ADSL TUTORIAL

Matthew J. Langlois, University of New Hampshire InterOperability Laboratory
121 Technology Drive, Suite 2, Durham, NH 03824 USA.
Extracted from the Introduction and Chapter 1 of **A G.hs Handshaking Protocol Analyzer For ADSL,**
a Master's Project by Matthew J. Langlois, May 2002.

INTRODUCTION

The demand for high-speed data networks in the "last mile" has driven the need for robust, interoperable, and easy to use multi-vendor Digital Subscriber Line (DSL) access solutions. DSL collectively refers to a group of technologies that utilize the unused bandwidth in the existing copper access network to deliver high-speed data services from the distribution center, or central office, to the end user. DSL technology is attractive because it requires little to no upgrading of the existing copper infrastructure that connects nearly all populated locations in the world. In addition, DSL is inherently secure due to its point-to-point nature. A simple diagram of a typical DSL system is shown in Figure 1 below:

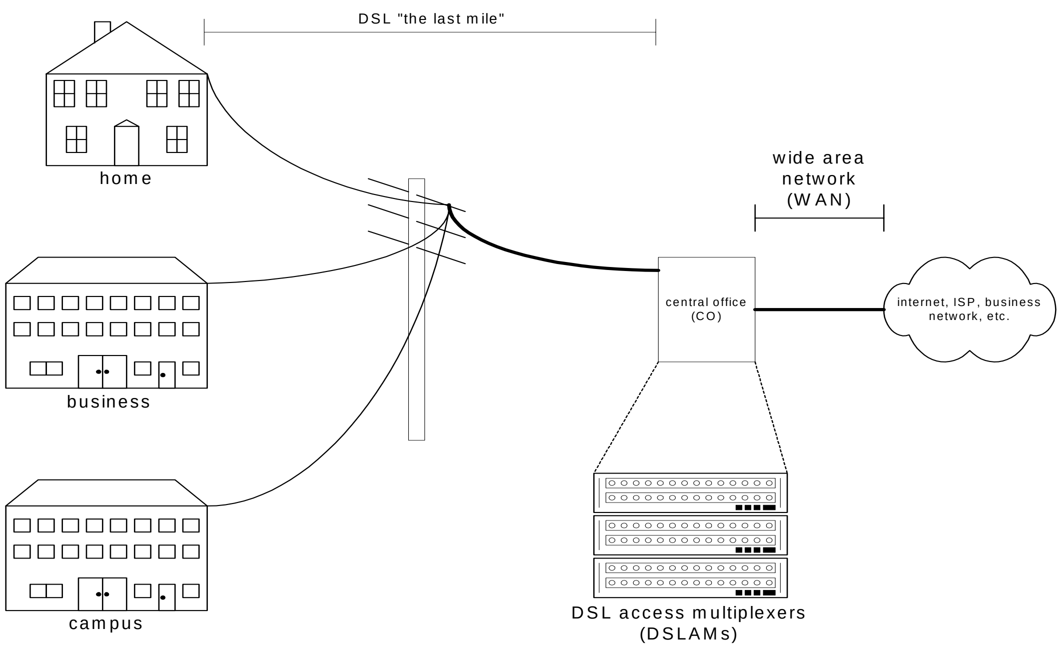


Figure 1: Typical DSL system.

There are many variations of DSL, each aimed at particular markets, all designed to accomplish the same basic goals. ADSL, or Asymmetric DSL, is aimed at the residential consumer market. ADSL provides higher data rates in the downstream direction, from the central office to the end user, than in the upstream direction, from the end user to the central office. Within the Internet connectivity-based residential environment, small requests by the end user often result in large transfers of data in the downstream direction. ADSL is a direct result of the asymmetric nature of the Internet and the needs of the end user, and was originally designed for video-on-demand applications.

ADSL possesses some distinct advantages when compared to traditional analog modems. One of them is the ability to operate alongside existing Plain Old Telephone Service (POTS) on a single pair of wires without disruption. POTS is the basic service that provides all phone lines with access to the Public Switched Telephone Network (PSTN). POTS provides the means for all voice-band related applications and technologies, such as telephony, caller identification, call waiting, analog facsimile, analog modem, etc.. ADSL systems allow the end user to access any POTS associated services and ADSL services simultaneously. ADSL also has the ability to dynamically adapt to varying channel conditions. ADSL systems automatically measure the characteristics of the channel and decide upon an appropriate data rate that can be effectively maintained according to a predefined acceptable bit-error rate (BER).

ADSL: ANSI T1.413-1998

The American National Standards Institute (ANSI) Telecommunications Committee created the first standardized ADSL specification. The current version of this specification is ANSI T1.413-1998 “Network and Customer Installation Interfaces – Asymmetric Digital Subscriber Line (ADSL) Metallic Interface.” ANSI T1.413-1998 defines the minimum set of requirements for satisfactory performance of ADSL systems utilizing the Discrete Multi-Tone (DMT) line code. The DMT line code, as defined in ANSI T1.413-1998, divides the useful bandwidth of the standard two wire copper medium used in the PSTN, which is 0 to 1104kHz, into 256 separate 4.3125kHz wide bins called sub-carriers. Each sub-carrier is associated with a discrete frequency, or tone, indicated by 4.3125kHz * n, where n = 1 to 256, and is essentially a single distinct data channel.

