Design and Implementation Guide-Part 2

December 1995

Jeff Baher
jbaher@cisco.com
Technical Marketing

Networked Multimedia

V. Quality of Service

Besides delivering a wide-range of hardware platforms and software features that help to deliver bandwidth to the Campus LAN and WAN, Cisco also offers a variety of Quality of Service (QoS) features for network multimedia applications.

Data, voice and video each have different QoS requirements. Real-time multimedia applications, such as videoconferencing, impose requirements on an internetwork because the traffic they produce must be delivered on a certain schedule or it becomes useless. Unlike traditional “best-effort” data services, such as File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), or X Windows, in which variations in latency often go unnoticed, video and audio data are useful only if delivered within a specified time period. Later delivery only impedes the usefulness of other information within the stream.

At a general level, the two primary forces working against network multimedia applications, besides the bandwidth issues, are latency and jitter.

Latency

Real-time, interactive applications such as desktop conferencing are sensitive to accumulated delay, which is referred to as latency. For example, 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 it takes information to travel the distance of the line. This period 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 it takes to send the packet across a given media. Transmission delay is determined by the speed of the media and the size of the packet (e.g. LocalTalk vs. Ethernet vs. FDDI)
- Store-and-forward delay—The length of time it takes for an internetworking device such as a switch, bridge, or router to receive a packet before it can send it.
- Processing delay—The time required by a networking device for route lookup, changing the header, and other switching tasks. In some cases, the packet also must be manipulated; for instance, changing the encapsulation type, changing the hop count, and so on. Each of these steps can contribute to the processing delay.

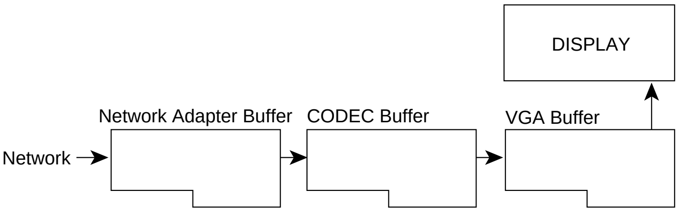
Jitter

If a network provides variable latency for different packets or cells, it introduces jitter, which is particularly disruptive to audio communications because it can cause audible pops and clicks. Multimedia networks should provide techniques to minimize jitter for traffic that it adversely affects; these techniques include increasing bandwidth or leveraging capabilities such as Cisco’s custom and priority queuing to improve usage of the existing network infrastructure.

Many applications provide utilities for minimizing jitter. The most common among these places the data in a buffer, from which the display software or hardware pulls data. The insulating 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 are therefore not normally perceivable on the output side. Figure 39 shows a typical buffering strategy that helps to minimize latency and jitter inherent in a given network.

Collectively, latency and jitter can be minimized in part by implementing some sort of buffering technique. Buffering can be performed within the network itself as well as with host equipment. Buffering, in this sense, acts as a regulator to offset inherent irregularities (latency/jitter) that occur during transmission. Take for example a client connecting 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. The net 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. Buffering can play a large role in displaying video, especially over today's existing networks. Keep in mind, buffering alone will not solve all the problems. This is due mostly to the fact that most buffers are not that large compared to the amount of data in a video file.

Figure 1. Buffering



There are several Cisco IOS features that can help deliver varying degrees of QoS. And as always, Cisco is continuing to develop new techniques for delivering a consistent quality of service.

Cisco IOS Priority Output Queuing and Custom Output Queuing

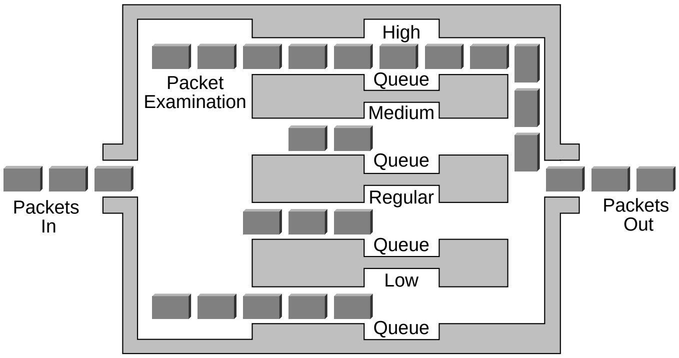
Cisco IOS currently offers two queuing features: Priority Output Queuing and Custom Queuing. Both of these queuing methods have their roots in integrating SNA traffic onto LANs. With SNA, the challenge is to preserve the predictable response time of the SNA network while ensuring the throughput requirements of LAN applications.

One approach to providing predictable performance is to increase line speeds to assure that adequate bandwidth is available during peak traffic conditions. While this may be a reasonable approach for backbone links, it may not be a cost-effective method of attaching remote sites to the backbone. A better approach may be to use lower-speed lines and give mission-critical data priority over less critical transmissions during peak traffic conditions.

To achieve this either Cisco IOS Priority Output Queuing or Custom Output Queuing can be used.

Priority output queuing allows a 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, traffic on the next highest-priority queue is transmitted, and so on. This mechanism assures that during congestion, the highest-priority data does not get delayed by lower-priority traffic. However, if the traffic sent to a given interface exceeds the bandwidth of that interface, lower-priority traffic can experience significant delays.

Figure 2. Priority Output Queuing Process



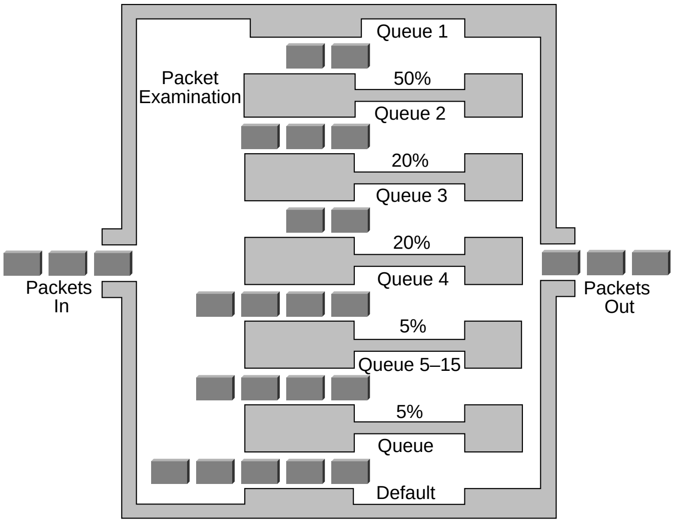
Below is a sample configuration and network design in which Priority Queuing is used to guarantee a consistent QoS level for Intel ProShare video conferencing.

```
!
hostname bertha
!
enable password ######
!
username bertha password 7 2394943E02B17
!
interface Ethernet0
ip address 171.68.158.49 255.255.255.248
!
interface Serial0
ip address 171.68.158.26 255.255.255.248
encapsulation PPP
ppp authentication chap
priority-group 1
!
ip route 131.108.0.0 255.255.0.0 171.68.158.25
ip route 171.69.0.0 255.255.0.0 171.68.158.25
access-list 101 permit ip any any
!
priority-list 1 protocol IP high TCP 23
priority-list 1 protocol UDP 5715 medium
!
```

For networks that need to provide a guaranteed level of service for all traffic, Cisco offers Custom Output Queuing. Custom Output Queuing allows a customer to reserve a percentage of a link, a variable bandwidth for specified protocols. Network managers can define up to 16 output queues for normal data and an additional queue for system messages such as LAN keepalive messages (routing packets are not assigned to the system queue). Cisco routers service each queue sequentially, transmitting a configurable percentage of traffic on each queue before moving on to the next one. Custom Queuing guarantees that mission-critical data is always assigned a certain percentage of the bandwidth, but also assures predictable throughput for other traffic.

