

Designing Internetworks for Multimedia

Networked multimedia applications are rapidly being deployed in campus LAN and WAN environments. From the corporate perspective, network multimedia applications, such as network TV or videoconferencing, hold tremendous promise as the next generation of productivity tools. The use of digital audio and video across corporate network infrastructures has tremendous potential for internal and external applications. The World Wide Web is a good example of network multimedia and its manifold capabilities.

More than 85 percent of personal computers sold are multimedia capable. This hardware revolution has initiated a software revolution that has brought a wide range of audio- and video-based applications to the desktop. It is not uncommon for computers to run video editing or image processing applications (such as Adobe Premiere and Photoshop) in addition to basic “productivity” applications (word processing, spreadsheet, and database applications).

The proliferation of multimedia-enabled desktop machines has spawned a new class of multimedia applications that operate in network environments. These network multimedia applications leverage the existing network infrastructure to deliver video and audio applications to end users, such as videoconferencing and video server applications. With these application types, video and audio streams are transferred over the network between peers or between clients and servers.

To successfully deliver multimedia over a network, it is important to understand both multimedia and networking. Three components must be considered when deploying network multimedia applications in campus LAN and WAN environments:

- *Bandwidth*—How much bandwidth do the network multimedia applications demand and how much bandwidth can the network infrastructure provide?
- *Quality of service*—What level of service does the network multimedia application require and how can this be satisfied through the network?
- *Multicasting*—Does the network multimedia application utilize bandwidth-saving multicasting techniques and how can multicasting be supported across the network?

This chapter addresses the underpinnings of effectively deploying network multimedia applications. Specifically, this chapter addresses the following topics:

- Multimedia Basics, including analog video, digital video, video compression, and digital audio standards
- Using Networked Multimedia Applications, including bandwidth and quality of service requirements
- Understanding Multicasting, including Internet Group Management Protocol, Distance Vector Multicast Routing Protocol, Multicast Open Shortest Path First, Protocol Independent Multicast, and Simple Multicast Routing Protocol

- Network Designs for Multimedia Applications, including traditional LAN designs, WAN designs, and high-speed LAN designs

Multimedia Basics

Much of today’s video starts out as an analog signal, so a working knowledge of analog standards and formats is essential for understanding digital video and the digitization process. The following topics are fundamental for understanding analog video:

- Broadcast Standards
- Video Signal Standards
- Video Storage Formats
- Digitizing Video
- Digitizing Audio

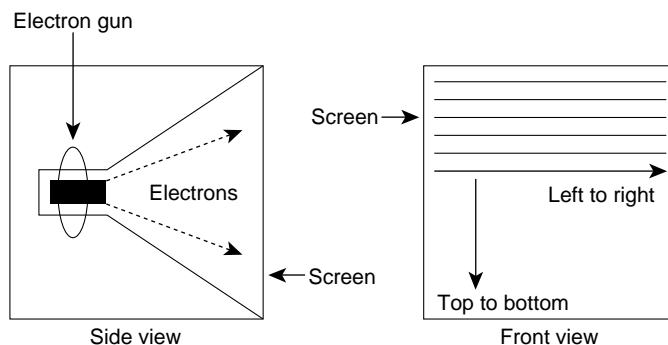
Broadcast Standards

The principal standards for analog broadcast transmission are as follows:

- *National Television Standards Committee (NTSC)*—The broadcast standard in Canada, Japan, the United States, and Central America. NTSC defines 525 vertical scan lines per frame and yields 30 frames per second. The scan lines refer to the number of lines from top to bottom on the television screen. The frames per second refer to the number of complete images that are displayed per second.
- *Phase Alternation Line (PAL)*—The broadcast standard in Europe and in the Middle East, Africa, and South America. PAL defines 625 vertical scan lines and refreshes the screen 25 times per second.
- *Système Electronique pour Couleur Avec Mémoire (SECAM)*—The broadcast standard in France, Russia, and regions of Africa. SECAM is a variant of PAL but it delivers the same number of vertical scan lines as PAL and uses the same refresh rate.

To produce an image on a television screen, an electron gun scans across the television screen from left to right moving from top to bottom, as shown in Figure 13-1.

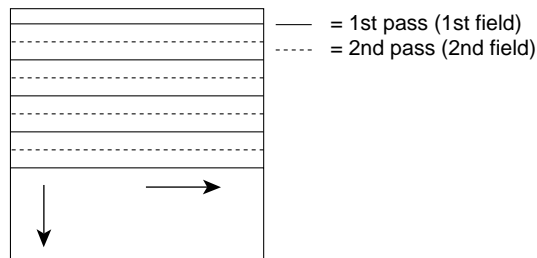
Figure 13-1 Television scan gun operation.



Early television sets used a phosphor-coated tube, which meant that by the time the gun finished scanning all the lines that the broadcast standard required, the lines at the top were starting to fade. To combat fading, the NTSC adopted an interlace technique so that on the first pass from top to bottom, only every other line is scanned. With NTSC, this means that the first pass scans 262 lines. The second pass scans another 262 lines that are used to fill in the rest of the TV image.

A frame represents the combination of the two passes, known as *fields*, as Figure 13-2 indicates. For NTSC to deliver 30 frames per second, it must generate 60 fields per second. The rate at which fields are delivered depends on the clocking source. NTSC clocks its refresh intervals from AC power. In the United States, the AC power runs at 60 hertz or 60 oscillations per second. The 60 hertz yields 60 fields per second with every two fields yielding a frame. In Europe, AC power clocks at 50 hertz. This yields 50 fields per second or 25 frames per second.

Figure 13-2 Interlace scan process.



Video Signal Standards

Black-and-white televisions receive one signal called *luminance* (also known as the *Y signal*). Each screen pixel is defined as some range of intensity between white (total intensity) and black (no intensity). In 1953, the NTSC was faced with the task of revising their standard to handle color. To maintain compatibility with older black-and-white sets, the NTSC set a color standard that kept the luminance signal separate and that provided the color information required for newer color television sets.

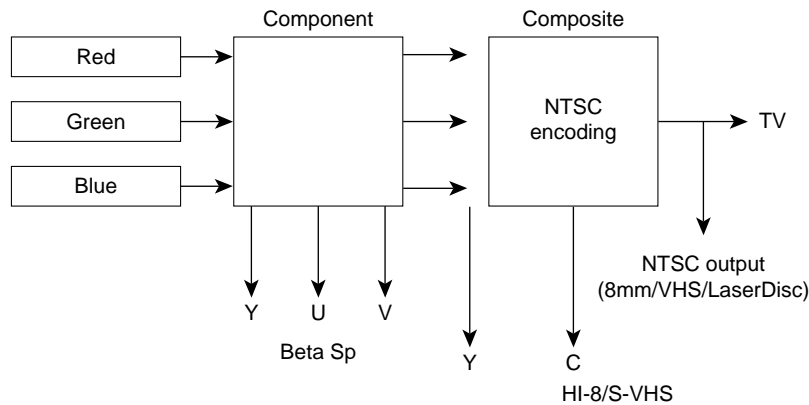
In the digital world, colors are typically expressed using red, green, and blue (RGB). The analog world has also embraced the RGB standard, at least on the acquisition side, where most cameras break the analog signal into RGB components.

Unfortunately, the NTSC could not use RGB as the color television standard because the old black-and-white sets could not decode RGB signals. Instead, they had to send a luminance signal for black-and-white sets and fill in the color information with other signals, called hue and saturation, (also known as the *U* and *V signals*). For this reason, digital color technology uses RGB and analog color technology, especially broadcast television, uses YUV (*Y*, *U*, and *V signals*).

Figure 13-3 traces an analog video signal from capture to NTSC output. On the far left is the RGB capture in which storage channels are maintained for each of the three primary colors. RGB, however, is an inefficient analog video storage format for two reasons:

- First, to use RGB, all three color signals must have equal bandwidth in the system, which is often inefficient from a system design perspective.
- Second, because each pixel is the sum of red, green and blue values, modifying the pixel forces an adjustment of all three values. In contrast, when images are stored as luminance and color formats (that is, YUV format), a pixel can be altered by modifying only one value.

Figure 13-3 RGB to NTSC encoding.



Component video maintains separate channels for each color value, both in the recording device and the storage medium. Component video delivers the highest fidelity because it eliminates noise that would otherwise occur if two signals were combined in one channel.

After NTSC encoding, the hue and saturation channels (U and V signals) are combined into one chrominance channel, the C channel. A video signal, called *S-Video*, carries separate channels for the luminance and chrominance signals. *S-Video* is also known as *Y/C video*.

All color and other information must be combined into one YUV channel, called the *composite signal*, to play on old black-and-white televisions. Technically, a composite signal is any signal that contains all the information necessary to play video. In contrast, any one individual channel of component or Y/C video is not sufficient to play video.

A video signal can be transmitted as composite, *S-Video*, or component video. The type of video signal affects the connector that is used. The composite signal, which carries all the information in one electrical channel, uses a one-hole jack called the *RCA Phono connector*. The *S-Video* signal, composed of two electrical channels, uses a four-pin connector called the *mini-DIN connector*. Finally, the component signal uses three connectors.

Video Storage Formats

There are six video storage formats: 8 mm, Beta SP, HI-8, Laserdisc, Super VHS (SVHS), and VHS. The six formats use different signals to store color. The composite signal provides the lowest quality because all signals are combined, which in turn has the highest potential for noise. The *S-Video* signal produces less noise because the two signals are isolated in separate channels. The component signal provides the highest quality signal because all components are maintained in separate channels. The image quality that a video capture board produces can only be as good as the signal it accepts. Table 13-1 lists the analog capture and storage standards for video.

Table 13-1 Analog Video Storage Formats

	Beta SP	SVHS/HI-8	VHS/8mm	Laserdisc
Color signal	Component	Y/C	Composite	Composite
Lines of resolution	750	400	200	400
Signal-to-noise ratio	50 db	47 db	47 db	47 db

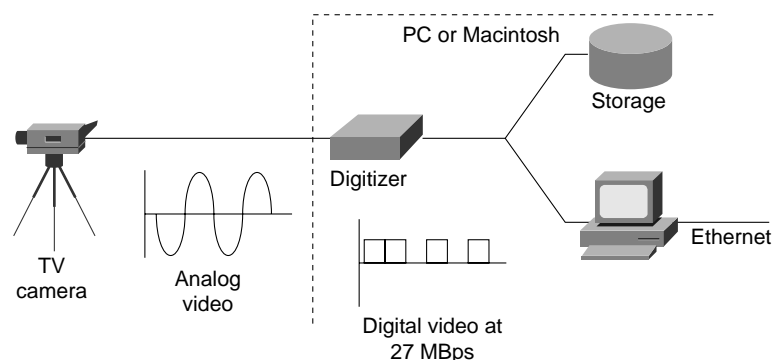
As Table indicates, the storage formats deliver different lines of resolution. Resolution is a measure of an image's quality. From the viewer's perspective, an image with higher resolution yields sharper picture quality than a lower resolution image.

Most consumer televisions display roughly 330 lines of horizontal resolution. Broadcast environments typically used high-end cameras to capture video. These cameras and their associated storage formats can deliver horizontal resolutions of approximately 700 lines. Each time a copy is made, the copied image loses some of its resolution. When an image is recorded in high-resolution, multiple generations of the video can be copied without a noticeable difference. When an image is recorded in a lower resolution, there is less room to manipulate the image before the viewer notices the effects.

Digitizing Video

Digitizing video involves taking an analog video signal and converting it to a digital video stream using a video capture board, as shown in Figure 13-4. Today, a variety of computer platforms, including PC, Macintosh, and UNIX workstations, offer video capture capabilities. In some cases, though, the capture equipment is a third-party add-on. The analog video source can be stored in any video storage format or it can be a live video feed from a camera. The source can be connected to the video capture card using any three connectors types (component, S-Video, or composite) depending on the connector type that the card supports.]

Figure 13-4 Analog-to-digital video conversion.



When capturing and digitizing video, the following components are critical:

- **Resolution**—The horizontal and vertical dimensions of the video session. A full-screen video session is typically 640 horizontal pixels by 480 vertical pixels. Full-screen video uses these dimensions because it yields the 4:3 aspect ratio of standard television. Of the 525 vertical scan lines in the NTSC standard, 483 lines are used to display video. The other lines are used for signaling and are referred to as the *vertical blanking interval*. Because the NTSC standard uses 483 vertical lines, capturing at 640 by 480 means that three lines are dropped during the digitization process.
- **Color depth**—The number of bits that are used to express color. At the high end is 24-bit color, which is capable of displaying 16.7 million colors and is the aggregate of 8 bits of red, 8 bits of green, and 8 bits of blue. The 8 bits are used to express color intensity from 0 to 255. Other common color depths are 16-bit and 8-bit, which yield roughly 65,000 and 256 colors, respectively.
- **Frame rate**—The number of frames that are displayed per second. To deliver NTSC-quality video, 30 frames per second are displayed. PAL and SECAM display 25 frames per second.

Based on these criteria, it is a simple mathematical operation to determine how much bandwidth a particular video stream requires. For example, to deliver uncompressed NTSC-quality digitized video to the network, a bandwidth of approximately 27 megabytes per second (Mbps) is needed. This number is derived from the following calculation:

$$640 \times 480 \times 3 \times 30 = 27.648 \text{ MBps (or 221.184 megabits per second [Mbps])}$$

where 640 and 480 represent the resolution in pixels, 3 represents 24-bit color (3 bytes), and 30 represents the number of frames per second.

As this calculation indicates, full-motion, full-color digital video requires considerably more bandwidth than today's typical packet-based network can support. Fortunately, two techniques reduce bandwidth consumption:

- Video Capture Manipulation
- Video Compression

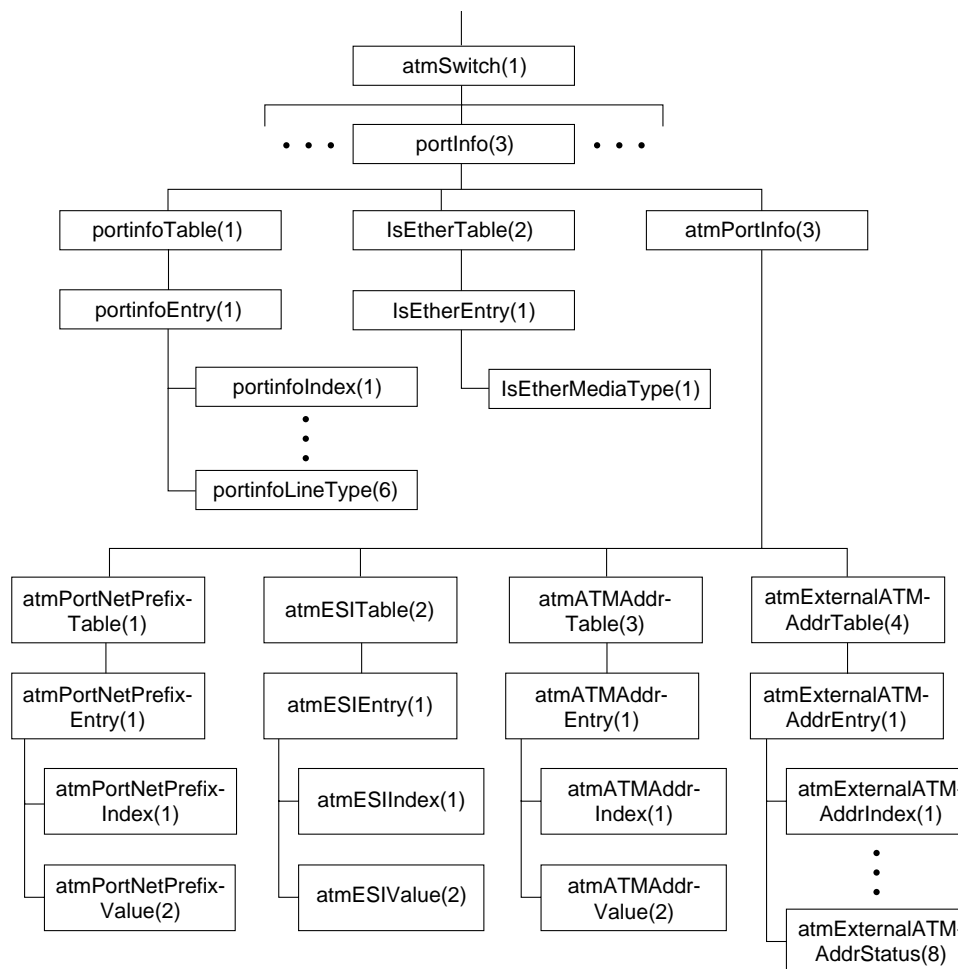
Video Capture Manipulation

Manipulating video capture parameters involves changing resolution, color depth, and frame rate. To reduce bandwidth consumption, all three variables are often changed. For example, some multimedia applications capture video at 320×240 with 8-bit color and at a frame rate of 15 frames per second. With these parameters, bandwidth requirements drop to 9.216 Mbps. Although this level of bandwidth is difficult for a 10-Mbps Ethernet network to achieve, it can be provided by 16-Mbps Token Ring, 100-Mbps Fast Ethernet, and other higher-speed technologies.

