j^R) \right) \tag{3.11}$$

If all external variables $(x_1^L, \ldots, x_k^L, x_1^R, \ldots, x_l^R)$ are renamed as $(x_1, x_2, \ldots, x_{k+l})$, and if the symbol S_Φ is a constraint set for graph Φ in Figures 3.3 and 3.4, then constraint set S_Φ will satisfy (3.12) below:

$$[(x_1, x_2, \ldots, x_{k+l}) \in S_\Phi] = [(x_0, x_1^L, \ldots, x_k^L) \in S_L] \cap [(x_0, x_1^R, \ldots, x_l^R) \in S_R]. \tag{3.12}$$

Consider the expression of the extrinsic probability for x_i^L *w.r.t.* the graph Φ can also be derived similar as equation (3.11).

Combining (3.11) and (3.12), obtains a generalized expression of the extrinsic probability for x_i *w.r.t.* the graph Φ as follows:

$$P_\Phi{}^{ex}(x_i) = C_{x_i}' \cdot \sum_{\sim\{x_i\}} \sum_{(x_1, x_2, \ldots, x_{k+l}) \in S_\Phi} \left(\prod_{\substack{j=1 \\ j \neq i}}^{k+l} P_\Phi{}^{in}(x_j) \right) \text{ for } i = 1, 2, \ldots, k+l. \tag{3.13}$$

A. Graphs without cycles

If a graph is always divided into two disconnected sub graphs when cut at any edge, then the graph is a cycle-free graph.

Consider a graph Φ is divided into two subgraphs $\Phi_{\mathcal{L}}$ and $\Phi_{\mathcal{R}}$, which are connected by an internal edge x_i. Then take these two subgraphs as two 'nodes' shown in Figure 3.4 the extrinsic probabilities for the graph can be obtained by computing them for subgraphs $\Phi_{\mathcal{L}}$ and $\Phi_{\mathcal{R}}$ using (3.13).

Continuously, we can take each of these subgraphs in turn (e.g., $\Phi_{\mathcal{L}}$) and then choose an internal edge (e.g., $x_j^{\mathcal{L}}$) which connects two smaller graphs (e.g., $\Phi_{\mathcal{LL}}$ and $\Phi_{\mathcal{LR}}$). Computations for the appropriate messages passing in both directions along the internal edge $x_j^{\mathcal{L}}$ enable the extrinsic probabilities for the graph $\Phi_{\mathcal{L}}$ to be obtained by calculating the extrinsic probabilities for the two smaller graphs $\Phi_{\mathcal{LL}}$ and $\Phi_{\mathcal{LR}}$ using (3.13). By executing this scheme recursively, the

extrinsic probabilities for the graph Φ can be obtained by implementing calculations for each individual node and then for messages passing between the nodes. These conclusion for the cycle-free graphs can be also found in [33, 5, 11].

As a result, the message-passing algorithm can be summarized as follows:

Message-passing algorithm

- step1. **Initialization**. If any arbitrary node N in the graph Φ is connected to k nodes $N_1, N_2, \ldots, N_k$ through edges $x_1, x_2, \ldots, x_k$, then
  Input: the intrinsic probabilities from the edge x_j are $P_\Phi{}^{in}(x_j)$ for $j = 1, 2, \ldots, k$
  Output: the input message along the edge x_j are $\mu_{N_j \to N}(x_j) = P_\Phi{}^{in}(x_j)$
- step2. **Sum-product update**. The output message from node N to any node N_i via edge x_i, $\mu_{N \to N_i}(x_i)$ for $i = 1, 2, \ldots, k$, is obtained from the input messages $\mu_{N_j \to N}(x_j)$ along the all other $k-1$ edges ($j = 1, 2, \ldots, k$ and $j \neq i$), i.e.,

$$\mu_{N \to N_i}(x_i) = C_{x_i} \sum_{\substack{\sim\{x_i\} \\ (x_1, x_2, \ldots, x_k) \in S_N}} \prod_{\substack{j=1 \\ j \neq i}}^{k} \mu_{N_j \to N}(x_j) \quad \text{for all } i = 1, 2, \ldots, k. \tag{3.14}$$

  Equation (3.14) applies to all nodes in the graph.
- step3. **Message-passing**. For any node in the graph, the output message from the node passed over each edge will become the input message for the receiving node.
- step4. **Repeat step2 and step3**. Repeat step2 and step3 for all nodes in the graph in terms of the message-passing schedule until the stopping criterion has been reached.

Since the expression (3.14) consist of the sum of the products, it is commonly called the sum-product update rule. The sum-product algorithm has been used for the message-passing algorithm by many authors [14, 18]. The message-passing schedule determines the order in which the messages are passed. The stopping criterion is used to determine whether to stop passing messages and declare that decoding is complete or failed.

By utilizing this algorithm we can obtain the extrinsic probabilities for the external edges.

For some applications, the final required information may be the posterior probabilities, which can be derived by calculating the products of the intrinsic and extrinsic probabilities for the external edges using equation (3.8).

B. Graphs with cycles

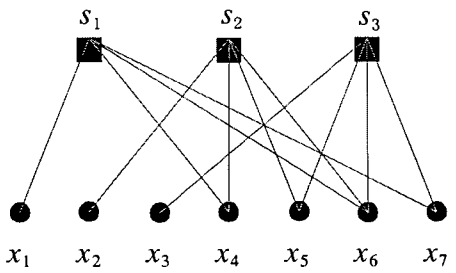
A graph is said to have cycles if it is possible to trace a path from a node back to itself by moving along the edges of the graph in only one direction. The length of a cycle is the number of edges along the cycle's path.

Cycles are common in many applications. If a graph includes cycles the assumption of independence for variables is invalidated, and the message-passing algorithm can only be used as an estimation approach without the guarantee of convergence. Fortunately, it has been found that for many cases of graphs with cycles, such as in the techniques of turbo and LDPC codes [13, 14, 20, 22, 30], the message-passing algorithm creates accurate estimates for the desired probabilities with much lower computation and complexity than an exact decoding. However, short cycles (e.g., of length 4) result in violation of the independence assumptions. In other words, the larger the cycle lengths the better performance of the algorithm.

3.3.4 Binary block codes

Any (n, k) binary block code with k message bits and $n-k$ check bits may be expressed by a bipartite graph. We assume a graph with n left bit nodes $\mathbf{x}$ (also called message nodes) and $n-k$ right check nodes $\mathbf{s}$ (also called parity-check nodes). Figure 3.5 gives an example of a bipartite graph for a $(7, 4)$ Hamming code with parity-check matrix $\mathbf{H}$.

In Figure 3.5 there are two basic types of nodes, the bit nodes $\boldsymbol{x}$ and check nodes $\mathbf{s}$. According to the schemes of normal graphs in previous sections, and because the bit node and check node have different types of constraint sets, we will discuss the normal graph message-passing rules for each.



$$\begin{cases} s_1 = x_1 + x_4 + x_6 + x_7 \\ s_2 = x_2 + x_4 + x_5 + x_6 \\ s_3 = x_3 + x_5 + x_6 + x_7 \end{cases}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Figure 3.5 A bipartite graph for a $(7, 4)$ Hamming code

A. Bit node

We use the model of a one node normal graph (e.g., Figure 3.2) to analyze a bit node B. The constraint set for bit node B is $S_B = \{(x_0, x_1, \ldots, x_k) | (x_0 = x_1 = \cdots = x_k)\}$ where variables $x_0, x_1, \ldots, x_k$ belong to the same alphabet A. Figure 3.6 shows a normal graph with a bit node. By using (3.9) the output along any edge-variable x_i can be derived as

$$\mu_{B \to N_i}(x_i = \zeta) = C_{x_i} \prod_{\substack{j=0 \\ j \neq i}}^{k} \mu_{N_j \to B}(x_j = \zeta) \text{ for } i = 0, 1, \ldots, k \tag{3.15}$$

where C_{x_i} is the normalization constant.

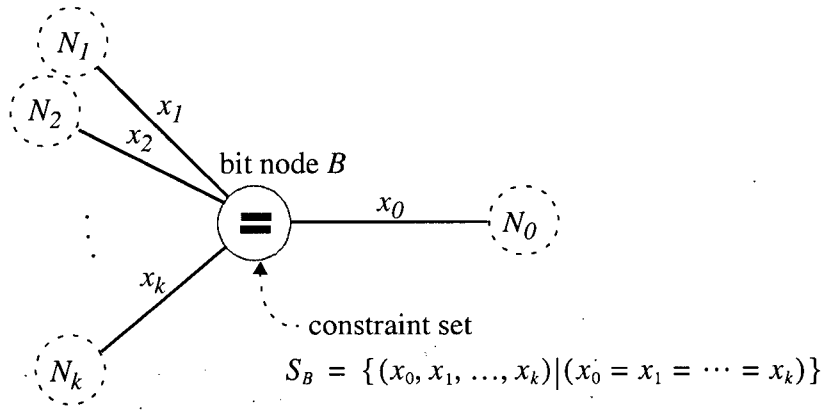


Figure 3.6 A normal graph with a bit node

Comparing (3.9) with (3.15), one sees that (3.15) simplifies the sum of products to a product of the intrinsic probabilities.

B. Check node

In any check node, a local constraint set specifies that all connected edge-variables sum to zero. If we use the same model of one node normal graph (e.g., Figure 3.2) to analyze the check node C and denote $\oplus$ as addition over GF_2, then the constraint set for the check node C is $S_c = \{(x_0, x_1, \ldots, x_k) | (x_0 \oplus x_1 \oplus \ldots \oplus x_k) = 0\}$ where variables $x_0, x_1, \ldots, x_k$ belong to the same alphabet A. A normal graph with a check node is shown in Figure 3.7, then the output along any edge-variable x_i is given by (3.16) below:

$$\mu_{C \to N_i}(x_i) = C_{x_i} \sum_{\substack{\sim\{x_i\} \\ (x_0, x_1, \ldots, x_k) \in S_C}} \prod_{\substack{j=0 \\ j \neq i}}^{k} \mu_{N_j \to C}(x_j) \tag{3.16}$$

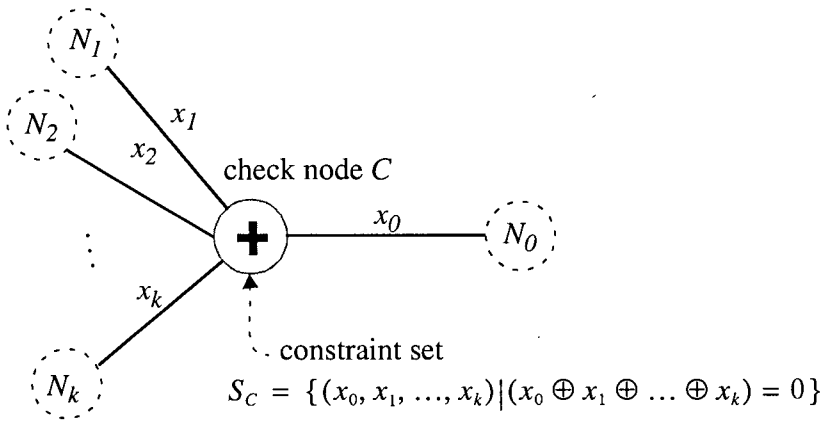


Figure 3.7 A normal graph with a check node

For binary block codes, $A = \{0, 1\}$. For a check node, the constraint set S_C constitutes 2^k configurations for $k+1$ variables. In equation (3.16), for a fixed x_i the summation is over 2^{k-1} elements so that the computation appears complicated. Fortunately, we can find a way to

simplify computation by using the binary additive law and induction rule.

Suppose the input messages to check node C correspond to the intrinsic probabilities defined as $p_i = P^{in}(x_i = 1)$; then $\begin{cases} \mu_{N_i \to C}(x_i = 1) = p_i \\ \mu_{N_i \to C}(x_i = 0) = 1-p_i \end{cases}$. If we choose $i = 0$ to investigate the solution it will have similar generalized results. Therefore, the output message along edge-variables x_0 is the extrinsic probability *w.r.t.* C, i.e.,

$$\begin{cases} \mu_{C \to N_0}(x_0 = 1) = P(x_1 \oplus \cdots \oplus x_k = 1) \\ \mu_{C \to N_0}(x_0 = 0) = P(x_1 \oplus \cdots \oplus x_k = 0) \end{cases}. \tag{3.17}$$

To calculate (3.17), we first consider the value of $P(x_1 \oplus \cdots \oplus x_k = 0)$ as follows:

- For $k = 2$, we have $P(x_1 \oplus x_2 = 0) = p_1p_2 + (1-p_1)(1-p_2)$; then
$$2P(x_1 \oplus x_2 = 0) - 1 = (1-2p_1)(1-2p_2) \tag{3.18}$$
- Assume the equation (3.18) can be generalized to the case when $k = K-1$ for $K \geq 3$,
$$2P(x_1 \oplus x_2 \oplus \cdots \oplus x_{K-1} = 0) - 1 = \prod_{i=1}^{K-1}(1-2p_i) \tag{3.19}$$
- Under the hypothesis of equation (3.19), when $k = K$,
let $X_K = x_1 \oplus x_2 \oplus \cdots \oplus x_K$ and $X_{K-1} = x_1 \oplus x_2 \oplus \cdots \oplus x_{K-1}$ then
$2P(X_K = 0) - 1 = 2P(X_{K-1} \oplus x_K = 0) - 1 = [2P(X_{K-1} = 0) - 1][1-2p_k]$.
- By using (3.19) we obtain
$$2P(x_1 \oplus x_2 \oplus \cdots \oplus x_K = 0) - 1 = \prod_{i=1}^{K}(1-2p_i). \tag{3.20}$$
- According to induction rules, the expression (3.20) is valid for all $K \geq 2$. Furthermore,
$$P(x_1 \oplus x_2 \oplus \cdots \oplus x_k = 0) = \frac{1}{2}\left(1 + \prod_{i=1}^{k}(1-2p_i)\right). \tag{3.21}$$

Substitution of (3.21) to (3.17) yields

$$\mu_{C\to N_0}(x_0 = 0) = \frac{1}{2}\left(1 + \prod_{i=1}^{k}(1-2p_i)\right) \text{ and} \tag{3.22}$$

$$\mu_{C\to N_0}(x_0 = 1) = \frac{1}{2}\left(1 - \prod_{i=1}^{k}(1-2p_i)\right) \tag{3.23}$$

where $p_i = \mu_{N_i\to C}(x_i = 1)$ are the input message probabilities.