A maximum of 255 sub-carriers can be used to modulate data in the downstream direction. Sub-carrier 256, the downstream Nyquist frequency, and sub-carrier 64, the downstream pilot frequency, are not available for user data, thus limiting the total number of available downstream sub-carriers to 254. Each of these 254 sub-carriers can support the modulation of 0 to 15 bits. Since the ADSL DMT data frame rate is 4000 frames per second, the maximum theoretical downstream data rate of an ADSL system is 15.24Mbps. Due to limitations in system architecture, specifically the maximum allowable Reed-Solomon codeword size (255 bytes), the maximum achievable downstream data rate is 8.16Mbps. However, in real world systems at least one byte of each Reed-Solomon codeword will be used for framing overhead, thus limiting the maximum achievable downstream data rate to 8.128Mbps.

The limitation of maximum allowable Reed-Solomon codeword size can be overcome if the interleaved data path is used (with S=1/2, where S is the number of data frames per RS codeword). Using the interleaved data path, which will be discussed in detail later, two Reed-Solomon codewords can be mapped to a single FEC output data frame. This equates to a maximum Reed-Solomon codeword size of ~510 bytes, which results in a maximum supportable downstream data rate of ~16Mbps. It should be noted that although the S=1/2 method yields a maximum supportable downstream data rate of 16Mbps, the theoretical maximum downstream data rate remains 15.24Mbps due to the fact that DMT systems are limited to 254 sub-carriers, each of which is capable of modulating a maximum of 15 bits. It should also be noted that support for this mode of operation is optional. See section 6.6.3 of ANSI T1.413-1998 for more information on this mode of operation.

Similarly, a maximum of 31 low frequency sub-carriers can be used to modulate data in the upstream direction. Sub-carrier 32, the upstream Nyquist frequency, and sub-carrier 16, the upstream pilot frequency, are again not available for user data, limiting the total number of available upstream sub-carriers to 30. Each of these 30 sub-carriers can support the modulation of 0 to 15 bits. Since the ADSL DMT data frame rate is 4000 frames per second, the maximum theoretical upstream data rate of an ADSL system is 1.8Mbps. Again, due to limitations in system architecture, specifically the POTS splitter cut-off frequencies and the duplexing method used (FDM or echo cancellation), the maximum achievable upstream data rate is typically less than 1Mbps. Figure 3 shows the basic plot of a DMT ADSL system in the frequency domain with approximate frequencies.

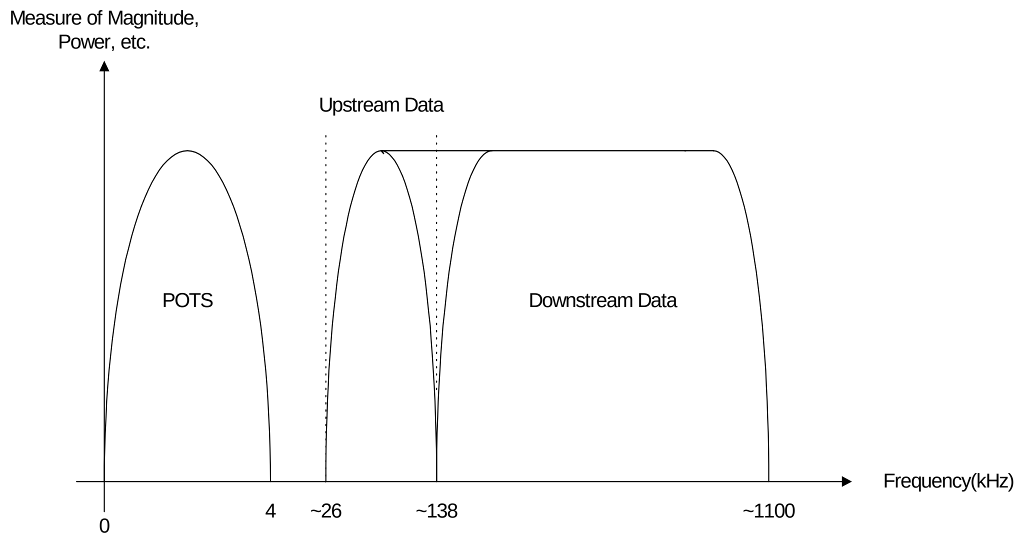


Figure 3. DMT based ADSL in the frequency domain.

Frequency Division Multiplexing (FDM) is a duplexing method that splits the available spectrum into two non-overlapping parts, one for upstream data and one for downstream data. FDM requires the analog hybrid circuit in each transceiver to effectively decouple, or split, the upstream and downstream portions of the analog DMT signal. The cut-off frequencies of these splitters are not formally defined and are therefore left to the discretion of the vendor. As a result, FDM splitting can adversely effect the upstream and downstream portions of the spectrum.