Video Compression

Video compression is a process whereby a collection of algorithms and techniques replace the original pixel-related information with more compact mathematical descriptions. Decompression is the reverse process of decoding the mathematical descriptions back to pixels for display. At its best, video compression is transparent to the end user. The true measure of a video compression scheme is how little the end user notices its presence, or how effectively it can reduce video data rates without adversely affecting video quality. An example of post-digitization video compression is shown in Figure 13-5.

Figure 13-5 Post-digitization video compression.



Video compression is performed using a CODEC (Coder/Decoder or Compressor/Decompressor). The CODEC, which can be implemented either in software or hardware, is responsible for taking a digital video stream and compressing it and for receiving a precompressed video stream and decompressing it. Although most PC, Macintosh, and UNIX video capture cards include the CODEC, capture and compression remain separate processes.

There are two types of compression techniques:

- *Lossless*—A compression technique that creates compressed files that decompress into exactly the same file as the original. Lossless compression is typically used for executables (applications) and data files for which any change in digital makeup renders the file useless. In general, lossless techniques identify and utilize patterns within files to describe the content more efficiently. This works well for files with significant redundancy, such as database or spreadsheet files. However, lossless compression typically yields only about 2:1 compression, which barely dents high-resolution uncompressed video files. Lossless compression is used by products such as STAC and Double Space to transparently expand hard drive capacity, and by products like PKZIP to pack more data onto floppy drives. STAC and another algorithm called Predictor are supported in the Cisco IOS software for data compression over analog and digital circuits.
- *Lossy*—Lossy compression, used primarily on still image and video image files, creates compressed files that decompress into images that look similar to the original but are different in digital makeup. This “loss” allows lossy compression to deliver from 2:1 to 300:1 compression.

Lossy compression cannot be used on files, such as executables, that when decompressed must match the original file. When lossy compression is used on a 24-bit image, it may decompress with a few changed pixels or altered color shades that cannot be detected by the human eye.

When used on video, the effect of lossy compression is further minimized because each image is displayed for only a fraction of a second (1/15 or 1/30 of a second, depending on the frame rate).

A wide range of lossy compression techniques is available for digital video. This simple rule applies to all of them: the higher the compression ratio, the higher the loss. As the loss increases, so does the number of artifacts. (An *artifact* is a portion of a video image for which there is little or no information.)

In addition to lossy compression techniques, video compression involves the use of two other compression techniques:

- *Interframe compression*—Compression between frames (also known as *temporal compression* because the compression is applied along the time dimension).
- *Intraframe compression*—Compression within individual frames (also known as *spatial compression*).

Some video compression algorithms use both interframe and intraframe compression. For example, Motion Picture Experts Group (MPEG) uses Joint Photographic Experts Group (JPEG), which is an intraframe technique, and a separate interframe algorithm. Motion-JPEG (M-JPEG) uses only intraframe compression.

Interframe Compression

Interframe compression uses a system of key and delta frames to eliminate redundant information between frames. Key frames store an entire frame, and delta frames record only changes. Some implementations compress the key frames, and others don't. Either way, the key frames serve as a reference source for delta frames. Delta frames contain only pixels that are different from the key frame or from the immediately preceding delta frame. During decompression, delta frames look back to their respective reference frames to fill in missing information.

Different compression techniques use different sequences of key and delta frames. For example, most video for Windows CODECs calculate interframe differences between sequential delta frames during compression. In this case, only the first delta frame relates to the key frame. Each subsequent delta frame relates to the immediately preceding delta frame. In other compression schemes, such as MPEG, all delta frames relate to the preceding key frame.

All interframe compression techniques derive their effectiveness from interframe redundancy. Low-motion video sequences, such as the head and shoulders of a person, have a high degree of redundancy, which limits the amount of compression required to reduce the video to the target bandwidth.

Until recently, interframe compression has addressed only pixel blocks that remained static between the delta and the key frame. Some new CODECs increase compression by tracking moving blocks of pixels from frame to frame. This technique is called *motion compensation* (also known as *dynamic carry forwards*) because the data that is carried forward from key frames is dynamic. Consider a video clip in which a person is waving an arm. If only static pixels are tracked between frames, no interframe compression occurs with respect to the moving parts of the person because those parts are not located in the same pixel blocks in both frames. If the CODEC can track the motion of the arm, the delta frame description tells the decompressor to look for particular moving parts in other pixel blocks, essentially tracking the moving part as it moves from one pixel block to another.

Although dynamic carry forwards are helpful, they cannot always be implemented. In many cases, the capture board cannot scale resolution and frame rate, digitize, and hunt for dynamic carry forwards at the same time.

Dynamic carry forwards typically mark the dividing line between hardware and software CODECs. Hardware CODECs, as the name implies, are usually add-on boards that provide additional hardware compression and decompression operations. The benefit of hardware CODECs is that they do not place any additional burden on the host CPU in order to execute video compression and decompression.

Software CODECs rely on the host CPU and require no additional hardware. The benefit of software CODECs is that they are typically cheaper and easier to install. Because they rely on the host's CPU to perform compression and decompression, software CODECs are often limited in their capability to use techniques such as advanced tracking schemes.

Intraframe Compression

Intraframe compression is performed solely with reference to information within a particular frame. It is performed on pixels in delta frames that remain after interframe compression and on key frames. Although intraframe techniques are often given the most attention, overall CODEC performance relates more to interframe efficiency than intraframe efficiency. The following are the principal intraframe compression techniques:

- *Run Length Encoding (RLE)*—A simple lossless technique originally designed for data compression and later modified for facsimile. RLE compresses an image based on “runs” of pixels. Although it works well on black-and-white facsimiles, RLE is not very efficient for color video, which have few long runs of identically colored pixels.
- *JPEG*—A standard that has been adopted by two international standards organizations: the ITU (formerly CCITT) and the ISO. JPEG is most often used to compress still images using discrete cosine transform (DCT) analysis. First, DCT divides the image into 8×8 blocks and then converts the colors and pixels into frequency space by describing each block in terms of the number of color shifts (frequency) and the extent of the change (amplitude). Because most natural images are relatively smooth, the changes that occur most often have low amplitude values, so the change is minor. In other words, images have many subtle shifts among similar colors but few dramatic shifts between very different colors.

Next, quantization and amplitude values are categorized by frequency and averaged. This is the lossy stage because the original values are permanently discarded. However, because most of the picture is categorized in the high-frequency/low-amplitude range, most of the loss occurs among subtle shifts that are largely indistinguishable to the human eye.

After quantization, the values are further compressed through RLE using a special zigzag pattern designed to optimize compression of like regions within the image. At extremely high compression ratios, more high-frequency/low-amplitude changes are averaged, which can cause an entire pixel block to adopt the same color. This causes a blockiness artifact that is characteristic of JPEG-compressed images. JPEG is used as the intraframe technique for MPEG.

- *Vector quantization (VQ)*—A standard that is similar to JPEG in that it divides the image into 8×8 blocks. The difference between VQ and JPEG has to do with the quantization process. VQ is a recursive, or multistep algorithm with inherently self-correcting features. With VQ, similar blocks are categorized and a reference block is constructed for each category. The original blocks are then discarded. During decompression, the single reference block replaces all of the original blocks in the category.

After the first set of reference blocks is selected, the image is decompressed. Comparing the decompressed image to the original reveals many differences. To address the differences, an additional set of reference blocks is created that fills in the gaps created during the first estimation. This is the self-correcting part of the algorithm. The process is repeated to find a third

set of reference blocks to fill in the remaining gaps. These reference blocks are posted in a lookup table to be used during decompression. The final step is to use lossless techniques, such as RLE, to further compress the remaining information.

VQ compression is by its nature computationally intensive. However, decompression, which simply involves pulling values from the lookup table, is simple and fast. VQ is a public-domain algorithm used as the intraframe technique for both Cinepak and Indeo.

End-User Video Compression Algorithms

The following are the most popular end-user video compression algorithms. Note that some algorithms require dedicated hardware.

- *MPEG1*—A bit stream standard for compressed video and audio optimized to fit into a bandwidth of 1.5 Mbps. This rate is special because it is the data rate of uncompressed audio CDs and DATs. Typically, MPEG1 is compressed in non-real time and decompressed in real time. MPEG1 compression is typically performed in hardware; MPEG1 decompression can be performed in software or in hardware.
- *MPEG2*—A standard intended for higher quality video-on-demand applications for products such as the “set top box.” MPEG2 runs at data rates between 4 and 9 Mbps. MPEG2 and variants are being considered for use by regional Bell carriers and cable companies to deliver video-on-demand to the home as well as for delivering HDTV broadcasts. MPEG2 chip sets that perform real-time encoding are available. Real-time MPEG2 decompression boards are also available. A specification for MPEG2 adaptation over ATM AAL5 has been developed.
- *MPEG4*—A low-bit-rate compression algorithm intended for 64-Kbps connections. MPEG4 can be used for a wide range of applications including mobile audio, visual applications, and electronic newspaper sources.
- *M-JPEG (Motion-JPEG)*—The aggregation of a series of JPEG-compressed images. M-JPEG can be implemented in software or in hardware.
- *Cell B*—Part of a family of compression techniques developed by Sun Microsystems. Cell B is designed for real-time applications, such as videoconferencing, that require real-time video transmission. Cell A is a counterpart of Cell B that is intended for non-real time applications where encoding does not need to take place in real time. Both Cell A and Cell B use VQ and RLE techniques.
- *Indeo*—Developed by Intel. Indeo uses VQ as its intraframe engine. Intel has released three versions of Indeo:
 - *Indeo 2.1*—Focused on Intel’s popular capture board, the Smart Video Recorder, using intraframe compression.
 - *Indeo 3.1*—Introduced in late 1993 and incorporated interframe compression.
 - *Indeo 3.2*—Requires a hardware add-on for video compression but decompression can take place in software on a high-end 486 or Pentium processor.
- *Cinepak*—Developed by SuperMatch, a division of SuperMac Technologies. Cinepak was first introduced as a Macintosh CODEC and then migrated to the Windows platform in 1993. Like Indeo, Cinepak uses VQ as its intraframe engine. Of all the CODECs, Cinepak offers the widest cross-platform support, with versions for 3D0, Nintendo, and Atari platforms.
- *Apple Video*—A compression technique used by applications such as Apple Computer’s QuickTime Conferencing.

- *H.261*—The compression standard specified under the H.320 videoconferencing standard. H.261 describes the video coding and decoding methods for the moving picture component of audio-visual services at the rate of $p \times 64$ Kbps, where p is in the range 1 to 30. It describes the video source coder, the video multiplex coder, and the transmission coder. H.261 defines two picture formats:
 - *Common Intermediate Format (CIF)*—Specifies 288 lines of luminance information (with 360 pixels per line) and 144 lines of chrominance information (with 180 pixels per line).
 - *Quarter Common Intermediate Format (QCIF)*—Specifies 144 lines of luminance (with 180 pixels per line) and 72 lines of chrominance information (with 90 pixels per line). The choice between CIF and QCIF depends on available channel capacity—that is, QCIF is normally used when p is less than 3.

The actual encoding algorithm of H.261 is similar to (but incompatible with) MPEG. Also, H.261 needs substantially less CPU power for real-time encoding than MPEG. The H.261 algorithm includes a mechanism for optimizing bandwidth usage by trading picture quality against motion so that a quickly changing picture has a lower quality than a relatively static picture. When used in this way, H.261 is a constant-bit-rate encoding rather than a constant-quality, variable-bit-rate encoding.

Hardware Versus Software CODECs

In many cases, the network multimedia application dictates the video compression algorithm used. For example, Intel's ProShare videoconferencing application uses the Indeo standard or H.261, and Insoft Communique! uses Cell B compression. In some cases, such as Apple Computer's QuickTime Conferencing, the end user can specify the compression algorithm.

In general, the more CPU cycles given to video compression and decompression, the better the performance. This can be achieved either by running less expensive software CODECs on fast CPUs (Pentium, PowerPC, or RISC processors) or by investing more money in dedicated hardware add-ons such as an MPEG playback board. In some cases, the application dictates hardware or software compression and decompression. Insoft's INTV! video multicast package, for instance, uses a hardware-based compressor in the UNIX workstation, but uses a software-based decompressor for the PC workstations. The implication is that to use INTV, the PCs might need to be upgraded to deliver the requisite processing capabilities.

Compression Ratios

Any of the compression standards discussed in this chapter are helpful in reducing the amount of bandwidth needed to transmit digital video. In fact, digital video can be compressed up to 20:1 and still deliver a VHS-quality picture. Table 13-2 shows digital video compression ratios and the approximate quality that they yield in terms of video formats.

Table 13-2 Image Quality as a Function of Compression Ratio

Video Compression Ratio	Analog Picture Quality Equivalent
20:1	VHS
10:1	SVHS/Hi-8
04:1	Broadcast quality

As Table 13-2 indicates, fairly high video compression ratios can be used while still preserving high-quality video images. For example, a typical MPEG1 video stream (640×480, 30 frames per second) runs at 1.5 Mbps.

Digitizing Audio

Many of today’s multimedia applications include audio support. Some applications include hardware for digitizing audio, and other applications rely on third-party add-ons for audio support. Check with the application vendor to learn how audio is handled.

Like digital video, digital audio often begins from an analog source, so an analog-to-digital conversion must be made. Converting an analog signal to a digital signal involves taking a series of samples of the analog source. The aggregation of the samples yields the digital equivalent of the analog sound wave. A higher sampling rate delivers higher quality because it has more reference points to replicate the analog signal.

The sampling rate is one of three criteria that determine the quality of the digital version. The other two determining factors are the number of bits per sample and the number of channels.

Sampling rates are often quoted Hertz (Hz) or Kilohertz (KHz). Sampling rates are always measured per channel, so for stereo data recorded at 8,000 samples per second (8 KHz), there would actually be 16,000 samples per second (16 KHz). Table lists common sampling rates.

Table 13-3 Common Audio Sampling Rates

Samples per Second	Description
08,000	A telephony standard that works with μ -LAW encoding.
11 K	Either 11025 (a quarter of the CD sampling rate) or half the Macintosh sampling rate (perhaps the most popular rate on Macintosh computers).
16,000	Used by the G.722 compression standard
18.9 K	CD-ROM/XA standard
22 K	Either 22050 (half the CD sampling rate) or the Macintosh rate, which is precisely 22254.545454545454.
32,000	Used in digital radio; Nearly Instantaneous Compandable Audio Matrix (NICAM) (IBA/BREMA/BB), and other TV work in the U.K.; long play Digital Audio Tape (DAT); and Japanese HDTV.
37.8 K	CD-ROM/XA standard for higher quality.
44,056	Used by professional audio equipment to fit an integral number of samples in a video frame.
44,100	CD sampling rate. DAT players recording digitally from CD also use this rate.
48,000	DAT sampling rate for domestic rate.

An emerging tendency is to standardize on only a few sampling rates and encoding styles, even if the file formats differ. The emerging rates and styles are listed in Table 13-4.

Table 13-4 Sample Rates and Encoding Styles

Samples Per Second	Encoding Style
08,000	8-bit μ -LAW mono
22,050	8-bit linear unsigned mono and stereo

Samples Per Second	Encoding Style
44,100	16-bit linear unsigned mono and stereo

Audio Compression

Audio data is difficult to compress effectively. For 8-bit data, a Huffman encoding of the deltas between successive samples is relatively successful. Companies such as Sony and Philips have developed proprietary schemes for 16-bit data. Apple Computer has an audio compression/expansion scheme called ACE on the Apple IIGS and called MACE on the Macintosh. ACE/MACE is a lossy scheme that attempts to predict where the wave will go on the next sample. There is very little quality change on 8:4 compression, with somewhat more quality degradation at 8:3 compression. ACE/MACE guarantees exactly 50 percent or 62.5 percent compression.