Equations (3.22) and (3.23) represent simplified update version of (3.16) when $i = 0$. They can be generalized to common expressions for any $i = 0, 1, \ldots, k$ as follows:

$$\mu_{C\to N_i}(x_i = 0) = \frac{1}{2}\left(1 + \prod_{\substack{j=0 \\ j\neq i}}^{k}(1-2p_j)\right); \tag{3.24}$$

$$\mu_{C\to N_i}(x_i = 1) = \frac{1}{2}\left(1 - \prod_{\substack{j=0 \\ j\neq i}}^{k}(1-2p_j)\right). \tag{3.25}$$

3.3.5 LDPC codes

LDPC codes are a specific example of linear block codes. By using the update equations for the bit nodes and check nodes described in section 3.3.4, we can exploit decoding on the graph for an LDPC code, as shown in Figure 3.8.

A parity matrix **H** has n columns and m rows; therefore assume that the bit nodes are B_i for $i = 1, 2, \ldots, n$ and the check nodes are C_j for $j = 1, 2, \ldots, m$. If there is an internal edge between the nodes B_i and C_j, then the edge is labeled by the edge-variable $u_{i,j}$. For an (n, ρ, γ) LDPC code, there are $\rho\gamma$ internal edges between those nodes in the graph. In addition, there is an

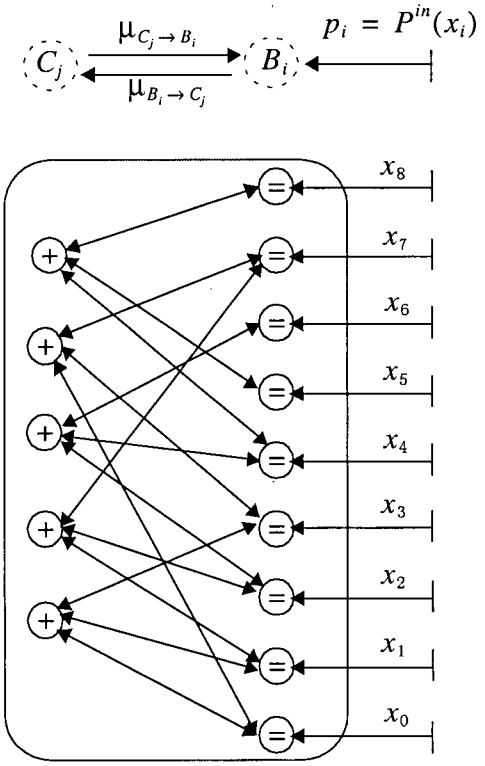


Figure 3.8 A normal graph for an LDPC decoder

external edge for each bit node B_i that is associated with the variable x_i. The intrinsic probability *w.r.t.* bit nodes B_i is denoted as $p_i = P^{in}(x_i)$.

If we denote $\alpha(i)$ as the set of parity checks in which the bit node B_i is involved and $\beta(j)$ as the set of bit nodes connected to the j-th check node C_j, then $\alpha(i)$ is also the set of row locations in the i-th column of the parity-check matrix that contain a "1", and $\beta(j)$ is also the set of columns in the j-th row that contain a "1".

A. Probabilities representations

The message-passing algorithm starts with a set of incoming probabilities $p_i = P^{in}(x_i)$. In term of (3.15), the message from bit node B_i to check node C_j is obtained as follows:

$$\mu_{B_i \to C_j}(u_{i,j} = \zeta) = c_{ij} P^{in}(x_i = \zeta) \cdot \prod_{j' \in \alpha(i) \setminus \{j\}} \mu_{C_{j'} \to B_i}(u_{i,j'} = \zeta) \text{ for } \zeta = 0, 1 \tag{3.26}$$

where $\alpha(i) \setminus \{j\}$ means the set $\alpha(i)$ except j, and c_{ij} is the normalization constant.

According to (3.24) and (3.25), the message from check node C_j to bit node B_i is given as

$$\mu_{C_j \to B_i}(u_{i,j} = 0) = \frac{1}{2}\left(1 + \prod_{i' \in \beta(j) \setminus \{i\}} [1 - 2\mu_{B_{i'} \to C_j}(u_{i',j} = 1)]\right); \tag{3.27}$$

$$\mu_{C_j \to B_i}(u_{i,j} = 1) = \frac{1}{2}\left(1 - \prod_{i' \in \beta(j) \setminus \{i\}} [1 - 2\mu_{B_{i'} \to C_j}(u_{i',j} = 1)]\right). \tag{3.28}$$

The message passed from bit node B_i to check node C_j is the probability that B_i has a certain value given the observed value of that bit node, and all the values communicated to B_i in the prior round from check nodes other than C_j incident to B_i. On the other hand, the message passed from C_j to B_i is the probability that B_i has a certain value given all the messages passed to C_j in the previous round from bit nodes other than B_i.

Equations (3.26), (3.27), and (3.28) express the message-passing algorithm for the LDPC code. We use some representations for the related variables as follows where $b = 0, 1$:

$$\begin{cases} p_i^b = P^{in}(x_i = b) \\ q_{ij}^b = \mu_{B_i \to C_j}(u_{i,j} = b) \\ s_{ij}^b = \mu_{C_j \to B_i}(u_{i,j} = b) \\ q_i^b = P^{post}(x_i = b) \end{cases}$$

We can draw message-passing flows for LDPC codes as shown in Figure 3.9. The update equation (3.26) for a message from bit node B_i to check node C_j can be rewritten as

$$q_{ij}^b = c_{ij} p_i^b \prod_{j' \in \alpha(i) \setminus \{j\}} s_{ij'}^b \text{ for } b = 0, 1. \tag{3.29}$$

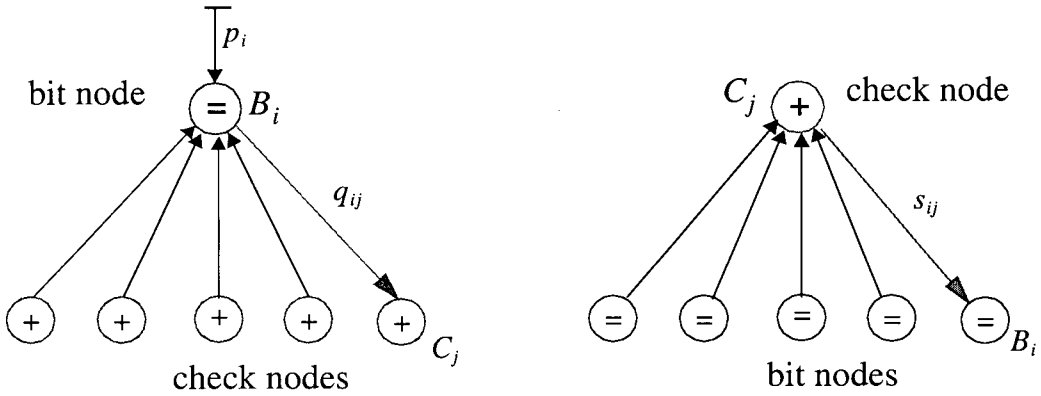


Figure 3.9 Message-passing flows for an LDPC decoder

The update equations (3.27) and (3.28) for the message from check node C_j to bit node B_i can be rewritten as follows:

$$s_{ij}^{0} = \frac{1}{2}\left(1 + \prod_{i' \in \beta(j)\setminus\{i\}} \delta q_{i'j}\right) \tag{3.30}$$

$$s_{ij}^{1} = \frac{1}{2}\left(1 - \prod_{i' \in \beta(j)\setminus\{i\}} \delta q_{i'j}\right) \tag{3.31}$$

where $\delta q_{ij} = q_{ij}^{0} - q_{ij}^{1} = 1 - 2q_{ij}^{1}$.

Combining equations (3.29) and (3.8), the posterior probabilities are provided as follows:

$$q_i^b = c_i \, p_i^b \prod_{j' \in \alpha(i)} s_{ij'}^b \quad \text{for } b = 0, 1. \tag{3.32}$$

After a number of iterations of the message-passing algorithm, it is expected that these estimate of the posterior probabilities converge to the ideal posterior probabilities *w.r.t.* the given LDPC code.

For binary LDPC codes, it is sometimes advantageous to work with likelihoods, or even log-likelihoods instead of probabilities.

B. Likelihood representations

For a binary random variable u let $L(p)$ in (3.33) be the likelihood ratio of u;

$$L(p) = \frac{P(u=0)}{P(u=1)} = \frac{1-p}{p}, \tag{3.33}$$

where $p = P(u=1)$ for $0 \le p \le 1$. Applying (3.29) to (3.33) yields

$$L(q_{ij}) = L(p_i) \cdot \prod_{j' \in \alpha(i) \setminus \{j\}} L(s_{ij'}). \tag{3.34}$$

Combining (3.30), (3.31)and (3.32) gives

$$L(s_{ij}) = \frac{1 + \prod_{i' \in \beta(j) \setminus \{i\}} \delta q_{i'j}}{1 - \prod_{i' \in \beta(j) \setminus \{i\}} \delta q_{i'j}}; \tag{3.35}$$

$$L(q_i) = L(p_i) \cdot \prod_{j' \in \alpha(i)} L(s_{ij'}). \tag{3.36}$$

C. Log-likelihood representations

Define the log-likelihood ratio of u as $LLR(p)$; then

$$LLR(p) = \ln\frac{P(u=0)}{P(u=1)} = \ln\frac{1-p}{p}. \tag{3.37}$$

Application of (3.29) to (3.37) yields

$$LLR(q_{ij}) = LLR(p_i) + \sum_{j' \in \alpha(i) \setminus \{j\}} LLR(s_{ij'}). \tag{3.38}$$

From (3.37), by using $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and $2\operatorname{atanh}(x) = \ln\left(\frac{1+x}{1-x}\right)$ for $|x| < 1$, we

obtain that $1-2p = \tanh\left(\frac{1}{2}LLR(p)\right)$. Thus substitution of (3.30)-(3.32) into (3.38) yields

$$LLR(s_{ij}) = 2\,\mathrm{atanh}\left[\prod_{i' \in \beta(j)\setminus\{i\}} \tanh\left(\frac{1}{2}LLR(q_{i'j})\right)\right]; \tag{3.39}$$

$$LLR(q_i) = LLR(p_i) + \sum_{j' \in \alpha(i)} LLR(s_{ij'}). \tag{3.40}$$

In comparing these three representations for the message-passing algorithm, one sees that the log-likelihood representations as shown in equations (3.38)-(3.40) simplify the computation considerably. Equation (3.40) recalls that the intrinsic, extrinsic and posterior LLRs *w.r.t.* the LDPC decoder satisfy the equation (3.8). Based on this posterior probability $LLR(q_i)$, bit decisions can be made and a decoded word $\hat{\mathbf{x}}$ is then able to be found.

The parity-check equations ($\mathbf{H}\hat{\mathbf{x}}^{\mathrm{T}} = 0$) are used to evaluate the decoded word. If all parity checks are satisfied, then the iteration stops; otherwise, it continues to exploit the updated message-passing procedures until all parity checks are satisfied or the iteration counter reaches its limit and declares a decoding failure.

If the multiplications in (3.39) can be replaced by additions, then the complexity of implementation for this algorithm will be greatly reduced.

Any product of real numbers can be expressed as follows:

$$\prod_i x_i = \left(\prod_i \mathrm{sgn}(x_i)\right) \cdot \exp\left(\sum_i \ln|x_i|\right). \tag{3.41}$$

Assume that for any $x > 0$, a function $\Gamma(x)$ is defined as follows:

$$\Gamma(x) = -\ln[\tanh(x/2)] = \ln\frac{e^x+1}{e^x-1}. \tag{3.42}$$

Then

$$\Gamma(x) = \Gamma^{-1}(x) \text{ for } x > 0. \tag{3.43}$$

By using (3.41)-(3.42), equation (3.39) can be written as follows:

$$LLR(s_{ij}) = \left(\prod_{i' \in \beta(j) \setminus \{i\}} \operatorname{sgn}(LLR(q_{i'j})) \right) \cdot \Gamma\left(\sum_{i' \in \beta(j) \setminus \{i\}} \Gamma(|LLR(q_{i'j})|) \right), \tag{3.44}$$

which consists of sign and amplitude parts. The sign part is given by $\prod_{i' \in \beta(j) \setminus \{i\}} \operatorname{sgn}(LLR(q_{i'j}))$. The amplitude is determined from $\Gamma\left(\sum_{i' \in \beta(j) \setminus \{i\}} \Gamma(|LLR(q_{i'j})|) \right)$, which is dominated by a large value in the summation, i.e., $\Gamma\left(\sum_{i' \in \beta(j) \setminus \{i\}} \Gamma(|LLR(q_{i'j})|) \right) \approx \Gamma\left(\max_{i' \in \beta(j) \setminus \{i\}} (\Gamma(|LLR(q_{i'j})|)) \right)$. Note that the function $\Gamma(x)$ is a monotonically decreasing for $x > 0$, such that $\Gamma\left(\sum_{i' \in \beta(j) \setminus \{i\}} \Gamma(|LLR(q_{i'j})|) \right) \approx \Gamma\left[\Gamma\left(\min_{i' \in \beta(j) \setminus \{i\}} (|LLR(q_{i'j})|) \right) \right]$.

Using (3.43), yields that $\Gamma(\Gamma(x)) = x$, thus (3.44) can be rewritten as

$$LLR(s_{ij}) = \left(\prod_{i' \in \beta(j) \setminus \{i\}} \operatorname{sgn}(LLR(q_{i'j})) \right) \cdot \min_{i' \in \beta(j) \setminus \{i\}} (|LLR(q_{i'j})|). \tag{3.45}$$

By using these "min" approximations, the decoding computation is reduced significantly with only a small degradation in performance. Such reduction in complexity is useful as LDPC codes find their way numerous applications, including for wireless communications and data storage channels.

The unique advantages of these "min" approximations for decoding LDPC codes for DSL transmission are described in Chapter 4.

Chapter 4 LDPC Codes in DSL Systems

Based on the structure of DMT systems introduced in Chapter 2, and the concept of LDPC codes and the iterative message-passing algorithm for their decoding discussed in Chapter 3, we now deal with the issues related to the implementation of LDPC codes in DSL systems. Three aspects are emphasized: the construction of LDPC codes, the encoding process, and the decoding process.

4.1 Constructing an LDPC Code

When the LDPC codes are designed properly, they can outperform conventional *error-correction codes* (ECC), such as turbo-codes. For example very long LDPC codes which are within 0.0045dB of the Shannon limit have been constructed by Chung [7]. The method was proposed by Gallager [14] to define a binary (n, ρ, γ) LDPC code by an $(m \times n)$ sparse parity-check matrix **H** built by ρ sub-matrices of size $\left(\frac{m}{\rho}\right) \times n$, where m must be a multiple of ρ.

