Engineering Literature Survey
On
High-Speed Digital Subscriber Line Generation 2 (HDSL2) Modems

Sultan Ahmed
George Benavides
Khalid Islam

March 13, 1998

EE382C: Embedded Software Systems
Spring 1998

Prof. Brian L. Evans
Department of Electrical and Computer Engineering
The University of Texas at Austin

Abstract: The emerging HDSL2 standard promises reliable, low-cost, high-speed digital communications for telecommuting, home office transactions, enterprise computing, internet access, internet service providing and video conferencing over the existing telephone network. This technology is slated to offer two-way data communication at a speed of 1.544 Mbps over a single twisted pair copper wire. This capacity far exceeds existing voice band maximum limit of 64 Kbps and ISDN limit of 144 Kbps and is comparable to that of T1 lines at a much lower cost.

We plan to perform simulation of an HDSL2 modem using the Ptolemy environment. Since HDSL2 technology is still in an evolving stage, such a simulation required an understanding and evaluation of the various proposals presented to the HDSL2 overseeing committee for adaptation. This document contains a survey of available information on this emerging technology. It also contains an outline of our simulation plan. Once the simulation plan is implemented, we shall summarize our results, findings, and issues in a separate document.

I. Introduction and Context of Research

We intend to simulate an HDSL2 modem using the Ptolemy design and simulation environment. We have researched information on the HDSL2 technology and proposals for the HDSL2 standard. Most of the competing proposals are in the area of coding schemes and consequently our research has focussed on evaluating the different schemes and on understanding the contents of the various blocks of an HDSL2 modem.

II. Objectives

Our first objective of the literature survey was to reach a basic understanding of the various blocks of an HDSL2 modem shown in figure1 below [1]. Very briefly, the framer converts T1 format to serial format, the scrambler randomizes the data, the serial

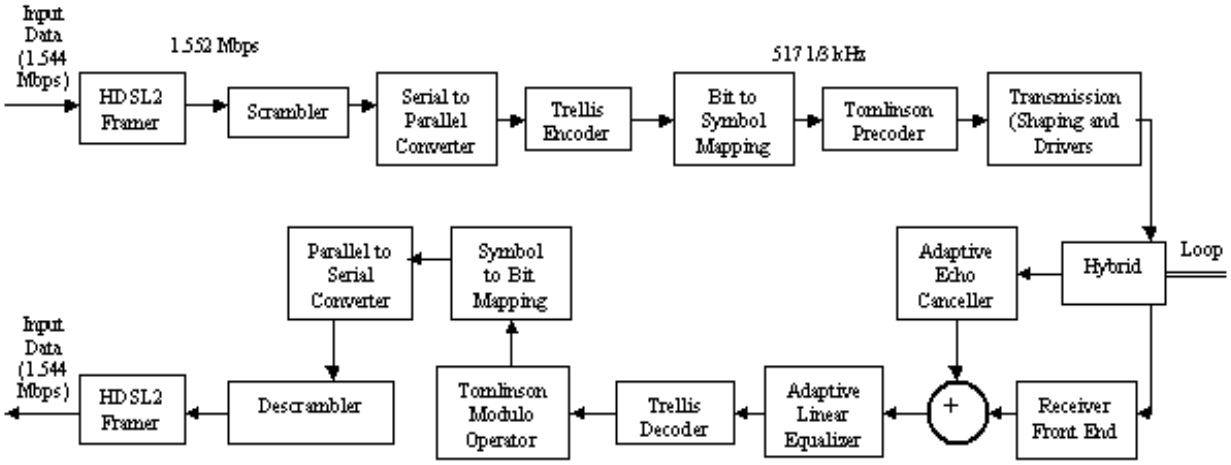

Figure 1. Top-level block diagram of an HDSL2 modem

to parallel converter produces 3 information bits per symbol time, the Trellis encoder adds 1 bit of redundant information resulting in 4 coded bits per symbol, the bit to level mapper maps these bits into one of 16 levels in a PAM (pulse amplitude modulation) constellation and the Tomlinson precoder precodes to counter the effect of the loop transfer function. The blocks in the receiver can be viewed as having the reverse functionality of the above mentioned blocks. This block diagram is taken from the Simple Coded PAM (SC-PAM) proposal for HDSL2 [1]. Our second objective was to evaluate

the various coding schemes to choose an appropriate one for our simulation. This is described in detail below.

