# Simulation and Modeling of an ADSL modem – Channel Model and Receiver Initialization

Magesh Valliappan

**Abstract**

In this project, I design a channel model and the systems needed for receiver initialization of a ADSL G.lite modem in the Ptolemy design environment. One of the significant problems in ADSL modem design is the inherent complexity of the system. The objective of this project is to build a high level abstraction of the specification of an ADSL G.lite modem. This will enable simulation, performance analysis, and synthesis of modem designs. A channel model is needed to accurately simulate the effect of transmission of an electrical signal through a twisted pair of copper cables. For optimal performance, the receiver and transmitter must adapt to the channel and noise characteristics during initialization. The receiver must estimate the channel and noise, perform channel equalization and also synchronize to the transmitter.

# 1 Introduction

The demand for bandwidth for high speed data communications for Internet access and interactive media transmission has been growing explosively. Digital subscriber line (DSL) technologies reuse the existing twisted pair copper cables that already reach most of the customers for telephone connections to provide cost-effective, dedicated uninterrupted high data rate connections that are secure and are always on. Asymmetric digital subscriber line (ADSL) is a form of DSL that caters to the typical asymmetric bandwidth requirements of residential customers and small businesses. A splitter is needed to separate the voiceband from the ADSL data transmission. Splitterless ADSL uses lower power to reduce interference and can therefore only support data rates of up to 1.5Mb/s downstream. This version of ADSL, called G.lite was standardized by the International Telecommunication Union recently [1].

One of the significant problems in ADSL modem design is the inherent complexity of the system. The objective of this project is to build a high level abstraction of the specification of an ADSL G.lite modem. This will enable simulation, performance analysis, and synthesis of modem designs. My contributions are a channel model and the systems needed for the initialization of an ADSL receiver. The channel model is needed to accurately simulate the effect of transmission of an electrical signal through a twisted pair of copper cables. For optimal performance the receiver and transmitter must adapt to the channel and noise characteristics during initialization. The Ptolemy design environment supports the required heterogeneous design with different models of computation [2].

# 2 Background

## 2.1 Channel Characteristics

Channel characteristics are determined by several factors including cable length, cable gauge, bridge taps, different resistor-capacitor terminations and shunt resistances [3]. The signal is subject to electromagnetic interference, called crosstalk, from other cables in the same bundle. Crosstalk can be modeled as colored additive Gaussian noise [4]. Relatively lower distortion is further caused by external sources and thermal noise.

## 2.2 G.lite ADSL modem

A bandwidth of 28 kHz to 552 kHz is used by ADSL G.lite modems [1]. Lower frequencies are avoided to prevent interference from voiceband data. The lower part of this bandwidth up to 138kHz is used by upstream data and the rest by downstream data. Using discrete multitone modulation (DMT), the modem modulates data into 128 subbands each of bandwidth 4.3125kHz. The transmitter encodes multiple bitstreams using baseband quadrature amplitude modulation (QAM). DMT is achieved by a 256-point inverse fast Fourier transform (IFFT). One of the subchannels is dedicated to a pilot signal used for synchronization. Inter symbol interference (ISI) is caused by a channel with memory. So a cyclic prefix is added to preserve the periodic nature of the IFFT. The receiver performs time domain equalization using an FIR filter. After the cyclic prefix is removed, an FFT is computed. The gain of the channel is canceled in a frequency domain equalizer before QAM demodulation to extract bits.