Based on Gallager's work, MacKay [20] constructed binary LDPC codes without 4-cycles by randomly generating columns, each of which contains exactly ρ ones. These random constructions provide LDPC codes with reasonable distance properties. One criticism of randomly generated LDPC codes is that the complexity of the encoding process for such codes is quadratic in block length. Richardson *et al.* [27] have discussed the design of LDPC codes using a low-complexity encoding process.

Alternative constructions are deterministic, which can be easy to describe using a small number of parameters. Eleftheriou et al. [9] proposed a scheme for deterministic construction of LDPC codes which relies on "array codes" [10]. Array codes are ECCs that have the capability to detect and correct burst errors using an algebraic decoder. These binary array codes have sparse parity-check matrices, and they are decodable as LDPCs, making them suitable for use in soft

iterative decoding schemes. Therefore, binary array codes provide the framework for deterministically constructing a class of LDPC codes.

Let us define a regular LDPC code by three parameters: a prime number p and two integers ρ and γ where $\gamma > \rho$ such that $\rho, \gamma \leq p$. Let $\mathbf{H}_1$ be the $(p\rho \times p\gamma)$ matrix that is defined as follows:

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \alpha^{\gamma-1} & \mathbf{I} & \alpha & \cdots & \alpha^{\rho-2} & \alpha^{\rho-1} & \cdots & \alpha^{\gamma-2} \\ \alpha^{2(\gamma-2)} & \alpha^{2(\gamma-1)} & \mathbf{I} & \cdots & \alpha^{2(\rho-3)} & \alpha^{2(\rho-2)} & \cdots & \alpha^{2(\gamma-3)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \alpha^{(\rho-1)(\gamma-\rho+1)} & \alpha^{(\rho-1)(\gamma-\rho+2)} & \cdots & \cdots & \mathbf{I} & \alpha^{\rho-1} & \cdots & \alpha^{(\rho-1)(\gamma-\rho)} \end{bmatrix} \tag{4.1}$$

where $\mathbf{I}$ is the $(p \times p)$ identity matrix and α is a $(p \times p)$ permutation matrix indicating a single left/right cyclic shift, e.g., when $p = 5$, $\alpha = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ or $\alpha = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$.

The parameter ρ and γ give the column weight and row weight of $\mathbf{H}_1$, respectively. Since there are no two rows that have overlapping ones in more than one position, matrix $\mathbf{H}_1$ is a sparse 4-cycle free parity-check matrix which can be decoded utilizing the message-passing algorithm. If we triangulize the matrix $\mathbf{H}_1$ simply by replacing the lower-triangular elements of its $(p\rho \times p\rho)$ leftmost sub-block (dashed in (4.1)) by zeros. The resulting matrix $\mathbf{H}$ is created as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \Theta & \mathbf{I} & \alpha & \alpha^2 & \cdots & \alpha^{\rho-2} & \alpha^{\rho-1} & \cdots & \alpha^{(\gamma-2)} \\ \Theta & \Theta & \mathbf{I} & \alpha^2 & \cdots & \alpha^{2(\rho-3)} & \alpha^{2(\rho-2)} & \cdots & \alpha^{2(\gamma-3)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Theta & \Theta & \Theta & \cdots & \Theta & \mathbf{I} & \alpha^{\rho-1} & \cdots & \alpha^{(\rho-1)(\gamma-\rho)} \end{bmatrix} \tag{4.2}$$

where Θ is the $(p \times p)$ null matrix.

The LDPC codes defined by **H** in (4.2) are irregular LDPC codes since both their row weights and column weights in **H** are not uniform. These codes have codeword length of $n = p\gamma$, number of parity checks $m = p\rho$ and information block length $k = p(\gamma - \rho)$. An LDPC code with $n' < n$ is simply obtained by discarding the $n - n'$ rightmost columns of **H**. Then efficient encoding can be performed from H without computing the generator matrix of the code. An n-tuple **x** is an LDPC code if and only if $\mathbf{H} \cdot \mathbf{x}^T = \mathbf{0}$, where **0** is a $p\rho \times 1$ null vector. Let the code vector be $\mathbf{x}^T = \begin{bmatrix} \mathbf{p} \\ \mathbf{s} \end{bmatrix}$ where the $p\rho \times 1$ vector **p** describes the parity part and $p(\gamma - \rho) \times 1$ vector **s** describes the systematic information part of code **x**. By utilizing advantages of the triangular structure and sparse density of the parity-check matrix **H** these LDPC codes are fast encodable. We propose to adopt this construct of LDPC codes in our simulations.

4.2 Encoding process for DMT systems

As described in standard [1] of DMT-ADSL and a draft trial-use standard [8] of VDSL, the DSL systems impose the data into physical frames and superframe structure. Each frame is encoded in DMT as a single DMT symbol. The number of bits contained in each DMT symbol is determined during initialization and held unchanged at that value until the next initialization is implemented.

The length of a DSL frame is determined by the adapted bit rate of the interface [29]. DMT symbols are encoded at the rate of 4000 *bps*, i.e., one complete transition per 250 *ms*. The actual amount of data that can be encoded in one transition is variable, depending on the line conditions (such as SNRs) which are found at initialization, the number of tones supported in the channel and the amount of data assigned to each tone. Tones located in noisy areas of the spectrum are suppressed, and the complexity of the symbols' constellations encoded in each tone needs to be set to optimize transmission over the channel. Each frame contains the data of one DMT symbol at one time. The data contained in each frame can be from both the fast data and

interleaved data buffers, and can include as well as overhead bits used for error correction, administration and management of the DSL link [1, 3, 8, 29].

4.2.1 LDPC encoding and constellation mapping

In DSL systems, the data frame buffer consists of the fast and interleaved data buffer. Eleftheriou et al. [9] proposed a structure for LDPC encoding and symbol mapping which is redrawn in Figure 4.1. The data frame buffer is split into two parts to be denoted as buffers ψ_u and ψ_c. Before symbol mapping the bits assigned to buffer ψ_c will be LDPC encoded, while the bits assigned into buffer ψ_u will remain uncoded. If the fast buffer is not present, the LSB of the interleaved data buffer will be assigned to the LSB of the buffer ψ_c. Each codeword with k bits from buffer ψ_c is input to the LDPC encoder and the output of encoder is a codeword with n bits.

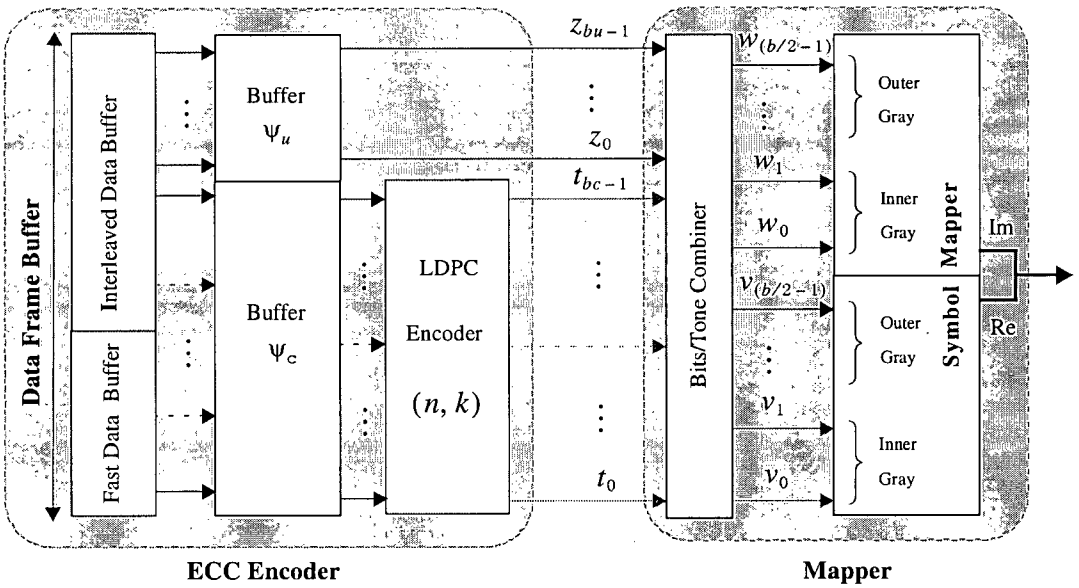


Figure 4.1 Overall LDPC encoding and symbol mapping

By using bit loading schemes, a total of b bits for each specific tone i is extracted from the buffer ψ_u and from the output of LDPC encoder as follows:

- bc bits are extracted from LDPC encoder output and denoted by $(t_{bc-1}, \ldots, t_0)$;

- $bu = b - bc$ bits are extracted from buffer ψ_u and denoted by $(z_{bu-1}, \ldots, z_0)$,

where b can be either 1 or a even integer between 0 and 14 inclusive, and

1. when $b = 2, 4, 6, 8, 10, 12, 14$,
   - $(t_{bc-1}, \ldots, t_0)$ is partitioned as $\langle t_{bc-1}, \ldots, t_{bc_v} \mid t_{bc_v-1}, \ldots, t_0 \rangle = \langle \dot{t}_w \mid \dot{t}_v \rangle$, and
   - $(z_{bu-1}, \ldots, z_0) = \langle z_{bu-1}, \ldots, z_{(b/2)-bc_v} \mid z_{(b/2)-bc_v-1}, \ldots, z_0 \rangle = \langle \bar{z}_w \mid \bar{z}_v \rangle$. Then the extracted $b = bc + bu$ bits are input to the "Bits/Tone Combiner". The combiner creates two binary $b/2$-tuples $\mathbf{v} = (v_{(b/2-1)}, \ldots, v_0) = \langle \bar{z}_v \mid \dot{t}_v \rangle$ and $\mathbf{w} = (w_{(b/2-1)}, \ldots, w_0) = \langle \bar{z}_w \mid \dot{t}_w \rangle$ that are inputted to the "symbol mapper".
   - $bc = bc_v + bc_w$, and usually $bc_v = bc_w$
2. when $b = 1$, a bit t_0 is extracted and assigned to $\mathbf{v} = (v_0) = (t_0)$, and word $\mathbf{w}$ is not used.
3. when $b = 0$, no information bits are assigned to the corresponding tone (subchannel).

For the first case, b is non-zero and even. Let $L = 2^{b/2}$, then the two binary words $\mathbf{v}$ and $\mathbf{w}$ are independently assigned to two L-ary real valued symbols (PAM symbols) that are mapped to the real and imaginary components of the complex QAM symbol, respectively. Each L-ary real PAM symbol is included in the set:

$$\mathbf{C} = \{C_l = 2l - L + 1\} \text{ for } l = 0, 1, \ldots, L-1. \tag{4.3}$$

Set $\mathbf{C}$ is effectively partitioned into 2^{bc_v} and 2^{bc_w} subsets, such that the minimum Euclidean distance between the symbols in each subset is maximized. The bc_v LSBs of $\mathbf{v}$ and bc_w LSBs of $\mathbf{w}$, which are coded by LDPC coder, label the subsets of $\mathbf{C}$ according to a Gray coding rule ("inner Gray" as shown in Figure 4.1). The remaining MSBs, which are uncoded, label symbols within a subset following a separate Gray coding rule ("outer Gray").

The real and imaginary components of the complex symbol to be transmitted are generated by this "double Gray-coded mapper".

If the length of an LDPC codeword is more than the length of one DMT symbol, then the buffers ψ_u and ψ_c correspond to the contents of more than one data frame buffer. In other words,

the "fast data buffer" in Figure 4.1 indicates the contents of successive fast data buffers, and "interleaved data buffer" indicates the contents of successive interleaved data buffer.

4.3 Decoding process for DMT systems

In conventional DSL systems, bits are mapped to constellation points in the frequency domain. Each received DMT symbol in the frequency domain consists of a frame of complex numbers that are associated with all DMT subchannels. According to the definition of the QAM constellation [1] used for signalling in each subchannel, we can select the constellation point closest in Euclidian distance to the received signal in each subchannel. Then, based on the constellation point being chosen the bit sequence can be determined by demapper. The detailed procedure has been described in Section 2.2.2.

For LDPC decoding schemes in DSL, the decoder deploys the iterative message-passing algorithm based on a normal graph. From the received signals the probabilistic information for each bit of the bit sequence of the DMT symbol is extracted as the input message for the decoder.

In our work, double Gray mapping is used for multilevel constellation mapping in the LDPC encoder. The LSBs labeled by "inner Gray" in Figure 4.1, which are all coded by LDPC, can be reliably decoded by utilizing the iterative message-passing algorithm on a normal graph. The uncoded MSBs are produced by simply applying a maximum likelihood estimate based on given values of corresponding LSBs.

The study of LDPC codes in this thesis is applied mainly in DSL systems over two different channel detectors. The one is the ideal (flat) AWGN channel detector, the other is the frequency-selective AWGN channel detector. In either case, we utilize the outputs from channel detector as the initial message inputs to perform message-passing algorithm for LDPC decoding.

4.3.1 Ideal AWGN channel

For an ideal AWGN channel, there is no attenuation distortion. A system with coding, mapping, and an ideal AWGN channel is shown in Figure 4.2, the LDPC encoder takes a message bit sequence $\mathbf{u}$ to an encoded bit sequence $\mathbf{v}$; then $\mathbf{v}$ is mapped to a sequence of symbols $\mathbf{x}$, for

transmission to the channel. The received sequence of symbols **y** will be decoded by a channel detector. Following our discussion in Section 3.3.3, for any edge between nodes, the extrinsic probability *w.r.t.* one node gives the intrinsic probability *w.r.t.* the other node. In Figure 4.2, the message passing schemes are applied between the three nodes, i.e., the channel detector N_C, a demapper N_D and LDPC decoder N_L.