III. Forward Error Correction (FEC) Scheme for HDSL2: Trellis Encoding

FEC means adding redundancy to the transmitted data so that a receiver can detect and recover from errors. Trellis coding is used for forward error correction in the HDSL2 standard. One reason for this is that Trellis coding lends itself to soft decoding as opposed to hard decoding required by block codes [2]. Soft decoding is a scheme in which the unquantized demodulator output is decoded directly using the probabilistic maximum-likelihood (ML) approach. This prevents the creation of errors that a separate quantization block would incur by making "hard" decisions when a demodulated value falls in the middle of two decision regions. For binary channel data, soft decoding results in a 2 dB signal-to-noise (SNR) gain over hard decoding [2].

The symbol rate of Trellis coded data is equal to that of uncoded data since redundancy is achieved by adding new values to the symbol constellation. This has the advantage of not increasing transmission bandwidth. Usually, one redundant bit is added to each symbol resulting in a doubling of the number of constellation values. Also, transmission power can be held constant by reducing inter-symbol spacing so as to keep total symbol space required the same as that in the uncoded case [3].

Trellis coding provides SNR improvement without sacrificing the data content of a symbol, transmission bandwidth or transmission power. We do however pay a cost in hardware complexity and latency.

IV. Picking a Trellis Coding Scheme for HDSL2

What we are looking for in a Trellis coding scheme to be adopted in the HDSL2 standard is one that will give us an SNR gain close to the theoretical maximum of 6 dB [3] while having a manageable complexity and tolerable latency. The SC-PAM proposal for HDSL2 presented in section II of this paper employs a simple 8-state Systematic Feedback Trellis encoder shown in figure 2 [1]. This scheme boasts minimal complexity and latency. However, in terms of SNR margin, its 3.2 dB SNR gain is not sufficient to

meet the 12,000 ft range requirement for the HDSL2 standard in the presence near end crosstalk (NEXT) from 49 other DSL sources (ADSL, HDSL or HDSL2). The reason the HDSL2 standard requires 49 disturbers to be tolerated is that phone companies bundle together 50 copper loops in a single binder. Note that the 8 states arise because there are 3 delay blocks, hence the number of states is 2^3.

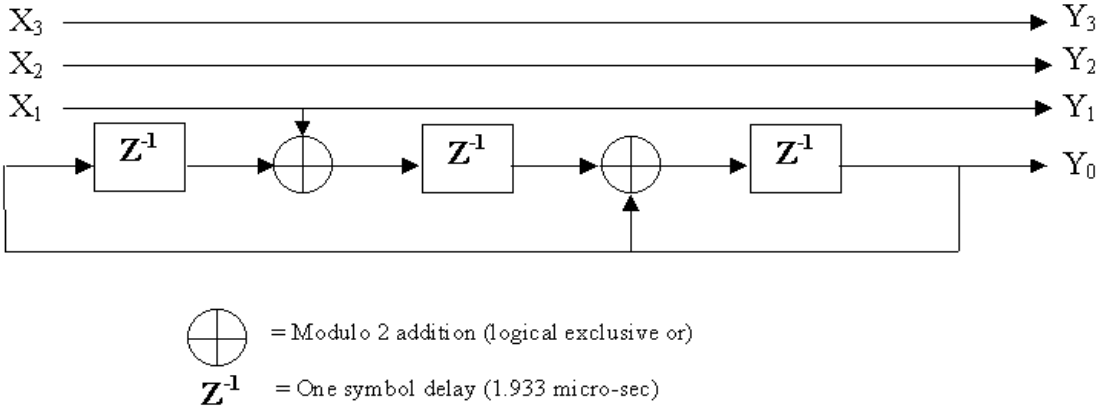

Figure 2: 8 state Systematic Feedback Trellis Encoder

Another Trellis scheme presented for being adopted as part of the HDSL2 standard is a 512 state Feedforward Convolutional encoder [4]. As with the 8 state Systematic Feedback encoder, this takes a 3 bit symbol and adds one bit and thus lends itself to 16-PAM as well. It gives us a good SNR gain of 5.1 dB at the cost of a higher but still tolerable latency at 217us. This encoder is shown in figure 3.