Public standards for voice compression using Adaptive Delta Pulse Code Modulation (ADPCM) are as follows:

- CCIT G.721 sampling at 32 Kbps
- CCIT G.723 sampling at 24 Kbps and 40 Kbps
- GSM 06.10 is a European speech encoding standard that compresses 160 13-bit samples into 260 bits (33 bytes), or 1,650 bytes per second (at 8,000 samples per second).

There are also two U.S. federal standards:

- 1016 using code excited linear prediction (CELP) at 4,800 bits per second)
- 1015 (LPC-10E) at 2,400 bits per second)

Using Networked Multimedia Applications

There is a wide range of network multimedia applications to choose from, so it is important to understand why a particular application is being deployed. Additionally, it is important to understand the bandwidth implications of the chosen application. Table 13-5 lists some of the popular network multimedia applications.

Table 13-5 Popular Network Multimedia Applications

Application	Type	Platform
Apple QuickTime Conferencing	Videoconferencing	Macintosh
AT&T Vistium	Videoconferencing	PC
CU-seeMe	Videoconferencing	Macintosh/PC/UNIX
InPerson	Videoconferencing	UNIX
Insoft Communique!	Videoconferencing	PC/UNIX
Intel CNN at Work	LAN broadcast	PC
Intel ProShare	Videoconferencing	PC
InVision	Videoconferencing	PC
Novell Video for NetWare	Video server	NetWare
PictureTel	Videoconferencing	PC
Starlight Starworks	Video server	UNIX/NetWare

Types of Applications

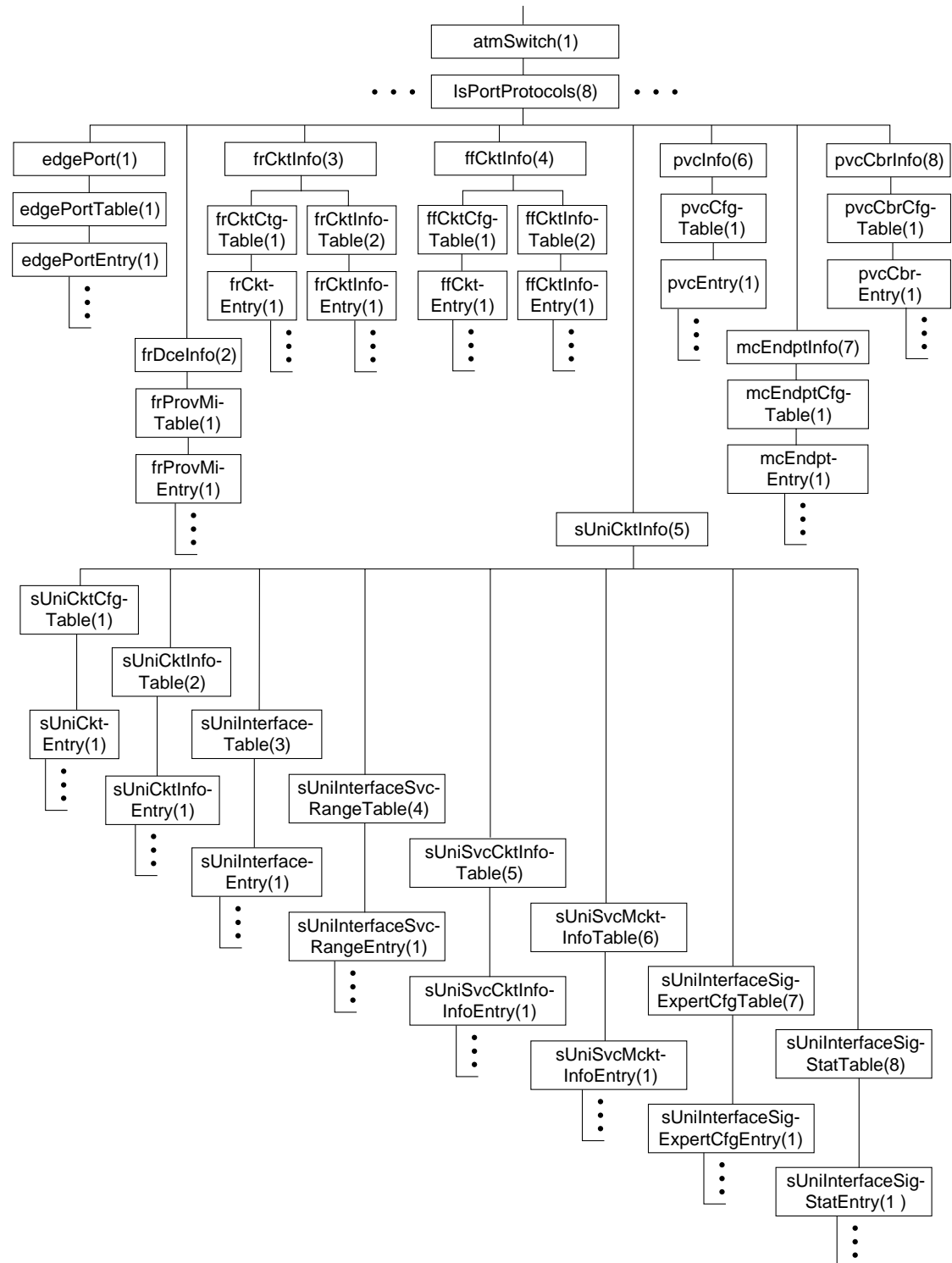
Network multimedia applications fall into the following categories:

- Point-to-Point Bidirectional Applications
- Point-to-Multipoint Bidirectional Applications
- Point-to-Point Unidirectional Applications
- Point-to-Multipoint Unidirectional Applications

Point-to-Point Bidirectional Applications

Point-to-point bidirectional applications, as shown in Figure 13-6, deliver real-time, point-to-point communication. The process is bidirectional, meaning that video can be transmitted in both directions in real time.

Figure 13-6 Point-to-point bidirectional applications.



Examples of point-to-point bidirectional applications include the following:

- Audio and videoconferencing
- Shared whiteboard

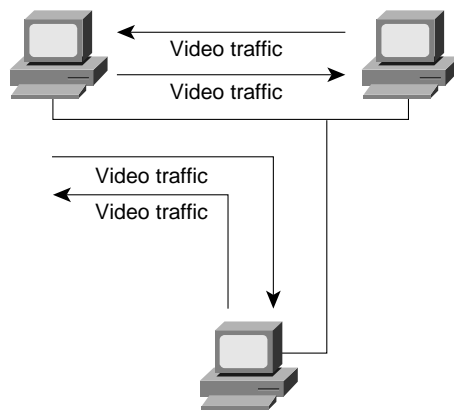
- Shared application

Audio and videoconferencing applications provide a real-time interactive environment for two users. Often, these applications also include a shared whiteboard application or an application-sharing functionality. Shared whiteboard applications provide a common area that both users can see and draw on. Shared whiteboards (also known as *collaborative workspaces*) are particularly useful in conversations where “a picture is worth a thousand words.” Application sharing is also a useful and productive tool. With application sharing, one user can launch an application, such as Microsoft Access, and the user at the other end can view and work with it as though the application were installed on that user’s computer. Coworkers at opposite ends of a network can collaborate in an application regardless of where the application resides.

Point-to-Multipoint Bidirectional Applications

Point-to-multipoint bidirectional applications as shown in Figure 13-7, use multiple video senders and receivers. In this model, multiple clients can send and receive a video stream in real time.

Figure 13-7 Point-to-multipoint bidirectional applications.



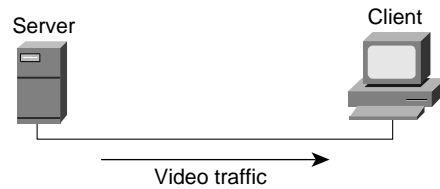
Interactive video, such as video kiosks, deliver video to multiple recipients. The recipients, however, can interact with the video session by controlling start and stop functions. The video content can also be manipulated by end-user interaction. Some kiosks, for example, have a touch pad that delivers different videos based on the user’s selection. Examples of point-to-multipoint bidirectional applications include the following:

- Interactive video
- Videoconferencing

Like a telephone call in which multiple listeners participate, the same can be done with certain videoconferencing applications. For example, a three-way video conference call can occur in which each person can receive video and audio from the other two participants.

Point-to-Point Unidirectional Applications

Point-to-point unidirectional applications, as shown in Figure 13-8, use point-to-point communications in which video is transmitted in only one direction. The video itself can be a stored video stream or a real-time stream from a video recording source.

Figure 13-8 Point-to-point unidirectional applications.

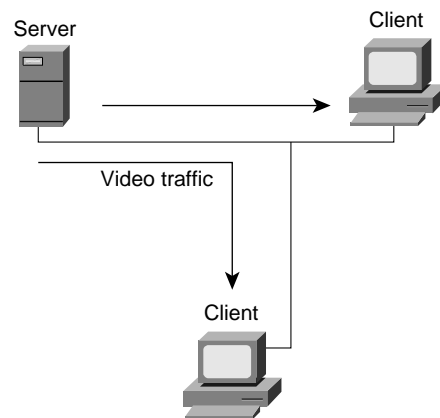
Examples of point-to-point unidirectional applications include the following:

- Video server applications
- Multimedia-enabled email applications

In point-to-point unidirectional applications, compressed video clips are stored centrally. The end user initiates the viewing process by downloading the stream across the network to the video decompressor, which decompresses the video clip for viewing.

Point-to-Multipoint Unidirectional Applications

Point-to-multipoint unidirectional applications, as shown in Figure 13-9, are similar to point-to-point unidirectional applications except that the video is transmitted to a group of clients. The video is still unidirectional. The video can come from a storage device or a recording source.

Figure 13-9 Point-to-multipoint unidirectional applications.

Examples of point-to-multipoint unidirectional applications include the following:

- Video server applications
- LAN TV

Both of these applications provide unidirectional video services. Video server applications deliver to multiple clients video streams that have already been compressed. LAN TV applications deliver stored video streams or real-time video from a camera source. Distance learning, in which classes are videotaped and then broadcast over the LAN and WAN to remote employees, is a popular example of a point-to-multipoint unidirectional video application.

Quality of Service Requirements

Data and multimedia applications have different quality of service requirements. Unlike traditional “best effort” data services, such as File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and X Window, in which variations in latency often go unnoticed, audio and video data are useful only if they are delivered within a specified time period. Delayed delivery only impedes the usefulness of other information in the stream. In general, *latency* and *jitter* are the two primary forces working against the timely delivery of audio and video data.

Latency

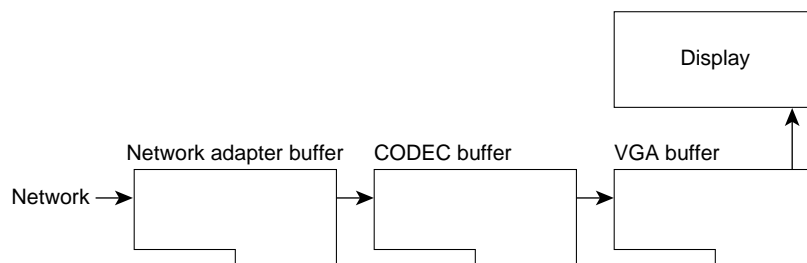
Real-time, interactive applications, such as desktop conferencing, are sensitive to accumulated delay, which is known as *latency*. Telephone networks are engineered to provide less than 400 milliseconds (ms) round-trip latency. Multimedia networks that support desktop audio and videoconferencing also must be engineered with a latency budget of less than 400 ms per round-trip. The network contributes to latency in several ways:

- *Propagation delay*—The length of time that information takes to travel the distance of the line. Propagation delay is mostly determined by the speed of light; therefore, the propagation delay factor is not affected by the networking technology in use.
- *Transmission delay*—The length of time a packet takes to cross the given media. Transmission delay is determined by the speed of the media and the size of the packet.
- *Store-and-forward delay*—The length of time an internetworking device (such as a switch, bridge, or router) takes to send a packet that it has received.
- *Processing delay*—The time required by a networking device for looking up the route, changing the header, and other switching tasks. In some cases, the packet also must be manipulated. For example, the encapsulation type or the hop count must be changed. Each of these steps can contribute to the processing delay.

Jitter

If a network delivers data with variable latency, it introduces jitter. Jitter is particularly disruptive to audio communications because it can cause pops and clicks that are noticeable to the user. Many multimedia applications are designed to minimize jitter. The most common technique is to store incoming data in an insulating buffer from which the display software or hardware pulls data. The buffer reduces the effect of jitter in much the same way that a shock absorber reduces the effect of road irregularities on a car: Variations on the input side are smaller than the total buffer size and therefore are not normally perceivable on the output side. Figure 13-10 shows a typical buffering strategy that helps to minimize latency and jitter inherent in a given network.

Figure 13-10 Hardware buffering minimizes latency and jitter.



Buffering can also be performed within the network itself. Consider a client that connects to a video server. During the video playback session, data moving from the video server to the client can be buffered by the network interface cards and the video decompressor. In this case, buffering acts as a regulator to offset inherent irregularities (latency/jitter) that occur during transmission. The overall effect is that even though the traffic may be bursty coming over the network, the video image is not impaired because the buffers store incoming data and then regulate the flow to the display card.

Buffers can play a large role in displaying video, especially over existing networks, but because they are not large enough to accommodate the entire audio or video file, the use of buffers cannot guarantee jitter-free delivery. For that reason, multimedia networks should also make use of techniques that minimize jitter.

One way of providing predictable performance is to increase line speeds to assure that adequate bandwidth is available during peak traffic conditions. This approach may be reasonable for backbone links, but it may not be cost effective for other links. A more cost-effective approach may be to use lower-speed lines and give mission-critical data priority over less critical transmissions during peak traffic conditions through the use of queuing techniques. The Cisco IOS software offers the following queuing strategies:

- Priority Queuing
- Custom Queuing
- Weighted Fair Queuing

Priority Queuing

Priority queuing allows the network administrator to define four priorities of traffic—high, normal, medium, and low—on a given interface. As traffic comes into the router, it is assigned to one of the four output queues. Packets on the highest priority queue are transmitted first. When that queue empties, packets on the next highest priority queue are transmitted, and so on.

Priority queuing ensures that during congestion, the highest-priority data is not delayed by lower-priority traffic. Note that, if the traffic sent to a given interface exceeds the bandwidth of that interface, lower-priority traffic can experience significant delays.

Custom Queuing

Custom queuing allows the network administrator to reserve a percentage of bandwidth for specified protocols. Cisco IOS Software Release 11.0 allows the definition of up to 16 output queues for normal data (including routing packets) with a separate queue for system messages, such as LAN keepalive messages. The router services each queue sequentially, transmitting a configurable percentage of traffic on each queue before moving on to the next queue. Custom queuing guarantees that mission-critical data is always assigned a certain percentage of the bandwidth but also assures predictable throughput for other traffic. For that reason, custom queuing is recommended for networks that need to provide a guaranteed level of service for all traffic.

Custom queuing works by determining the number of bytes that should be transmitted from each queue, based on the interface speed and the configured percentage. When the calculated byte count from a given queue has been transmitted, the router completes transmission of the current packet and moves on to the next queue, servicing each queue in a round-robin fashion.

With custom queuing, unused bandwidth is dynamically allocated to any protocol that requires it. For example, if SNA is allocated 50 percent of the bandwidth but uses only 30 percent, the next protocol in the queue can take up the extra 20 percent until SNA requires it. Additionally, custom queuing maintains the predictable throughput of dedicated lines by efficiently using packet-switching technologies such as Frame Relay.

Weighted Fair Queuing

Weighted fair queuing was introduced with Cisco IOS Software Release 11.0. Weighted fair queuing is a traffic priority management algorithm that identifies conversations (traffic streams) and then breaks up the streams of packets that belong to each conversation to ensure that capacity is shared fairly between individual conversations. By examining fields in the packet header, the algorithm automatically separates conversations.

Conversations are sorted into two categories—those that are attempting to use a lot of bandwidth with respect to the interface capacity (for example, FTP) and those that need less (for example, interactive traffic). For streams that use less bandwidth, the queuing algorithm always attempts to provide access with little or no queuing and shares the remaining bandwidth between the other conversations. In other words, low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally.