### 2.2.1 Receiver Initialization

During initialization known data sequences are transmitted to train the receiver. The G.lite standard specifies a detailed list of states and the signals to transmit and expect to receive while in each state [1]. The various steps in a typical initialization in chronological order are -

- **Handshake** - The receiver negotiates settings during handshake. Detection of the handshake signals needs systems that can detect activity in individual subchannels.
- **Automatic gain control** - The receiver measures the incoming signal amplitude to adapt the automatic gain control.
- **Synchronization** - The receiver acquires approximate frame synchronization by estimating the channel delay. Sample synchronization is achieved by using the phase of a demodulated pilot signal.
- **Time domain equalization** - A linear filter model of the channel is estimated. This is used to compute an FIR filter that performs equalization.
- **Noise and gain estimation** - The gain and noise in each subchannel is computed by observing received samples obtained from a wideband pseudo-random signal transmitted with cyclic prefix. The subchannel gain estimate is used to compute the frequency domain equalizer.
- **Bit loading** - Signal-to-noise ratio (SNR) of each sub channel is estimated. The optimal bit allocation for the subchannels is computed using this data.

The receiver exchanges these settings with the transmitter in subsequent operations before entering normal operations.

# 3 Modeling

## 3.1 Channel Model

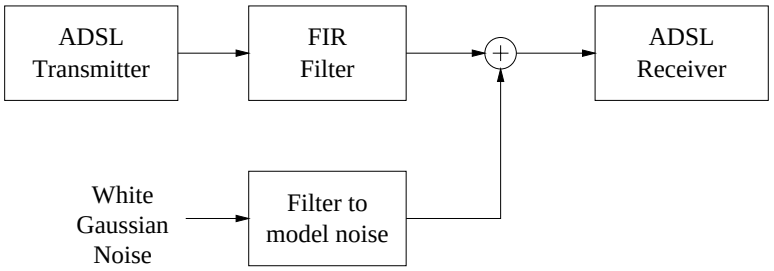


Figure 1: Block diagram of the channel model. (The graph is homogeneous)

The effect of transmission of the signal over the channel is modeled by a discrete time finite impulse response (FIR) filter determined by the parameters of the cable. The channel model is shown in Fig. 1. Crosstalk is modeled as additive white Gaussian noise filtered by an FIR filter. Additive white Gaussian noise is used to model static noise. The channel model is an homogeneous synchronous dataflow (SDF) graph [5].

## 3.2 Receiver Initialization

The transceiver is modeled in two sections. The front end performs sample input and output, dynamic addition and removal of cyclic prefix and provides dynamic adjustment

of the frame boundaries. This requires maintaining a dynamic buffer of samples. To model the synchronous interface to the analog domain the transceiver performs sample input and output through one SDF block. This block is followed by dynamic dataflow to manage buffers and extract frames. The sampling phase error is estimated by the back end and can be used in a phase locked loop tied to the sampling clock.

The back end of the transceiver processes the incoming frames and generates frames of outgoing data. This part of the transceiver is modeled as an SDF graph driving statically schedulable boolean dataflow (BDF) blocks [5]. At every iteration one frame of input and output is processed. This makes the design relatively simpler, but introduces a signal processing delay in the system. The BDF systems control dataflow to computation intensive subsystems implemented in SDF blocks. An SDF representation is well suited for fast implementations. The boolean control variables are generated by a finite state machine (FSM). The FSM also controls the generation of transmitter data frames. The FSM transfers information about frame boundaries and makes requests for addition or removal of the cyclic prefix to the front end.

# 4 Implementation of Channel Model

The FIR filter that models the channel is computed at the start of the simulation using a program compiled from publicly available source code `linemod` [6]. The program offers several options for cable parameters like gauge, length, bridge taps and additional circuit elements. The coloring filter for crosstalk noise is also computed at the start of simulations

as an FIR filter based on the number of disturbers in the cable bundle.

# 5 Implementation of Receiver Initialization

The front end of the receiver is implemented in boolean dataflow, but requires a dynamic scheduler. The front end would normally perform gain scaling at the analog interface. However, to enable periodic updates, the gain scaling is performed at the back end. Sample synchronization is assumed and the sampling phase error estimate is not used. The simulation currently uses floating point data.