To provide this feature, Cisco routers determine how many bytes 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.

Figure 3. Custom Output Queuing Process



Independent tests of Cisco routers have shown that when a line is saturated with traffic from multiple protocols, traffic carried on the line maintains allocated bandwidth percentages to within a few percentage points of what was specified in the configuration.

A key advantage of Cisco's "bandwidth reservation" technique is that unused bandwidth can be 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.

Below is a sample configuration and network design in which Custom Output Queuing is used to ensure a certain QoS level for Intel ProShare video conferencing.

```
!
hostname bertha
!
enable password ######
!
username bertha password 7 2394943E02B17
!
interface Ethernet0
ip address 171.68.158.49 255.255.255.248
!
interface Serial0
ip address 171.68.158.26 255.255.255.248
encapsulation PPP
ppp authentication chap
custom-queue-list 1
!
ip route 131.108.0.0 255.255.0.0 171.68.158.25
ip route 171.69.0.0 255.255.0.0 171.68.158.25
access-list 101 permit ip any any
!
queue-list 1 queue 1 byte-count 579
queue-list 1 queue 2 byte-count 193
queue-list 1 queue 3 byte-count 193
!
queue-list 1 protocol IP 1 UDP 5715
queue-list 1 protocol IP 2 TCP 23
queue-list 1 protocol DECNET 3
!
```

Priority Output Queuing and Custom Queuing Latest Enhancements:

With Release 11.0 of Cisco IOS, the number of queues that can be used for Priority Output Queuing and Custom Queuing has been increased to 16. Release 11.0 also adds a Management Information Base (MIB) for providing detailed access to Priority and Custom queuing information. This MIB provides information currently available via the "show queue" EXEC command.

Weighted Fair Queuing

In addition to Priority Output Queuing and Custom Queuing, Release 11.0 of Cisco IOS (available in September 1995) offers a new queuing method: Weighted Fair Queuing. Weighted Fair Queuing is a sophisticated traffic priority management algorithm that identifies conversations (traffic streams) and then breaks up the trains of packets belonging to each conversation to ensure that the capacity is shared fairly between individual conversations. The algorithm automatically sorts among conversations without the need for the user to define access lists. This separation is accomplished by examining sufficient fields in the packet header to identify unique 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.

Fair queuing provides an automated way to stabilize network behavior during congestion and results in increased performance and reduced retransmission. Weighted Fair Queuing does not require the user to define access lists like in the past with priority and custom queuing. In most cases, this feature will provide much better and smoother end-to-end performance over a given link and may reduce the need to resolve link congestion without an expensive increase in bandwidth.

Weighted Fair Queuing is available for all platforms running Cisco IOS. It will be the default for most serial interfaces; however, it will be possible to configure Priority Queuing or Custom Queuing for serial interfaces as needed.

Note: Fair Queuing is disabled by default for all LAN interfaces and for serial interfaces configured for X.25, LAPB, and SDLC.

Advanced Flow Control: Resource Reservation Protocol (RSVP)

The following discussion of flow control and RSVP is taken from the Network Flow Management document available at http://cio.cisco.com/warp/customer/614/18.html

Resource Reservation Protocol, RSVP for short, is quickly gaining recognition as an advanced method for dynamically allocating bandwidth to network-based applications. Cisco recognizes RSVP's tremendous potential to deliver a Quality of Service (QoS) framework to traditional packet-based networks. To that end, Cisco has committed to integrating RSVP into Cisco IOS and the weighted fair queuing framework around the Q2 1996 timeframe.

RSVP in particular is valuable for constant bit rate multimedia applications (see ATM QoS section for constant bit rate definition). With RSVP a network-application can make a request to the network for a specific quality of service. It is the responsibility of the internetworking devices (i.e. routers) to respond to the RSVP request and to establish a connection through the network that can support the requisite quality of service.

To understand how RSVP works it is important to first discuss the concept of "flow" within a network. In general, traffic in an Internet falls into two major groups and each major group has two subgroups. The two major groups are elastic and real-time traffic.

Elastic traffic is either interactive or transaction oriented and therefore sensitive to delay, or is a bulk transfer of data and therefore sensitive to bandwidth. In ATM literature, elastic traffic is also called available bit rate (ABR) traffic. This refers to its other defining characteristic: not only can applications that use it speed up or slow down according to the bit rate available, but bulk transfers will tune themselves to use it all.

Real-time traffic lacks this elasticity, but also has definable average and peak data rates and loss and delay expectations. It either requires that the network rarely lose traffic and deliver consistent performance, or it accepts jitter and moderate loss of traffic in exchange for priority to maintain timeliness of delivery. These service classes are called guaranteed and predictive service, respectively.

One of the fundamental problems in dealing with Internet traffic is that hosts tend to emit traffic in a bursty fashion, which results in closely spaced sequences of messages that keep getting in each other's way. Seen another way, these sequences of messages are flows of information moving through the Internet; they are the commerce of related applications. These relationships, the formulated requirements they impose, and the message streams they exchange, are called flows. They may be unicast (system to system) or multicast (system to set of systems). RFC 1190 describes a flow as "a directed tree carrying traffic away from a source to all destinations."

Requirements of Multimedia Applications

As mentioned earlier, multimedia applications impose requirements on an internetwork, because the traffic they produce must be delivered on a certain schedule or it becomes useless. Unlike elastic traffic, whose data is valid without respect to time, video and audio data is useful only if delivered within the frame window. If delivered later, it only impedes the usefulness of a later frame. Silent periods in voice traffic (periods of no traffic) can be stretched somewhat, but stretched sounds make words difficult to understand and can change their meanings.

As discussed earlier, the applications account for variability in the network by buffering data. This is not without cost, however; if an MPEG CODEC can produce 12 MBps, then 1.5 megabits are necessary to buffer for one second. Applications must trade off the amount of available buffer against the network's ability to satisfy them. When sufficient buffering is not possible, the application must request the network to induce less jitter (variability in interframe arrival timing), and prepare itself to accept frame loss if necessary to honor its request.

We have observed that multimedia applications may have specific bandwidth needs. These may crowd out other traffic by reducing its available bandwidth, or other traffic may render the bandwidth unavailable by filling it. When the flow has hard requirements, the correspondents must communicate their needs to the network to ensure that they can be met.

Interacting with each of these is the requirement for quality. Applications producing high-quality, real-time graphics use a large buffer and ask the network to avoid dropping traffic. When the cost of quality is exceeds its utility or the network's ability to provide it, the user will settle for lower quality. This Quality of Service (QoS) requirement is likewise a necessary parameter.

Flow Specifications

Elastic traffic is difficult to characterize beyond describing its general attributes. Real-time traffic, however, can be described in reasonably precise terms. The output rate of video CODECs can be measured, for example MPEG CODECs tend to average about 3 to 7 MBits and peak to 12 in normal, 30-frame-per-second video. Such a statement of requirements is called a flow specification.

RFC 1363 describes a flow specification as "a data structure used by internetwork hosts to request special services of the internetwork, often guarantees about how the internetwork will handle some of the hosts' traffic. In the future, hosts are expected to have to request such services on behalf of distributed applications such as multimedia conferencing."

