# - Cable and DSL Technologies -

## ***Cable Technologies***

Traditional **Cable Television (CATV)** provided a unidirectional transport of video channels to home consumers. Competition from satellite TV providers forced the development of a full-duplex data infrastructure, across a **Hybrid Fiber/Coaxial (HFC)** network.

Open-standard specifications for data communication across this Broadband Cable network are defined by the **Data Over Cable Service Interface Specification (DOCSIS)**. DOCSIS was developed by the **Multimedia Cable Network System (MCNS) Partners** (now CableLabs)**,** and defines the **Layer-1** and **Layer-2** specifications for cable communication.

There are three current implementations of this specification:

- DOCSIS v1.0
- DOCSIS v1.1
- DOCSIS v2.0

While DOCSIS v2.0 is the *newest* specification, the most widespread implementation (currently) is still v1.1. (DOCSIS v3.0 is under development, presently). DOCSIS defines two key devices necessary for broadband cable communication:

- **Cable Modem Termination System (CMTS) –** Usually located at the central office (or **headend)**, modulates and demodulates signals to and from the Cable Modem. A single CMTS can accommodate thousands of cable modems, and provides the connection point to the Internet backbone.

- **Cable Modem (CM)** – Usually located at the customer premises, terminates the cable line, and modulates/demodulates signals to and from the CMTS.

Data flowing from the CMTS to the Cable Modem is deemed **downstream traffic**. Data from the Cable Modem to the CMTS is **upstream traffic.**

A DOCIS binary configuration file provides the appropriate ISP parameters for cable modems to connect to the network.

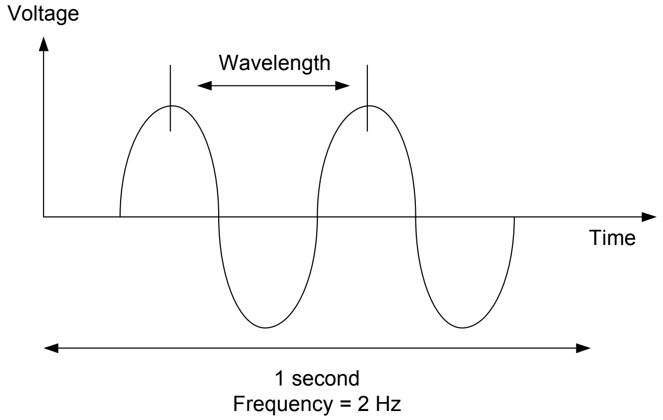
(Reference: http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/cable.pdf; http://www.cablemodem.com/)

## ***Cable Modulation***

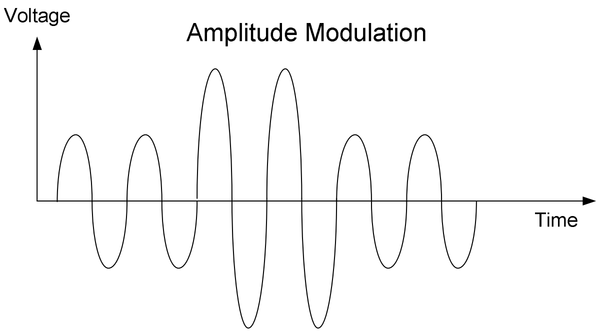
Recall that **downstream** traffic refers to data flowing *from* the CMTS *to* the cable modem. The actual *physical* signal sent across the HFC network is in the form of **Radio Frequency (RF) waves**, which constitute a segment of the electromagnetic spectrum.

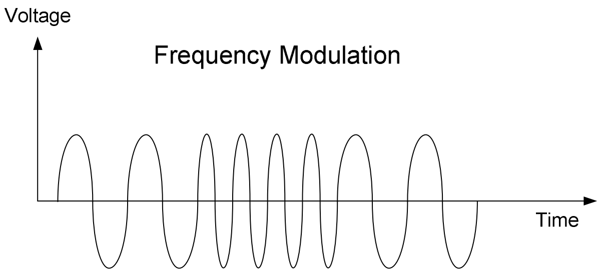
Electromagnetic signals are measured by the **frequency** of their waves, or the number of cycles completed per a given time period. The standard frequency measurement unit is the **hertz (Hz),** or one *cycle* per *second*.