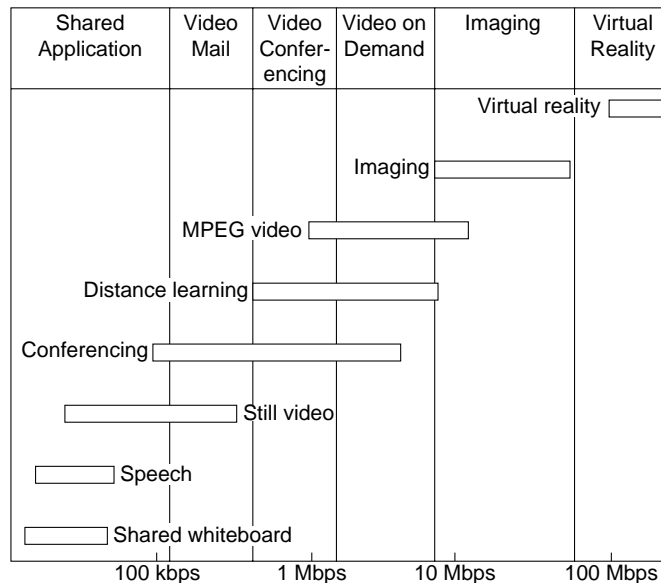
Weighted fair queuing provides an automatic way of stabilizing network behavior during congestion and results in increased performance and reduced retransmission. In most cases, weighted fair queuing provides smooth end-to-end performance over a given link and, in some cases, may resolve link congestion without an expensive increase in bandwidth.

Note Weighted fair queuing is enabled by default on most serial interfaces; priority queuing or custom queuing can be configured instead. By default, weighted fair queuing is disabled on serial interfaces that are configured for X.25, LAPB, and SDLC, and on all LAN interfaces.

Bandwidth Requirements

Bandwidth requirements for network multimedia applications can range anywhere from 100 Kbps to 70 or 100 Mbps. Figure 13-11 shows the amount of bandwidth that the various types of network multimedia applications require.

Figure 13-11 Network bandwidth usage.



As Figure 13-11 indicates, the type of application has a direct impact on the amount of LAN or WAN bandwidth needed. Assuming that bandwidth is limited, the choice is either to select a lower quality video application that works within the available bandwidth, or consider modifying the network infrastructure to deliver more overall bandwidth.

Understanding Multicasting

Traditional network applications, including most of today's network multimedia applications, involve communication only between two computers. A two-user videoconferencing session using Intel ProShare, for example, is a strictly unicast transaction. However, a new breed of network multimedia applications, such as LAN TV, desktop conferencing, corporate broadcasts, and collaborative computing, requires simultaneous communication between groups of computers. This process is known generically as *multipoint communications*.

When implementing multipoint network multimedia applications, it is important to understand the traffic characteristics of the application in use. In particular, the network designer needs to know whether an application uses unicast, broadcast, or multicast transmission facilities, defined as follows:

- *Unicast*—In a unicast design, applications can send one copy of each packet to each member of the multipoint group. This technique is simple to implement, but it has significant scaling restrictions if the group is large. In addition, unicast applications require extra bandwidth, because the same information has to be carried multiple times—even on shared links.
- *Broadcast*—In a broadcast design, applications can send one copy of each packet and address it to a broadcast address. This technique is even simpler than unicast for the application to implement. However, if this technique is used, the network must either stop broadcasts at the LAN boundary (a technique that is frequently used to prevent broadcast storms) or send the broadcast everywhere. Sending the broadcast everywhere is a significant burden on network resources if only a small number of users actually want to receive the packets.
- *Multicast*—In a multicast design, applications can send one copy of each packet and address it to a group of computers that want to receive it. This technique addresses packets to a group of receivers (at the multicast address) rather than to a single receiver (at a unicast address), and it depends on the network to forward the packets to only those networks that need to receive them. Multicasting helps control network traffic and reduces the amount of processing that hosts have to do.

Many network multimedia applications, such as Insoft INTV! 3.0 and Apple QuickTime Conferencing 1.0, implement multicast transmission facilities because of the added efficiency that multicasting offers to the network and to the client. From the network perspective, multicast dramatically reduces overall bandwidth consumption and allows for more scalable network multimedia applications.

Consider an MPEG-based video server. Playback of an MPEG stream requires approximately 1.5 Mbps per client viewer. In a unicast environment, the video server send $1.5 \times n$ (where n =number of client viewers) Mbps of traffic to the network. With a 10-Mbps connection to the server, roughly six to seven streams could be supported before the network runs out of bandwidth. In a multicast environment, the video server need send only one video stream to a multicast address. Any number of clients can listen to the multicast address and receive the video stream. In this scenario, the server requires only 1.5 Mbps and leaves the rest of the bandwidth free for other uses.

Multicast can be implemented at both OSI Layer 2 and OSI Layer 3. Ethernet and Fiber Distributed Data Interface (FDDI), for example, support unicast, multicast, and broadcast addresses. A host can respond to a unicast address, several multicast addresses, and the broadcast address. Token Ring also supports the concept of multicast addressing but uses a different technique. Token Rings have functional addresses that can be used to address groups of receivers.

If the scope of an application is limited to a single LAN, using an OSI Layer 2 multicast technique is sufficient. However, many multipoint applications are valuable precisely because they are not limited to a single LAN.

When a multipoint application is extended to an Internet consisting of different media types, such as Ethernet, Token Ring, FDDI, Asynchronous Transfer Mode (ATM), Frame Relay, SMDS, and other networking technologies, multicast is best implemented at OSI Layer 3. OSI Layer 3 must define several parameters in order to support multicast communications:

- *Addressing*—There must be an OSI Layer 3 address that is used to communicate with a group of receivers rather than a single receiver. In addition, there must be a mechanism for mapping this address onto OSI Layer 2 multicast addresses where they exist.
- *Dynamic registration*—There must be a mechanism for the computer to communicate to the network that it is a member of a particular group. Without this capability, the network cannot know which networks need to receive traffic for each group.
- *Multicast routing*—The network must be able to build packet distribution trees that allow sources to send packets to all receivers. A primary goal of packet distribution trees is to ensure that only one copy of a packet exists on any given network—that is, if there are multiple receivers on a given branch, there should be only one copy of each packet on that branch.

IP Multicast

The Internet Engineering Task Force (IETF) has developed standards that address the parameters that are required to support multicast communications:

- *Addressing*—The IP address space is divided into four sections: Class A, Class B, Class C, and Class D. Class A, B, and C addresses are used for unicast traffic. Class D addresses are reserved for multicast traffic and are allocated dynamically.
- *Dynamic registration*—RFC 1112 defines the Internet Group Management Protocol (IGMP). IGMP specifies how the host should inform the network that it is a member of a particular multicast group.
- *Multicast routing*—There are several standards for routing IP multicast traffic:
 - Distance Vector Multicast Routing Protocol (DVMRP) as described in RFC 1075.
 - Multicast Open Shortest Path First (MOSPF), which is an extension to Open Shortest Path First (OSPF) that allows it to support IP multicast, as defined in RFC 1584.
 - Protocol Independent Multicast (PIM), which is a multicast protocol that can be used with all unicast IP routing protocols, as defined in the two Internet standards-track drafts entitled *Protocol Independent Multicast (PIM): Motivation and Architecture* and *Protocol Independent Multicast (PIM): Protocol Specification*.

IP Multicast Group Addressing

Figure 13-12 shows the format of a Class D IP multicast address.

Figure 13-12 Class D address format.



Unlike Class A, B, and C IP addresses, the last 28 bits of a Class D address have no structure. The multicast group address is the combination of the high-order 4 bits of 1110 and the multicast group ID. These are typically written as dotted-decimal numbers and are in the range 224.0.0.0 through 239.255.255.255. Note that the high-order bits are 1110. If the bits in the first octet are 0, this yields the 224 portion of the address.

The set of hosts that responds to a particular IP multicast address is called a *host group*. A host group can span multiple networks. Membership in a host group is dynamic—hosts can join and leave host groups. For a discussion of IP multicast registration, see the section called “Internet Group Management Protocol” later in this chapter.

Some multicast group addresses are assigned as well-known addresses by the Internet Assigned Numbers Authority (IANA). These multicast group addresses are called *permanent host groups* and are similar in concept to the well-known TCP and UDP port numbers. Address 224.0.0.1 means “all systems on this subnet,” and 224.0.0.2 means “all routers on this subnet.”

Table 13-6 lists the multicast address of some permanent host groups.

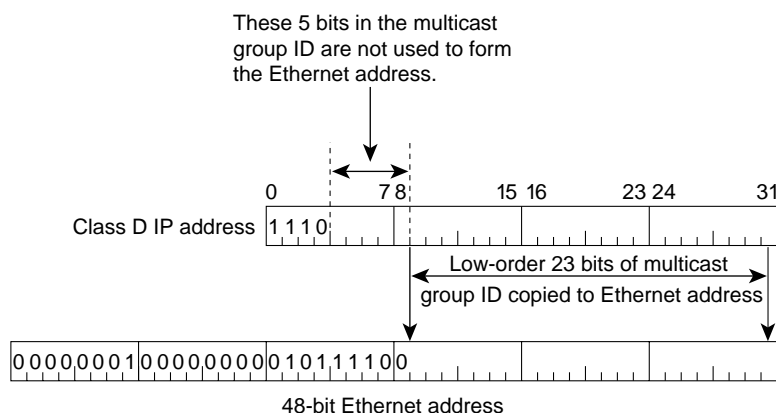
Table 13-6 Example of Multicast Addresses for Permanent Host Groups

Permanent Host Group	Multicast Address
Network Time Protocol	224.0.1.1
RIP-2	224.0.0.9
Silicon Graphics Dogfight application	224.0.1.2

The IANA owns a block of Ethernet addresses that in hexadecimal is 00:00:5e. This is the high-order 24 bits of the Ethernet address, meaning that this block includes addresses in the range 00:00:5e:00:00:00 to 00:00:5e:ff:ff:ff. The IANA allocates half of this block for multicast addresses. Given that the first byte of any Ethernet address must be 01 to specify a multicast address, the Ethernet addresses corresponding to IP multicasting are in the range 01:00:5e:00:00:00 through 01:00:5e:7f:ff:ff.

This allocation allows for 23 bits in the Ethernet address to correspond to the IP multicast group ID. The mapping places the low-order 23 bits of the multicast group ID into these 23 bits of the Ethernet address, as shown in Figure 13-13. Because the upper five bits of the multicast address are ignored in this mapping, the resulting address is not unique. Thirty-two different multicast group IDs map to each Ethernet address.

Figure 13-13 Multicast address mapping.



Because the mapping is not unique and because the interface card might receive multicast frames in which the host is really not interested, the device driver or IP modules must perform filtering.

Multicasting on a single physical network is simple. The sending process specifies a destination IP address that is a multicast address, and the device driver converts this to the corresponding Ethernet address and sends it. The receiving processes must notify their IP layers that they want to receive datagrams destined for a given multicast address and the device driver must somehow enable reception of these multicast frames. This process is handled by joining a multicast group.

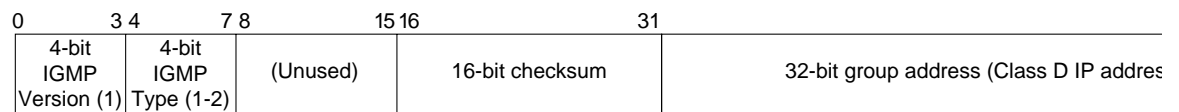
When a multicast datagram is received by a host, it must deliver a copy to all the processes that belong to that group. This is different from UDP where a single process receives an incoming unicast UDP datagram. With multicast, multiple processes on a given host can belong to the same multicast group.

Complications arise when multicasting is extended beyond a single physical network and multicast packets pass through routers. A protocol is needed for routers to know if any hosts on a given physical network belong to a given multicast group. This function is handled by the Internet Group Management Protocol.

Internet Group Management Protocol

The Internet Group Management Protocol (IGMP) is part of the IP layer and uses IP datagrams (consisting of a 20-byte IP header and an 8-byte IGRP message) to transmit information about multicast groups. IGMP messages are specified in the IP datagram with a protocol value of 2. Figure 13-14 shows the format of the 8-byte IGMP message.

Figure 13-14 IGMP message format.



The value of the *version* field is 1. The value of the *type* field is 1 for a query sent by a multicast router and 2 for a report sent by a host. The value of the *checksum* field is calculated in the same way as the ICMP checksum. The group address is a class D IP address. In a query, the group address is set to 0, and in a report, it contains the group address being reported.

The concept of a process joining a multicast group on a given host interface is fundamental to multicasting. Membership in a multicast group on a given interface is dynamic (that is, it changes over time as processes join and leave the group). This means that end users can dynamically join multicast groups based on the applications that they execute.

Multicast routers use IGMP messages to keep track of group membership on each of the networks that are physically attached to the router. The following rules apply:

- A host sends an IGMP report when the first process joins a group. The report is sent out the same interface on which the process joined the group. Note that if other processes on the same host join the same group, the host does *not* send another report.
- A host does not send a report when processes leave a group, even when the last process leaves a group. The host knows that there are no members in a given group, so when it receives the next query, it doesn't report the group.
- A multicast router sends an IGMP query at regular intervals to see whether any hosts still have processes belonging to any groups. The router sends a query out each interface. The group address in the query is 0 because the router expects one response from a host for every group that contains one or more members on a host.

- A host responds to an IGMP query by sending one IGMP report for each group that still contains at least one process.

Using queries and reports, a multicast router keeps a table of its interfaces that have one or more hosts in a multicast group. When the router receives a multicast datagram to forward, it forwards the datagram (using the corresponding multicast OSI Layer 2 address) on only those interfaces that still have hosts with processes belonging to that group.

The Time to Live (TTL) field in the IP header of reports and queries is set to 1. A multicast datagram with a TTL of 0 is restricted to the same host. By default, a multicast datagram with a TTL of 1 is restricted to the same subnet. Higher TTL field values can be forwarded by the router. By increasing the TTL, an application can perform an expanding ring search for a particular server. The first multicast datagram is sent with a TTL of 1. If no response is received, a TTL of 2 is tried, and then 3, and so on. In this way, the application locates the server that is closest in terms of hops.

The special range of addresses 224.0.0.0 through 224.0.0.255 is intended for applications that never need to multicast further than one hop. A multicast router should never forward a datagram with one of these addresses as the destination, regardless of the TTL.

Multicast Routing Protocols

A critical issue for delivering multicast traffic in a routed network is the choice of multicast routing protocol. Three multicast routing protocols have been defined for this purpose:

- Distance Vector Multicast Routing Protocol
- Multicast OSPF
- Protocol Independent Multicast

The goal in each protocol is to establish paths in the network so that multicast traffic can effectively reach all group members.

Distance Vector Multicast Routing Protocol

Distance Vector Multicast Routing Protocol (DVMRP) uses a technique known as *reverse path forwarding*. When a router receives a packet, it floods the packet out all paths except the path that leads back to the packet's source. Reverse path forwarding allows a data stream to reach all LANs (possibly multiple times). If a router is attached to a set of LANs that does not want to receive a particular multicast group, the router sends a "prune" message up the distribution tree to prevent subsequent packets from traveling where there are no members.

New receivers are handled by using grafts. Consequently, only one round-trip time (RTT) from the new receiver to the nearest active branch of the tree is required for the new receiver to start getting traffic.

To determine which interface leads back to the source of the data stream, DVMRP implements its own unicast routing protocol. This unicast routing protocol is similar to RIP and is based on hop counts. As a result, the path that the multicast traffic follows might not be the same as the path that the unicast traffic follows. The need to flood frequently means that DVMRP has trouble scaling. This limitation is exacerbated by the fact that early implementations of DVMRP did not implement pruning.

DVMRP has been used to build the MBONE—a multicast backbone across the public Internet—by building tunnels between DVMRP-capable machines. The MBONE is used widely in the research community to transmit the proceedings of various conferences and to permit desktop conferencing.

Multicast OSPF

Multicast OSPF (MOSPF) is an extension of the OSPF unicast routing protocol and works only in internetworks that use OSPF. OSPF works by having each router in a network understand all of the available links in the network. Each OSPF router calculates routes from itself to all possible destinations. MOSPF works by including multicast information in OSPF link-state advertisements so that an MOSPF router learns which multicast groups are active on which LANs.

MOSPF builds a distribution tree for each source-group pair and computes a tree for active sources sending to the group. The tree state is cached and must be recomputed when a link state change occurs or when the cache times out.

MOSPF works well in environments that have relatively few source-group pairs active at any given time. It works less well in environments that have many active sources or in environments that have unstable links.