Attributes of a Flow

The information needed by the network to determine bandwidth reservation, then, is as follows:

- Mean data rate
- The largest amount of data the router will keep in queue
- Minimum quality of service

The mean data rate informs the router how much bandwidth it must reserve for the traffic in question. This will affect its queuing algorithm; flows requiring high bandwidth will be given different weights than traffic requiring lower bandwidth.

The largest amount of data the router will keep in queue determines latency. A new message arriving will be delayed by a time proportional to the amount of traffic that is ahead of it in the queue. To guarantee latency bounds, the router must be able to predict queue depths.

Minimum quality of service is the question of whether traffic requires preemptive service ("predictive" service) or whether it requires only that it be isolated from the rogue behavior of others ("guaranteed" service). The router may and usually will give better service than is required.

Identifying the Affected Traffic

The network also needs appropriate information to determine which traffic is subject to that reservation. This data is provided in an information element called a "filter specification." A filter specification includes:

- Who is going to send it
- Who wants to receive it from which senders
- Other identifying characteristics

From an IP (layer 3) perspective, these requirements translate to:

- IP Source Address
- IP Destination Address
- IP Protocol (usually UDP)
- DP Port Number

Style of Reservation to Install

Beyond this, though, there is an association that needs to be made between the flow specification and the variety of reservation it creates. Audio and video applications have very different reservation requirements.

Wildcard Filters

The term "wildcard" implies a filter specification that selects all senders. A wildcard reservation automatically extends to new senders to the session as they appear.

A wildcard-filter-style reservation creates a single resource "pipe" along each link that is shared by data packets from all senders for the given session. The "size" of this pipe is the largest of the resource requests for that link from all receivers, independent of the number of senders using it.

The typical use of such a filter is audio traffic, in which there are many potential speakers in a conversation, but (if it is a polite conversation) only one of them speaks at a time. Thus, regardless of the number of speakers, bandwidth need be reserved for exactly one.

Fixed Filters

A fixed-filter-style reservation request creates one reservation per specified sender. If a receiver specifies a reservation for each of five senders, for example, five separate reservations are installed. The typical use of this is video traffic, in which several senders may be simultaneously originating traffic, and the receiver needs to receive it all regardless of who is “holding the floor” at the moment.

Note, however, that multiple receivers each indicating a need to receive from a given sender do not install separate reservations. Rather, the largest of the presented reservations is granted, and the rest are assumed to be using the same bandwidth. This is a valid assumption in multicast applications, wherein a single message in might be forwarded to many receivers.

Dynamic Filters

A dynamic-filter-style reservation decouples reservations from filters. Each such reservation request specifies several distinct reservations to be made using the same flow specification. The number of reservations that are actually made in a particular node is dependent on the number of senders upstream from the node.

Once a dynamic-filter-style reservation has been established, the receiver can change the set of filter specifications to specify a different selection of senders without a new admission control check. This is known as “channel switching,” in analogy with a television set.

To provide assured channel switching, each node along the path must reserve enough bandwidth for all channels, even though some of this bandwidth may be unused at any one time. If the number of required reservations changes (because the receiver changed or because the number of upstream sources changed), or if the common flow specification changes, the refresh message is treated as a new reservation that is subject to admission control and may fail.

Like a fixed-filter-style reservation, a dynamic-filter-style reservation causes distinct reservations for different senders. Like a fixed-filter-style reservation, the dynamic filter is intended for use in video applications.

RSVP’s Relationship with Other DDN Protocols

Like other protocols in the Defense Digital Networking suite, RSVP does not attempt to solve the whole problem. The whole problem includes: the encapsulation and forwarding of unicast and multicast datagrams, addressing of sender and receivers, routing of flows, multicast group membership, and the like. RSVP solves the resource reservation problem, leaving solutions to other parts to the following:

- IP Datagram Transport
- IP Routing
- Internet Group Management Protocol
- Multicast Routing

IP Datagram Transport, in this context, refers both to the algorithms for forwarding unicast IP datagrams described in RFC 791, and the algorithms for multicast datagram forwarding described in RFC 1112 and its successors.

IP Routing is familiar; OSPF, BGP-4, Enhanced IGRP, and other protocols provide this service. It determines how unicast traffic will travel from a specific source to a specific destination.

The Internet Group Management Protocol (IGMP) is also described in RFC 1112; it enables senders and receivers to advertise that they want to be members of specific IP Multicast groups.

Multicast routing is provided by such protocols as Protocol Independent Multicast (PIM). In certain cases, multicast routing uses the same path that a unicast would follow. It may even inspect the unicast route table to use its decision. In other cases, it may optimize for the multicast group.

RSVP System Responsibilities

RSVP permits participants in flows (potentially any flow, but primarily targeting multimedia flows) to advise the network of their needs, and for the network to configure itself to meet them. The participants are identified as senders, receivers, and network elements.

At each “node” (router or host) along the path, RSVP passes a new resource reservation request to an admission control routine to determine whether there are sufficient resources available. If there are, the node reserves the resources and updates its packet scheduler and classifier control parameters to provide the requested Quality of Service.

RSVP’s design goals include:

- To support multicast or unicast data delivery
- To reserve resources for simplex data streams
- To minimize the application state
- For data flow receivers, to initiate and maintain the resource used for that flow
- To maintain a “soft state” in the routers
- To gracefully support dynamic membership changes
- To automatically adapt to routing changes
- To provide several reservation models to fit a variety of applications.
- To operate transparently through routers that do not support it

Minimization of application state is a goal that drives many aspects of the architecture. Alternatives include having each sender know and inform the network of all receivers and having the network cache all information. In some applications, all receivers are also senders; however, it is not uncommon for there to be a few senders and many receivers. In a sender-oriented or network-oriented design, the latter class of flow would cache significant state in places where it is not especially useful. In a receiver-oriented design, each participant in the flow caches just enough information to play its role.

We now turn to the responsibilities of the different varieties of participants in a flow.

RSVP Receivers

Systems that receive multimedia traffic (which may also be senders, but that is another context) have a simple set of responsibilities. They must know that a particular IP unicast or multicast address is being used to send data that they are interested in. They use the Internet Group Management Protocol (IGMP), as defined in RFC 1112, to join this multicast group, or they open a TCP or UDP connection for unicast traffic. From the RSVP “Path Message” that they start to receive, or from the TCP connection information, they can now determine how to identify the messages that they want to receive, which is usually a matter of UDP or TCP port numbers.

Armed with this information, they use the RSVP Reserve (RESV) message to periodically advertise to the network their interest in the flow; later, they will either (passively) stop sending RESVs or (actively) use the information to advertise a decision to stop receiving.

The RESV message contains the source address of the requester and the destination IP address that the data stream is using. It goes on to specify some number of coupled flow and filter specifications. On the basis of these specifications, the network installs the necessary bandwidth reservations and queuing parameters.

RSVP Senders

RSVP senders periodically emit a message called a PATH message. The PATH message indicates that the system is a sender and contains the information required by the network to route RESVs and the information required by the receiver to build the RESV message. This information includes:

- Destination address (IP multicast address)
- Reservation ID
- Previous-hop IP address (used in forwarding RESV messages)
- Templates for identifying traffic from that sender
- Flow specification describing the sender’s output

This message is sent down the multicast path of the network just as the data it describes is sent. The routers capture it, however, and update the previous-hop IP address before sending it on. This way, the reservation information is consolidated and presented to all receivers.

The flow specification carried in this message, however, does not necessarily result in a reservation of like magnitude being installed. The receiver is expected to use it as an upper bound. If, for example, the sender is using an MPEG CODEC to encode video data, the network may not have enough bandwidth to carry 5 MBps of data through it and might refuse all reservations. The receiver will then have to determine what quality is acceptable and reduce the reservation accordingly.