**Wavelength** refers to the physical length of a frequency cycle. Frequency and wavelength are *inversely* related to each other. The higher the frequency, the shorter the wavelength will be.



**Modulation** is the method of altering a signal (usually by varying its **amplitude, frequency,** or **phase),** to convey a message or data stream:





(Reference: http://www-scm.tees.ac.uk/users/u0000408/Async/async.htm)

## ***Cable Modulation (continued)***

IP traffic across the cable infrastructure is typically encapsulated as **MPEG (Moving Picture Experts Group)** frames. CMTS devices and Cable Modems *modulate* MPEG frames onto an RF signal, using **QAM (Quadrature Amplitude Modulation)**.

QAM implementations typically used for downstream traffic include:

- 64-QAM (6 bits per **symbol -** amount of data sent at a given interval)
- 256-QAM (8 bits per symbol)

**Upstream traffic** refers to data from the Cable Modem to the CMTS, and typically utilizes **QPSK (Quadrature Phase Shift Keying)** modulation, or **16-QAM.**

(Reference: http://www.indiana.edu/~emusic/acoustics/frequency.htm; http://en.wikipedia.org/wiki/Quadrature_amplitude_modulation)

## ***Cable Standards***

Three cable standards are used heavily worldwide:

- **NTSC (National Television Standards Committee)** – cable standard used in the United States.
- **SECAM (Système Electronic Couleur avec Mèmoire) –** cable standard used in France and other European countries.
- **PAL (Phase-Alternating Line)** – cable standard used in most other countries.

The NTSC standard allocates 6 megahertz (MHz) for each TV channel. Thus, when “flipping” through TV channels, one is actually scanning up or down RF frequencies.

Downstream data traffic is *also* assigned a 6 MHz channel. Upstream data traffic can utilize as small as a 2 MHz channel.

Traffic is generally assigned the following frequencies:

- 5 MHz to 42 MHz – allocated for *upstream* traffic
- 50 MHz to 860 MHz – allocated for *downstream* traffic.

(Reference: http://www.cisco.com/univercd/cc/td/doc/product/cable/cab_rout/cr72hig/ub72fqcy.htm)

## ***Components of a Cable Modem***

Most standard cable-modems are comprised of several key parts:

- Tuner
- Demodulator
- Modulator
- MAC
- CPU
- Local interface (Ethernet or USB)

The **tuner** physically terminates the coaxial cable, and utilizes a *diplexer* to simultaneously manage both upstream and downstream frequencies.

The **demodulator** provides several functions for *downstream* traffic:

- Demodulates the QAM encoding from the signal.
- Provides an analog-to-digital conversion.
- Performs error correction on the converted digital signal.
- Synchronizes the MPEG frames to ensure data is delivered in the correct order.

The **modulator** performs the encoding necessary to send *upstream* traffic.

(Reference: http://www.cable-modems.org/tutorial/; http://computer.howstuffworks.com/cable-modem.htm)

## ***Cable Modem Initiation***

When a cable modem is powered on, it communicates with the CMTS and performs the following provisioning process:

1. Scans for a downstream channel and establishes synchronization with the CMTS.
2. Obtains upstream parameters from the CMTS.
3. Performs ranging and automatic adjustments, including the required power levels (usually between 40 – 50 dBmV). Layer-1 and Layer-2 connectivity is established at this stage.
4. Establishes IP connectivity using DHCP.
5. Establishes Time of Day (ToD).
6. Acquires the DOCSIS parameter file via TFTP.
7. Registers with the CMTS.
8. Initializes baseline privacy.

Services such as TFTP, DHCP, and ToD must be made available at the provider's headend for the provisioning process to complete successfully.

Once the process is complete, the customer equipment (such as a router or workstation) can request network layer information from the cable provider, including:

- An IP address (usually public)
- A subnet mask
- A default gateway

(Reference: http://www.cisco.com/warp/public/109/initialization_pdf_wallchart.pdf)

## DSL Technologies

**DSL (Digital Subscriber Line)** technologies utilize unused (and *higher*) frequencies on existing telephone lines for data communication. Both voice and data services can be used *simultaneously* on the same copper wire. DSL is a **Layer-1** technology only.