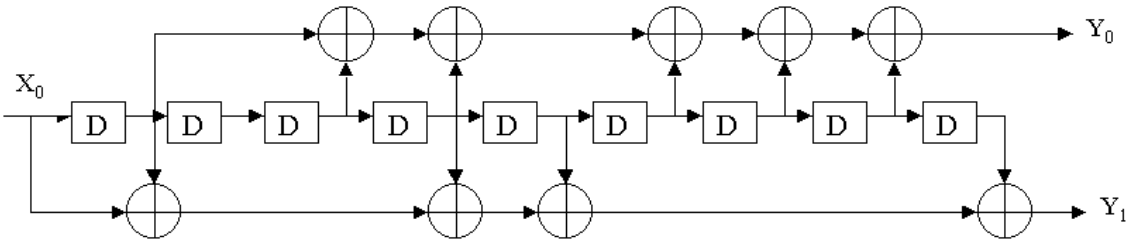

Figure 3: 512 State Feedforward Convolutional Trellis Encoder

Companies trying to get the maximum SNR gain possible have tried various other Trellis coding schemes. A two stage scheme (512 states + 16 states) employing a 128 QAM (quadrature amplitude modulation) also achieves a 5.1 dB SNR gain at a latency of

445us [5]. Another scheme with constraint length 13 (8k states) also achieves the same gain [6]. One suggestion has been to leave the encoder programmable.

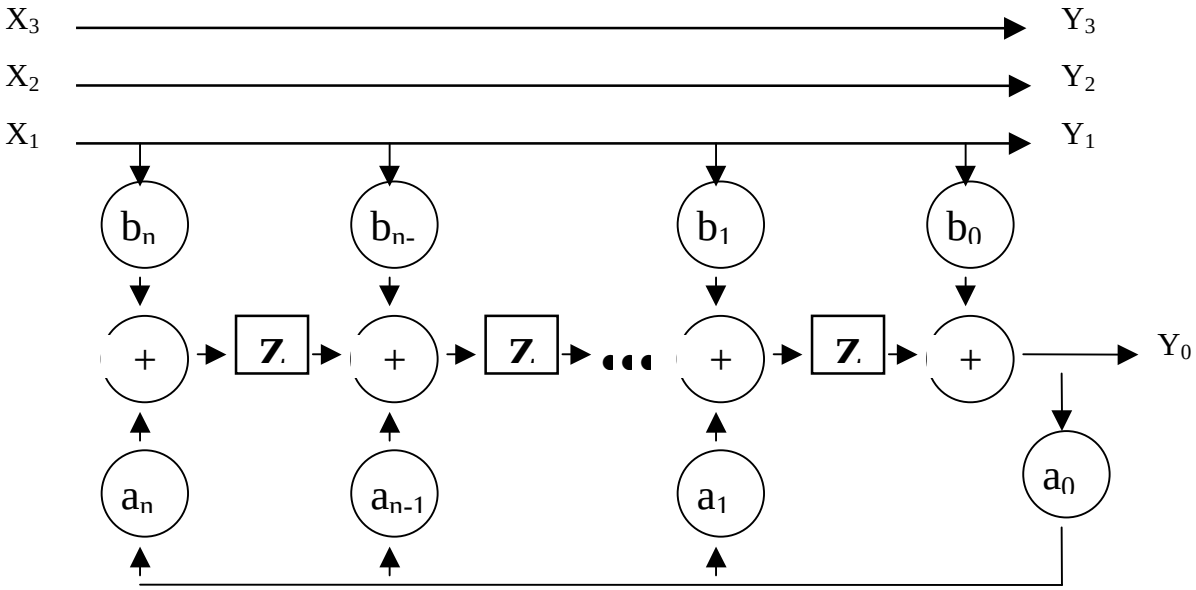

Figure 4: Programmable Trellis Encoder

This enables future migration and allows flexibility in decoding [7]. The program constants are determined at power up. This scheme is diagrammed in figure 4.