RSVP Networks

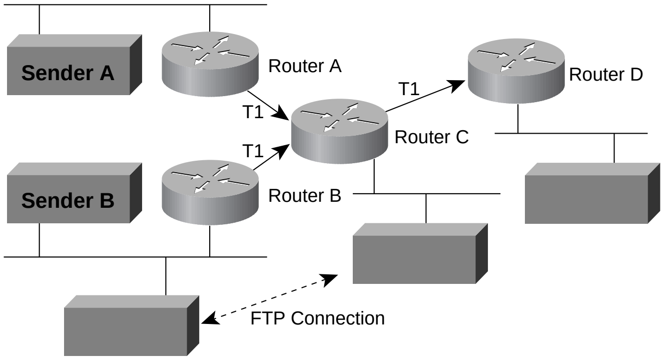
These responsibilities and messages described above meet in the network. The routers understand the current network topology, the characteristics of the interfaces they support, and the amount of current reservation. Thus, when a receiver indicates a need to hear from a sender, the router can adjust its parameters to provide a viable service for the receiver or tell the receiver that the capability just isn’t there.

RSVP Examples

Selective Optimization—Example 1

There is a network multicast running with two senders (two systems generating video data) each generating 600-kbps video feeds and some number of receivers (see Figure 42). A new receiver wants to start receiving this video information.

Figure 4. Example Network #1



In addition, there is other traffic in the network; notably, several lengthy file transfers using one of the necessary lines.

Because this is a multicast network, the new receiver uses IGMP to ask to be added to the video multicast, and a multicast tree is built that starts delivering the data on a best-effort basis. The user receives traffic from Sender A, but not from Sender B.

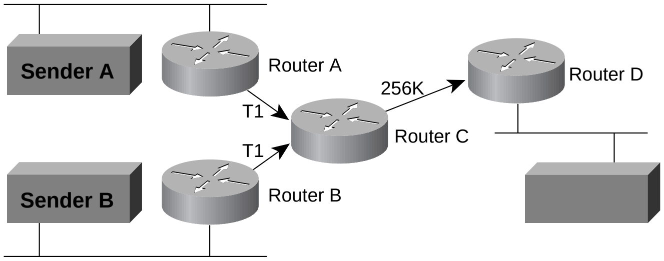
Senders A and B are periodically sending RSVP PATH messages that indicate what sort of traffic they are producing. The receiver sends a single RESV message asking to reserve bandwidth for the traffic from Sender A and Sender B.

- Router D determines that the Ethernet connecting it to the new receiver has adequate bandwidth available, and sends the RESV to Router C.
- Router C determines that there is sufficient bandwidth (600 kbps + 600 kbps = 1.2 MBps) on the T1 to Router D (although there is but 336K left over) and installs the queuing parameters giving that traffic 78 percent of the line.
- It also determines that there are different paths to Senders A and B. It divides the RESV into two messages, and forwards them to Routers A and B.
- Router A determines that there is enough bandwidth (600 kbps) available for Sender A's traffic, installs the queuing parameters giving that traffic 39 percent of the line, and forwards the RESV to Sender A. The effect is nominal in the absence of other traffic.
- Router B determines that there is enough bandwidth (600 kbps) available for Sender B's traffic, installs the queuing parameters giving that traffic 39 percent of the line, and forwards the RESV to Sender B. The effect on the file transfers is that they no longer can "hog" the line and are forced to slow down to give preference to the video data.

Selective Optimization—Example 2

There is a network multicast running, with two senders (two systems generating video data) each generating 150-kbps video feeds and some number of receivers (see Figure 43). A new receiver wants to start receiving this video information.

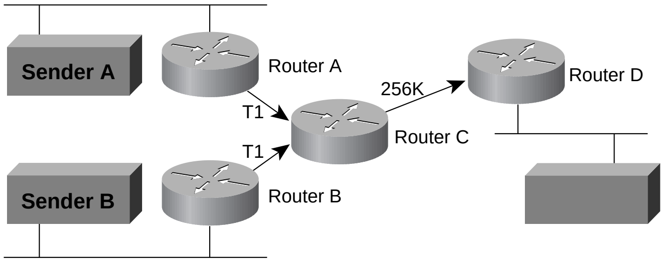
Figure 5. Example Network #2



Because this is a multicast network, the new receiver uses IGMP to ask to be added to the video multicast, and a multicast tree is built that starts delivering the data on a best-effort basis. However, because the total data being sent exceeds the capacity of the network to carry it, and because there may be other traffic, the service is not what the receiver would like to see.

Senders A and B are periodically sending RSVP PATH messages that indicate what sort of traffic they are producing. The receiver now decides that Sender A's traffic is more important (that's where the most useful information is coming from), and wants to improve the quality from there. Upon receiving a PATH message from A, the sender can now initiate a reservation toward A and tell the network how to improve the flow from A (see Figure 44).

Figure 6. Installed Reservations from A to New Receiver



The effect of these reservations is probably not great on the T1 line from Router A to Router C, because it represents a small percentage (10 percent) of the total bandwidth. However, the effect on the 256-kbps line is dramatic. Router C adjusts its queuing to ensure that the video feed has all the bandwidth it needs, at the expense of all other data.

If, however, the new receiver attempts to make the same reservation toward Sender B, Router C advises that the bandwidth simply is not there.

ATM QoS

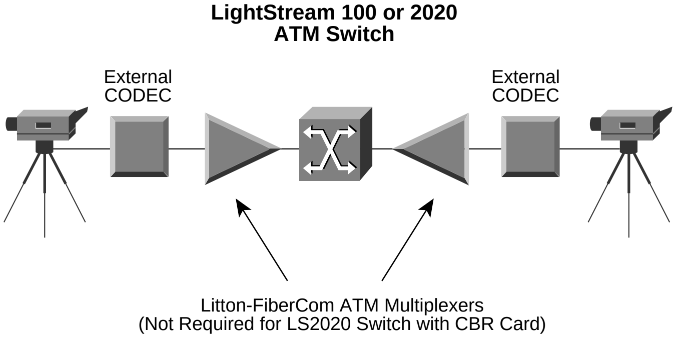
Much of the discussion of ATM QoS is taken from Anthony Alles' *ATM Internetworking* document available at http://cio.cisco.com/warp/customer/614/12.html.

Any discussion about QoS would not be complete without a discussion of ATM and ATM's inherent QoS features. Within an ATM network, connections are categorized into various ATM QoS types: CBR (constant bit rate), VBR (variable bit rate), ABR (available bit rate), and UBR (unspecified bit rate), depending upon the nature of the QoS guarantee desired and the characteristics of the expected traffic types.

For the most part network multimedia applications will be either CBR or VBR. Constant bit rate video applications are designed to run over traditional 64Kbps or multiple 64Kbps lines. With ATM, CBR video will be transported using circuit emulation. As a result, the ATM switch must support circuit emulation. The LightStream 2020 does.

For ATM switches that do not have CBR line cards, a service multiplexer is needed. The multiplexer has inputs for CBR traffic at T1/E1 and T3/E3 speeds and can adapt those streams to ATM. Litton-FiberCom makes an ATM multiplexer which provides ATM adaptation with an OC-3 (155 Mbps) ATM port.