At the back end, pilot detection and loss of signal detection are performed for every frame. The parent SDF system generates a reference signal to collect statistics of the input at every subchannel for channel and noise estimation [3]. An ISI minimizing equalizer has been implemented [7]. A practical bit loading algorithm has also been implemented [8]. The FSM is also implemented as an SDF block for convenience.

# 6 Simulation Results

The test results were obtained by a stand-alone simulation of the downstream receiver with a precomputed input sequence. The implementation is scheduled using a fast DDF scheduler present in Ptolemy. The simulation time is about 0.06 seconds per frame of input (4300 samples/second) on a PC powered by an AMD K6-2 running at 450MHz. For a full simulation of initialization that would correspond to approximately 40 minutes. Fig. 2 shows the estimated channel and the equalized channel.

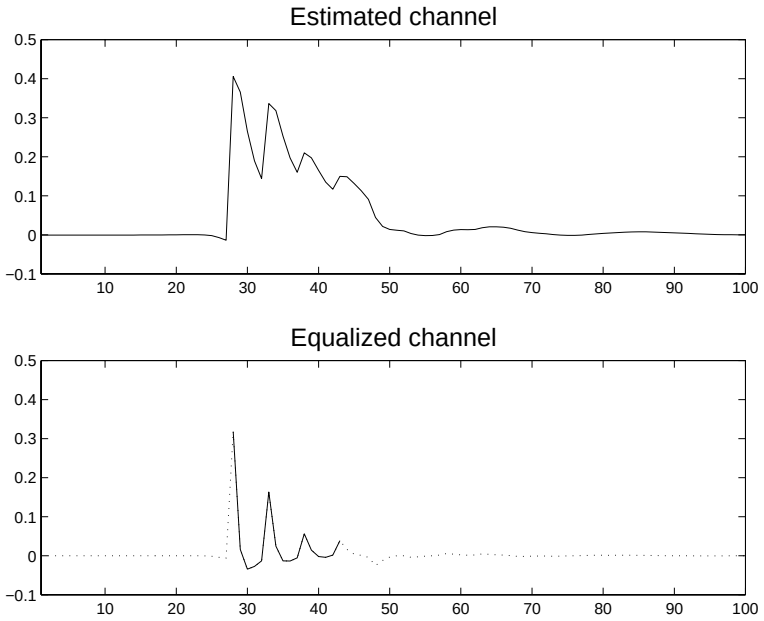


Figure 2: The estimated channel (top) is obtained after 512 frames of input. The equalized channel is the effective channel after equalization. The dotted line represents ISI.

# 7 Conclusions

In this project, I design a channel model and the systems required for initialization of an ADSL G.lite modem. The channel model is configurable and models both crosstalk and additive noise. The design uses signals identical to the standard specified initialization training sequences. The receiver is modeled in two sections. The front end being dynamically scheduled to provide a supply of data frames to the back end. The back end is statically scheduled A finite state machine is used to direct the operation of the receiver. The system is extensible and can be used for evaluation of the design of an ADSL receiver under standard compliant input conditions.

# References


[1] ITU-T G.992.2, "Splitterless asymmetric digital subscriber line (ADSL) transceivers."

[2] *Ptolemy 0.7 Users's Manual.* University of California at Berkeley, 1990.

[3] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding Digital Subscriber Line Technology.* Prentice-Hall PTR, 1999.

[4] K. J. Kerpez, "Near end crosstalk is almost Gaussian," *IEEE Trans. on Comm.*, vol. 41, pp. 670–672, May 1993.

[5] B. L. Evans, "Class notes for EE382C: Embedded software systems, Spring 2000," http://www.ece.utexas.edu/ bevans/courses/ee382c.

[6] Information Systems Laboratory, Stanford University, "The linemod program." http://www-isl.stanford.edu/people/cioffi/linemod/linemod.html.

[7] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. on Comm.*, vol. 44, pp. 1662–1672, Dec. 1996.

[8] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *IEEE J. on Selected Areas in Comm.*, vol. 43, pp. 773–775, Feb./Mar./Apr 1995.