An optional duplexing method, echo cancellation, can also be utilized in ADSL systems. Echo cancellation allows the upstream and downstream portions of the spectrum to overlap, improving downstream performance by allowing more low attenuation low frequency sub-carriers to be utilized for downstream data transport. "An "echo" is a reflection of the transmit signal into the near end received. Echoes are of concern because the signals that correspond to both directions of digital transmission coexist on the twisted-pair transmission line, so that the echo is unwanted noise.[1]" The upstream and downstream portions of the signal are again decoupled by the analog hybrid circuit. Echo cancellation is then achieved by subtracting an estimate of the unwanted echo from the decoupled receive signal. In ADSL systems, good echo cancellers can, and must, achieve 70dB of rejection.

[1] Understanding Digital Subscriber Line Technology, pages 140 and 141.

As defined in ANSI T1.413-1998, DMT supports asynchronous (ATM) or synchronous (STM) based bearer services, “the transport of data at a certain rate without regard to its content, structure, or protocol,” through the use of bearer channels. A bearer channel is “a user data stream of a specified data rate that is transported transparently by an ADSL system in ASx or LSx, and carries a bearer service.” Bearer channels deal strictly with data rates and services and are logical channels that use the underlying sub-carriers as a transport mechanism. ANSI T1.413-1998 provides for the simultaneous transport of seven bearer channels, with up to four dedicated downstream bearer channels, denoted as ASx, where x = 1, 2, 3, or 4, and up to three upstream bearer channels, denoted as LSx, where x = 1, 2, or 3. ASx bearer channels are simplex, whereas LSx bearer channels are duplex and can be configured to carry both downstream and upstream data. For the transport of both STM and ATM data, Table 1 shows the bearer channel data range multiples for all ASx and LSx. Support for multiples higher than those shown in Table 1 and for multiples other than 32kbps (n bytes * 4000 data frames per second = n * 32kbps) is optional.

Bearer Channels	Lowest Required Multiple	Largest Required Multiple	Corresponding Highest Required Data Rate
AS0	1	$n_0 = 192$	6144 kbps
AS1	1	$n_1 = 144$	4608 kbps
AS2	1	$n_2 = 96$	3072 kbps
AS3	1	$n_3 = 48$	1536 kbps
LS0	1	$m_0 = 20$	640 kbps
LS1	1	$m_1 = 20$	640 kbps
LS2	1	$m_2 = 20$	640 kbps

Table 1. Required 32kbps multiples for transport of ATM and STM[2].

DMT based ADSL systems support two latency paths for data transmission, the interleaved path and the fast path (no data interleaving). The latency mode of an ADSL system does not need to be the same in both the downstream and upstream directions. For single latency, only support for bearer channel AS0 in the downstream direction and bearer channel LS0 in the upstream direction is required. See ANSI T1.413-1998 Section 5 for provisions regarding bearer channel latency path assignments.

Bearer channel data is partitioned into individual data frames, each data frame consisting of two parts, the fast data buffer and the interleaved data buffer. Each data buffer contains a certain number of bytes, according to Table 1, from each of the bearer channels that are in use and that are assigned to that latency path, plus overhead. There are four framing modes, 0, 1, 2, and 3, each mode aimed at reducing the total amount of overhead required. Modes 0 and 1 are classified as full overhead framing, with and without synchronization control capabilities, respectively, and modes 2 and 3 are classified as reduced overhead framing with separate fast and sync bytes and

[2] Taken from ANSI T1.413-1998.

with merged fast and sync bytes, respectively. Starting with framing mode 0, each mode requires progressively less overhead. For STM transport, support for framing mode 0 is required and support for framing modes 1, 2, and 3 is optional. Likewise, for ATM transport, support for framing modes 0 and 1 is required while support for framing modes 2 and 3 is optional. Reduced overhead framing modes apply "when there are only single channels in each direction, or secondarily, when only a single fast channel is in use and a single interleaved channel is used.[3]" All framing modes are formally defined in section 6.4 of ANSI T1.413-1998.

The basic structures of the fast and interleaved data buffers, in both the downstream and upstream directions, with full overhead framing (mode 0), are shown in Figures 4, 5, 6, and 7. These figures show representations of data frames at various stages in the transceiver reference model of Figure 9. These figures were taken from ANSI T1.413-1998, sections 6.4.1.2 and 7.4.1.2. The upstream data buffers differ from the downstream data buffers because only the duplex LSx bearer channels are available for upstream data transmission; therefore no ASx bytes are required. It should be noted that allocation of the AEX, LEX, fast, and sync bytes depend upon the selected framing mode and data buffer allocation; AEX and LEX bytes are used to identify the bearer channels used, fast and sync bytes are reserved for overhead. The use of these fields within the fast and interleaved data buffers is formally defined in ANSI T1.413-1998, sections 6.4.1.2 and 7.4.1.2.