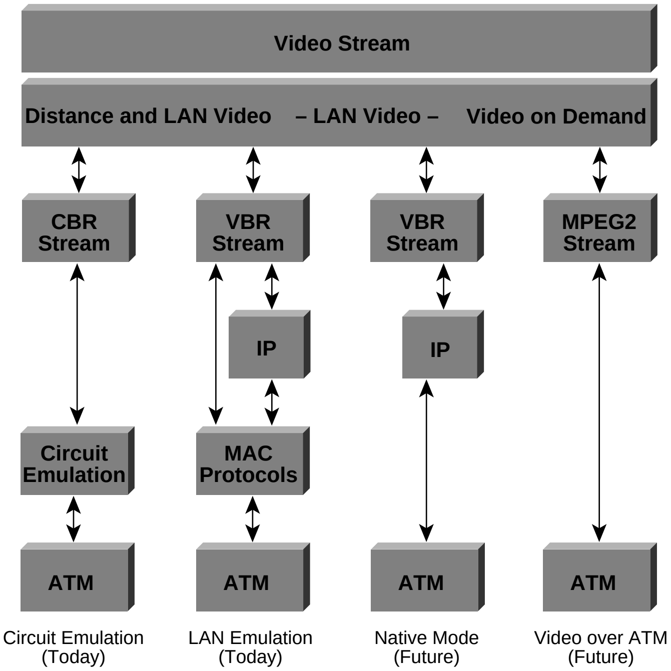
Figure 7. Service Multiplexers for CBR Video


LightStream 100 or 2020
ATM Switch
External
CODEC
External
CODEC
Litton-FiberCom ATM Multiplexers
(Not Required for LS2020 Switch with CBR Card)

Variable bit rate video applications which make up the lot of today's multimedia applications are more bursty in nature than CBR applications. VBR applications, often time referred to as packetized video, are commonly seen in traditional LAN environments. The video compression algorithm used, MPEG for instances, generates variable bit rate output (based on key frames and delta frames) which is packetized onto the LAN. In ATM, VBR applications can run using LAN emulation (LANE) or by running natively using IP over ATM.

MPEG2 is a special VBR case which can run directly on ATM, bypassing LANE and IP altogether. In this case, there is an MPEG-2-to-ATM convergence layer in which MPEG-2 information is translated into ATM cells. Refer to the diagram below for CBR and VBR mappings into ATM.

Figure 8. Video Stream Protocol Mappings



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

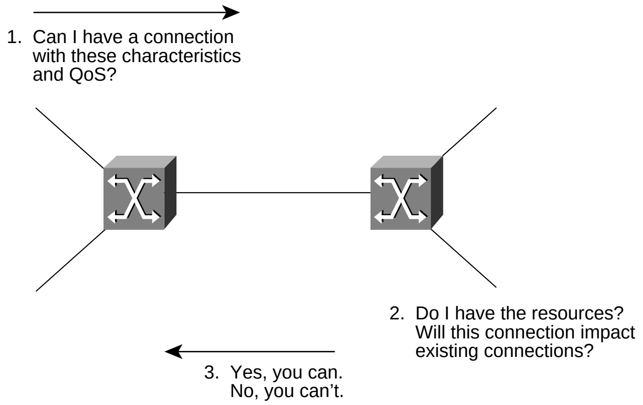
Note: In UNI 3.0/3.1, the traffic parameters and requested QoS for a connection cannot be negotiated at set-up, or changed over the lifetime of the connection. UNI 4.0 will support connection QoS negotiation; how this will be supported within P-NNI is for future study.

QoS and the P-NNI

Based on P-NNI signaling (Private-Network to Network Interface), ATM switches implement a function known as connection admission control (CAC) to deliver QoS guarantees. Whenever a connection request is received by the switch, the switch performs the CAC function. That is, based upon the traffic parameters and requested QoS of the connection, the switch determines whether setting up the connection violates the QoS guarantees of established connections (for example, by excessive contention for switch buffering). The switch accepts the connection only if violations of current guarantees are not reported. CAC is a local switch function, and is dependent on the architecture of the switch and local decisions on the strictness of QoS guarantees.

The VC (virtual channel) routing protocol must ensure that a connection request is routed along a path that leads to the destination and has a high probability of meeting the QoS requested in the connection set up—that is, of traversing switches whose local CAC will not reject the call.

Figure 9. Connection Admission Control


1. Can I have a connection with these characteristics and QoS?
2. Do I have the resources? Will this connection impact existing connections?
3. Yes, you can. No, you can't.

To install QoS parameters, the P-NNI protocol uses a topology state routing protocol in which nodes flood QoS and reachability information so that all nodes obtain knowledge about reachability within the network and the available traffic resources within the network. Such information is passed within P-NNI topology state packets (PTSP), which contain various type-length-value (TLV) encoded P-NNI topology state elements (PTSE). This is similar to current link state routing protocols such as OSPF. Unlike these, however, which only have rudimentary support for QoS, the P-NNI protocol supports a large number of link and node state parameters that are transmitted by nodes to indicate their current state at regular intervals, or when triggered by particular events.

There are two types of link parameters: non-additive link attributes used to determine whether a given network link or node can meet a requested QoS; and additive link metrics that are used to determine whether a given path, consisting of a set of concatenated links and nodes (with summed link metrics), can meet the requested QoS.

The current set of link metrics are:

- Maximum cell transfer delay (MCTD) per traffic class.
- Maximum cell delay variation (MCDV) per traffic class
- Maximum cell loss ratio (MCLR) for CLP=0 cells, for the CBR and VBR traffic classes
- Administrative Weight: This is a value set by the network administrator and is used to indicate the desirability or otherwise of a network link.

The current set of link attributes are:

- Available Cell Rate (ACR): A measure of the available bandwidth in cells per second, per traffic class
- Cell Rate Margin (CRM): A measure of the difference between the effective bandwidth allocation per traffic class, and the allocation for sustainable cell rate; this is a measure of the safety margin allocated above the aggregate sustained rate
- Variance Factor (VF): A relative measure of CRM margin normalized by the variance of the aggregate cell rate on the link

There is currently some controversy as to whether the CRM and VF add much value to the GCAC (Generic CAC, see below)—the traffic passing through ATM switches may prove to be so irregular (for example, cell peaks may be bunched) that such second order statistics may prove to be too volatile and yield little useful information. Calculating such statistics is also non-trivial, particularly in the presence of aggregation.

All network nodes can obtain an estimate of the current state of the entire network through flooded PTSPs that contain such information as listed above. Unlike most current link state protocols, the P-NNI protocol advertises not only link metrics, but also nodal information. Typically, PTSPs include bi-directional information about the transit behavior of particular nodes based upon entry and exit port, and current internal state. This is particularly important in cases where the node represents an aggregated network. In such a case, the node metrics must attempt to approximate the state of the entire aggregated network. This internal state is often at least as important as that of the connecting links for QoS routing purposes.

The need to aggregate network elements and their associated metric information also has important consequences on the accuracy of such information, as discussed below.

Two approaches are possible for routing a connection through the network: hop-by-hop routing and source routing. Hop-by-hop routing is used by most current network layer protocols such as IP or IPX, where a packet is routed at any given node only to another node—the "next hop"—closer to the final destination. In source routing, the initial node in the path determines the entire route to the final destination.

Hop-by-hop routing is a good match for current connectionless protocols because they impose little packet processing at each intermediate node. The P-NNI protocol, however, uses source routing for a number of reasons. For instance, it is very difficult to do true QoS-based routing with a hop-by-hop protocol since each node needs to perform local CAC and evaluate the QoS across the entire network to determine the next hop. Hop-by-hop routing also requires a standard route determination algorithm at each hop to preclude the danger of looping.