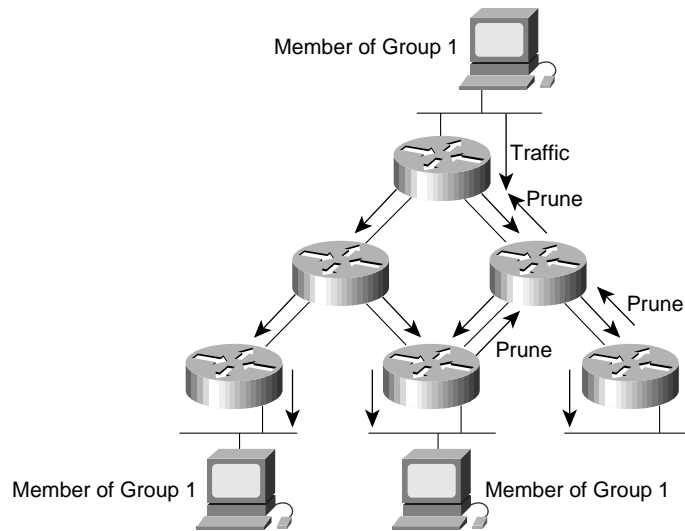
Protocol Independent Multicast

Unlike MOSPF, which is OSPF dependent, Protocol Independent Multicast (PIM) works with all existing unicast routing protocols. Unlike DVMRP, which has inherent scaling problems, PIM solves potential scalability problems by supporting two different types of multipoint traffic distribution patterns: dense mode and sparse mode. Dense mode is most useful when the following conditions occur:

- Senders and receivers are in close proximity to one another.
- There are few senders and many receivers.
- The volume of multicast traffic is high.
- The stream of multicast traffic is constant.

Dense-mode PIM uses reverse path forwarding and is similar to DVMRP. The most significant difference between DVMRP and dense-mode PIM is that PIM works with whatever unicast protocol is being used—it does not require any particular unicast protocol.

In dense mode, PIM floods the network and prunes back based on multicast group member information. Dense mode is effective, for example, in a LAN TV multicast environment because it is likely that there will be a group member on each subnet. Flooding the network is effective because little pruning is necessary. An example of PIM dense-mode operation is shown in Figure 13-15.

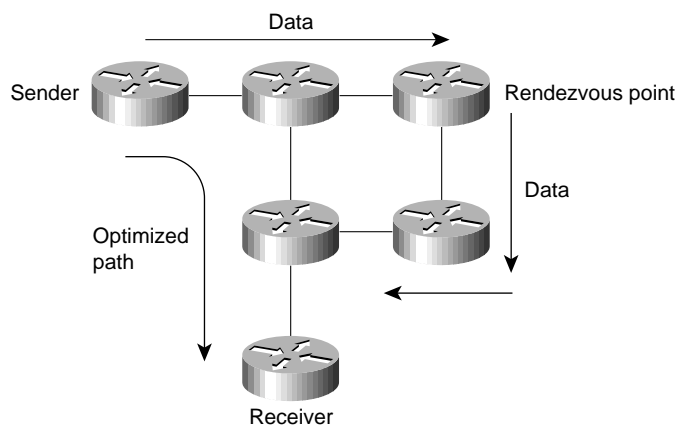
Figure 13-15 PIM dense-mode operation.

Sparse-mode PIM is most useful when the following conditions occur:

- There are few receivers in a group.
- Senders and receivers are separated by WAN links.
- The stream of multicast traffic is intermittent.

Sparse-mode PIM is optimized for environments where there are many multipoint data streams. Each data stream goes to a relatively small number of the LANs in the internetwork. For these types of groups, reverse path forwarding would make inefficient use of the network bandwidth.

In sparse-mode, PIM assumes that no hosts want the multicast traffic unless they specifically ask for it. It works by defining a rendezvous point (RP). The RP is used by senders to a multicast group to announce their existence and by receivers of multicast packets to learn about new senders. When a sender wants to send data, it first sends the data to the RP. When a receiver wants to receive data, it registers with the RP. Once the data stream begins to flow from sender to RP to receiver, the routers in the path automatically optimize the path to remove any unnecessary hops. An example of PIM sparse-mode operation is shown in Figure 13-16.

Figure 13-16 PIM sparse-mode operation.

Note The administrators of the MBONE plan to adopt PIM because it is more efficient than DVMRP.

Simple Multicast Routing Protocol

Simple Multicast Routing Protocol (SMRP) is a transport layer multicast protocol standard for multicast AppleTalk and IPX traffic.

Note Initial support for SMRP is provided by Cisco IOS Software Release 11.0 or later for AppleTalk only.

With SMRP, a router on each local network segment is elected as the primary node. The primary node handles requests from local devices to create multicast groups on that segment. When it wants to send multicast data, a device sends a Create Group Request packet to ask the primary node to assign a group address. The primary node responds by sending to the requesting device a Create Group Response packet that contains the assigned group address.

Devices that want to receive multicast data from this group send a Join Request packet to ask their local router to join the group. The local router forwards the Join Request to the primary node that created the group. The primary node responds by sending a Join Response.

Multicast data sent by the source is forwarded by router downstream interfaces toward receivers. Receivers can join and leave a group at any time, and a sender can delete the group at any time. The routers ensure that multicast data is transmitted as efficiently as possible, without duplication, from senders to receivers.

Routers maintain and update SMRP multicast groups by periodically sending Creator Query and Member Query packets to poll the network for the presence of senders and receivers. A router that detects the disappearance of a sender deletes the group. A router that senses the disappearance of a receiver informs its upstream neighbor to stop forwarding multicast data if no other receivers exist on the segment. Each router periodically informs its neighbors of its presence by sending Hello packets.

Network Designs for Multimedia Applications

This section examines network designs that work well with network multimedia applications. The following topics are covered:

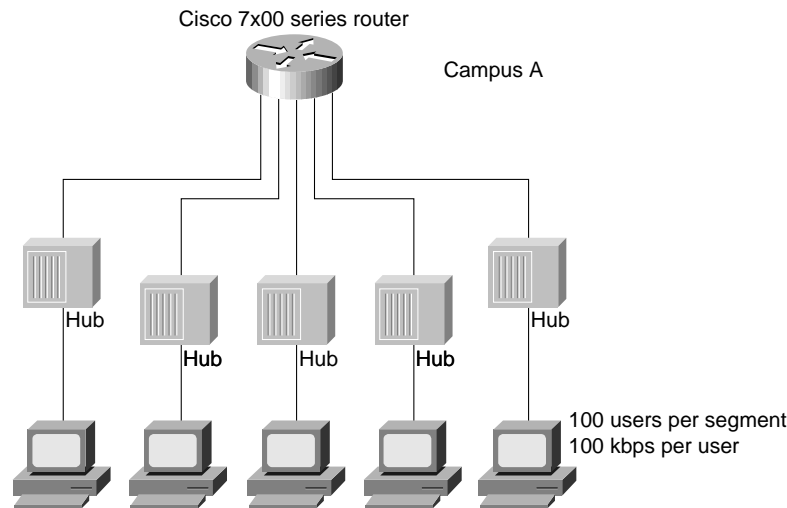
- Traditional LAN Designs
- WAN Designs
- High-Speed LAN Designs

Traditional LAN Designs

Some campus LAN environments already have adequate bandwidth for running certain network multimedia applications, but most do not. In many cases, lack of bandwidth is *not* caused by a slow LAN medium—instead, lack of bandwidth is caused by inefficient LAN design and segmentation. A considerable amount of bandwidth can be gained by using switches to resegment the campus LAN environment.

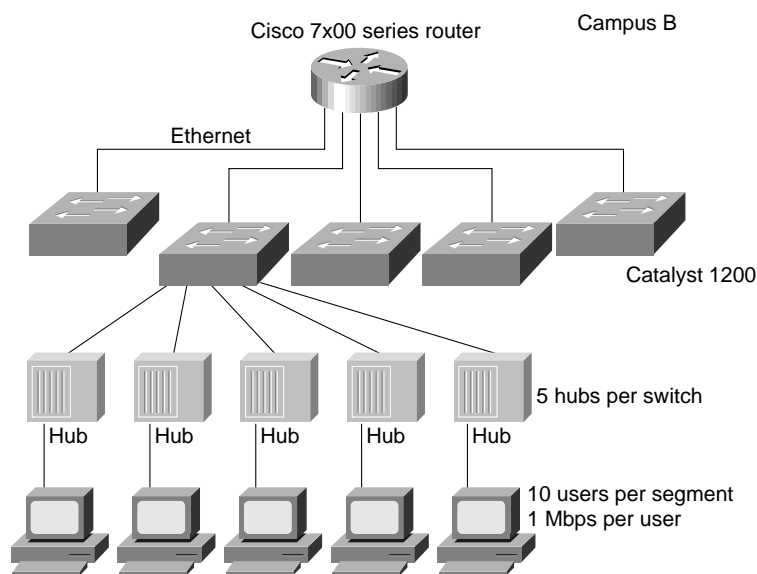
Consider three different campus designs. In Figure 13-17, Campus A has 500 users on five separate 100-node shared Ethernet segments. Each of the five segments are connected via a Cisco 7x00 series router.

Figure 13-17 Shared Ethernet campus LAN design.



With 100 users per segment, the net bandwidth per user is 100 Kbps. Using the graph shown in Figure 13-17, an audio conferencing package is the most that Campus A can handle. In Figure 13-18, Campus B uses a combination of shared Ethernet hubs (repeaters) and Ethernet switches to deliver substantially more bandwidth per user.

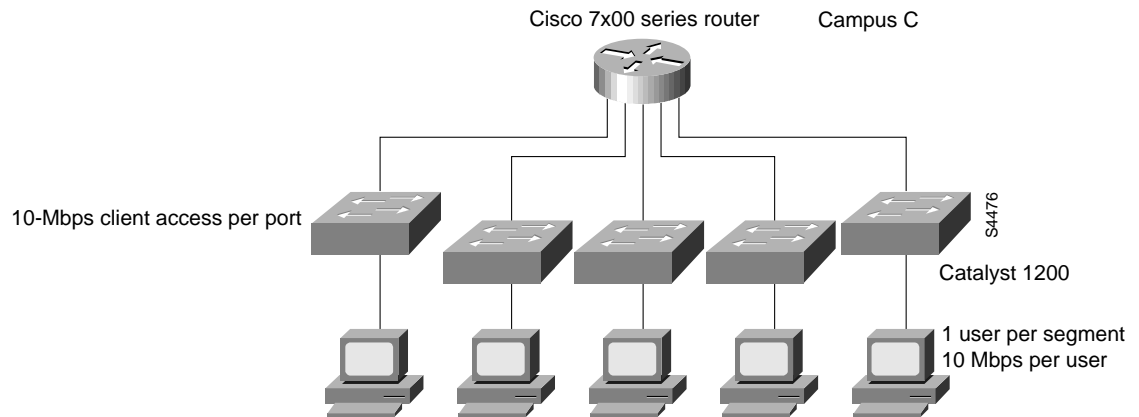
Figure 13-18 Shared Ethernet and switched Ethernet campus LAN design.



In Figure 13-18, ten users are connected to a shared Ethernet hub. The hub is then connected to dedicated 10-Mbps Ethernet switch ports. Each of the Ethernet switches is connected together over a routed Ethernet backbone. In this scenario, each hub gets 10 Mbps, which yields roughly 1 Mbps for each of the ten users on the hub. Based on this network design, Campus B can run medium-quality video applications.

Campus C, shown in Figure 13-19, eliminates the shared Ethernet hubs. Each user has a dedicated 10-Mbps connection to the LAN via a direct connection to an Ethernet switch port. Like Campus B, the switches are interconnected over a routed Ethernet backbone. With 10 Mbps of bandwidth per user, Campus C can easily support high-quality network multimedia applications.

Figure 13-19 Switched Ethernet campus LAN design.



The comparison of Campus A, Campus B, and Campus C illustrates that the first step in delivering more bandwidth is *not* ripping out the existing Ethernet or Token Ring infrastructure and moving to a 100-Mbps technology. Instead, the proper first step is to deploy switches thereby improving bandwidth per user by assigning a small number of users to each switch port or by assigning one user to each switch port, thereby providing dedicated 10-Mbps bandwidth to that user. This technique is known as *microsegmenting*.

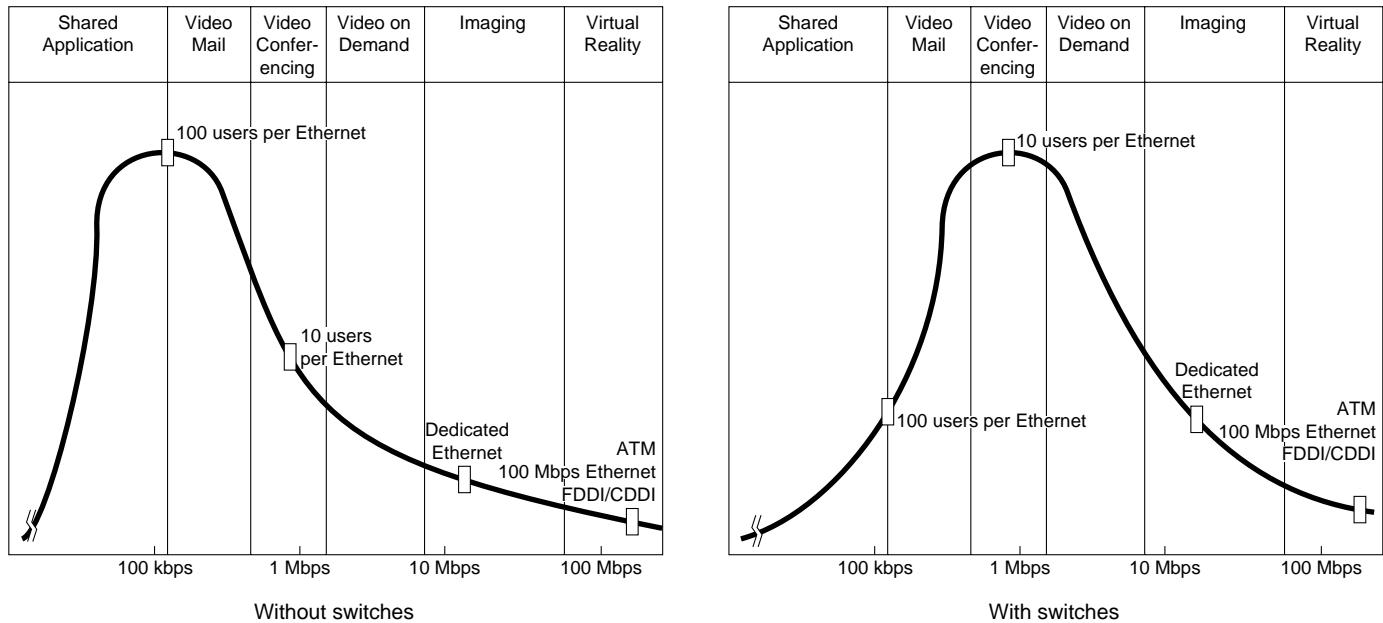
The majority of today's network multimedia applications require less than 10 Mbps for operation, so Ethernet is still an acceptable LAN medium. The problem with Ethernet is that more of its 10 Mbps needs to be delivered to each user than is delivered by the typical shared network.

Figure 13-20 shows how microsegmentation can affect per-user bandwidth, thus allowing network multimedia applications that have high bandwidth requirements to run.

When using LAN switches to design networks to support multimedia applications, it is important to remember the following design constraints:

- Multicast packets are basically equivalent to broadcast packets.
- Switches flatten the network and cause broadcast packets (and multicast packets) to be flooded throughout the network.
- Virtual LANs (VLANs) can be used to control the size and scope of the broadcast domain and, therefore, the networks on which multicast packets are sent.
- Routers are required to allow VLANs to communicate with each other and to control the spread of multicast packets.
- VLANs and routers are required for scalability in switched LAN networks.
- Because it supports IGMP, the Catalyst 1200 switch is well-suited for networks that support network multimedia applications.

Figure 13-20 Effect of switches on usage patterns.



For more information about using LAN switches in your network design, see Chapter 12, “Designing Switched LAN Internetworks.”

WAN Designs

Although there are many different ways to increase LAN bandwidth, increasing WAN bandwidth is not so easy. Because it is expensive, WAN bandwidth is a scarce resource in many environments. Running multimedia applications across a WAN is a challenge.

If additional bandwidth is needed in the WAN, first look at available circuit-switched technologies: switched-56, switched-T1, and ISDN. With these services, charges are based on connect time, which in the case of multimedia means that charges will be based on the length of the multimedia session. In cases where the circuit-switched service is used with another connecting WAN service (switched or leased), the circuit-switched service can be configured as a backup service.