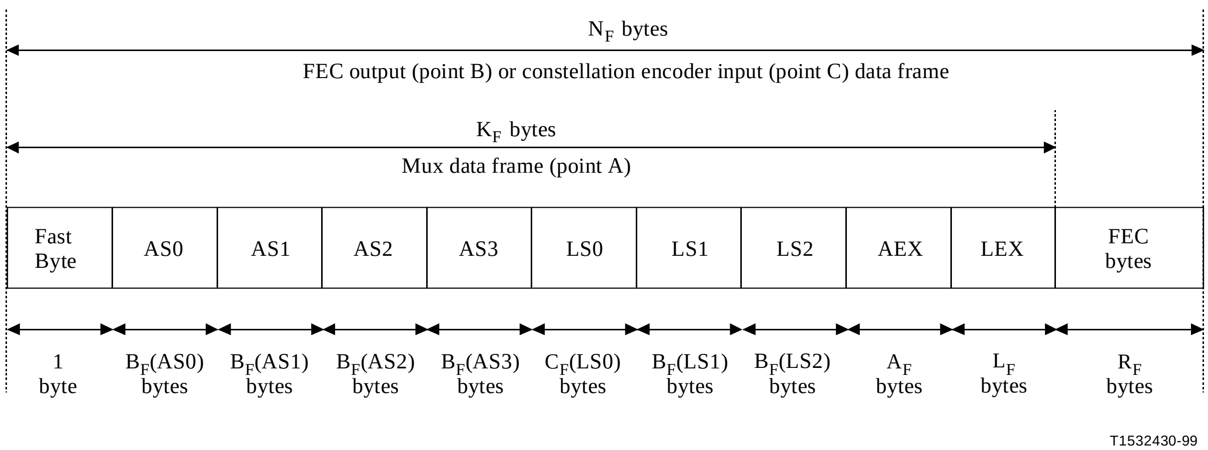


Figure 4. Fast data buffer – ATU-C transmitter.

[3] Summers, page 58.

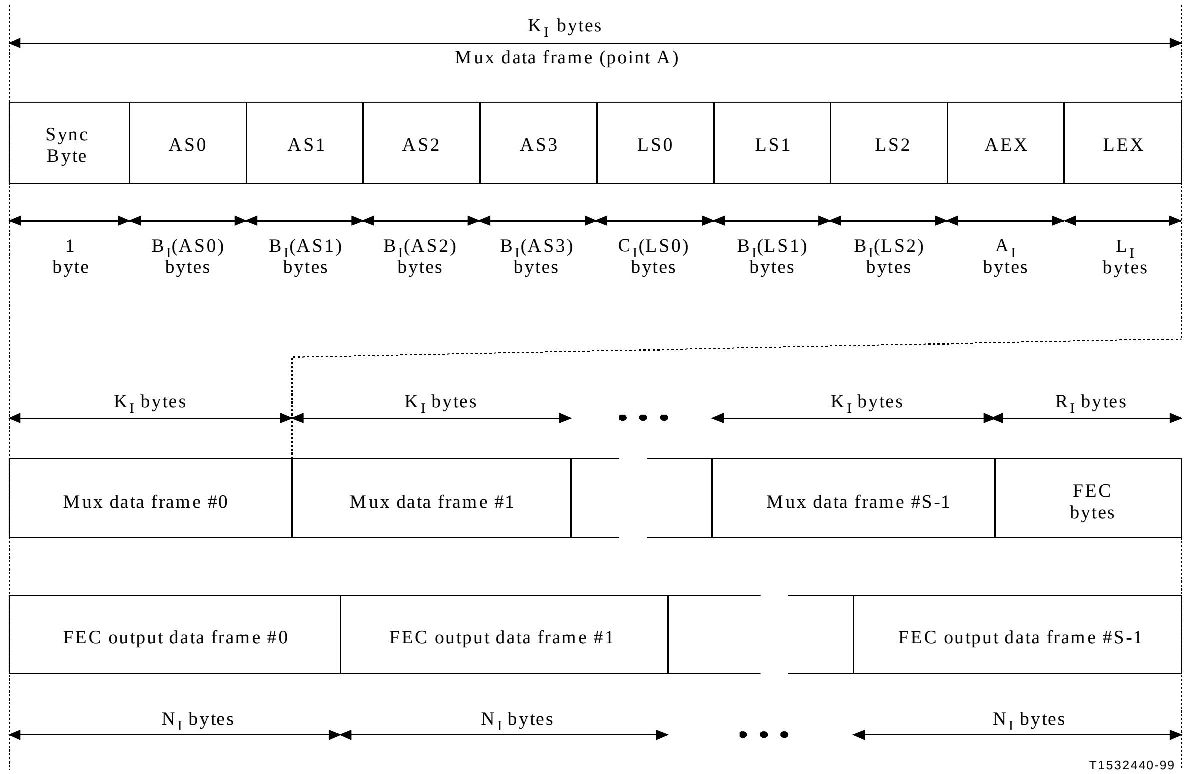


Figure 5. Interleaved data buffer – ATU-C transmitter.