However, in a source-routed protocol, only the first node would ideally need to determine a path across the network, based upon the requested QoS and its knowledge of the network state, which is gained from the PTSPs. It could then insert a full source routed path into the signaling request that would route it to the final destination. Ideally, intermediate nodes would only need to perform local CAC before forwarding the request. Also, since it is easy to preclude loops when calculating a source route, a particular route determination algorithm does not need to be standardized, leaving this as another area for vendor differentiation.

This description is only ideal, however because in practice, the source routed path that is determined by a node can only be a best guess. This is because in any practical network, any node can have only an imperfect approximation to the true network state because of the necessary latencies and periodicity in PTSP flooding. As discussed in the next section, the need for hierarchical summarization of reachability information also means that link parameters must also be aggregated. Aggregation is a "lossy" process, and necessarily leads to inaccuracies. Furthermore, as noted above, CAC is a local matter. In particular, this means that the CAC algorithm performed by any given node is both system dependent and open to vendor differentiation.

The P-NNI protocol tackles these problems by defining a Generic CAC (GCAC) algorithm. This is a standard function that any node can use to calculate the expected CAC behavior of another node, given that node's advertised additive link metrics, described above, and the requested QoS of the new connection request. The GCAC is an algorithm that was chosen to provide a good prediction of a typical node-specific CAC algorithm, while requiring a minimum number of link state metrics. Individual nodes can control the degree of stringency of the GCAC calculation involving the particular node by controlling the degree of laxity or conservativeness in the metrics advertised by the node.

The GCAC actually uses the additive metrics described above; indeed these metrics were selected to support the GCAC algorithm chosen for the P-NNI protocol. Individual nodes (physical or logical) will need to determine and then advertise the values of these parameters for themselves, based upon their internal structure and loading. Note, however, that the P-NNI Phase 1 GCAC algorithm is primarily designed for CBR and VBR connections; variants of the GCAC are used depending upon the type of QoS guarantees requested and the types of link metrics available, yielding greater or lesser degrees of accuracy.

The only form of GCAC done for UBR connections is to determine whether a node can support such connections. For ABR connections, a check is made to determine whether the link or node is authorized to carry any additional ABR connections and to ensure that the ACR for the ABR traffic class for the node is greater than the Minimum Cell Rate specified by the connection.

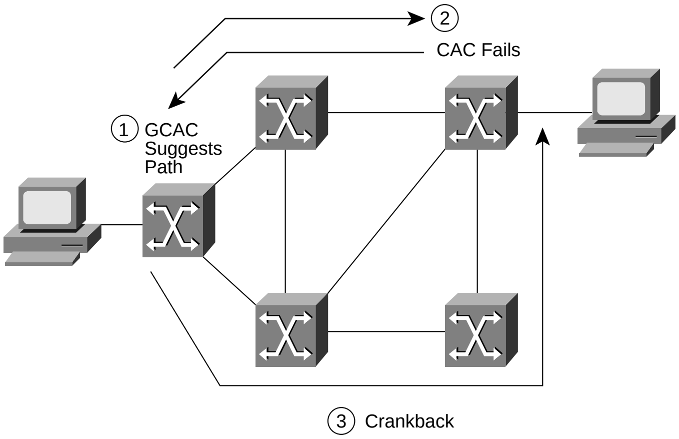
Using the GCAC, a node presented with a connection request (which passes its own CAC) processes the request as follows:

1. All links that cannot provide the requested ACR, and those whose CLR exceeds that of the requested connection, are "pruned" from the set of all possible paths using the GCAC.
2. From this reduced set, along with the advertised reachability information, a shortest path computation is performed to determine a set of one or more possible paths to the destination.
3. These possible paths are further pruned by using the additive link metrics, such as delay, and possibly other constraints. One of the acceptable paths would then be chosen. If multiple paths are found, the node may optionally perform tasks such as load balancing.
4. Once such a path is found (note that this is only an "acceptable" path to the destination, not the "best" path, the protocol does not attempt to be optimal), the node constructs a designated transit list (DTL) that describes the complete route to the destination (the structure of the DTL is described below) and inserts this into the signaling request. The request is then forwarded along this path.

This, however, is not the end of the story. Each node in the path still performs its own CAC on the routed request because its own state may have changed since it last advertised its state within the PTSP used for the GCAC at the source node. Its own CAC algorithm is also likely to be somewhat more accurate than the GCAC. Hence, notwithstanding the GCAC, there is always the possibility that a connection request may fail CAC at some intermediate node. This becomes even more likely in large networks with many levels of hierarchy, since QoS information cannot be accurately aggregated in such cases. To allow for such cases, without excessive connection failures and retries, the P-NNI protocol also supports the notion of crankback.

Crankback is where a connection which is blocked along a selected path is rolled back to an intermediate node, earlier in the path. This intermediate node attempts to discover another path to the final destination, using the same procedure as the original node, but uses newer, or hopefully more accurate network state. This is another mechanism that can be much more easily supported in a source-routed protocol than in a hop-by-hop protocol.

Figure 10. Operation of Crankback



One of the concerns with P-NNI route generation is that most commonly used routing algorithms (such as Dijkstra calculations) were designed for single, cumulative metrics such as link weightings or counts. Since P-NNI uses a number of complex link parameters for link pruning, path selection may often not generate any acceptable paths. In such cases, sophisticated algorithms may use a technique known as fallback, where particular attributes (such as delay) are selectively relaxed, and paths are recalculated in order to find a path that meets some minimal set of desired attributes. In general, path selection, like CAC, is an area with considerable scope for vendor differentiation.

P-NNI Phase 1 Adoption Status

While the P-NNI Phase 1 protocol is extremely powerful, it is also quite complex. For this reason, the ATM Forum's work on the protocol is unlikely to be completed until the second half of 1995. Actual interoperable implementations are unlikely to be widely deployed until well into 1996. For instance, as of the time of writing, many vendors currently had yet to fully roll out implementations of UNI 3.0 signaling, despite the fact that this standard was completed in September 1993. Clearly, the P-NNI Phase 1 protocol is much more complex than UNI 3.0.

Unfortunately, without a P-NNI protocol, there is no standard way for users to build interoperable multivendor ATM networks. Many users are not willing to wait until 1996 for such interoperability since they have pressing needs to test multiple vendors' switches within the ATM test beds that they are currently running. To solve this short-term protocol, Cisco Systems proposed to the ATM Forum that it develop a very simple, UNI-based signaling protocol for switch interoperability.

Originally designated the P-NNI Phase 0 protocol, this was later renamed the Interim Inter-Switch Signaling Protocol (IISP) to avoid confusion with the P-NNI Phase 1 protocol. This protocol was recently completed and approved by the ATM Forum [Forum6]. The IISP, as the name suggests, is essentially a signaling protocol for inter-switch communication. Given the fact that the UNI 3.0/3.1 signaling procedures are essentially symmetrical, it uses UNI signaling for switch-to-switch communication, with nodes arbitrarily taking the role of the network and user side across particular switch-to-switch links (known as IISP links).

Signaling requests are routed between switches using configured address prefix tables within each switch, which precludes the need for a VC routing protocol. These tables are configured with the address prefixes that are reachable through each port on the switch. When a signaling request is received by a switch, either across a UNI or an IISP link, the switch checks the destination ATM address against the prefix table and notes the port with the longest prefix match. It then forwards the signaling request across that port using UNI procedures.

The IISP protocol is very simple and does not require modification to UNI 3.0/3.1 signaling or any new VC routing protocol. It can leverage current development efforts on UNI signaling and hence can be deployed very quickly. The IISP, however, does not have anywhere near the same scalability as the Phase 1 protocol. For instance, manually configuring prefix tables limits its applicability to networks with only a small number of nodes. This is adequate for now, given that most ATM switches today are deployed in small test beds and not in large scale production networks.