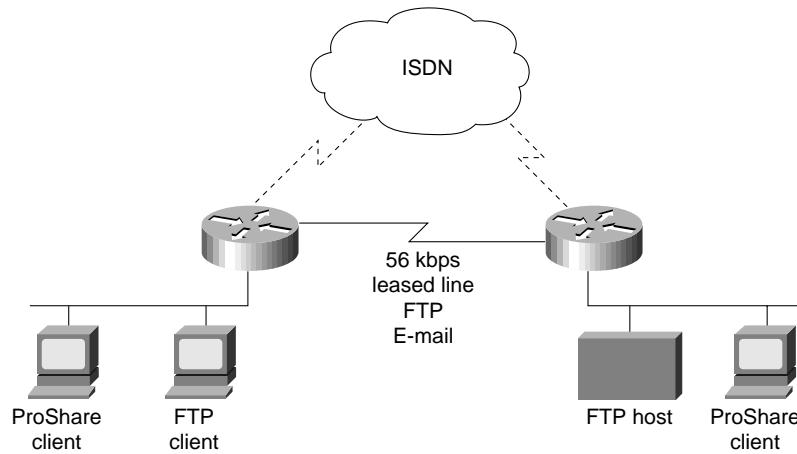
One way to improve utilization of WAN connections is to schedule WAN usage appropriately. On-demand applications (such as videoconferencing) typically consume WAN bandwidth during the working day, but other applications (such as video server applications) can be scheduled so that they consume bandwidth during off hours. A typical video server environment might have multiple video servers deployed in various sites. During the day, users access their local video server for training material or other video feeds. At night, when the WAN is idle, the video servers can replicate information and receive updates of new video content. By arranging to make use of unutilized WAN bandwidth at night, video servers can be maintained without adding to network traffic during the day.

Several Cisco IOS features can be used to control connect time and the type of data that flows over a WAN link, including snapshot routing, IPX and SPX spoofing, Name Binding Protocol (NBP) filtering, bandwidth on demand, and access lists. WAN connections should also take advantage of policy-based routing, which was introduced with Cisco IOS Software Release 11.0.

Policy-based routing is designed for networks in which both circuit-switched WAN and leased line connections are used. With policy-based routing, traffic can be routed over redundant WAN links based on traffic type (such as protocol or UDP port number). For example, policy-based routing can

be used to route email and FTP traffic over a serial link and to route Intel ProShare traffic across an ISDN link. In Figure 13-21, policy-based routing is used to configure a T1 interface for regular traffic and an ISDN interface for video-conferencing traffic.

Figure 13-21 Policy-based routing.



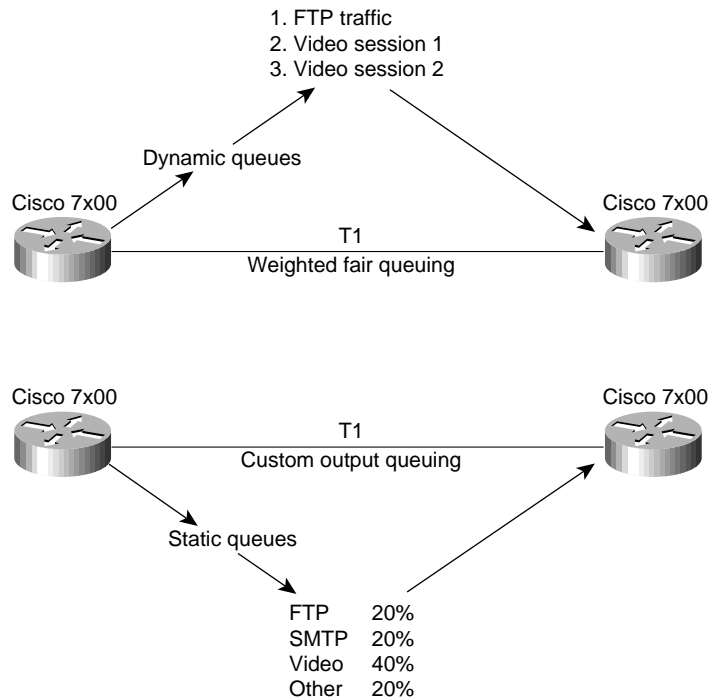
In Figure 13-21, the multimedia gets the required bandwidth from the circuit-switched service. Because the circuit-switched service is up only when the application is in use, WAN costs are controlled. Traditional LAN traffic runs separately on the leased line and experiences uninterrupted service.

Until WAN bandwidth becomes affordable at any speed, delivering bandwidth to applications over the WAN will remain a difficult task. Wherever possible, take advantage of circuit-switched technologies and Cisco IOS features such as policy-based routing and bandwidth on demand.

Note The Cisco IOS software includes two lossless data compression algorithms, STAC and Predictor, that compress data that is transmitted over WAN links. Neither algorithm should be used to compress video because they cannot achieve the compression ratios that are achieved by video and audio compression algorithms. In addition, do not use STAC and Predictor to compress video that has already been compressed. In most cases, instead of decreasing the size of a video or audio transmission, these algorithms increase it.

Additionally, take advantage of the priority queuing, custom queuing, and weighted fair queuing to optimize WAN traffic patterns. For example, set up a queue for a particular multicast session or use weighted fair queuing to dynamically queue the multicast stream, as shown in Figure 13-22.

Figure 13-22 WAN queuing techniques.



High-Speed LAN Designs

Many of today's network multimedia applications are packet-based audio or video applications. These applications are transmitted using the traditional OSI Layer 3 protocols: IP, IPX, and AppleTalk. Stream-based applications are best exemplified in ATM environments in which audio or video is captured and converted directly into ATM cells and transmitted *natively* using ATM through the ATM switch fabric. Typically, these multimedia applications are constant bit rate (CBR) and use AAL1 and circuit emulation for transmission.

It is important to ask the following questions of each network multimedia application in use:

- Is the application packet-based or stream-based?
- What are the bandwidth requirements?
- Does the application support multicast transmission?
- Does the application support quality of service parameters?

Designing a network to support packet-based video is quite different from designing a network for stream-based applications. Packet-based video is best deployed in networks built around switches and routers. To further tailor the network, virtual LAN (VLAN) technology can also be leveraged across the campus LAN and WAN.

In this model, ATM can be deployed as a backbone technology to interconnect different switches and VLANs. From an implementation standpoint, if IP is the only protocol on the network, the ATM part of the network can run classical IP over ATM, as defined in RFC 1577. However, if the ATM network needs to support additional protocols or IP multicast, the ATM network must run LAN Emulation (LANE) instead.

If resegmenting and microsegmenting an existing network, as described in the section “Traditional LAN Designs” earlier in this chapter, does not yield enough bandwidth to run network multimedia applications, or if a new network is being designed, consider the following high-speed LAN technologies:

- Fast Ethernet
- Fiber Distributed Data Interface and Copper Distributed Data Interface (FDDI and CDDI)
- Asynchronous Transfer Mode (ATM)

The combination of switches and routers interconnected using a high-speed backbone technology (Fast Ethernet, FDDI, or ATM) provides sufficient bandwidth for most network multimedia applications in the campus environment.

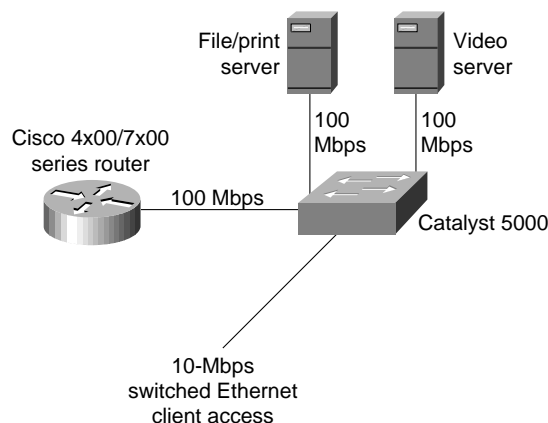
Fast Ethernet

Fast Ethernet (IEEE 802.3u), delivers 100-Mbps bandwidth over category 5 unshielded twisted-pair (UTP) wire or fiber-optic cable. Like 10-Mbps Ethernet, Fast Ethernet uses carrier sense multiple access collision detection (CSMA/CD) network access method. Perhaps the two best advantages of Fast Ethernet are that it is relatively inexpensive (assuming category 5 UTP is present) and that migration from traditional 10-Mbps Ethernet is simple. Fast Ethernet delivers bandwidth that allows for a variety of different network design scenarios:

- High-speed client-server connectivity
- High-speed interswitch communication
- High-speed backbone

High-speed client-server connectivity is a popular use for Fast Ethernet. In this scenario, servers (Novell NetWare, Windows NT, and SPARC servers) are on Fast Ethernet and transmit to clients connected via Fast Ethernet or switched 10-Mbps Ethernet. Fast Ethernet server connectivity works particularly well in video server environments where the server needs to deliver multiple video streams to its clients. The capability to take advantage of a high-speed connection is a product of the server’s architecture and the operating system that it runs. Novell NetWare, for example, can deliver substantial I/O caching, which in turn generates high-speed transfers. Figure 13-23 shows a design that gives users on 10-Mbps Ethernet access to file, print, and video servers located on 100-Mbps segments.

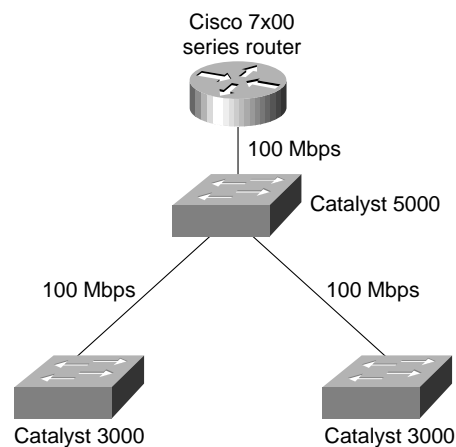
Figure 13-23 Fast Ethernet server access.



Using Fast Ethernet for high-speed client connectivity is also effective. Today, reasonably priced Fast Ethernet adapters are available for PCs (EISA and PCI) and SPARCstations (S-bus). Because installation is simple, Fast Ethernet provides a straightforward migration path to 100-Mbps bandwidth.

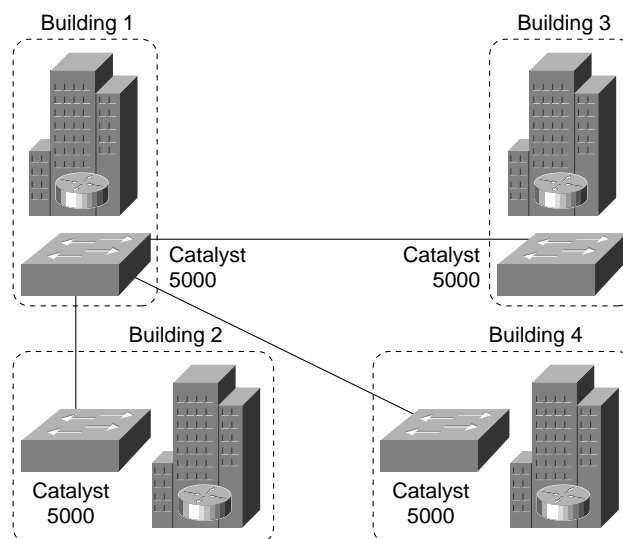
Fast Ethernet can also be used to interconnect Ethernet switch workgroups. In this scenario, a group of switches is interconnected using Fast Ethernet. This is particularly useful in a microsegmented environment in which each client has a dedicated 10-Mbps segment. With a Fast Ethernet connection between switches, a client can communicate with a client attached to a different switch without sacrificing bandwidth, as shown in Figure 13-24.

Figure 13-24 Fast Ethernet interswitch connections.



Fast Ethernet connections over category 5 UTP are limited to 100 meters in length. With fiber, Fast Ethernet can deliver connections up to two kilometers in length, allowing Fast Ethernet over fiber to be used as a backbone technology to interconnect various switched segments in a campus environment, as shown in Figure 13-25.

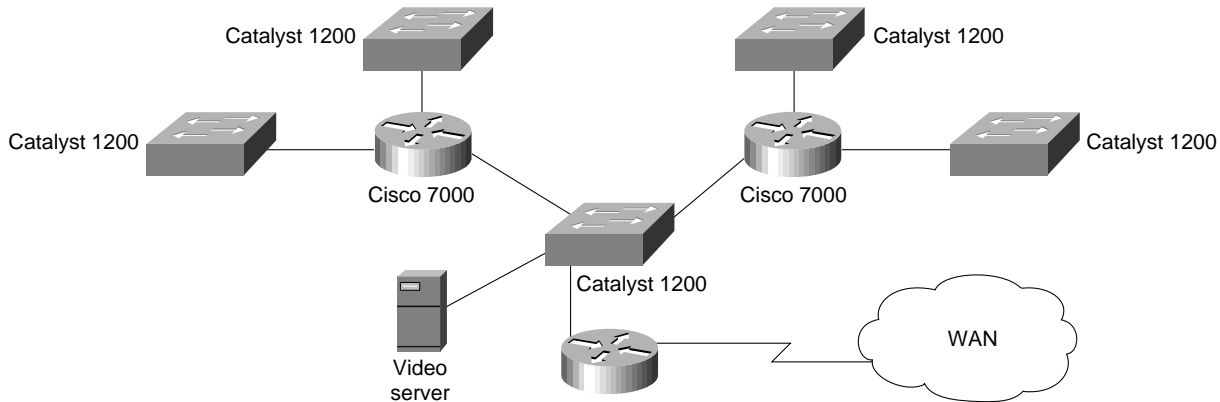
Figure 13-25 Fast Ethernet backbone.



In practice, Fast Ethernet is rarely used as a core backbone technology because FDDI and ATM offer advanced features that make them more viable for backbone implementations.

The design shown in Figure 13-26 works well for low-port density switched Ethernet environments, using switches for client and server access and routers for core connectivity. This design controls multicast traffic by deploying IGMP at the switch port, which allows multicast traffic to be sent only to ports that have registered an IGMP Join.

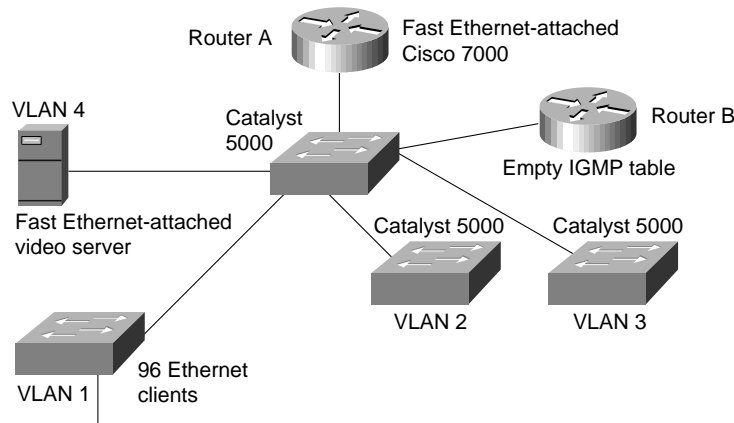
Figure 13-26 Low-port density design.



For high-port density Ethernet or Token Ring environments, a combination of routers and Catalyst 3000, Catalyst 1600, or Catalyst 5000 switches is effective. The design relies on VLAN technology to control multicast traffic. VLAN technology permits the creation of multiple bridge groups within a switch or across high-speed backbones with remote switches. With VLANs, multicast transmission can be limited to only the desired ports by creating a specific VLAN that includes only the multicast sender and the multicast recipients.

Designing VLANs to support multicast applications hinges largely on the application in use. Figure 13-27 is an example of a campus design that uses a single network TV multicast application.

Figure 13-27 Network TV multicast design.



In Figure 13-27, there is only one VLAN per switch, resulting in a large number of clients per VLAN. The video source resides on the high-speed backbone and is in its own VLAN. During the multicast transmission, the video source sends a video stream out the high-speed connection. Router A receives the video stream and sends it out its high-speed link to the VLANs on the Catalyst 5000 switches.