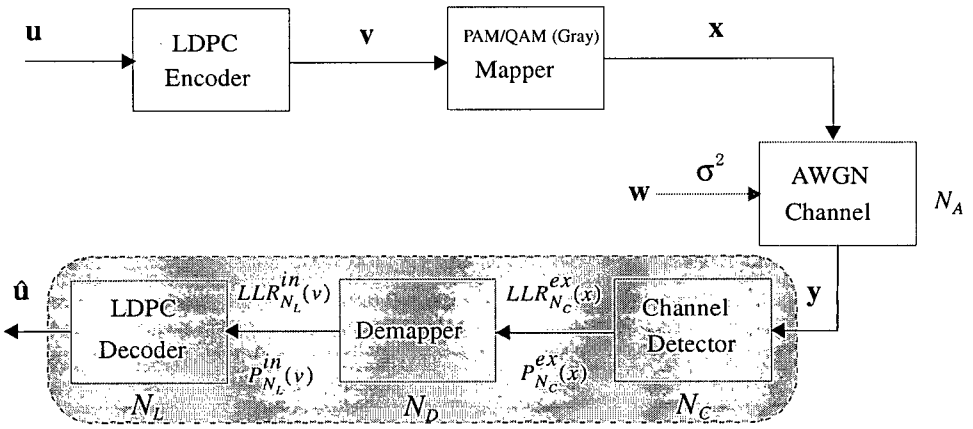


Figure 4.2 Diagram of a system with coding, mapping over an ideal AWGN channel

Let us consider a transmitted DMT symbol, which corresponds to 512 uniformly spaced time samples without a cyclic prefix. Therefore, each element of transmitted signal vector **x** is expressed as $x[i]$ for $i = 0, \ldots, 511$. When this transmitted symbol passes through the AWGN channel, the noise signal vector **w** is added to the transmitted signal, where vector **w** is an *i.i.d.* vector of AWGN variables. Thus, each element of **w**, denoted as $w[i]$ for $i = 0, \ldots, 511$, is Gaussian noise with zero-mean and variance of σ^2.

Suppose **y** is the received signal vector, then we have $\mathbf{y} = \mathbf{x} + \mathbf{w}$ and

$$y[i] = x[i] + w[i] \text{ for } i = 0, \ldots, 511. \tag{4.4}$$

In frequency domain, by converting (4.4) via DFT we can obtain

$$W_k = Y_k - X_k \text{ for } k = 0, \ldots, 511, \tag{4.5}$$

where $X_k = \frac{1}{\sqrt{512}}\sum_{i=0}^{511} x[i]\exp\left(-j\frac{2\pi ik}{512}\right)$, $Y_k = \frac{1}{\sqrt{512}}\sum_{i=0}^{511} y[i]\exp\left(-j\frac{2\pi ik}{512}\right)$ and

$$W_k = \sum_{i=0}^{511}\frac{w[i]}{\sqrt{512}}\exp\left(-j\frac{2\pi ik}{512}\right) = \sum_{i=0}^{511}\frac{w[i]}{\sqrt{512}}\cos\left(\frac{2\pi ik}{512}\right) - j\sum_{i=0}^{511}\frac{w[i]}{\sqrt{512}}\sin\left(\frac{2\pi ik}{512}\right). \quad (4.6)$$

If $Y_k = Y_k^R + jY_k^I$ is the received symbol associated to the k-th subchannel, then from (4.5) and (4.6) one concludes that all elements of the noise in frequency domain are also complex *i.i.d.* Gaussian variables. Each pair of real and imaginary components of the noise in each subchannel are independent of each other, and all has zero mean and variance of $\sigma_k^2 = \sigma^2/2$. Our purpose is to evaluate the probabilities of every bit in the coded binary tuples assigned to each subchannel.

A. QAM constellation

We can use a 2^b-QAM modulation scheme in each subchannel, in which case each subchannel is assigned exactly b bits (called binary b-tuples). For example, when $b = 4$, 16-QAM results Figure 4.3 shows signal points used in the calculation of the a posteriori probabilities for the associated bits [1].

Let the binary 4-tuples associated with the 16-QAM constellation points be labeled $B_i = (b_3, b_2, b_1, b_0)$ for $i = 0, 1, \ldots, 15$. We would like to compute $P(b_i = 0|Y_k = \zeta_k)$ and $P(b_i = 1|Y_k = \zeta_k)$ for $i = 0, \ldots, 3$, where ζ_k is a received complex value.

Regarding bit b_0, $P(b_0 = 0|Y_k = \zeta_k) = \sum_{\sim(b_0 = 1)} P(b_3, b_2, b_1, b_0|Y_k = \zeta_k)$, i.e.,

$$P(b_0 = 0|Y_k = \zeta_k) = \sum_{\substack{B_i|b_0 = 0 \\ i = 0, \ldots, 15}} P(B_i|Y_k = \zeta_k). \quad (4.7)$$

Assume that all constellation points are transmitted with equal probabilities; by using

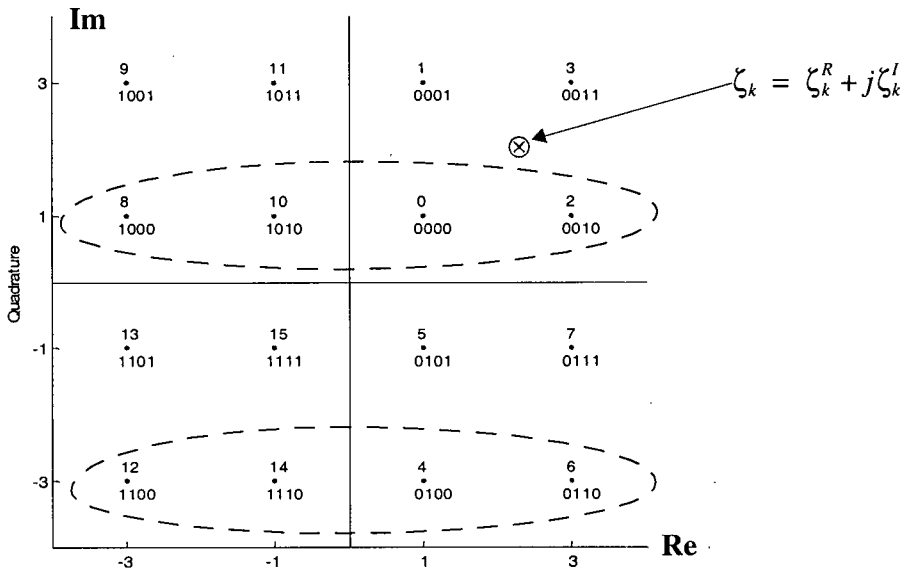


Figure 4.3 Diagram for calculation of a posteriori probabilities for 16-QAM

Bayes' rule we obtain the following:

$$P(b_0 = 0 \mid Y_k = \zeta_k) = \frac{\sum\limits_{\substack{B_i \mid (b_0 = 0) \\ i = 0, \ldots, 15}} P(Y_k = \zeta_k \mid B_i)}{\sum\limits_{B_i,\, i = 0, \ldots, 15} P(Y_k = \zeta_k \mid B_i)} = \frac{\sum\limits_{\substack{B_i \mid (b_0 = 0) \\ i = 0, \ldots, 15}} p(Y_k = \zeta_k \mid B_i)}{\sum\limits_{B_i,\, i = 0, \ldots, 15} p(Y_k = \zeta_k \mid B_i)}, \tag{4.8}$$

where B_i is a binary 4-tuples associated with the i-th constellation point for 16-QAM, and function $p(x|B_i)$ is *probability density function* (*p.d.f.*) conditioned on event B_i. Then

$$P(b_0 = 1 \mid Y_k = \zeta_k) = 1 - P(b_0 = 0 \mid Y_k = \zeta_k). \tag{4.9}$$

Alternatively we can compute the a posteriori probabilities for each of the bits corresponding to any subchannel. When $Y_k = \zeta_k = \zeta_k^R + j\zeta_k^I$, the likelihoods for all valid constellation points are calculable. For any specific value of bit $b_i \in \{0, 1\}$, we can always partition all constellation points into two subsets. For example, when $b_0 = 0$ in Figure 4.3, an associated subset of points is $S_{b_0 = 0} = \{B_0, B_2, B_4, B_6, B_8, B_{10}, B_{12}, B_{14}\}$ (eight points in the dash oval

circles), while the other subset of points $S_{b_0=1} = \{B_1, B_3, B_5, B_7, B_9, B_{11}, B_{13}, B_{15}\}$ is associated with $b_0 = 1$. We can obtain the a posteriori probability of $b_0 = 0$ by summing the likelihoods for the eight points in the first subset $S_{b_0=0}$, and then dividing the sum of the likelihoods over all 16 constellation points.

Consider the received symbol $Y_k = \zeta_k = \zeta_k^R + j\zeta_k^I$ centered on the constellation point which is associated with binary 4-tuples B_i. If we take the noise signal as the two-dimensional Gaussian noise, the conditional *p.d.f.* of the transmitted symbol $A_i = A_i^R + jA_i^I$ can be expressed as follows:

$$p(Y_k|A_i) = p(\zeta_k^R, \zeta_k^I|(A_i^R + jA_i^I)) = \frac{1}{2\pi\sigma_k^2}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2 + (\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right). \tag{4.10}$$

Substitution of (4.10) into (4.8), yields

$$P(b_0 = 0|Y_k = \zeta_k) = \frac{\sum_{\substack{B_i|b_0=0 \\ i=0,\ldots,15}} \exp\left(-\frac{(\zeta_k^R - A_i^R)^2 + (\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right)}{\sum_{B_i,\, i=0,\ldots,15} \exp\left(-\frac{(\zeta_k^R - A_i^R)^2 + (\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right)}. \tag{4.11}$$

Similarly, we can compute all values of $P(b_i = 0|Y_k = \zeta_k)$ and $P(b_i = 1|Y_k = \zeta_k)$ for $i = 0, \ldots, 3$.

Returning to diagram in Figure 4.2, the channel detector can evaluate the log-likelihood ratio of extrinsic probabilities for each bit in those 4-tuples B_i as follows:

$$LLR_{N_c}^{ex}(b_i) = \log\frac{P(b_i = 0 \mid Y_k = \zeta_k)}{P(b_i = 1 \mid Y_k = \zeta_k)} = \log\frac{\sum_{\substack{B_j \mid b_0 = 0 \\ j = 0, \ldots, 15}} \exp\left(-\frac{(\zeta_k^R - A_j^R)^2 + (\zeta_k^I - A_j^I)^2}{2\sigma_k^2}\right)}{\sum_{\substack{B_j \mid b_0 = 1 \\ j = 0, \ldots, 15}} \exp\left(-\frac{(\zeta_k^R - A_j^R)^2 + (\zeta_k^I - A_j^I)^2}{2\sigma_k^2}\right)}. \quad (4.12)$$

The demapper does not change the probabilities for any bit b_i. Thus the intrinsic probabilities and LLRs representations *w.r.t.* bit b_i inputted to LDPC decoder are given by equations (4.13) and (4.14), respectively:

$$P_{N_c}^{in}(b_i = 0 \mid Y_k = \zeta_k) = \frac{\exp(LLR_{N_c}^{ex}(b_i))}{1 + \exp(LLR_{N_c}^{ex}(b_i))}; \quad (4.13)$$

$$LLR_{N_c}^{in}(b_i) = LLR_{N_c}^{ex}(b_i) \text{ for } i = 0, \ldots, 3. \quad (4.14)$$

By applying (4.12)-(4.14) for all 4-tuples associated with every subchannel we calculate the intrinsic probabilities and LLRs for each bit assigned to each tone, such that we can decode all bits by using the message-passing algorithm on the normal graph constructed by the LDPC parity-check matrix.

Similarly, we can expand the application of the method described in this section to QAM constellations with any size of 2^b (such as 64-QAM, 256-QAM, 1024-QAM and so on).

The number of constellation points increases exponentially with the number of bits associated with said constellation; accordingly the computation of the intrinsic probabilities and LLRs for each bit of b-tuples associated with the constellation points will involve the computation of a large number of likelihoods or log-likelihoods. Therefore, it is impractical to apply this method for large QAM constellations.

In Eleftheriou's proposal [9], the set of b-tuples is split into two subsets of $b/2$-tuples. The first subset of $b/2$-tuples is mapped to the real part of the constellation point while the

second subset maps to the imaginary part of the constellation point. When $b \geq 2$, the two $b/2$-tuples independently select two $2^{b/2}$-ary real symbols. We present an example to discuss this strategy in part B below.

B. Gray-coded QAM constellation

For $b = 4$, 16-QAM is applied. The figure to depict the calculation of the a posteriori probabilities for the bits associated with the constellation [9] is shown in Figure 4.4.

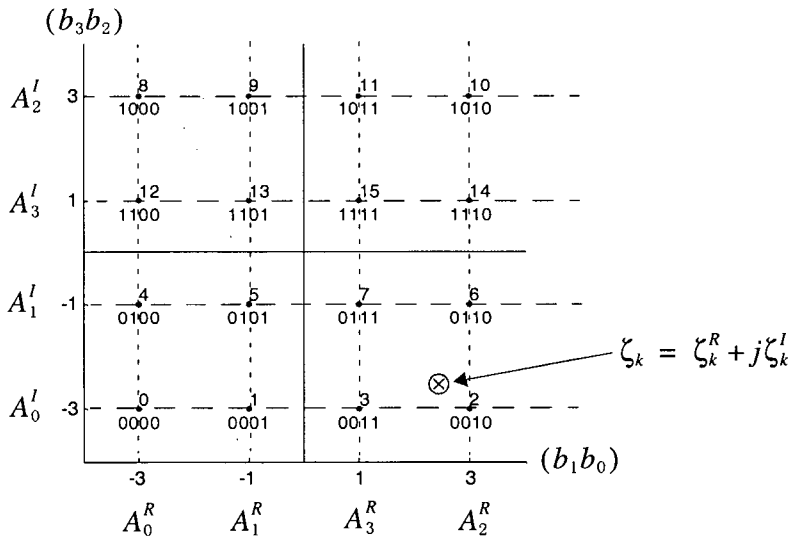


Figure 4.4 Diagram for 16-QAM with Gray coding

Since the binary 4-tuples denoted by (b_3, b_2, b_1, b_0) can be split into two 2-tuples as (b_1, b_0) and (b_3, b_2), we would like to compute $P(b_i = 0|Y_k^R = \zeta_k^R)$, $P(b_i = 1|Y_k^R = \zeta_k^R)$ for $i = 0, 1$ and $P(b_i = 0|Y_k^I = \zeta_k^I)$, $P(b_i = 1|Y_k^I = \zeta_k^I)$ for $i = 2, 3$, where ζ_k^R and ζ_k^I are the real and imaginary components of the received symbol in the specific subchannel.

Consider bit b_0, in which case $P(b_0 = 0|Y_k^R = \zeta_k^R) = \sum_{b_1 = 0}^{1} P(b_0 = 0, b_1|Y_k^R = \zeta_k^R)$. Assume next that all real components of constellation points are transmitted with equal probabilities.