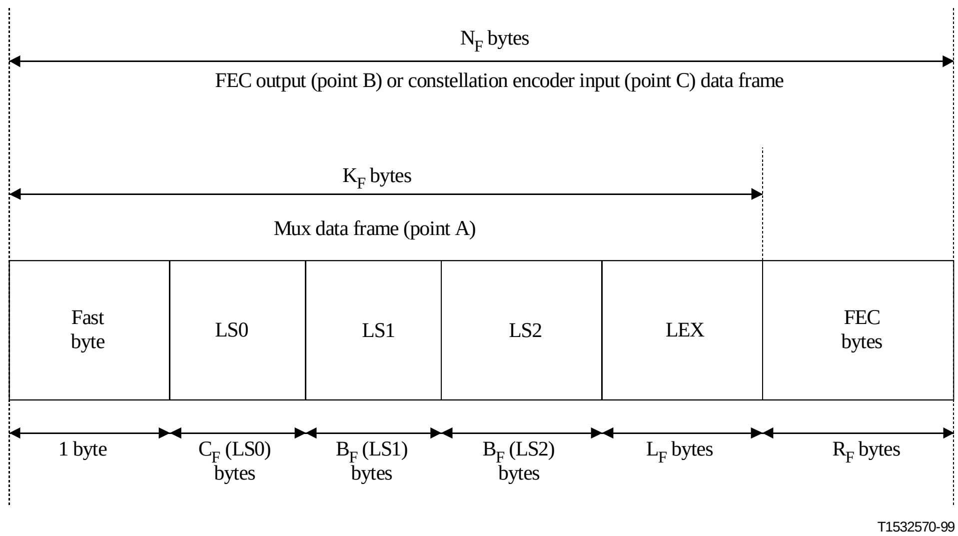


Figure 6. Fast data buffer – ATU-R transmitter.

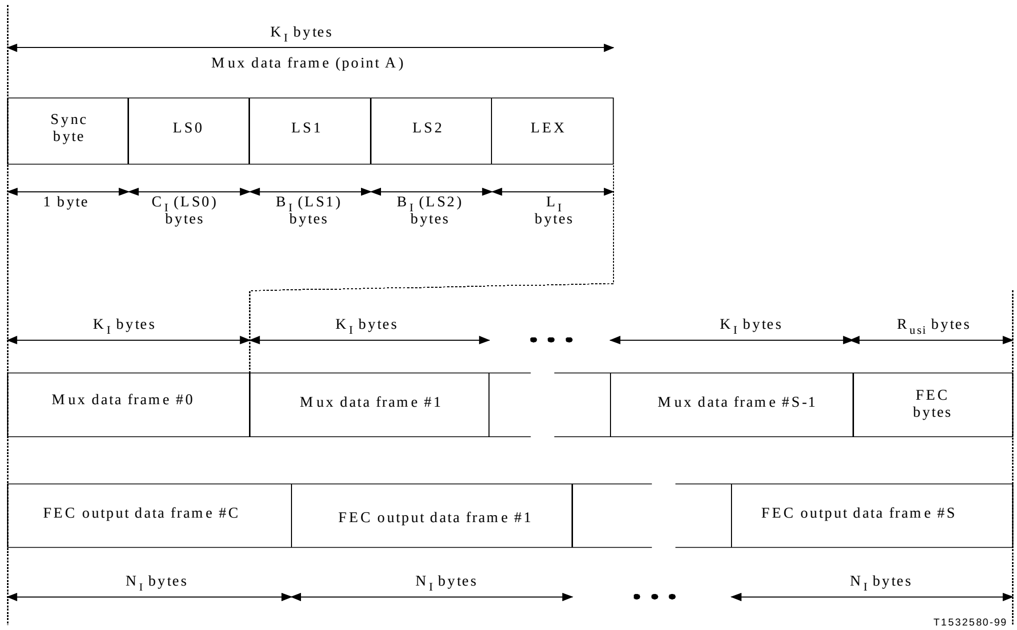


Figure 7.

Interleaved data buffer – ATU-R transmitter.

ADSL utilizes a superframe structure for data frame transmission. 68 DMT data frames, numbered from 0 to 67, are grouped together to form a superframe, as shown in Figure 8. Each superframe is actually 69 data frames; the 69th data frame is a synchronization symbol inserted by the DMT modulator to establish superframe boundaries. To allow for insertion of the synchronization symbol (while maintaining the 4000 frames per second data frame rate) the transmit frame rate is actually increased to 69/68 * 4000 frames per second. Figure 8 was taken from section 6.4.1.1 of ANSI T1.413-1998.

Figure 9 shows a basic block diagram of a typical DMT based ADSL transceiver.