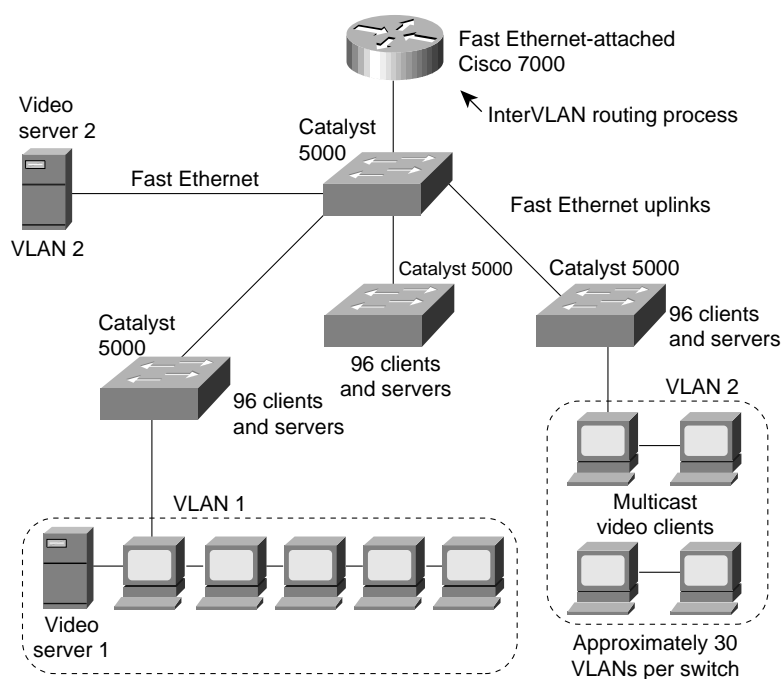
When a VLAN receives a multicast stream from the router, it forwards it to all members of that VLAN. Therefore, this design works well for environments in which every client tunes in to the network TV transmission. If only a few clients per VLAN tune in to the broadcast and the remaining clients task the network for other services, the multicast traffic can hinder overall network performance.

The routers support IGMP, which limits multicast traffic to only those interfaces that have registered IGMP Joins from clients. In Figure 13-27, Router B has no IGMP receivers in its table and therefore multicast traffic is not forwarded out any of its interfaces.

To impose greater control over multicast transmission, a microVLAN strategy can be used. In this scenario, a switch has multiple VLANs (thereby limiting the multicast traffic to fewer ports). MicroVLANs are best used in multipoint videoconferencing environments and environments where there are multiple multicast video sources. In these environments, many different multicast transmissions may occur simultaneously, which can impose some scalability issues unless the multicast traffic can be contained.

Figure 13-28 shows a microVLAN design in which the VLANs are aligned based on multicast demands. VLAN 1, for example, contains clients that primarily receive video from Video server 1. VLAN 1 also receives video from Video server 2, which is the corporate broadcast service.

Figure 13-28 MicroVLAN design.



The microVLAN approach minimizes the effects of multicast traffic by creating many small broadcast domains using VLANs.

One issue to keep in mind with the microVLAN design is that it might violate the 80/20 rule for designing VLANs. VLAN design is optimized when at least 80 percent of the traffic is intraVLAN and at most 20 percent of the traffic is interVLAN. Essentially, performance is optimized when traffic remains within the local VLAN. If VLANs are aligned based on multicast clients and servers, there is a good chance that access to other servers, such as the email server, will be interVLAN. Because interVLAN communication must be handled by a router, as interVLAN communication increases, route processing increases. Ultimately, the number of VLANs per router port should be determined by the multicast applications in use and their respective bandwidth requirements.

Compared with low-bandwidth multicast applications, high-bandwidth multicast applications place a greater constraint on the number of VLANs on a router interface. For additional information about VLANs, see Chapter 12, “Designing Switched LAN Internetworks.”

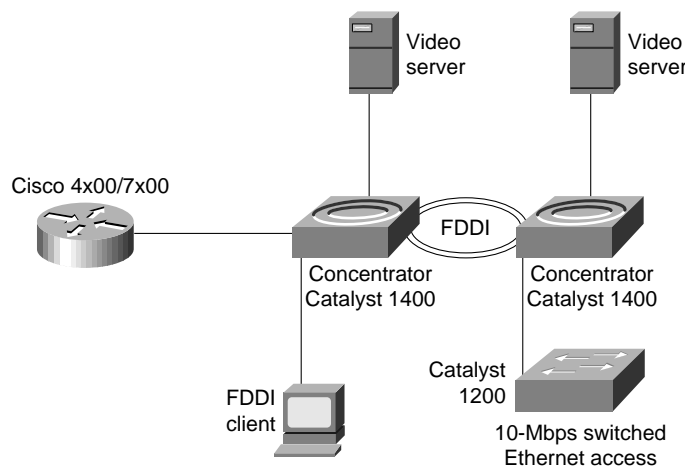
Fiber Distributed Data Interface and Copper Distributed Data Interface

Fiber Distributed Data Interface (FDDI) and Copper Distributed Data Interface (CDDI) deliver bandwidth that allows for a variety of different network design scenarios. FDDI is particularly attractive as a backbone technology for the following reasons:

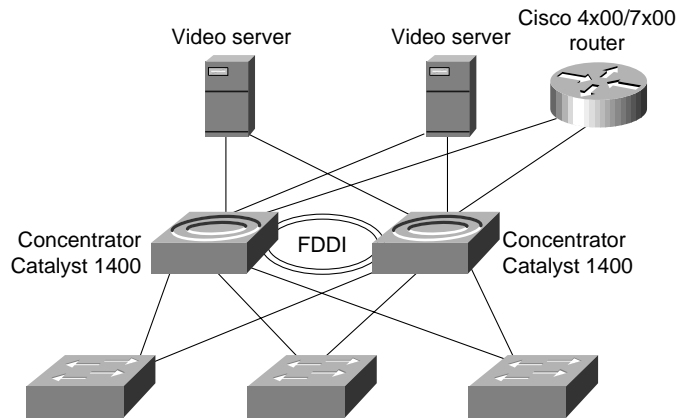
- *Distance capabilities*—With multimode fiber, an FDDI connection can span 2 kilometers. With single mode fiber, an FDDI connection can span 10 kilometers. This capability allows tremendous flexibility for interconnecting LAN segments in a campus environment.
- *Fault tolerance and redundancy*—FDDI’s inherent fault tolerance and its ability to support designs such as dual-homing also make the technology attractive in backbone environments.
- *Security*—Optical transmission makes it more difficult for hackers to tap into compared to traditional copper transmission.

Like Fast Ethernet, FDDI and CDDI can deliver high-speed client connectivity, but most often, FDDI and CDDI are used for server and backbone connections, especially in video server environments where multiple video streams are sent to video clients, as shown in Figure 13-29.

Figure 13-29 FDDI or CDDI server access.

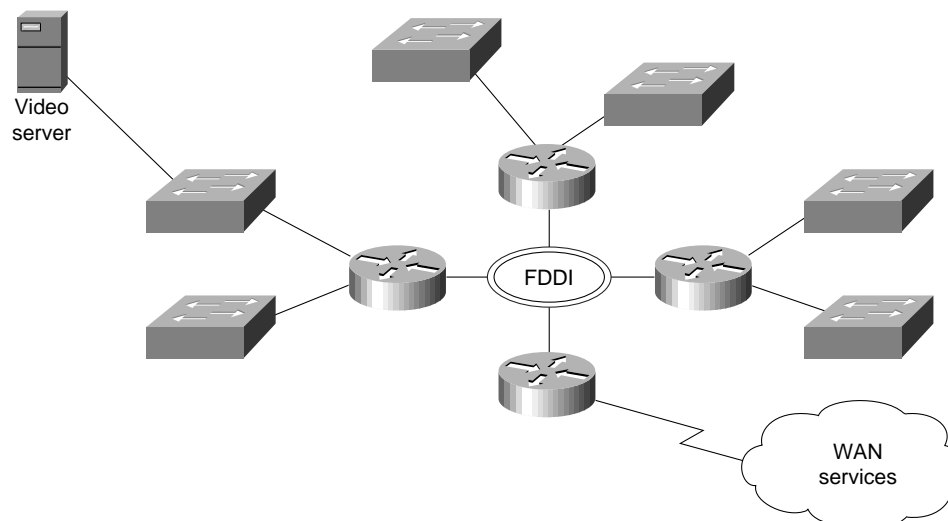


In addition to delivering high bandwidth, FDDI and CDDI deliver better redundancy than Fast Ethernet. With FDDI and CDDI, a server can be *dual-homed* to FDDI or CDDI concentrators, as shown in Figure 13-30. Dual-homing gives a server access to two FDDI or CDDI rings. Under normal circumstances, the server uses only one ring. If the primary ring fails, the server can fall back to the secondary ring, maintaining connectivity with no down time. Dual-homing requires that the server FDDI or CDDI adapter be a Dual Attached Station (DAS) adapter (as opposed to a Single Attached Station [SAS] connector, which provides a single physical connection).

Figure 13-30 FDDI dual-homed design.

Clients attached to different Ethernet switch workgroups can gain high-speed intercommunication, which allows a client connected to one Ethernet switch to access a video server or initiate a videoconferencing session with a resource connected to another Ethernet switch. In this design, dual-homing can be implemented. An FDDI-equipped switch can be dual-homed to two different concentrators, providing greater redundancy and fault tolerance.

The design shown in Figure 13-31 works for point-to-point applications that only impose bandwidth demands on the network, but it is vulnerable to multicast applications. The switch transmits OSI Layer 2 multicast frames to all ports in the same manner as it transmits OSI Layer 2 broadcast frames. For example, if a client accesses a multicast video stream on a server, the multicast transmission is forwarded to all switch ports, which undermines the performance benefits of switching.

Figure 13-31 Switch/router campus design.

Asynchronous Transfer Mode

Asynchronous Transfer Mode (ATM) has gained much attention as the next-generation LAN and WAN technology. Much of the excitement about ATM centers around the fact that ATM delivers an entirely switch-based fabric and offers high-speed connectivity (100-Mbps TAXI, 155-Mbps OC-3

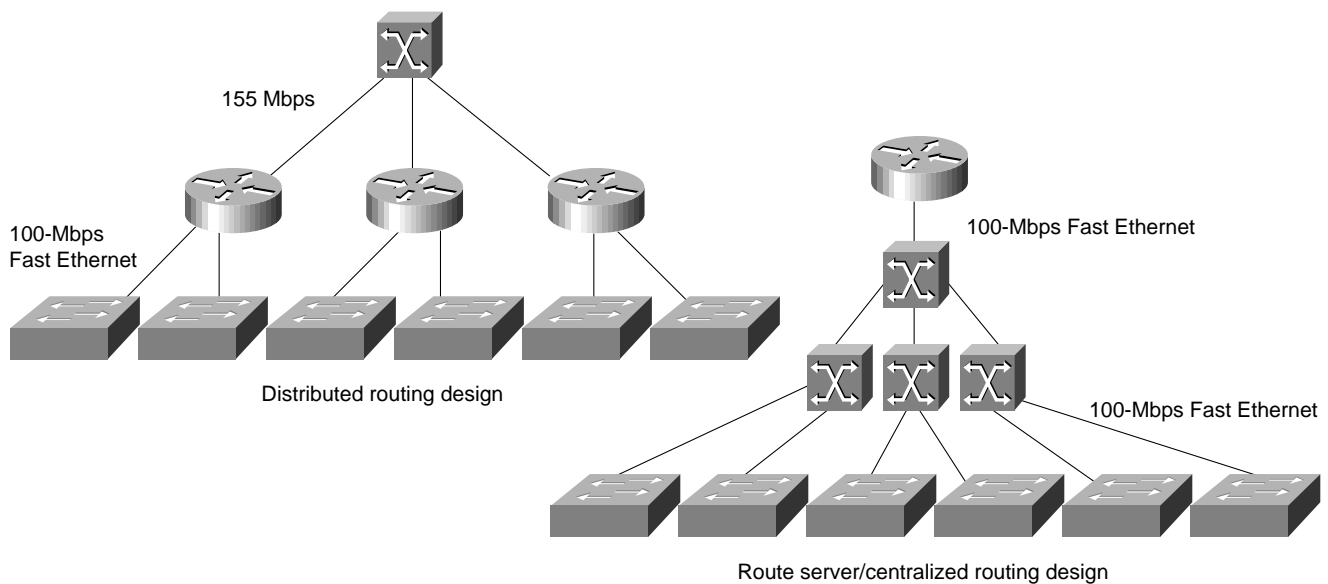
and in the future 622-Mbps OC-12). Besides the raw bandwidth that ATM provides, the technology also offers extensive support for transporting video, voice, and data. As Figure 13-32 illustrates, a variety of different design scenarios are possible using ATM equipment.

From a bandwidth perspective, ATM offers considerable flexibility for running network multimedia applications. Although ATM provides features, such as quality of service support, that make it an attractive environment in which to run network multimedia applications, ATM is not a prerequisite for running network multimedia applications. Rather, today’s existing LAN technologies can also support many network multimedia applications.

LAN Emulation

LAN Emulation (LANE) defines a service interface for Open Systems Interconnection (OSI) Layer 3 protocols that is identical to that of existing LANs and encapsulates data sent across the ATM network in the appropriate LAN MAC packet format. It makes no attempt to emulate the actual media access control protocol of the specific LAN concerned (that is, CSMA/CD for Ethernet or token passing for IEEE 802.5).

Figure 13-32 Enterprise ATM network design.

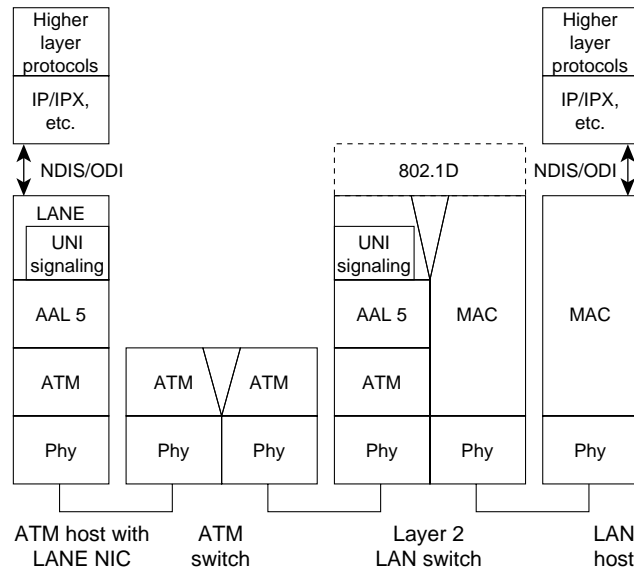


Currently, LANE does not define a separate encapsulation for FDDI. FDDI packets are mapped into Ethernet or Token Ring-emulated LANs (ELANs) using existing translational bridging techniques. Because they use the same packet formats, the two most prominent new LAN standards, Fast Ethernet (100BaseT) and IEEE 802.12 (100VG-AnyLAN), can be mapped unchanged into either the Ethernet or Token Ring LANE formats and procedures.

LANE supports a range of maximum packet (MPDU) sizes, corresponding to maximum size Ethernet, 4-Mbps and 16-Mbps Token Ring packets, and to the value of the default MPDU for IP over ATM. Typically, the size of the MPDU depends on the type of LAN that is being emulated and on the support provided by LAN switches bridged to the ELAN. An ELAN with only native ATM hosts, however, may optionally use any of the available MPDU sizes, even if a size does not correspond to the actual MPDU in a real LAN of the type being emulated. All LAN Emulation clients (LECs) within a given ELAN must use the same MPDU size. Put simply, LANE makes an ATM network look and behave like an Ethernet or Token Ring LAN—albeit one operating much faster than such a network.

The advantage of LANE is that it allows higher-layer protocols to work without modification over ATM networks. Because LANE presents the same service interface of existing MAC protocols to network-layer drivers (for example, an NDIS- or ODI-like driver interface), no changes are required in those drivers. See Figure 13-33 for a representation of the LANE protocol architecture.

Figure 13-33 LANE protocol architecture.



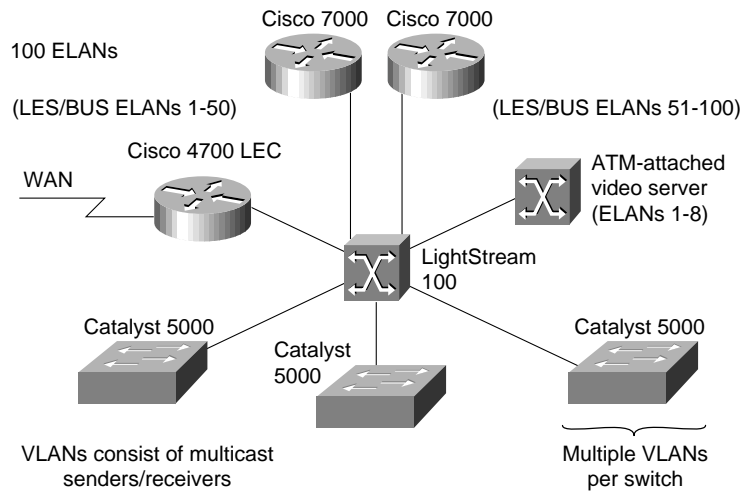
The goal of LANE is to accelerate the deployment of ATM at the same time that work continues on the full definition and implementation of native mode network-layer protocols.