Using Bayes' rule then yields

$$P(b_0 = 0|Y_k^R = \zeta_k^R) = \frac{\sum_{b_1} P(Y_k^R = \zeta_k^R|b_0 = 0, b_1)}{\sum_{b_1}\sum_{b_0} P(Y_k^R = \zeta_k^R|b_0, b_1)} = \frac{\sum_{b_1} p(Y_k^R = \zeta_k^R|b_0 = 0, b_1)}{\sum_{b_1}\sum_{b_0} p(Y_k^R = \zeta_k^R|b_0, b_1)}. \quad (4.15)$$

Consider that the transmitted real component mapped by 2-tuple $(b_1 b_0)$ is A_i^R for $i = 0, \ldots, 3$ (refer to Figure 4.4), then

$$p(Y_k^R = \zeta_k^R|A_i^R) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right); \quad (4.16)$$

$$P(b_0 = 0|Y_k^R = \zeta_k^R) = \frac{\sum_{i=0,2}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}{\sum_{i=0,\ldots,3}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}; \quad (4.17)$$

$$LLR(b_0) = \log\frac{\sum_{i=0,2}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}{\sum_{i=1,3}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}. \quad (4.18)$$

Similarly,

$$P(b_1 = 0|Y_k^R = \zeta_k^R) = \frac{\sum_{i=0,1}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}{\sum_{i=0,\ldots,3}\exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}; \quad (4.19)$$

$$LLR(b_1) = \log \frac{\sum_{i=0,1} \exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}{\sum_{i=2,3} \exp\left(-\frac{(\zeta_k^R - A_i^R)^2}{2\sigma_k^2}\right)}. \tag{4.20}$$

Assume that the transmitted imaginary component mapped by 2-tuple $(b_3 b_2)$ is A_i^I for $i = 0, \ldots, 3$ (refer to Figure 4.4), then

$$LLR(b_2) = \log \frac{\sum_{i=0,2} \exp\left(-\frac{(\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right)}{\sum_{i=1,3} \exp\left(-\frac{(\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right)}; \tag{4.21}$$

$$LLR(b_3) = \log \frac{\sum_{i=0,1} \exp\left(-\frac{(\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right)}{\sum_{i=2,3} \exp\left(-\frac{(\zeta_k^I - A_i^I)^2}{2\sigma_k^2}\right)}. \tag{4.22}$$

By applying (4.17)-(4.22) for all 4-tuples associated with every subchannel we calculate the intrinsic probabilities and LLRs for each coded bit assigned to each tone, such that we can decode all bits by using the message-passing algorithm.

It is clearly evident that this 16-QAM constellation with Gray coding simplifies the computation both for the probability and the LLR of every bit associated with the received symbol without any degradation in performance.

For large signal constellations of QAM, say 1024-QAM, the length of the tuples is $b = 10$. We use the strategy for LDPC encoding and constellation mapping described in Section 4.2.1. Let $bc_v = bc_w = 2$, by applying similar methods expressed in this section for all coded LSBs ($bc = 4$) of 10-tuples associated with every subchannel we calculate the intrinsic probabil-

ities and LLRs for them, such that we can decode all LSBs by using message-passing algorithm on the normal graph constructed by parity-check matrix. Once the LSBs are produced the corresponding MSBs ($bu = 6$) are determined by maximum likelihood estimation. The implementation is employed independently to the real and imaginary parts of the received symbols. Thus the computation is much simplified.

Until now we have discussed the detection and decoding for LDPC codes in DSL systems with a flat AWGN channel. If the channel transfer characteristic is not flat, all modulated subcarriers will be attenuated and rotated by different amounts, but they will not affect each other (i.e., they are orthogonal). This situation we consider next.

4.3.2 Frequency-selective AWGN channel

Recalling the introduction of discrete multitone modulation in Section 2.1, let the frequency-domain value of transmitted vector **x** be **X**, the value of received vector **y** be **Y** and the frequency-domain value of AWGN noise vector **w** be **W.** A frequency-domain diagram of Figure 2.4 is shown in Figure 4.5, where Λ is defined as following:

$$\Lambda = \begin{bmatrix} \lambda_{N-1} & 0 & \cdots & 0 \\ 0 & \lambda_{N-2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_0 \end{bmatrix} \tag{4.23}$$

and $\lambda_i = \frac{1}{\sqrt{N}} \sum_{k=0}^{v} h_k \exp\left(-j2\pi \frac{\pi k i}{N}\right)$ for $i = 0, 1, \ldots, N-1$; the sequence $h_0, h_1, \ldots, h_v$ is the impulse response of the channel.

Then equation (2.9) can be rewritten as following:

$$W_i = Y_i - \lambda_i X_i \text{ for } i = 0, 1, \ldots, N-1. \tag{4.24}$$

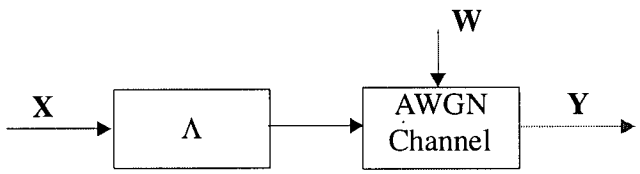


Figure 4.5 Frequency-domain diagram for a distorted AWGN channel

A. QAM constellation

Each element of transmitted symbol in (4.24) is attenuated by a specific element of frequency-domain response of the channel. Notwithstanding, a 2^b-QAM modulation scheme can still be used for this system. Detecting and decoding techniques as described in part A of Section 4.3.1 are applicable.

For $i = 0, 1, \ldots, N-1$ let each of the frequency-domain complex values of the channel characteristic be expressed as $\lambda_i = \lambda_i^R + j\lambda_i^I$ and assume a transmitted symbol is $A_i = A_i^R + jA_i^I$. Then

$$W_i = Y_k - \lambda_i A_i = (\zeta_k^R - \lambda_i^R A_i^R + \lambda_i^I A_i^I) + j(\zeta_k^I - \lambda_i^I A_i^R - \lambda_i^R A_i^I). \tag{4.25}$$

Thus, the *p.d.f.* of the two-dimensional Gaussian noise, centered on the constellation point A_i can be expressed as

$$p(Y_k|A_i) = \frac{1}{2\pi\sigma_k^2}\exp\left(-\frac{(\zeta_k^R - \lambda_i^R A_i^R + \lambda_i^I A_i^I)^2 + (\zeta_k^I - \lambda_i^I A_i^R - \lambda_i^R A_i^I)^2}{2\sigma_k^2}\right). \tag{4.26}$$

The corresponding formula *w.r.t.* equation (4.11) should be

$$P(b_0 = 0 | Y_k = \zeta_k) = \frac{\sum_{\substack{B_i | b_0 = 0 \\ i = 0, \ldots, 15}} \exp\left(-\frac{(\zeta_k^R - \lambda_i^R A_i^R + \lambda_i^I A_i^I)^2 + (\zeta_k^I - \lambda_i^I A_i^R - \lambda_i^R A_i^I)^2}{2\sigma_k^2} \right)}{\sum_{B_i, i = 0, \ldots, 15} \exp\left(-\frac{(\zeta_k^R - \lambda_i^R A_i^R + \lambda_i^I A_i^I)^2 + (\zeta_k^I - \lambda_i^I A_i^R - \lambda_i^R A_i^I)^2}{2\sigma_k^2} \right)}. \quad (4.27)$$

Similarly, we can compute all values of $P(b_i = 0 | Y_k = \zeta_k)$ and $P(b_i = 1 | Y_k = \zeta_k)$ for $i = 0, \ldots, 3$. The channel detector can evaluate the log-likelihood ratio of extrinsic probabilities for each bit in 4-tuples associated with a subchannel as follows:

$$LLR_{U_c}^{ex}(b_i) = \log \frac{\sum_{\substack{B_i | b_0 = 0 \\ j = 0, \ldots, 15}} \exp\left(-\frac{(\zeta_k^R - \lambda_i^R A_i^R + \lambda_i^I A_i^I)^2 + (\zeta_k^I - \lambda_i^I A_i^R - \lambda_i^R A_i^I)^2}{2\sigma_k^2} \right)}{\sum_{\substack{B_i | b_0 = 1 \\ j = 0, \ldots, 15}} \exp\left(-\frac{(\zeta_k^R - \lambda_i^R A_i^R + \lambda_i^I A_i^I)^2 + (\zeta_k^I - \lambda_i^I A_i^R - \lambda_i^R A_i^I)^2}{2\sigma_k^2} \right)}. \quad (4.28)$$

For all bits b_i the intrinsic probabilities and LLRs representations *w.r.t.* bit b_i input to the LDPC decoder are given by equations (4.13) and (4.14), respectively. By applying (4.12)-(4.14) for all 4-tuples associated with every subchannel, we can calculate the intrinsic probabilities and LLRs for all bits assigned to whole DMT symbol.

B. Gray-coded QAM constellation

Estimating the probability or LLR of each coded bit of the b bits using the same method as in part B of Section 4.3.1 will not be directly applicable here. Recall that the set of b-tuples is split into two subsets of $b/2$-tuples. The first subset of $b/2$-tuples is mapped to the real part of the constellation point while the second subset of $b/2$-tuples is mapped to the imaginary part of the constellation point. After the each complex element of the transmitted symbol is multiplied by a complex value λ_i, a problem is created, i.e., the real and imaginary parts of the channel noise are not mutually independent, as shown in equation (4.25). The real and imaginary parts of the symbols cannot be processed independently as described previously.

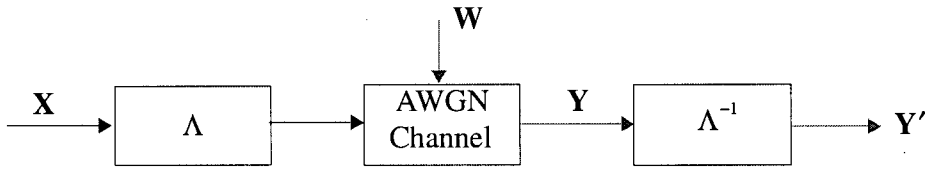


Figure 4.6 Frequency-domain zero-forcing equalizer for a distorted AWGN channel

Let Λ^{-1} be a transfer function defined below in equation (4.29):

$$\Lambda^{-1} = \begin{bmatrix} 1/\lambda_{N-1} & 0 & \cdots & 0 \\ 0 & 1/\lambda_{N-2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1/\lambda_0 \end{bmatrix}, \tag{4.29}$$

the received frequency-domain signal $\mathbf{Y}$ can be transformed into $\mathbf{Y}'$ as shown in Figure 4.6, i.e.,

$$\mathbf{Y}' = \Lambda^{-1}\mathbf{Y}. \tag{4.30}$$

Substitution of (2.9) and (4.29) into (4.30) yields

$$Y_i' = X_i + W_i' \text{ for } i = 0, 1, \ldots, N-1, \tag{4.31}$$

where $W_i' = \dfrac{W_i}{\lambda_i}$. Using $\lambda_i = \lambda_i^R + j\lambda_i^I$ and $W_i = W_i^R + jW_i^I$, the real and imaginary parts of W_i' are obtained as follows:

$$\begin{cases} Re(W_i') = \dfrac{\lambda_i^R W_i^R + \lambda_i^I W_i^I}{(\lambda_i^R)^2 + (\lambda_i^I)^2} \\ Im(W_i') = \dfrac{\lambda_i^R W_i^I - \lambda_i^I W_i^R}{(\lambda_i^R)^2 + (\lambda_i^I)^2} \end{cases}. \tag{4.32}$$

Since W_i^R and W_i^I are *i.i.d.* Gaussian variables with same zero mean and variance

$\sigma_i^2 = \sigma^2/2$, we find from (4.32) that $Re(W_i')$ and $Im(W_i')$ have same zero mean and variance

$$\sigma'^2_i = \frac{\sigma^2/2}{(\lambda_i^R)^2 + (\lambda_i^I)^2}. \tag{4.33}$$

Thus, under the function transformation Λ^{-1}, which is usually called frequency-domain *zero-forcing equalization* (ZFE), Y_k and σ_k^2 are transformed to Y_k' and σ'^2_k respectively. The method detailed in part B of Section 4.3.1 is applicable to obtain the *posteriori* probabilities and LLRs of all coded bits of b-tuples assigned to the DMT symbol.

Since the set of frequency-domain values $\lambda_i = \lambda_i^R + j\lambda_i^I$ for $i = 0, 1, \ldots, N-1$ are known for the fixed DSL channel, equations (4.32) and (4.33) imply that the receiver consists of a set of independent processors operating in parallel.

Chapter 5 Simulation Models and Results

In this thesis, we consider the use of LDPC codes in DSL systems that conventionally use the RS-TCM FEC coding schemes. The performance of different coding schemes is derived and analyzed. The term "*coding gain*" is usually employed to quantify the superior performance of one FEC scheme over another with the same transmission power, channel and BER.

In this chapter, we present the BER vs. E_b/N_0 performance curves for the different FEC coding schemes with various *bit allocation tables* (BATs). As described in Chapter 2 the BAT is determined during initialization according to channel characteristics [17] and background noises.

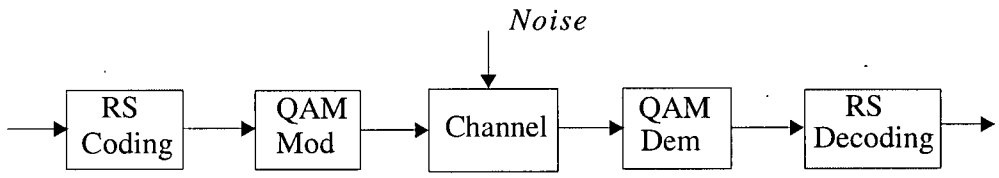
Our simulations use the Simulink platform. To simulate DSL systems which consist of many function blocks, we need to create some S-Function blocks to implement our algorithms. To reduce the run time, our S-Function blocks are programmed in C programming language.

First, we assume the channel is an ideal AWGN channel, in which case the frequency spectrum is a flat over the channel passband. Next, we choose a frequency-selective channel with distortion and interference. The performance of the different coding schemes in DSL systems with 256 DMT over these channels will be determined by system simulations.

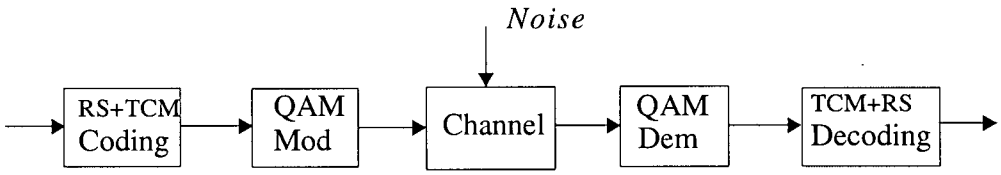
5.1 System Models

To compare the performance of different FEC coding schemes, we choose three different models for DSL systems to implement simulations as shown in Figure 5.1.