DSL service requires the use of a **DSL modem** at the customer premises. The DSL modem connects across the POTS network to a **DSL access multiplexer (DSLAM)** at the provider's Central Office (CO). A single DSLAM can service a large number of DSL modems.

Data flowing from the DSLAM to the DSL modem is deemed **downstream traffic**. Data from the DSL modem to the DSLAM is **upstream traffic.**

A filter is often used on lines connecting to a phone or fax to prevent the data signal (intended for the DSL modem) from interfering with analog voice communication.

There are two fundamental types of DSL:

- **ADSL (Asymmetric DSL)** – downstream rate is higher than upstream rate. ASDL is the most common DSL implementation.
- **SDSL (Symmetric DSL)** – downstream/upstream rates are identical

DSL is subject to distance limitations – usually a maximum of 18,000 feet. Additionally, the further the distance a customer is from the provider's CO, the slower the service.

Factors that can affect DSL service:

- **Signal attenuation** – the signal becomes progressively weaker as the distance increased from the CO.
- **Bridge Taps**
- **Load Coils**
- **Wire Gauge** – thicker wire is required for higher DSL speeds.
- **Impedance mismatch**
- **Crosstalk** – refers to noise introduced onto a wire from another closely bundled wire.
- **RF Interference**

## ***Flavors of DSL Technologies***

Recall that the two fundamental types of DSL are **ADSL** and **SDSL**. Each implementation of DSL supports multiple *flavors*, referred to collectively as *x*DSL. Flavors of ADSL include:

- **Standard ADSL** – provides a faster downstream rate (usually up to 8 Mbps) than upstream rate (usually up to 1 Mbps). Supports simultaneous use of voice and data on a wire.

- **RADSL (Rate-Adaptive DSL)** – dynamically adjusts the data rate depending on the current condition of the local loop. Supports simultaneous use of voice and data on a wire.

- **VDSL (Very-High-Rate DSL) –** provides very high speed service (up to 52 Mbps downstream, and 16 Mbps upstream) over much shorter distances (less than 5,000 feet). The **Cisco Long Reach Ethernet (LRE)** standard is based on VDSL. Supports simultaneous use of voice and data on a wire.

Flavors of SDSL include:

- **SDSL** – generic term referring to any *x*DSL that offers equal upstream and downstream data rates.

- **HDSL (High-Data-Rate DSL) –** delivers between 1.5 and 2.3 Mbps of symmetrical upstream and downstream speeds. Does *not* support simultaneous use of voice and data on a wire.

- **IDSL (ISDN DSL) –** operates over ISDN BRI technology, with a maximum of 144 Kbps (2B channels + 1D channel). Unlike ISDN, IDSL does not require call setup. However, IDSL does *not* support simultaneous use of voice and data on a wire.

| ***Flavor*** | ***Type*** | ***Data Rate (down/up)*** | ***Voice & Data?*** | ***Distance*** |
|---|---|---|---|---|
| ADSL | Asymmetric | 8 Mbps / 1 Mbps | Yes | 18,000 feet |
| RADSL | Asymmetric | Variable | Yes | 18,000 feet |
| VDSL | Asym./Sym. | 52 Mbps / 16 Mbps | Yes | ~ 4,500 feet |
| HDSL | Symmetric | 2.3 Mbps / 2.3 Mbps | No | 12,000 feet |
| IDSL | Symmetric | 144 Kbps / 144 Kbps | No | > 20,000 feet |

(Reference: http://electronics.howstuffworks.com/vdsl.htm)

## ***DSL Modulation***

The three most common ADSL modulation types are as follows:

- **CAP (Carrierless Amplitude Phase)**
- **DMT (Discrete MultiTone)**
- **G.Lite DMT** (or **Splitterless DMT)**

**CAP** modulation separates the signal into three **single-channel** bands. Notice that a buffer has been built in-between each band, to reduce possible interference:

- 0 kHz to 4 kHz – allocated for *POTS* (voice) services.
- 25 kHz to 160 kHz – allocated for *upstream* data traffic.
- 240 kHz to 1.5 MHz – allocated for *downstream* data traffic.