LANE Designs

When designing with LANE, the primary issues typically center on the scalability of LAN Emulation servers (LESs) and broadcast and unknown servers (BUSs). Currently, all multicast transmission relies on the BUS for delivery to all LAN Emulation clients (LECs) within a given ELAN.

In a Cisco ATM network, the router operates as the BUS for a given ELAN. If the router supports multiple ELANs, it runs multiple BUS processes. Router performance is a function of the number of ELANs the router is a member of and the number of BUS processes that it executes. In environments in which there are a large number of ELANs, additional routers should be deployed to handle BUS functionality for each ELAN. Essentially, BUS functionality is distributed across a set of routers in the ATM network, as shown in Figure 13-34.

Figure 13-34 Distributed LES/BUS design.



Currently, LANE is the only ATM technology that addresses multicast packet-based video. Classical IP over ATM (RFC 1577) has no provision for resolving OSI Layer 2 multicast addresses into ATM addresses. For more information about LANE, see Chapter 8, “Designing ATM Internetworks.”

Native Mode ATM

Native mode ATM protocols bypass the MAC address encapsulation of LANE. In native mode, address resolution mechanisms map network-layer addresses directly into ATM addresses, and the network-layer packets are then carried across the ATM network. Currently, IP is the only protocol for which extensive native-mode work has been done.

From the perspective of running network multimedia applications, one of the most compelling reasons for running native mode protocols is quality of service support. LANE deliberately hides ATM so any network-layer protocol that operates over ATM cannot gain access to the quality of service properties of ATM and must therefore use unspecified bit rate (UBR) or available bit rate (ABR) connections only. Currently, this is not a major restriction because all network protocols were developed for use over existing LAN and WAN technologies, none of which can deliver a guaranteed quality of service. Consequently, no existing network-layer protocol can request a specific quality of service from the network or deliver it to a higher-layer protocol or application. In turn, most network applications today do not expect to receive any guaranteed quality of service from the underlying network protocol, so they do not request it.

For a long time, IP has had optional support for type of service (TOS) indications within the IP header that could theoretically be used to provide a rudimentary form of quality of service support. In practice, however, almost no end-system or intermediate-system IP implementations have any support for TOS because TOS indications cannot be mapped into any common underlying networking technology. Few, if any, IP routing protocols use the TOS bits, and no applications set them.

At best, all current network-layer protocols expect and deliver only a “best effort” service—precisely the type of service that the ABR service was designed to offer. Just as LANE adapts the connection-oriented nature of ATM to offer the same type of connectionless service that is expected by network-layer protocols, so ABR hides the guaranteed quality of services features of ATM to offer the best effort service expected by these protocols. As such, ABR and LANE perfectly complement each other.

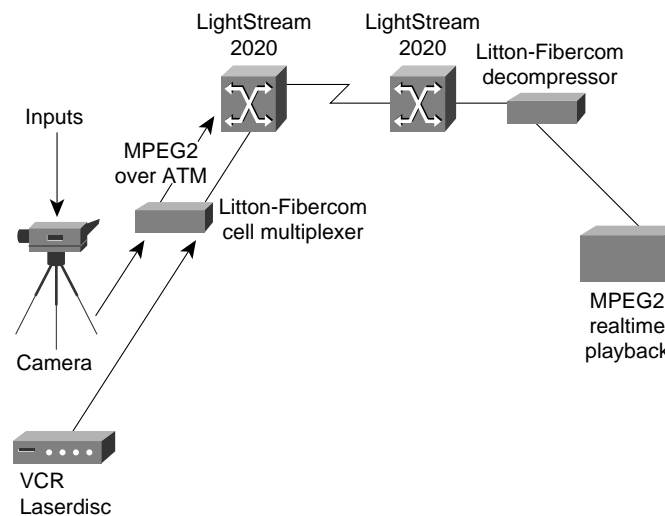
As ATM networks proliferate, it is likely that demand will grow to use the quality of service features of ATM, which will spur application development expressly designed to take advantage of ATM and ATM quality of service.

Native ATM Designs

As mentioned earlier in this chapter, LANE is best suited for “best effort” traffic (that is, ABR traffic) but is not well-suited for applications that require more predictable network service, such as CBR and VBR multimedia applications. For these applications, it is best to run native ATM. In a native ATM environment, digital video and audio is sent to a service multiplexer that segments the audio and video streams into cells and forwards them out to ATM-attached clients that receive the streams.

MPEG2, which is a VBR application, is a good example of a native ATM application. With MPEG2, video can be digitized and compressed in real time and then put into ATM cells for delivery to ATM-attached clients. Figure 13-35 shows an example of MPEG2 running over ATM.

Figure 13-35 MPEG2 over ATM.



Multimedia Applications in ATM Networks

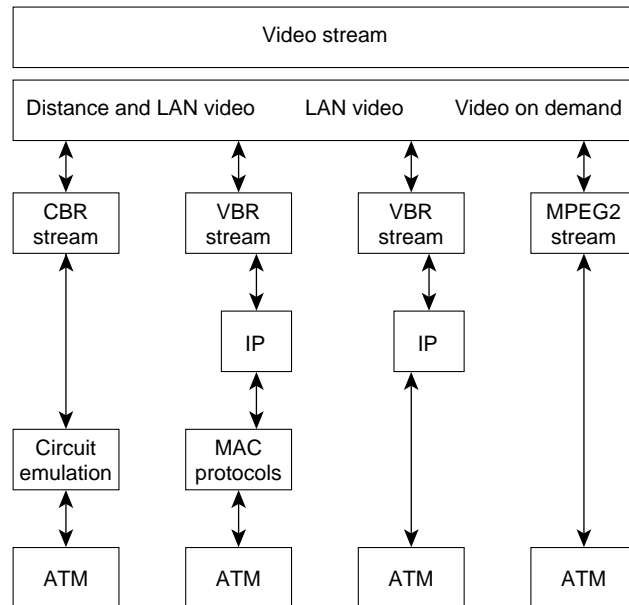
Within an ATM network, connections are categorized into various quality-of-service types: constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), and unspecified bit rate (UBR). For the most part, network multimedia applications are CBR or VBR. CBR video applications are designed to run over traditional 64-Kbps or multiple 64-Kbps lines. With ATM, CBR video is transported using circuit emulation, which means that the ATM switch must support circuit emulation.

ATM switches that do not have CBR line cards must have a service multiplexer. The multiplexer has inputs for CBR traffic at T1/E1 and T3/E3 speeds and can adapt those streams to ATM. For example, the Litton-FiberCom ATM multiplexer features real-time video encoding and provides ATM adaptation with an OC-3 (155 Mbps) ATM port.

VBR video applications, which are commonly seen in traditional LAN environments, are more bursty than CBR applications. VBR applications are often referred to as *packetized video*. The video compression algorithm, such as MPEG, generates VBR output that is packetized onto the LAN. In ATM, VBR applications can run using LANE or can run natively using IP over ATM.

MPEG2 is a special case of VBR that can run directly on ATM, bypassing LANE and IP altogether. In this case, there is an MPEG2-to-ATM convergence layer in which MPEG2 information is translated into ATM cells. Figure 13-36 shows how CBR and VBR map into ATM.

Figure 13-36 Video stream protocol mappings.



Depending on the type of ATM service requested, the network is expected to deliver guarantees on the particular mix of quality of service elements (such as cell loss ratio, cell delay, and cell delay variation) that are specified at the connection setup.

In UNI 3.0/3.1, the traffic parameters and requested quality of service for a connection cannot be negotiated at setup, nor can they be changed over the life of the connection. UNI 4.0 will support connection quality of service negotiation.

ATM Multicasting

There are two fundamental types of ATM connections:

- Point-to-point connections, which connect two ATM end systems. Such connections can be unidirectional or bidirectional.
- Point-to-multipoint connections, which connect a single source end system (known as the *root node*) to multiple destination end systems (known as *leaves*). Cell replication is done within the network by the ATM switches at which the connection splits into two or more branches. Such connections are unidirectional, permitting the root to transmit to the leaves but not permitting the leaves to transmit to the root, or to each other, on the same connection.

An analog to the multicasting or broadcasting capability common in many shared-media LAN technologies, such as Ethernet and Token Ring, is notably missing from these types of ATM connections. In such technologies, multicasting allows multiple end systems to receive data from other multiple systems and to transmit data to these multiple systems. Such capabilities are easy to implement in shared-media technologies such as LANs, where all nodes on a single LAN segment must necessarily process all packets sent on that segment. The obvious analog in ATM to a multicast LAN group would be a bidirectional, multipoint-to-multipoint connection. Unfortunately, this obvious solution cannot be implemented when using ATM Adaptation Layer 5 (AAL5), the most common ATM adaptation layer used to transmit data across ATM networks.

Unlike AAL3/4, with its Message Identifier (MID) field, AAL5 does not have any provision within its cell format for the interleaving of cells from different AAL5 packets on a single connection. Therefore, all AAL5 packets sent to a particular destination across a particular connection must be received in sequence, with no interleaving between the cells of different packets on the same connection, or the destination reassembly process would not be able to reconstruct the packets.

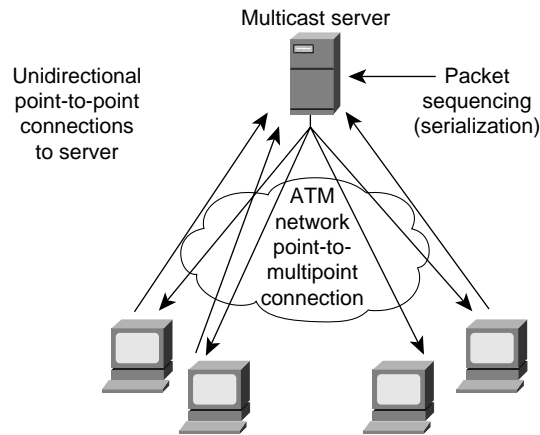
Despite the problems that AAL5 has with multicast support, it is not feasible to use AAL3/4 as an alternative for data transport. AAL3/4 is a much more complex protocol than AAL5 and would lead to much more complex and expensive implementations. Indeed, AAL5 was developed specifically to replace AAL3/4. Although the MID field of AAL3/4 could preclude cell interleaving problems, allowing for bidirectional, multipoint-to-multipoint connections, this would also require some mechanism for ensuring that all nodes in the connection use a unique MID value. There is no such mechanism currently in existence or development; the number of possible nodes within a given multicast group would also be severely limited due to the small size of the MID field.

ATM AAL5 point-to-multipoint connections can be only unidirectional because if a leaf node were to transmit an AAL5 packet onto the connection, it would be received by both the root node and all other leaf nodes. However, at these nodes, the packet sent by the leaf could be interleaved with packets sent by the root, and possibly other leaf nodes; this would preclude the reassembly of any of the interleaved packets. Clearly, this is not acceptable.

Notwithstanding this problem, ATM requires some form of multicast capability because most existing protocols (having been developed initially for LAN technologies) rely on the existence of a low-level multicast/broadcast facility. Three methods have been proposed for solving this problem:

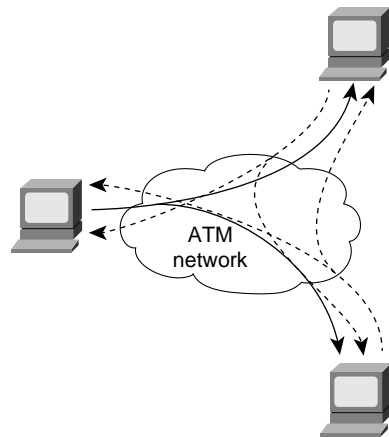
- *VP-multicasting*—With this mechanism, a multipoint-to-multipoint VP links all nodes in the multicast group, and each node is given a unique VCI value within the VP. Interleaved packets can be identified by the unique VCI value of the source. Unfortunately, this mechanism requires a protocol that uniquely allocates VCI values to nodes, and such a protocol does not currently exist. It is also not clear whether current segmentation and reassembly (SAR) devices could easily support such a mode of operation. Moreover, UNI 3.0/3.1 does not support switched virtual paths. UNI 4.0, however, should add this capability.
- *Multicast server*—With this mechanism as illustrated in Figure 13-37, all nodes wanting to transmit onto a multicast group set up a point-to-point connection with an external device known as a multicast server (perhaps better described as a resequencer or serializer). The multicast server, in turn, is connected to all nodes that want to receive the multicast packets through a point-to-multipoint connection. The multicast server receives packets across the point-to-point connections, and then retransmits them across the point-to-multipoint connection—but only after ensuring that the packets are serialized (that is, one packet is fully transmitted prior to the next being sent). In this way, cell interleaving is precluded.

Figure 13-37 Multicast server operation.



- *Overlaid point-to-multipoint connections*—With this mechanism, all nodes in the multicast group establish a point-to-multipoint connection with each of the other nodes in the group, and, in turn, become a leaf in the equivalent connections of all other nodes. Therefore, as shown in Figure 13-38, all nodes can both transmit to and receive from all other nodes.

Figure 13-38 Overlaid point-to-multipoint connections.



Overlaid point-to-multipoint connections require each node to maintain n connections for each group, where n is the total number of transmitting nodes within the group. The multicast server mechanism requires only two connections. Overlaid point-to-multipoint connections also require a registration process for telling a node that joins a group what the other nodes in the group are, so that the joining node can form its own point-to-multipoint connection. The other nodes also need to know about the new node so they can add the new node to their own point-to-multipoint connections. The multicast server mechanism is more scalable in terms of connection resources but has the problem of requiring a centralized resequencer, which is both a potential bottleneck and a single point of failure.

In short, there is no ideal solution within ATM for multicast. Higher layer protocols within ATM networks use both the multicast server solution and the overlaid point-to-multipoint connection solution. This is one example of why using existing protocols with ATM is so complex. Most current protocols, particularly those developed for LANs, implicitly assume a network infrastructure similar to existing LAN technologies—that is, a shared-medium, connectionless technology with implicit broadcast mechanisms. ATM violates all of these assumptions.

Work in Progress

In the case of IP, the IETF has developed the notion of an Integrated Services Internet, which envisages a set of enhancements to IP to allow it to support integrated or multimedia services. These enhancements include traffic management mechanisms that closely match the traffic management mechanisms of ATM. For instance, protocols such as Resource Reservation Protocol (RSVP) are being defined to allow for resource reservation across an IP network, much as ATM signaling does within ATM networks.

RSVP is an advanced method for dynamically allocating bandwidth to network-based applications running in traditional packet-based networks. RSVP will be particularly useful for CBR multimedia applications because it will allow a network application to request a specific quality of service from the network. It will be the responsibility of internetworking devices (such as routers) to respond to the RSVP request and to establish a connection through the network that can support the requested quality of service.

The IP Version 6 (IPv6) protocol (formally known as the IP Next Generation [IPng] protocol), which the IETF is now developing as a replacement for the current IPv4 protocol, incorporates support for a flow ID within the packet header. The network uses the flow ID to identify flows, much as VPI/VCI (virtual path identifier/virtual channel identifier) are used to identify streams of ATM cells. Protocols such as RSVP will be used to associate with each flow a flow specification that characterizes the traffic parameters of the flow, much as the ATM traffic contract is associated with an ATM connection.

The IETF is also in the process of developing a new transport protocol, the Real-Time Transport Protocol (RTP). RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data (such as audio, video, or simulation data) over multicast or unicast network services. RTP builds on protocols like RSVP for resource reservation and on transport technologies such as ATM for quality of service guarantees. The services provided by RTP to real-time applications include payload type identification, sequence numbering, time stamping, and delivery monitoring.

The concept of a Multicast Address Resolution Server (MARS), which can be considered the analog of the ARP server in RFC 1577, is also in development. A MARS serves a group of nodes known as a *cluster*. All end systems within the cluster are configured with the ATM address of the MARS. The MARS supports multicast through *multicast meshes* of overlaid point-to-multipoint connections, or through multicast servers.

Summary

This chapter addressed how to effectively deploy network multimedia applications. Specifically, this chapter addressed the following topics:

- Multimedia basics, including analog video, digital video, video compression, and digital audio standards
- Using networked multimedia applications, including bandwidth and quality of service requirements
- Understanding multicasting, including Internet Group Management Protocol, Distance Vector Multicast Routing Protocol, Multicast Open Shortest Path First, Protocol Independent Multicast, and Simple Multicast Routing Protocol
- Network designs for multimedia applications, including traditional LAN designs, WAN designs, and high-speed LAN designs