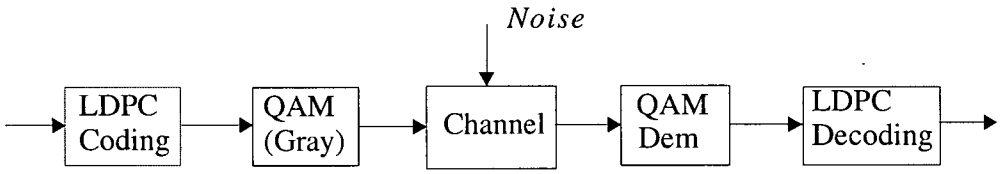
In Figure 5.1, in order to view different coding schemes we omitted many other blocks that work in DSL standard systems [1, 8]. As shown in Figure 5.1 (a) and (b), the randomly generated input bit sequence is split into the fast and interleaved path (refer to the Figure 2.5). The data in the two paths are independently scrambled and encoded by RS outer coder [1]. Our proposal for using LDPC codes in DSL systems as shown in Figure 5.1 (c) does not require the



(**a**) RS coding scheme



(**b**) RS-TCM coding scheme adopted by current DSL standards



(**c**) a proposed LDPC coding instead of current coding in DSL

Figure 5.1 Three system models for simulations

RS outer coder; the data in the fast and interleaved path are input directly to the LDPC encoder (refer to Section 4.2.1).

Before presenting our simulation results, we describe some parameters such as bit allocation table, code rate, SNR, and others.

5.2 Basic Parameters

5.2.1 Bit allocation table

The *bit allocation table* (BAT) is determined primarily by channel characteristics and background noises (refer to Sections 1.3 and 2.1) and the specifications of the DMT DSL

standards. The BAT is formed as an initialization activity. In our simulations, we choose 256 subchannels in a DMT symbol; the BAT specifies the number of encoded bits for every subchannel of a DMT symbol. Our study focuses on increasing the coding gain for appropriate FEC coding schemes. Some restrictions imposed on the BAT by DSL standards will be neglected. Therefore, the BAT has two types of existence. For ideal AWGN channel, DMT becomes OFDM, all subchannels of a DMT symbol have the same bandwidth and bit rate. For a frequency-selective channel the bit rate for each subchannel may be different.

5.2.2 Data rate, bit error rate and signal to noise ratio

Let N_I represent the number of information bits in a single DMT symbol, and let N_T be the total number of the bits transmitted in each DMT symbol. Then the overall code rate for the FEC scheme is

$$R = \frac{N_I}{N_T}. \tag{5.1}$$

In our simulation system, we choose 256 subchannels for each DMT symbol; the outputs of QAM mapper are 256 complex values in the frequency domain. After transformation of Hermitian symmetry and then IFFT, there are $S = 512$ real samples for ideal AWGN channel and $S = 512 + \nu$ real samples for frequency-selective AWGN channel where ν is the length of the cyclic prefix (or guard interval).

Assume that E_s is the average energy per subchannel symbol in time domain, $N_0/2$ is the two-sided power spectral density of the AWGN process, and E_b is the average energy per single bit of information. The variance of the AWGN is: $\sigma^2 = N_0/2$.

The relationship between E_s and E_b is

$$E_b = \frac{E_s S}{N_I}. \tag{5.2}$$

Therefore, we obtain

$$\frac{E_s}{N_0}(dB) = \frac{E_b}{N_0}(dB) + 10\log\frac{N_I}{S}, \tag{5.3}$$

where E_b/N_0 is commonly called the *signal to noise ratio* (SNR).

The probability of a bit error is estimated using *bit error rate* (BER) measurements. In the following simulations we present the performance of different FEC schemes by comparing the curves of BER vs. E_b/N_0.

5.3 Initialization

All simulations are implemented on the Simulink platform. Although we utilize different FEC coding schemes in the various DSL systems, the initialization procedures for all of them are almost identical. We describe some pivotal steps as follows:

1. Set up a system model including the channel model.
2. Load the BAT determined by SNRs for all subchannels and background noises.
3. Set appropriate total number of information bits based on the specifications of FEC schemes.
   - For the RS-TCM coding scheme, all information bits are split into the fast and interleaved paths by a pre-setting fraction. Set the parameters in every function block in both paths, these are CRC checks, RS coding, and Scrambler blocks. In addition, the parameters in the interleaving block need to be set in the interleaved path. According to the BAT and standards of DSL [1, 8], the parameters can be set for Wei's 16-state 4-D TCM coding block.
   - For the LDPC coding scheme, an outer coder to increase the performance is not required. Set up the parameters for the LDPC parity-check matrix which is generated using the construction technique by Eleftheriou and Olcer [9].
4. For frequency-selective AWGN channel models, the frequency response spectrum for the channel needs to be measured for all DMT subcarriers of the channel, which can also be derived if the BAT is available.
5. Set the value of output power at the transmitter and the value of variance for AWGN noise.

6. For LDPC coding scheme, the limit of iterations for decoding algorithm is set to be 30.
7. All simulations continue until the bit counter reaches 10^8 or at least 100 errors are detected.

5.4 Simulation Results

5.4.1 Results for DMT systems over ideal AWGN channel

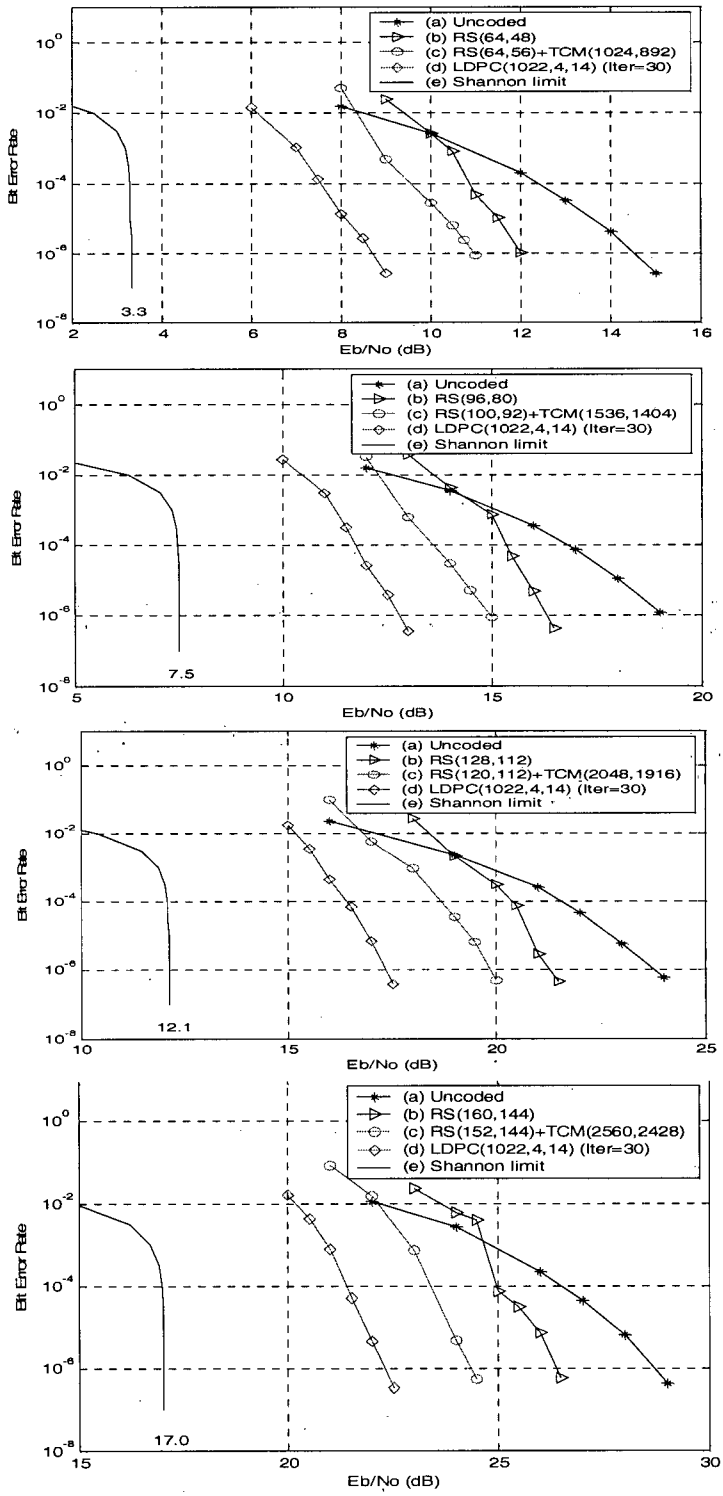
In this section, we present the simulation results in the form of BER vs. SNR curves. The performance of the different coding schemes for three system models as shown in Figure 5.1 will be compared. In each of the 256 subchannels of a DMT symbol, various sizes of the QAM constellations will be used; we choose 2^b-QAM constellation for $b = 4, 6, 8, 10$ corresponding to flat b-bits BAT. That means there are b bits assigned to each of 256 tones that are represented by 2^b-QAM symbols.

In each of Figures 5.2 (A)-(D), curves (b), (c), and (d) in each figure are BER vs. E_b/N_0 curves for three coding schemes (refer to Figure 5.1) which have approximately the same overall code rates in any figure. Curve (a) in each figure shows performance for corresponding uncoded 2^b-QAM modulation scheme in a 256-subchannel DMT system over an AWGN channel and curve (e) in each figure presents the Shannon limit[4] for such scheme over the AWGN channel.

Consider Figure 5.2 (A) with 16-QAM as an example to explain the coding schemes corresponding to the curves (b), (c), and (d). Curve (b) presents the performance of (64,48) RS code used in both fast and interleaved paths such that there are 32 redundant bytes per DMT symbol. The coding scheme corresponding to curve (c) concatenates (64, 56) RS coding used in both fast and interleaved paths with a (1024, 892) TCM coding block. There are 16 redundant bytes in RS code and 132 redundant bits in TCM code. For the curve (d), we use (1022, 4, 14) LDPC coding scheme with no outer RS code.

To identify the performance of different coding schemes, we choose a common value of

[4] For determination of Shannon limit see Appendix A.



(A) 16-QAM

(B) 64-QAM

(C) 256-QAM

(D) 1024-QAM

Figure 5.2 Performance curves of coding schemes over AWGN channel with 2^b-QAM: (A) 16-QAM; (B) 64-QAM; (C) 256-QAM; (D) 1024-QAM

10^{-6} for bit error rate; we then observe the corresponding values of E_b/N_0 from Figures 5.2 (A)-(D). The resulting difference in coding gains is summarized in Table 5.1.

Table 5.1 Gaps and gains for different coding schemes over flat AWGN channels

($P_e = 10^{-6}$)	Shannon limit (e) (dB)	Gap (a) to (e) (dB)	Gap (b) to (e) (dB)	Gap (c) to (e) (dB)	Gap (d) to (e) (dB)	Gain (c) to (a) (dB)	Gain (d) to (a) (dB)	Gain (d) to (c) (dB)
(A) 16-QAM	3.3	11.2	8.70	7.60	5.35	3.60	5.85	2.25
(B) 64-QAM	7.5	11.5	8.80	7.45	5.25	4.05	6.25	2.20
(C) 256-QAM	12.1	11.6	9.15	7.75	5.20	3.85	6.40	2.55
(D) 1024-QAM	17.0	11.7	9.40	7.35	5.25	4.35	6.45	2.10

From the values in last column of Table 5.1, at a BER of 10^{-6}, the LDPC coding scheme shows a gain of about 2.1~2.55 dB over that of the RS-TCM coding scheme, which in turn has a gain of about 4 dB over the uncoded QAM modulation.

Considering the LDPC codes with 16-QAM constellation over AWGN channel, the performance curves of two LDPC codes are shown in Figure 5.3. At the BER of 10^{-6}, the (2044,6,28) LDPC coding obtains 1 dB more gain than the (1022,3,14) LDPC coding under the same code rate. Therefore, the longer the length of LDPC code, the better the performance.

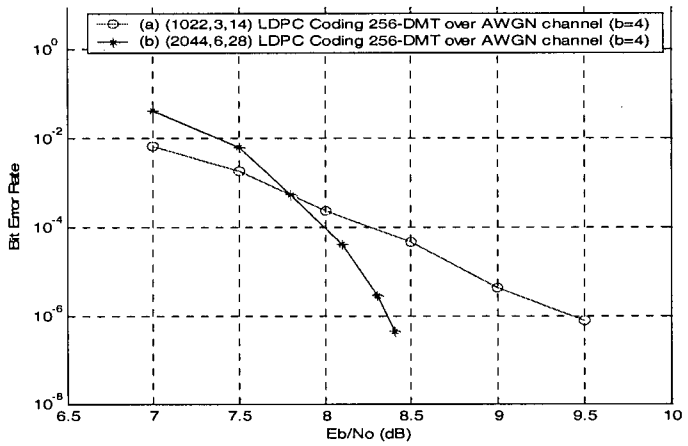


Figure 5.3 Comparison of performance curves of different length of LDPC coding schemes with 16-QAM over AWGN channel

5.4.2 Results for DMT systems over frequency-selective AWGN channel

For the DSL channel, the channel impulse response and noise power spectral density are relatively fixed over any communication session. The direct measurement of SNRs for each of the subchannels is employed during initialization and synchronization of DSL modems. By using the water-pouring algorithm based on given SNRs for each tone we can obtain a specific BAT for a specific DMT channel.

Given a BAT, we can obtain the discrete SNRs and frequency responses of the channel by inverse transformation of bit-loading as shown below in equations (5.4) and (5.5):

$$SNR_i = 10 \cdot \log((2^{b_i} - 1) \times \gamma) \text{ and} \tag{5.4}$$

$$H_i = 10^{SNR_i/20} \text{ for } i = 1, 2, \ldots, 256, \tag{5.5}$$

where b_i denotes the number of bits assigned to the i-th subchannel, and H_i denotes the frequency response at the i-th tone. Equations (5.4) and (5.5) apply to an ADSL system with BER $= 10^{-7}$, margin[5] = 6dB, and "gap" $\gamma \approx 14.0$ [3].

In this thesis, the BAT is assumed to be available for the DMT channel, such as BAT-1 or BAT-2 as shown in Figure 5.4. It is evident that the channel with BAT-2 has much more serious distortion than the channel with BAT-1. Before we compare with the performance of the systems over channels with BAT-1 and BAT-2, we first study the impact of using different cyclic prefix length on the system over the channel with BAT-1.

According to the analysis for DMT techniques in Section 2.1.3, the length v of cyclic prefix is a key variable. When v is set to a large value, the accuracy of channel estimation is high but the effective energy per bit is lowered. If v is set to a small value with a high energy per bit value, the accuracy of channel model is low. Therefore, the choice of v is traded off against these

---

[5] To guarantee a particular service to customers, DSL service providers require that data and error rates be achieved with all anticipated crosstalk and noise levels increased by some margin m. For ADSL and VDSL 6dB ($m = 4$) is the accepted value.