IISP implementations will not be interoperable with P-NNI Phase 1 implementations because IISP only uses UNI and not NNI signaling. Users will need to upgrade their switches when P-NNI Phase 1 becomes available. This was deliberately done to simplify the specification and accelerate the deployment of IISP, and to emphasize its interim nature.

The IISP also does not support QoS-based routing, although nodes may implement CAC; it does not support crankback, though nodes can be configured with redundant or alternate paths (the selection of such paths being a local matter). These limitations of the IISP, however, are not as restrictive as might first be imagined. While the Phase 1 protocol has extensive support for QoS routing, this is required only for routing VBR and CBR connections, where end systems can request a specific QoS. End systems that request either Unspecified Bit Rate (UBR) or Available Bit Rate (ABR) connections, however, can specify only very limited QoS capabilities. As such, the P-NNI protocol metrics do not apply to such connections and must be routed using some other criteria—such as shortest path.

More on QoS in ATM Networks

Currently ATM can be implemented using either native mode ATM protocols or by using LAN emulation (LANE). LANE works by defining a service interface for higher layer (Layer 3) protocols, which is identical to that of existing LANs, and that data sent across the ATM network are encapsulated in the appropriate LAN MAC packet format. It does not mean that any attempt is made to emulate the actual media access control protocol of the specific LAN concerned (that is, CSMA/CD for Ethernet or token passing for the IEEE 802.5 LAN protocol).

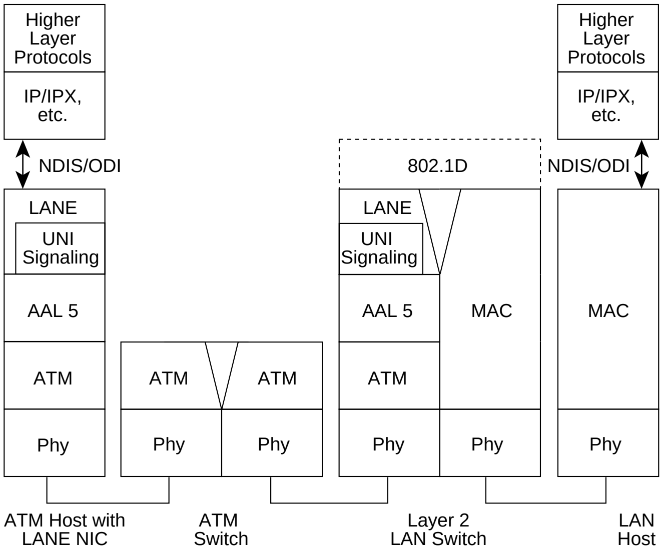
The current LANE protocol does not define a separate encapsulation for FDDI. FDDI packet must be mapped into either Ethernet or Token Ring emulated LANs, using existing translational bridging techniques. The two most prominent new LAN standards under consideration, Fast Ethernet (100BaseT) and 802.12 (100VG-AnyLAN) can both be mapped unchanged into either the Ethernet or Token Ring LANE formats and procedures, as appropriate, since they use the same packet formats.

The LANE protocol supports a range of maximum packet (MPDU) sizes, corresponding to maximum size Ethernet, and 4 Mbps and 16 Mbps Token Ring packets, and to the value of the default MPDU for IP over ATM. Typically the appropriate MPDU will be used depending upon what type of LAN is being emulated—and is supported on the LAN switches bridged to the ELAN (Emulated LAN). An ELAN with only native ATM hosts, however, may optionally use any of the available MPDU sizes, even if this does not correspond to the actual MPDU in a real LAN of the type being emulated. All LECs (LAN Emulation clients) within a given ELAN must use the same MPDU size.

Put simply, the LANE protocols make an ATM network look and behave like an Ethernet or Token Ring LAN—albeit one operating much faster than a real such network.

The rationale for LANE is that it requires no modifications to higher layer protocols to enable their operation over an ATM network. Since the LANE service 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 49). The intention is to accelerate the deployment of ATM, since considerable work remains to be done in fully defining native mode operation for the plethora of existing network layer protocols.

Figure 11. LANE Protocol Architecture



LANE Components and LANE Design Components

Below are the principal components of LAN emulation.

The LAN emulation components include the following:

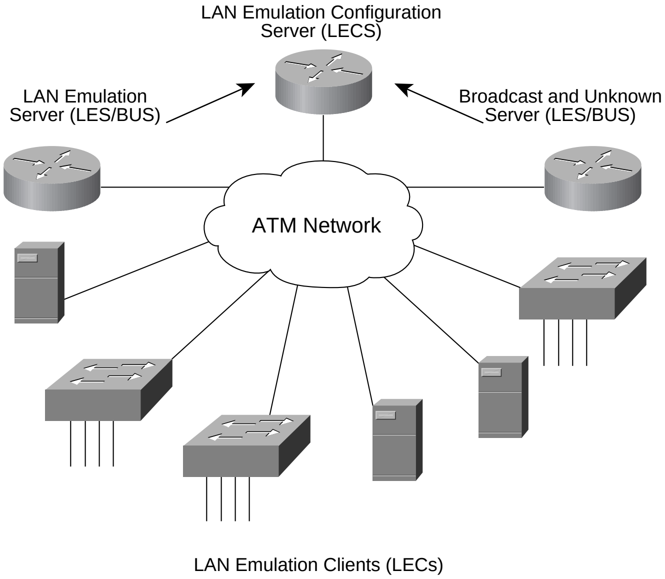
- *LAN Emulation Client*—End systems such as Network Interface Card (NIC)-connected workstations, Catalyst 5000s, or Cisco 7x00s that support LAN emulation require the implementation of a LAN emulation client (LEC). The LEC emulates an interface to a legacy LAN to the higher-level protocols. It performs data forwarding, address resolution, and registration of MAC addresses with the LAN emulation server (LES) and communicates with other LECs via ATM Virtual Channel Connections (VCCs).
- *LAN Emulation Server*—The LES provides a central control point for all LECs. LECs maintain a Control Direct VCC to the LES to forward registration and control information. The LES maintains a point-to-multipoint VCC, known as the Control Distribute VCC, to all LECs, and only control information is forwarded on this VCC. As new LECs join the ATM emulated LAN (ELAN), they are added as a leaf to the Control Distribute tree.
- *Broadcast and Unknown Server*—The broadcast and unknown server (BUS) acts as a central point to distribute broadcasts and multicasts. ATM is essentially a point-to-point technology without “any-to-any” or “broadcast” support. LAN emulation solved this problem by centralizing the broadcast support in the BUS. Each LEC must set up a Multicast Send VCC to the BUS. The BUS will then add the LEC as a leaf to its point-to-multipoint VCC, the Multicast Forward VCC.

The BUS also acts as a multicast server. LAN emulation is defined on ATM Adaptation Layer 5 (AAL5), which specifies a simple trailer to be appended to a frame before it is broken into ATM cells. The problem is that there is no way to differentiate between ATM cells from different senders when multiplexed on a virtual channel. It is assumed that cells received will be in sequence, and when the End of Message (EOM) cell arrives, you should just have to reassemble all of the cells that have already arrived.

The BUS must take the sequence of cells on each Multicast Send VCC and reassemble them into frames. When a full frame is received, it can be queued to send to all of the LECs on the Multicast Forward VCC. This way all the cells from a particular data frame can be guaranteed to be sent in order and not interleaved with cells from any other data frames on the point-to-multipoint VCC.