**DMT** (ANSI T1.413 or ITU G992.1) has mostly replaced CAP as the predominant ADSL modulation standard. As with CAP, DMT modulation separates the signal three separate bands:

- 0 kHz to 4 kHz – allocated for POTS (telephony) services
- 20 kHz to 130 kHz – allocated for *upstream* data traffic.
- 140 kHz to 1 MHz – allocated for *downstream* data traffic.

However, DMT additionally segments all three bands into 256 equal *subchannels* of 4 kHz, as opposed to CAP's single-channel implementation. If the quality of a particular subchannel is impaired due to noise, DMT will dynamically shift traffic to a different channel.

**G.Lite DMT** (ITU G992.2) is mostly used in home consumer markets, and allows the use of simultaneous voice/data **without splitters**. G.Lite supports only 128 channels, and thus cannot attain the same speeds as standard DMT.

## ***Layer-2 DSL Protocols***

The DSL standard is a Layer-1 technology. The most common Layer-2 protocols used over DSL include:

- **RFC 1483 Bridging**
- **Point-to-Point Protocol over Ethernet (PPPoE)**
- **Point-to-Point Protocol over ATM (PPPoA)**

RFC 1483 Bridging is not widely implemented due to scalability and security limitations. This guide will concentrate on PPPoE and PPPoA.

## ***PPPoE***

PPPoE combines the advantages of PPP (**authentication** and **link-control**) with Ethernet (robust **bridging**). The PPP frame is encapsulated *within* an Ethernet frame.

PPPoE utilizes the following PPP components:

- **HDLC** – for encapsulating packets into frames over serial lines.
- **LCP** – for establishing, maintaining, and terminating the link.
- **NCP** – allows multiple Layer-3 protocols (such as IP and IPX) to be encapsulated into frames.

The PPPoE session is initiated by a PPPoE-enabled client (usually a workstation or router) at the customer premises. The PPPoE client must *authenticate* with the PPPoE server (usually the DSLAM), before the session can be fully established.

The PPPoE connection process progresses through the following steps:

- The PPPoE client *broadcasts* out a PPPoE Active Discovery Initiation (PADI) packet, and specifies the type of service it requires.
- The PPPoE server responds with a *unicast* PPPoE Active Discovery Offer (PADO) packet.
- The PPPoE client then *unicasts* a PPPoE Active Discovery Request (PADR) packet.
- The PPPoE server then responds with a *unicast* PPPoE Active Discovery Session-Confirmation (PADS) packet.

After the PPPoE session has been established, PPP will perform its LCP and NCP tasks as normal.

To end the PPPoE session, either the client or server can send a PPPoE Active Discovery Terminate (PADT) packet.

The PPPoE MTU size ***must not*** be set higher than 1492 bytes. A typical ethernet frame is 1500 bytes, but the PPPoE and PPP Protocol ID information require 8 bytes:

| Ethernet Header | PPPoE Header (6 bytes) | PPP Protocol ID (2 bytes) | Data |
|---|---|---|---|

⟵ 1500 bytes ⟶

(Reference: http://www.cisco.com/en/US/tech/tk175/tk819/tsd_technology_support_protocol_home.html; http://www.cisco.com/warp/public/794/pppoe_arch.html)

## ***Configuring PPPoE***

A Cisco IOS device can function as a PPPoE client. First, Virtual Private Dialup Network (VPDN) must enabled for PPPoE:

```
Router(config)# vpdn enable

Router(config)# vpdn-group pppoe
Router(config-vpdn)# request-dialin
Router(config-vpdn-acc-in)# protocol pppoe
```

Next, the ethernet or ATM interface connecting to the ADSL modem must have PPPoE enabled:

```
Router(config)# interface ethernet 0/0
Router(config-if)# no ip address
Router(config-if)# pppoe enable
Router(config-if)# pppoe-client dial-pool-number 1

Router(config)# interface atm 0/0
Router(config-if)# no ip address
Router(config-if)# pvc 0/33
Router(config-if-vc)# pppoe-client dial-pool-number 1
```