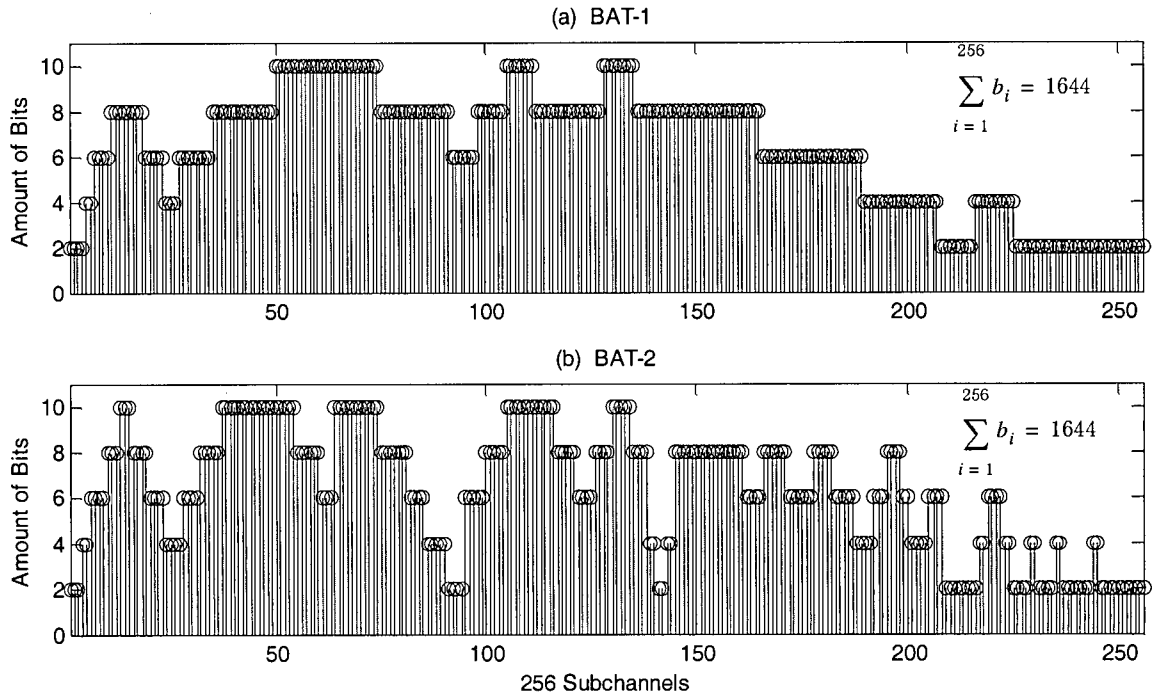


Figure 5.4 Two examples of BAT-1 and BAT-2 for 256 DMT channels

two factors.

To avoid the degradation of ISI in a DMT system the choice of ν is determined directly by the time-domain delay of the channel filter. Usually ν should be chosen equal to or greater than the number of channel filter taps. In our simulations, the length of guard interval ν is always set equal to the number of taps of the channel filter.

In the pre-load function of the Simulink simulation model, we choose $\nu = 16, 32, 64$ and 100 to establish channel filters. The corresponding frequency responses of channel filters are shown in Figure 5.5.

From Figure 5.5 we find it is the most accurate to describe measured SNRs by the frequency response of channel filter with 100 taps. With the same BAT-1 shown in Figure 5.4 (a) the performance curves of DMT system over the above four models of channel filters are given in Figure 5.6. In standard of ADSL the guard interval of cyclic prefix is $\nu = 32$ [1], which

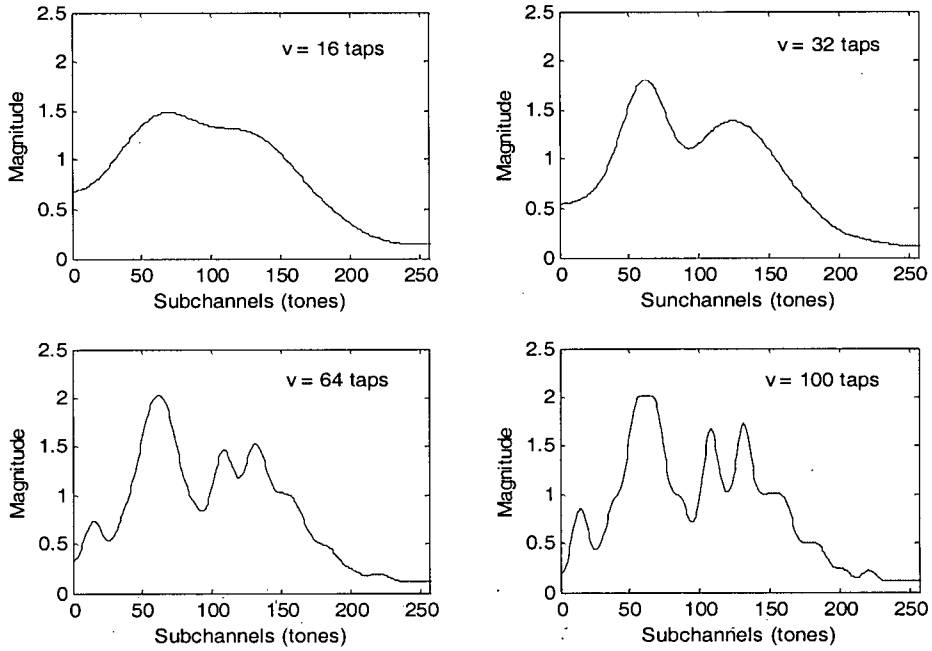


Figure 5.5 Frequency responses of the channel filter with different length of taps

corresponds to the curve (b). At the BER of 10^{-6} the gap is 0.38dB between (b) and (d). In later simulations we choose a 100 tap delay filter to study the performance of different coding

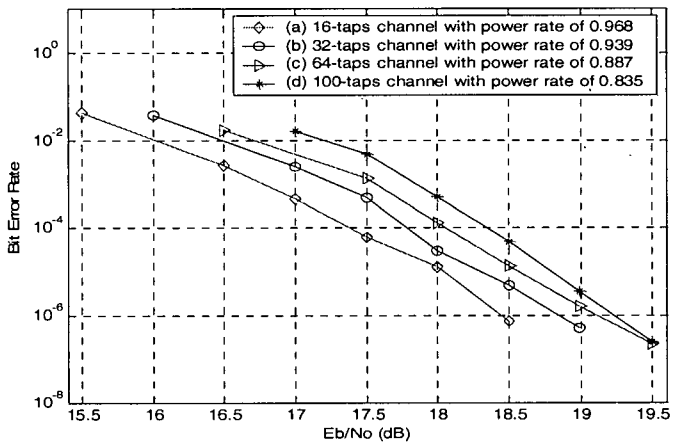


Figure 5.6 Comparison of performance curves of DMT systems over the channel filter with different length of taps

schemes.

Consider that the DMT system operates over the channel implemented by a 100 tap digital filter. Three FEC coding schemes with approximately the same overall code rate of 0.832 are chosen. The performance curves for BAT-1 and BAT-2 are shown in Figure 5.7; curve (a) presents the performance of uncoded QAM modulation scheme; curve (b) corresponds to the operation of (102,86) RS coding in both fast and interleaved paths; curve (c) presents behavior of concatenat-

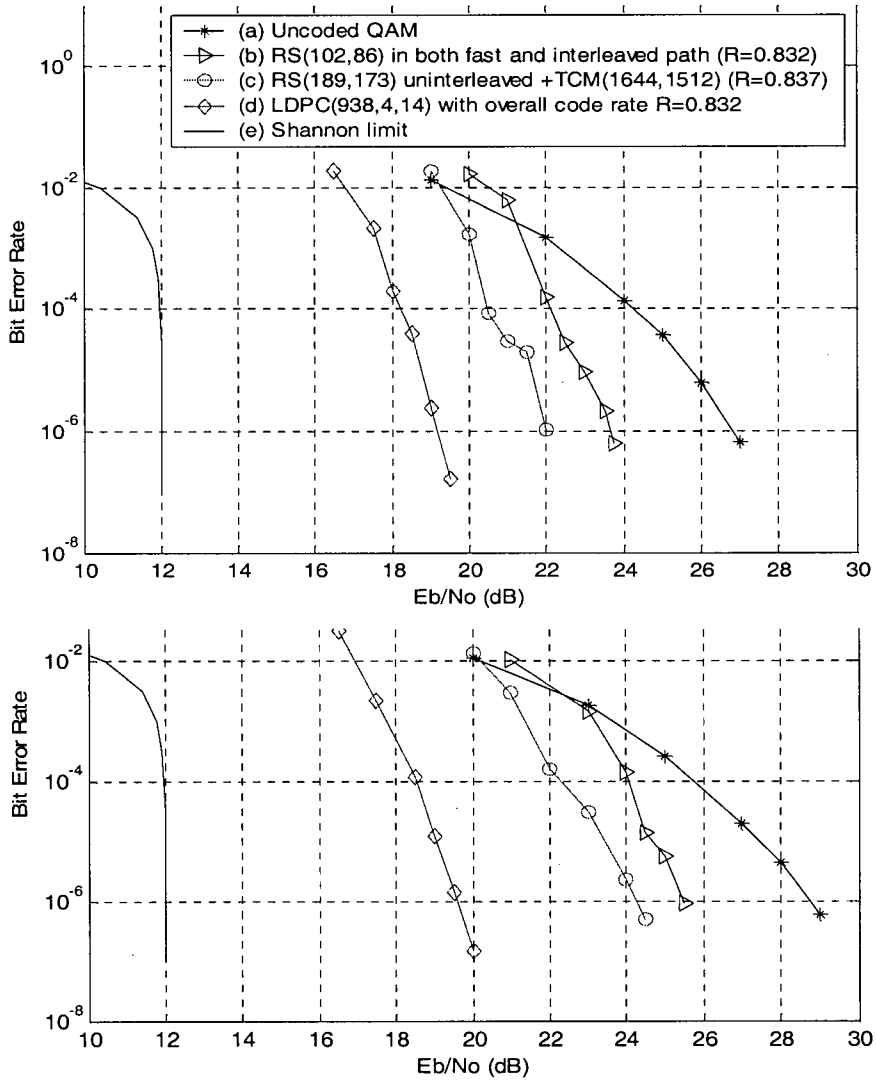


Figure 5.7 Comparison of performance of DMT systems with different coding schemes over the channel filter with 100 taps (a) BAT-1; (b) BAT-2

ing (189,173) RS code through uninterleaved path to a (1644,1512) TCM code; curve (d) expresses the action of (938,4,14) LDPC coding scheme; and curve (e) indicates the Shannon limit at this code rate for the AWGN channel[6].

Considering the difference in E_b/N_0 in Figure 5.7, the gaps of the various coding schemes from the Shannon limit and the gains from uncoded QAM are summarized in Table 5.2 for BER = 10^{-6}. From the values in last column of Table 5.2, for the BAT-1 channel, at a BER of 10^{-6} the LDPC coding scheme shows a gain of about 2.8dB over the RS-TCM coding scheme which in turn has a gain of about 4.86 dB over the uncoded dynamically-sized QAM modulation; for the BAT-2 channel, at a BER of 10^{-6} the LDPC coding scheme shows a gain of about 4.75dB over the RS-TCM coding scheme which in turn has a gain of about 4.5dB over the uncoded dynamically-sized QAM modulation.

Table 5.2 Gaps and gains of different coding schemes over two different channels (BAT-1 and BAT-2)

($P_e = 10^{-6}$)	Shannon limit (e) (dB)	Gap (a) to (e) (dB)	Gap (b) to (e) (dB)	Gap (c) to (e) (dB)	Gap (d) to (e) (dB)	Gain (c) to (a) (dB)	Gain (d) to (a) (dB)	Gain (d) to (c) (dB)
BAT-1	12	14.86	11.7	10	7.2	4.86	7.66	2.8
BAT-2	12	16.75	13.5	12.25	7.5	4.50	9.25	4.75

Compare these results to those described in the Section 5.4.1 for systems over a flat AWGN channel. The QAM modulation and LDPC coding strategy used with the frequency-selective channel model is effective in avoiding degradation of performance inherent in other FEC coding schemes, especially for channels like that where BAT-2 is used to overcome the channel's severe distortion.

5.4.3 Effect of using simplified LDPC decoding

Based on all same conditions for the (938, 4, 14) LDPC coding scheme, we compare its performance in decoding using two different arithmetic operations in the iterative message-

[6] Consider the computational complexity of the Shannon limit for frequency-selective AWGN channel (refer to Appendix A) and comparison of performance of different coding schemes working on the same systems. The curve of Shannon limit for AWGN channel shown here is only used for a relative reference.

passing algorithm. One operation uses strict sum-product update operation, while the other adopts the "min" approximation operation (described in Section 3.3.5).

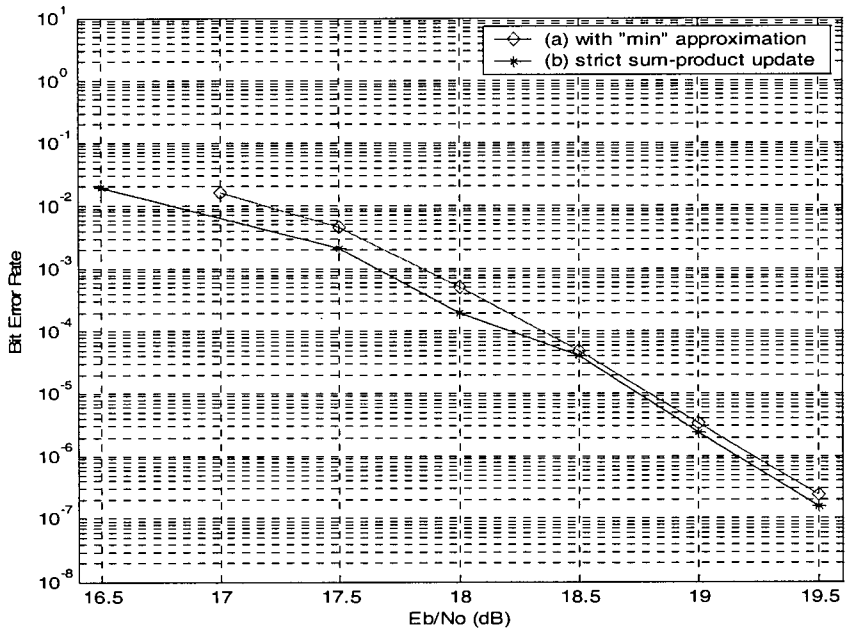


Figure 5.8 Comparison of performance of (938, 4, 14) LDPC coding by using two different arithmetic operations for iterative message-passing decoding algorithm

The simulation results are shown in Figure 5.8. When BER is less than 4×10^{-5} the gap between two curves is less than 0.1dB; thus the performance of "min" approximation is worse by less than 0.1dB in comparison with that of strict sum-product update.