Table 1 below summarizes the various Trellis coding schemes. The 512 state Feedforward Convolutional encoder is the clear winner giving the highest SNR gain

Coder Type	**States**	**Coding Gain**	**Latency**	**Modulation**
Systematic feedback	8	3.2dB	Relatively small	16-PAM
Feedfor. Convolut.	512	5.1dB	217 usec	16-PAM
2 stage	512, 32	5.1dB	445usec	128-QAM
Constraint length 13	8092	5.1dB		64-QAM
Programmable	variable	5.1dB+	217usec+/-	16-PAM

Table 1: Published Data on the Performance of Various Trellis Schemes

(5.1 dB) with lower latency than other schemes with the same gain.

V. Tomlinson-Harashima precoding

Since the proposed Trellis scheme approaches its theoretical limitations of 6db coding gain, we must use another method with less complexity to increase SNR. One

such method is the Zero Forcing Decision Feedback Equalizer (ZF-DFE) (See figure 5) which consists of a feedback filter (FBF), and a feed-forward filter (FFF).

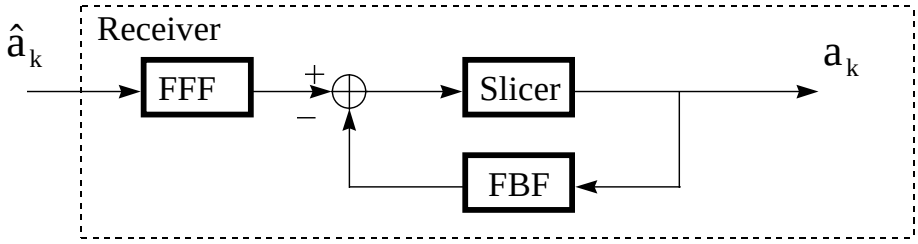

Figure 5. The ZF-DFE structure with symbol input $\hat{a}_k$ and output $\hat{a}_k$.

To incorporate the ZF-DFE with the Trellis scheme is difficult, since the ZF-DFE requires zero delay decisions or no latency to gain full advantage of its benefits. Also error propagation can be a problem due to the feedback filter. To overcome these problems, we can use another technique founded by Tomlinson and Harashima[8,9]. This technique simply splits the ZF-DFE and transfers the FBF into the transmitter (see figure 6 below).

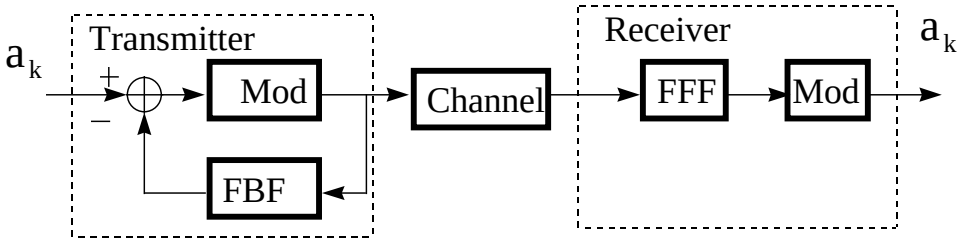

Figure 6. The Tomlinson-Harashima precoder (THP) structure.

The purpose of moving the FBF to the receiver removes the possibility of error propagation, since the Intersymbol Symbol Interference (ISI) is non-existent in the transmitter. The purpose of the 'Mod' operator is to add nother sequence of distinct integers(d_k) to limit the output symbol (x_k) between the range of $-M \le x_k \le M$ where M is the number of symbol levels (see figure 7).

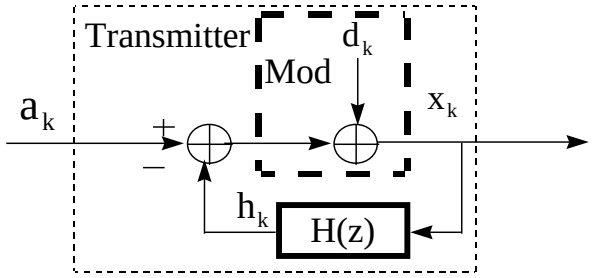

Figure 7. The details of THP in the transmitter.