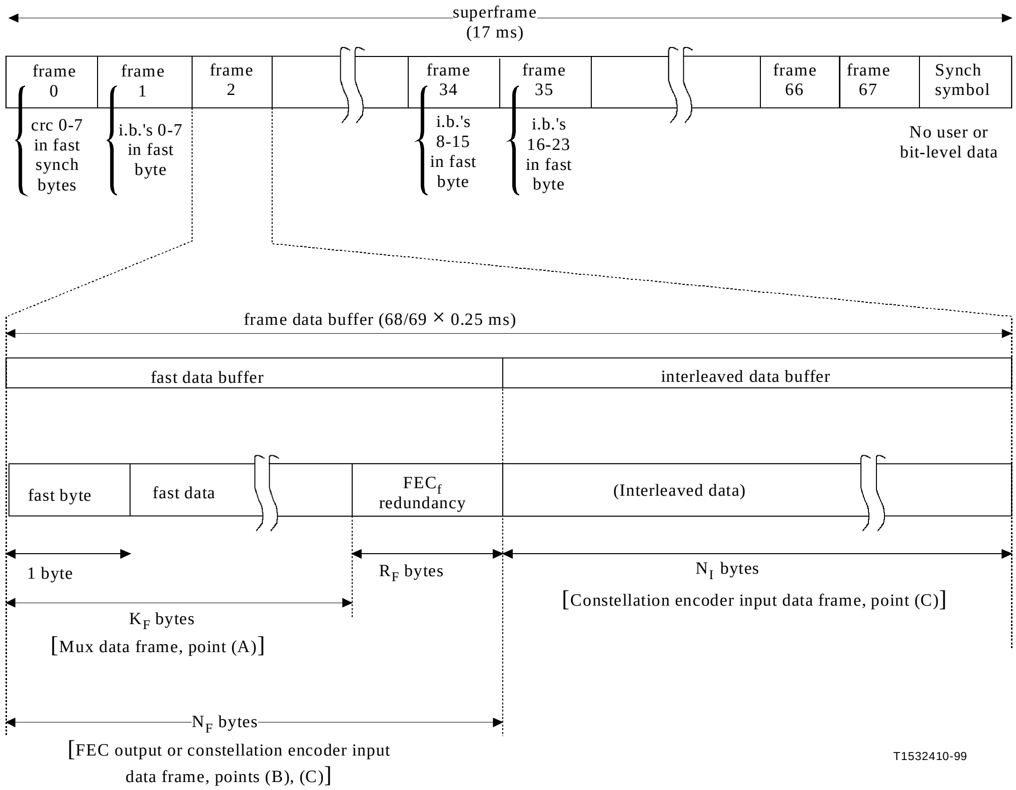
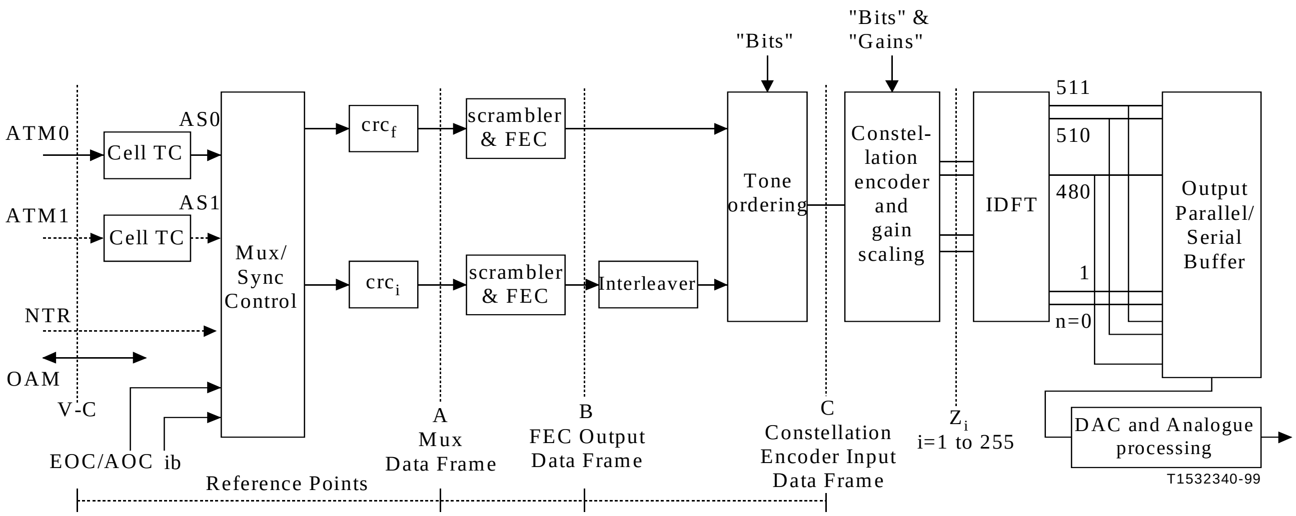


Figure 8. ADSL superframe structure.

A functional block diagram of a DMT based ADSL ATU-C transceiver is shown in Figure 9. The ATU-R transceiver is essentially the same, with the only major differences being the size of the IDFT block and the bearer channels utilized (LSx versus ASx). Figure 9 assumes ATM transport; STM based transceivers can be obtained from their ATM based counterparts by eliminating the ATM Cell Transmission Convergence (TC) blocks and utilizing the appropriate bearer channels. Figure 9 was taken from Section 4.2.2 of ANSI T1.413-1998.