A degradation of 0.1 dB is negligible. However, the "min" operation greatly simplifies the computation for iterative probabilistic decoding, especially for a long LDPC codes. Therefore, for small enough of BER, the "min" approximation is usable and has significant effect in reducing computational decoding complexity. This reality is of substantial practical importance.

Chapter 6 Summary and Conclusions

6.1 Summary

In Chapter 2 we summarized the DMT-DSL system and the RS-TCM scheme. The principle of DMT modulation and methods of coding and decoding for Wei's 16-state 4-D convolutional TCM codes were introduced.

Chapter 3 gave an introduction to Gallager LDPC codes and to the probabilistic decoding technique of these LDPC codes. The message-passing algorithm is expressed as probabilistic iterative decoding of binary block codes which are based on a normal graph. Three representations of the messages for performing the message-passing algorithm on LDPC codes were discussed. Finally, a simplified 'min' approximation of sum-product update for iterative decoding was derived.

The implementation of LDPC codes in DMT-DSL systems was discussed in Chapter 4. From LDPC encoding to symbol mapping we presented an overall LDPC encoder which was proposed by Eleftheriou [9]. Our study emphasized effective encoding and decoding of LDPC codes over channels with severe distortion. The channel model included AWGN and a linear filter was characterized by its frequency response.

To simplify modulation Double Gray mapping was applied from a square QAM to double PAMs. At the detector of receiver, a simple frequency domain zero-forcing equalizer was used. By combining corresponding demapping and probabilistic iterative decoding we created a means to use the LDPC FEC scheme on DMT-DSL systems operating on frequency-selective AWGN channels. Theoretically, almost the same performance using error correction on DMT systems is obtainable over various different channels.

Our simulations in Chapter 5 were implemented using the Simulink software package. Many existing blocks were employed including Bernoulli Random Binary Generator, CRC

Generator/Detector, Convolutional Interleaver/Deinterleaver, Scrambler & FEC/FEC & Descrambler, FFT/IFFT, QAM Modulator/Demodulator, AWGN Channel, and others. Some new functional blocks were created, including coding/decoding for TCM and LDPC, Gray mapping/demapping, tone ordering, and others. All simulations were implemented systematically.

6.2 Conclusions

In this thesis, we show how LDPC codes can work effectively on DMT-DSL systems. The FEC scheme based on LDPC codes and the message-passing algorithm operating on the normal graph are studied. Our scheme is applicable to both ideal and frequency-selective AWGN channels. For severe channel distortion, particularly, the error correction performance of the LDPC coding scheme gives significantly better performance than other FEC schemes.

Our experimental results indicate the following important conclusions:

- At the BER of 10^{-6}, the RS-TCM scheme yielded coding gains as follows over uncoded various QAM schemes: 4.0dB for AWGN channel, 4.86dB for BAT-1 channel, and 4.5dB for BAT-2 channel.
- At the BER of 10^{-6}, the LDPC coding scheme provided more than 2 dB coding gain over the RS-TCM FEC scheme which is the current standard for xDSL systems; specifically: (a) for ideal AWGN channels the coding gain is around 2.1~2.55dB; (b) for the BAT-1 channel the coding gain is 2.8dB; (c) for the BAT-2 channel the coding gain is 4.75 dB.
- At the BER of 10^{-6}, the coding gain for LDPC coding relative to uncoded QAM is much greater over the worse channels (such as BAT-2) than over the better channels (such as BAT-1). In other words, the performance curve for the LDPC coding scheme over the BAT-2 channel is almost same as that for the BAT-1 channel. However, for RS-TCM scheme, more than 2 dB degradation of performance occurs when channel changes from BAT-1 to BAT-2.
- The "min" approximation used for probabilistic decoding of longer LDPC codes greatly reduces computation with very small degradation in performance.
- Iterative decoding of LDPC codes does not create output burst errors when there is a

decoder failure; this result is different from that of the Viterbi decoder adapted for the decoding of TCM. An outer RS code can be omitted when LDPC codes are used for FEC.

6.3 Future Work

We propose some future research as follows:

- consider using LDPC codes of length sufficient to achieve near-capacity performance
- design an effective equalizer to improve further performance for LDPC codes
- combine *filtered multitone modulation* (FMT) with LDPC codes in DSL systems
- consider using LDPC codes for other applications such as *space-time coded* (STC) systems and CDMA systems

Appendix A The Calculation of the Shannon Limit

Let C represent the capacity of the AWGN channel being considered in bits/second. Let P denote the average signal power at receiver. According to the information capacity theorem [16] is stated as follows:

- The information capacity of a constinuous channel of bandwidth B hertz, perturbed by AWGN with power spectral density $N_0/2$ and limited in bandwidth to B, is given by

$$C = B\log_2\left(1 + \frac{P}{N_0 B}\right) \text{ bits per second.} \tag{A1.1}$$

If R_b is the information transfer rate in bits per second, then

$$R_b \leq C. \tag{A1.2}$$

The inequality (A1.2) is valid for error-free transmission (i.e., $P_e = 0$). If the error probability P_e is allowed not to be zero, then we can obtain the relationship between the maximum information rate and the channel capacity as follows:

$$R_b \leq \frac{C}{1 - H(P_e)} \tag{A1.3}$$

where $H(P_e)$ denotes the binary entropy at the probability P_e.

A1.1 AWGN channel

For the 256-DMT systems over AWGN channel, let R_s be the sampling rate in DMT symbols per second, substitution of (5.2) yields

$$P = R_s E_b N_I \tag{A1.4}$$

Since each subchannel in 256-DMT systems uses a 4kHz bandwidth, the bandwidth B of

the channel can be computed as $B = 256 \times 4kHz = 1.024MHz$. Therefore, combination of equations (A1.1), (A1.3) and (A1.4), and replacement of the inequality in (A1.3) with an equality yield

$$\frac{E_b}{N_0} = \frac{1.024MHz}{R_s N_I}\left(2^{\frac{R_b(1-H(P_e))}{256\times 4000}} - 1\right). \tag{A1.5}$$

The equation (A1.5) implies the Shannon limit curve for the AWGN channel.

A1.2 Frequency-selective AWGN channel

For the 256-DMT systems over frequency-selective AWGN channel, let R_s be the sampling rate in DMT symbols per second. For $k = 1, 2, \ldots, 512$, let $H(f_k)$ be the k-th component of the channel frequency response and P_k be the average received signal power in each subchannel assigned b_k bits within Δf bandwidth. The information capacity of the k-th subchannel is

$$C_k = \Delta f \log_2\left(1 + \frac{P_k}{N_0 \Delta f}\right) \tag{A1.6}$$

where

$$P_k = \frac{|H(f_k)|^2 R_s N_I b_k E_b}{N_T}. \tag{A1.7}$$

The total capacity of the overall channel is approximately given as follows:

$$C \cong \sum_{k=1}^{512} C_k = \Delta f \sum_{k=1}^{512} \log_2\left(1 + \frac{|H(f_k)|^2 R_s N_I b_k}{N_T \Delta f} \cdot \frac{E_b}{N_0}\right). \tag{A1.8}$$

Then substitution of (A1.3) with replacement of the inequality in (A1.3) with an equality yields

$$R_b[1 - H(P_e)] = \Delta f \sum_{k=1}^{512} \log_2\left(1 + \frac{|H(f_k)|^2 R_s N_I b_k}{N_T \Delta f} \cdot \frac{E_b}{N_0}\right). \quad \text{(A1.9)}$$

The definitions of N_I, N_T, E_b and E_s refer to Section 5.2.2. Therefore, we can theoretically get the Shannon limit curve for DMT systems over frequency-selective AWGN channel. However, it is a quite complicated process to obtain the Shannon limit curve by using equation (A1.9).

Glossary of Acronyms

ADSL	Asymmetric Digital Subscriber Line
ANSI	American National Standards Institute
AWGN	Additive White Gaussian Noise
BAT	Bit Allocation Table
BRI	Basic Rate Interface
BSC	Binary Symmetric Channel
CO	Central Office
CDMA	Code Division Multiple Access
DFT	Discrete Fourier Transform
DMT	Discrete Multitone
DSL	Digital Subscriber Line
ECC	Error-Correction Codes
FEC	Forward Error Correction
FEXT	Far-End Crosstalk
FMT	Filtered Multitone modulation
GF	Galois Field
GUI	Graphical User Interface
HDSL	High bit-rate Digital Subscriber Line
IDFT	Inverse Discrete Fourier Transform
i.i.d.	independent and identically distributed
ISI	Intersymbol Interference
ISDN	Integrated Services Digital Network
LDPC	Low-Density Parity-Check

LLR	Log-Likelihood Ratio
LSB	Least Significant Bit
MCM	Multicarrier Modulation
ML	Maximum Likelihood
MSB	Most Significant Bit
MSED	Minimum Squared Euclidean Distance
NEXT	Near-End Crosstalk
OFDM	Orthogonal Frequency Division Multiplexing
ONU	Optical Network Unit
PAM	Pulse Amplitude Modulation
POTS	Plain Old Telephone System
PSD	Power Spectral Density
QAM	Quadrature Amplitude Modulation
RFI	Radio Frequency Interference
RS Code	Reed Solomon code
SNR	Signal to Noise Ratio
STC	Space Time Coded
TCM	Trellis Coded Modulation
UTP	Unshielded Twisted Pair
VDSL	Very high bit-rate Digital Subscriber Line
VOD	Video On Demand
ZFE	Zero-Forcing Equalization
w.r.t.	with respect to
xDSL	A generic term which applies to all DSL technologies
XTalk	Crosstalk

Selected Bibliography

[1] American National Standard for Telecommunications - Network and Customer Installation Interfaces, "Asymmetric digital subscriber line (ADSL) metallic interface," T1E1.4/98-007R5, Dec. 1998.

[2] C. Berrou, A Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes," *Proc. IEEE International Communications Conference*, 1993, Page(s) 1064-1070.

[3] J. A. C. Bingham, *ADSL, VDSL, and Multicarrier Modulation*, Palo Alto, California: John Wiley & Sons, 2000.

[4] J. A. C. Bingham, "Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come," *IEEE Communication Magazine*, May 1990, Page(s): 5-14.

[5] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Transactions on Information Theory*, vol. 47, 2001.

[6] P. S. Chow and J. M. Cioffi, "Bandwidth optimization for high speed data transmission over channels with severe intersymbol interference," *IEEE GLOBECOM '92. 'Communication for Global Users'.*, Vol. 1, Dec. 1992, Page(s): 59 -63

[7] S. Chung, G. D. Forney, and T. Richardson, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, Feb. 2001, Vol 5, No. 2, Page(s) 58-60.

[8] Draft Trial-Use Standard for Telecommunications - Interface Between Networks and Customer Installation, "Very-high bit-rate digital subscriber line (VDSL) metallic interface," T1E1.4/2000-013R3, Nov. 2000.

[9] E. Eleftheriou and S. Olcer, "G.gen: G.dmt.bis: G.lite.bis: Proposed text on LDPC coding for inclusion in draft recommendation," *ITU-Telecommunication Standardization Sector*, Temporary Document SC-065, August 2001.

[10] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int'l Symposium on Turbo Codes and Related Topics*, Brest, France, Sept. 2000, Page(s) 543-546.

[11] G. D. Forney, Jr., "Codes on Graphs: Normal Realizations," *IEEE Transactions on Information Theory*, Feb. 2001, Vol. 47, Page(s) 520-548.

[12] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, Vol. 61, Mar. 1973, Page(s): 268-278

[13] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," *IEEE J. Select. Areas Communications,* May 2001, Vol. 19, Page(s) 908-917

[14] R. G. Gallager, "Low Density Parity Check Codes," *IRE Transaction on Information Theory*, 1962, Page(s) 21-28.

[15] R. G. Gallager, Low Density Parity Check Codes, MIT Press, Cambridge, MA, 1963.

[16] S. Haykin, *Communication Systems, 4th Edition*, McMaster University, ON: John Wiley & Sons, 2000.

[17] I. Kalet, "The Multitone Channel", *IEEE Transactions on Communications*, pp. 119-124, February 1989.

[18] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, Feb. 2001, Vol. 47, Page(s) 498-519.

[19] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information Theory*, Vol. 47, No. 2, Feb. 2001, Page(s): 585-598

[20] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, March 1999, Vol. 45, Page(s) 399-431.

[21] D.J.C. MacKay and R.M. Neal, "Near Shannon limit performance of Low Density Parity Check Codes," *Electronics Letters*, Aug. 1996, Vol. 32, Page(s) 1645-1655.

[22] D. J. C. Mackay, R. J. McEliece, and J. F. Cheng, "Turbo decoding as an instance of pearl's "belief propagation" algorithm," *IEEE J. on Select Areas in Communication*, 1997.

[23] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers,1988.

[24] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," *Acoustics, Speech, and Signal Processing*, IEEE International Conference on ICASSP '80., Vol. 5, Apr. 1980, Page(s): 964 -967

[25] T.J. Richardson, M.A. Shokrollahi and R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-checkcodes," *IEEE Transactions on Information Theory*, Feb. 2001, Vol. 47, No. 2, Page(s) 619-637.

[26] T.J. Richardson and R.L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, Feb. 2001, Vol. 47, No. 2, Page(s) 599-618.

[27] T.J. Richardson and R.L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, Vol. 47, No. 2, Feb. 2001, Page(s) 638-656.

[28] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, 1948, Vol. 27, Page(s) 379-423, 623-656.

[29] T. Starr, J. M Cioffi, and P. Silverman, *Understanding Digital Subscriber Line technology.* Upper Saddle River, NJ 07458: Prentice Hall, 1999.

[30] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, Sept. 1981, Vol. 27, Page(s) 533-547.

[31] P. P. Vaidyanathan, "Filter banks in digital communications," *IEEE Circuits and Systems Magazine*, Volume: 1 Issue: 2, Second Quarter 2001, Page(s) 4-25.

[32] L. F. Wei, "Trellis-Coded Modulation with Multidimensional Constellations," *IEEE Transactions on Information Theory*, Vol. IT-33, No.4, July 1987, Page(s) 483-501.

[33] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. Thesis, Link-oping University, Sweden, 1996.

[34] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Englewood Cliffs, NJ 07632: Prentice Hall, 1995.