The VPI/VCI must match the provider's. Next, a virtual DSL dialer interface must be created and configured:

```
Router(config)# interface dialer 1
Router(config-if)# encapsulation ppp
Router(config-if)# dialer pool 1
Router(config-if)# ip mtu 1492
Router(config-if)# ppp authentication chap
Router(config-if)# ppp chap hostname MYNAME
Router(config-if)# ppp chap password MYPASSWORD
```

Unlike ISDN, a *dialer-list* is not required for DSL, as the connection is always on. Notice that the *mtu* has been reduced to *1492*.

The provider's DSL modulation type is *auto*-detected by default, but can be manually specified on ATM interfaces:

```
Router(config)# interface atm 0/0
Router(config-if)# dsl operating-mode auto
Router(config-if)# dsl operating-mode ansi-dmt
Router(config-if)# dsl operating-mode itu-dmt
Router(config-if)# dsl operating-mode splitterless
```

(http://www.cisco.com/en/US/tech/tk175/tk15/technologies_configuration_example09186a0080126dc0.shtml)

## ***Additional PPPoE Configuration***

NAT Overload is often configured on the client PPPoE router:

```
Router(config)#  interface ethernet 0/1
Router(config-if)#  ip nat inside

Router(config)#  interface dialer 1
Router(config-if)#  ip nat outside

Router(config)#  access-list 10 permit ip 192.168.123.0 0.0.0.255
Router(config)#  ip nat inside source list 10 interface ethernet0/0 overload
```

DHCP can be configured to provide IP addressing to clients *behind* the router:

```
Router(config)#  ip dhcp pool MYPOOL
Router(dhcp-config)#  network 192.168.123.0 255.255.255.0
Router(dhcp-config)#  default-router 192.168.123.1
Router(dhcp-config)#  import all
```

The *import all* parameter allows DNS information pulled from the provider to be passed to internal clients.

Finally, traffic must be directed out the dialer interface using a default route:

```
Router(config)#  ip route 0.0.0.0 0.0.0.0 dialer1
```

## PPPoA

Unlike PPPoE (and RFC 1483 Bridging), which are both bridging technologies, PPPoA **routes** traffic from the PPPoA-enabled client (again, usually a workstation or router), to the provider's aggregating router.

All other PPP-related functions are identical between PPPoE and PPPoA.

## Configuring PPPoA

A Cisco IOS device can function as a PPPoA client. First, Virtual Private Dialup Network (VPDN) must enabled for PPPoA:

```
Router(config)#  vpdn enable

Router(config)#  vpdn-group pppoa
Router(config-vpdn)#  request-dialin
Router(config-vpdn-acc-in)#  protocol pppoa
```

Next, the ATM interface connecting to the ADSL modem must have PPPoA enabled:

```
Router(config)#  interface atm 0/0
Router(config-if)#  no ip address
Router(config-if)#  dsl operating-mode auto

Router(config-if)#  pvc 0/33
Router(config-if-vc)#  encapsulation aal5mus ppp dialer
Router(config-if-vc)#  dialer pool-member 1
```

The VPI/VCI must match the provider's. Next, a virtual DSL dialer interface must be created and configured:

```
Router(config)#  interface dialer 1
Router(config-if)#  encapsulation ppp
Router(config-if)#  dialer pool 1
Router(config-if)#  ppp authentication chap
Router(config-if)#  ppp chap hostname MYNAME
Router(config-if)#  ppp chap password MYPASSWORD
```

Notice that reducing the *mtu* to *1492* is not required for PPPoA.

## ***Troubleshooting PPPoE and PPPoA Connections***

To view the current status of PPPoE sessions:

```
Router#  show pppoe session
```

To view the establishment and termination of PPPoE sessions in real-time:

```
Router#  debug pppoe events
```

To debug PAP or CHAP authentication issues:

```
Router#  debug ppp authentication
```

To troubleshoot Layer-1 information, including whether the local router is communicated with the DSLAM:

```
Router#  show dsl interface atm 0/0
```