With reference to Figure 9, the Cell TC block handles all ATM specific requirements, including header error control (HEC) generation, idle cell insertion, cell payload scrambling, bit timing and ordering, cell delineation, and HEC verification. The ATM Cell TC block essentially handles the conversion of ATM data to ADSL bearer channel data. Again, in ADSL systems transporting STM data, the ATM Cell TC block is not utilized. Following the ATM Cell TC block, data is routed, through bearer channels, to the Mux/Sync Control block. The Mux/Sync Control block synchronizes data to the 4kHz ADSL data frame rate and multiplexes data into the fast and/or interleaved data buffers.



NOTE – Solid versus dashed lines are used to indicate required versus optional capabilities respectively. This figure is not intended to be complete in this respect, see clauses 6 and 7 for specific details.

Figure 9. Reference model for an ATU-C transceiver supporting ATM transport.

A cyclic redundancy check (CRC) is performed on the contents of each data buffer on a superframe basis. The CRC that is generated from each data buffer per superframe will be transmitted in the fast and sync bytes, of the fast and interleaved data buffers, respectively, of the first data frame (frame 0) of the following superframe. The fast and sync bytes of the remaining data frames within a specific superframe are used to transmit embedded operations channel (EOC), synchronization, ADSL overhead control (AOC) channel, and indicator bit information.

The following algorithm is used to scramble the contents of each data buffer:

$$d_n' = d_n \oplus d_{n-18}' \oplus d_{n-23}'$$

where d_n is the n-th output from the fast or interleaved data buffer, and d_n' is the n-th output from the corresponding scrambler. "Scrambling . . . eliminates repetitious data patterns such as all 0's or 1's. Random bit patterns reduce both radio frequency interference and signal crosstalk from one pair to another pair in the cable.[4]" Forward error correction (FEC) capabilities are also applied to the scrambled data in each data buffer through the use of Reed-Solomon (RS) Coding. RS check bytes are added to a certain number of data bytes to produce an RS codeword. As mentioned earlier, the maximum RS codeword size is 255 bytes. Sections 6.6 and 7.6 of ANSI T1.413-1998 outline the minimum RS coding capabilities of an ATU-C and ATU-R, respectively.

Convolutional interleaving is performed only on data in the interleaved data buffer. Data in the fast data buffer differs only in that it is not interleaved, and therefore "fast" because there is no latency associated with the interleaving process. Interleaving essentially "weaves" RS codewords (data frames with RS check bytes appended) together to reduce the effects of noise on data transmission. Interleaving spreads the damage, if you will, from an impulsive noise hit across multiple RS codewords, reducing the likelihood that consecutive data bytes within a single RS codeword will be errored. For example, instead of having 10 bytes in a single RS codeword corrupted, which may be too large an error too to be corrected by FEC, 1 byte in 10 consecutive RS codewords will be corrupted, greatly reducing the size of the error per RS codeword and allowing for correction by FEC. Although interleaving provides better error correction capabilities, it introduces a significant amount of delay to the data path.

The next block in the reference model of Figure 9 is the “Tone Ordering” block. Tone ordering assigns the bits in both the fast and interleaved data buffers to individual sub-carriers according to a pre-defined ordered “bits and gains” table. The original “bits and gains” table is determined during initialization and essentially defines the number of bits each sub-carrier can modulate and the relative gain of each sub-carrier. The ordered “bits and gains” table is a modified version (according to the tone ordering algorithm) of the original “bits and gains” table. The tone ordering algorithm first assigns all of the bits from the fast data buffer to the sub-carriers with the least number of bits assigned to them, and then assigns the bits in the interleaved data buffer to the sub-carriers with the greatest number of bits assigned to them. This algorithm ensures that the sub-carriers that are most likely to be clipped (clipping can be considered an impulsive error) by the digital-to-analog converter (DAC), which are the tones with the highest number of bits assigned to them, contain bits from the interleaved data buffer. Tone ordering allows impulsive clipping errors to be successfully corrected by the FEC. The tone ordering algorithm will vary slightly if trellis coding is enabled.

The “Constellation Encoder and gain scaling” block follows tone ordering in the reference model of Figure 9. Prior to the actual constellation encoding, a bit extraction algorithm extracts bits from the data frame buffer according to the ordered “bits and gains” table. A bit conversion algorithm is also implemented for the purpose of trellis coding, if trellis coding is enabled (the use of trellis coding is optional and effects both the tone ordering algorithm and the constellation encoding algorithm). During bit extraction, the bits from two consecutive ordered sub-carriers are used to form the binary word *u*, which is then converted, through a convolutional code, into two binary words *v* and *w*, which are input to the constellation encoder. Section 6.8 of ANSI T1.413-1998 contains specific information regarding trellis coding. For each sub-carrier, the constellation encoder selects a point (in the complex plane) from a grid of points based on the binary data assigned to that sub-carrier. Constellation encoding converts binary data into a complex number. Gain scaling, as per the “bits and gains” table, is applied to all data carrying sub-carriers - each complex point is multiplied by a gain in the range of approximately 0.75 to 1.33; i.e. $Z_i = g_i * (X_i + jY_i)$. It should be noted that gain scaling is not used during transmission of the synchronization symbol.