Without the 'Mod' operator, the resulting output symbol will grow indefinitely. For the H(z) filter, we need to send a training sequence to the receiver which will then send the appropriate coefficients back to the transmitter. Essentially the H(z) filter will consist of the channel response, transmitter shaping filter and receiver filter. Using the H(z) in the feedback, eliminates the effects of the channel before transmitting the symbol. In theory, the symbol received at the receiver should not contain any ISI. At the receiver, we need to filter the receive symbol (c_k) to whiten the noise added by the channel (see figure 8).

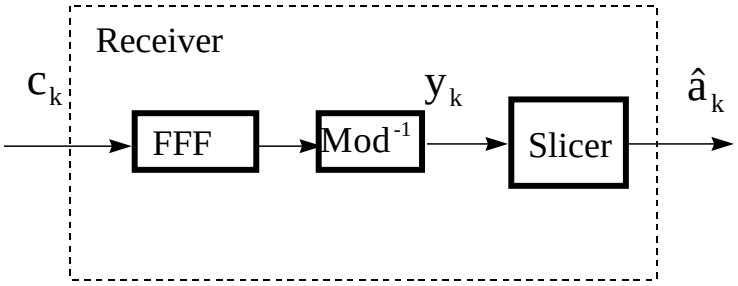

Figure 8. THP at the receiver

Whitening the noise insures that the received signal is spectrally flat. This flatness is needed for the inverse 'Mod' operator, otherwise, errors could occur in the inverse translation[10]. The slicer simply obtains the input and maps or quantizes it to the appropriate symbol. The result of combining THP with the Trellis has been studied extensively [10,11]. These results show about 3db improvement for 128 Trellis coded QAM. With the combined coding gain of the Trellis and THP, we can invariably achieve the upper bounds of channel capacity. This attainable capacity makes HDSL2 a viable alternative to HDSL.

VI. Implementation Plan and Conclusion

For the HDSL2 modem simulation project, we plan to implement all the transmitter and receiver blocks in Ptolemy using synchronized dataflow (SDF) graphs except for the adaptive linear equalizer and the adaptive echo canceller which will be added if time permits. We have also decided not to deal with crosstalk interference for the sake of saving time.

At this time, we have worked out most of the block details. We still have a few pending research issues regarding some of the blocks, including Trellis decoder (Viterbi decoding), OPTIS (signal shaping for transmission) and the adaptive linear equalizer.

VII. References

1. R. Goodson, K. Schneider and J. Moore, "Proposal for single loop HDSL using simple coded PAM," Adtran contribution, T1E1.4/96-037, Apr. 1996.
2. G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67, Jan. 1982.
3. E. Lee and Messerschmidt, "Digital data communications."
4. M. Tu and J. Liu, "A 512-state PAM TCM code for HDSL2," PairGain contribution, T1E1.4/97-300, Apr. 1997.
5. M. Tu and J. Liu, "A comparison of MS/TCM coding schemes for HDSL2," PairGain contribution, T1E1.4/97-299, Apr. 1997.
6. J. Liu and G. Zimmerman, "Proposal for HDSL2 transmission: spectra, line code and coding," PairGain contribution, T1E1.4/97-074, Apr. 1997.
7. R. Goodson, "Proposal for programmable trellis encoder for HDSL2," Adtran contribution, T1E1.4/97-338, Apr. 1997.
8. M. Tomlinson, "New automatic equalizer employing modulo arithmetic,"*Electronics Letters*, vol. 7,pp. 138-139,Mar. 1971.
9. H. Harashima and M. Miyakawa, "Matched-transmission technique for channels with rational spectra," *IEEE Transactions Communications*, vol. COM-20, pp. 774-780, Aug. 1972.
10. A. Aman and R. Cupso, "Combined Trellis coding and DFE through Tomlinson Precoding", *IEEE Journal on Selected Areas in Communciations*, vol. 9, pp. 876-883.
11. G. Pottie and M. Eyuboglu, "Combined coding and Precoding for PAM and QAM HDSL sytems", *IEEE Journal on Selected Areas in Communciations*, vol 9, pp861-869.