Following the “Constellation Encoder and gain scaling” block, modulation is performed by the inverse discrete Fourier Transform (IDFT) operation. Recall that in the downstream direction the “Constellation Encoder and gain scaling” block produces 255 complex Z_i values. To produce real values, x_n, using the IDFT and the 255 complex Z_i values, Z must have Hermitian symmetry, thus for i =257 to 511, $Z_i = Z_{(512-i)}*$. Thus, in the downstream direction, the IDFT generates 512 real values, x_n. Likewise, in the upstream direction the “Constellation Encoder and gain scaling” block produces 31 complex Z_i values. To produce real values, x_n, using the IDFT and the 31 complex Z_i values, Z must have Hermitian symmetry, thus for i =33 to 63, $Z_i = Z_{(64-i)}*$. Thus, in the downstream direction, the IDFT generates 64 real values, x_n. Sections 6.11.2 and 7.11.2 of ANSI T1.413-1998 define modulation by the IDFT in more detail. Parallel to serial conversion, DAC processing, and analog processing follow the modulation block.

Many of the functional blocks mentioned above require parameters to be determined and set that depend on specific knowledge of the communication channel. These parameters are negotiated during initialization, when an ADSL ATU-R is first connected to an ATU-C (DSLAM). Initialization will be discussed in detail later.

ADSL: ITU-T G.992.1 (G.dmt):

ITU-T G.992.1 (G.dmt) is the ITU's nearly identical version of ANSI T1.413-1998. The major differences between G.dmt and ANSI T1.413-1998 are: G.dmt defines DMT based ADSL operation in different regions such as North America, Europe, and Asia, whereas ANSI T1.413-1998 defines operation only in North America, and G.dmt utilizes ITU-T G.994.1 (G.hs) for the handshaking portion of initialization, whereas ANSI T1.413-1998 does not. G.dmt addresses different region specific requirements for DMT based ADSL systems because of the inherent differences in telecommunications systems in different parts of the world. For example, the telecommunication infrastructure in Asia is quite different than that of North America or even Europe. There are also different requirements in different parts of the world, such as support for DMT based ADSL systems operating in the frequency band above ISDN, which is very common in Europe. Since G.dmt is an international specification it must address all of these differences and issues. The use of G.hs in G.dmt compliant devices is necessitated by the need to negotiate through all of the possible options defined in G.dmt, which may all be supported by a single ATU-R or ATU-C.

ADSL: ITU-T G.992.2 (G.lite)

ITU-T G.992.2 (G.lite) is a variation of G.dmt and ANSI T1.413-1998. The intent of G.lite is to provide a less complicated version of DMT based ADSL that can be easily deployed and installed by the end user. The tradeoff associated with decreased complexity is decreased bandwidth. G.lite supports maximum data rates of about 2Mbps downstream an 800kbps upstream – far less than its "full rate" counterpart. There are a few other major differences between G.lite and G.dmt, such as: G.lite does not require a splitter at the remote end (ATU-R); G.lite utilizes only the interleaved data path, thus eliminating many options associated with dual latency and with the fast data path in general; G.lite incorporates extra measures to combat impedance problems due to the omission of the remote splitter. It should be noted that unlike G.dmt, there is no ANSI based G.lite type specification.

SHDSL: ITU-T G.991.2 (G.shdsl)

ITU-T G.991.2 (G.shdsl) is a symmetric type of DSL that, like G.dmt and G.lite, utilizes G.hs as a handshake procedure. G.shdsl is rate adaptive, providing from 192kbps to 2.312Mbps, in increments of 8kbps, in both directions. G.shdsl does not support analog splitting and simultaneous POTS or ISDN in the same pair of wires, but is designed to be spectrally compatible with other DSL technologies. G.shdsl does allow for an optional 4 wire mode of operation, and also defines signal regenerators, or repeaters, for the purpose of alleviating the most common limitation in DSL systems: loop length. G.shdsl is based on 16-level trellis coded pulse amplitude modulation (16 TCPAM), reducing overall complexity versus the DMT modulation scheme utilized by ADSL systems. Although G.shdsl utilizes G.hs, initialization in G.shdsl systems is quite different than in ADSL systems. This difference is mainly due to the differences in physical layer modulation schemes.

REFERENCES

[1] Starr, Thomas, John M. Cioffi, and Peter J. Silverman. Understanding Digital Subscriber Line Technology. Upper saddle River, NJ: Prentice-Hall, Inc., 1999.

[2] American National Standards Institute, Inc., Standards Committee T1 – Telecommunications. Standard/Recommendation T1.413-1998, Network and Customer Installation Intefaces – Asymmetric Digital Subscriber Line (ADSL) Metallic Interface. November 1998.

[3] Summers, Charles K. ADSL: Standards, Implementation, and Architecture. New York, NY: CRC Press LLC, 1999. [58]