- *LAN Emulation Configuration Server*—This server maintains a database of LECs and the ELANs that they belong to. It accepts queries from LECs and responds with the appropriate VLAN identifier, namely the ATM address of the LES that serves the appropriate VLAN/ELAN. This database is maintained by the network administrator.

Figure 12. LAN Emulation


LAN Emulation Configuration Server (LECS)
LAN Emulation Server (LES/BUS)
Broadcast and Unknown Server (LES/BUS)
ATM Network
LAN Emulation Clients (LECs)

For a more in-depth analysis of LANE Design, refer to Harbrinder Kang's *Switched LAN Design Guide.*

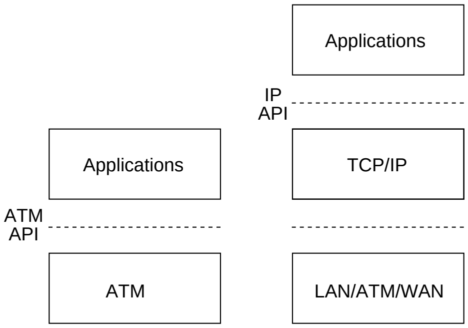
The other option for running ATM is to use native mode protocols, bypassing LANE's MAC address encapsulation. In native mode, address resolution mechanisms are used to map network layer addresses directly into ATM addresses, and the network layer packets are then carried across the ATM network. Currently the only protocol for which extensive work has been done in this area is IP (RFC 1577). Novell has publicly discussed a protocol known as Connection Oriented IPX (CO-IPX) which will adapt IPX specifically for ATM network, and will add QoS support, but full development of this protocol is not expected for some while.

From the perspective of running network multimedia applications, one of the most compelling reasons for running native mode protocols is QoS support. LANE deliberately hides ATM so any network layer protocol that operates over ATM cannot gain access to the QoS properties of ATM and must, therefore, use UBR or ABR connections only. At the moment, this is not a major restriction because all current network protocols were developed for use over existing LAN and WAN technologies, none of which can deliver a guaranteed QoS. Consequently, no existing network layer protocol can request a specific QoS from the network, or deliver such to a higher layer protocol or application. Hence, in turn, most network applications today do not expect to receive, and do not request, any guaranteed QoS from the underlying network protocol.

IP has long had optional support for Type of Service (TOS) indications within the IP header, which could theoretically be used to provide a rudimentary form of QOS support. In practice, however, almost no end system or intermediate system IP implementations have any support for TOS since they cannot be mapped into any common underlying networking technology. Few, if any, IP routing protocols use the TOS bits, for instance, and no applications set them.

At best, therefore, all current network layer protocols today 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 ATM's connection-oriented nature to offer the same type of connectionless service that is expected by network layer protocols, so ABR hides the guaranteed QoS features of ATM to offer the best effort service expected by these protocols. As such, ABR and LANE perfectly complement each other.

Figure 13. Native and Conventional Applications



The figure above illustrates the architectural difference between running native ATM applications versus Conventional applications. In the future, however, this distinction is unlikely to prevail, especially as ATM networks proliferate, it is likely that demand will grow to utilize their QoS benefits, since this is one of ATM's major selling points. This, in turn, will trigger application development expressly designed to take advantage of ATM and ATM QoS.

IP, ATM, and QoS

In the specific case of IP, the IETF has developed the notion of an Integrated Services Internet. This 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 the Resource Reservation Protocol (RSVP) are being defined to allow for resource reservation across an IP network, much as ATM signaling allows this within ATM networks.

As discussed earlier, RSVP is a control protocol, much like ICMP, that will be used by applications within IP end-systems to indicate to nodes transmitting to them the nature (such as bandwidth, jitter, maximum burstiness, etc.) of the packet streams that they wish to receive. Intermediate systems, along the path from the source to the destination IP end-systems, will also interpret RSVP control packets in order to perform admission control (analogous to ATM CAC) and allocate the resources required to support the requested traffic flows. Such systems will maintain "soft-state" about such traffic flows, much as ATM switches maintain connection state, and will perform packet level traffic shaping, scheduling, and so on, in the same manner that ATM switches groom cell streams so as to provide the guaranteed QoS. RSVP can hence be thought of as providing very much the same traffic contract specification functions with respect to packet level traffic flows that ATM UNI and NNI signaling play with respect to cell flows.

One significant difference between RSVP and ATM signaling is that RSVP uses a receiver oriented model, where the receiving node indicates to the network and the transmitting node the nature of the traffic flow that the node is willing and able to receive, whereas in ATM, the transmitting node indicates to the receiving nodes and network the nature of the cell streams that it desires to transmit. The former model is more application oriented, while the latter is more network oriented. Methods of reconciling these two differing paradigms are currently under study.

RSVP is fundamentally built upon a multicast paradigm, and routes traffic flows along source routed point-to-multipoint paths (with unicast handled as a special case of multicast). New multicast protocols like Protocol Independent Multicast (PIM), and their associated unicast packet routing protocols, will hence be closely coupled with RSVP, much as VC routing protocols are closely coupled with UNI and NNI signaling.

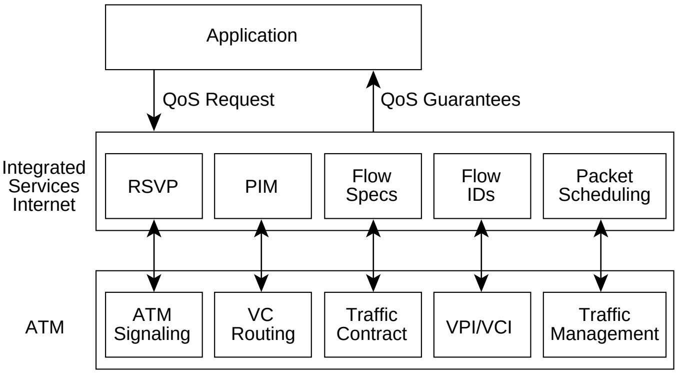
Such protocols rely upon a flow specification to characterize the expected traffic patterns for a stream of IP packets between two applications, which the network can process through packet-level policing, shaping, and scheduling mechanisms to deliver a requested QoS. The flow can be thought of as a layer 3 connection, since it identifies and characterizes a stream of packets between two or more nodes, even though the protocol remains ostensibly connectionless.

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, which the network can use to identify flows, much as VPI/VCI (virtual path identifier/virtual channel identifier) are used to identify streams of ATM cells. Protocols like RSVP will be used to associate with each flow a flowspec that characterizes the traffic parameters of the flow, much as the ATM traffic contract is associated with an ATM connection.

It is certain that IPv6 will incorporate full support for integrated services through the use of such mechanisms and the definition of protocols like RSVP. Such support might also be extended to the current IPv4 protocol. It is likely that IPv6, and other protocol components of the Integrated Service Internet, will be fully standardized by the end of 1995, and components may be deployed perhaps even earlier.

Figure 52 below shows the mapping between RSVP and more generally Integrated Services Internet into ATM.

Figure 14. Mapping of the Integrated Services Internet into ATM



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 realtime data, such as audio, video or simulation data, over multicast or unicast network services, and builds upon protocols like RSVP for resource reservation, and upon transport technologies like ATM for QoS guarantees. The services provided by RTP to real time applications include payload type identification, sequence numbering, timestamping and delivery monitoring. Closely tied to the RTP protocol functions is the RTP control protocol (RTCP), to monitor the quality of service and to convey information about the participants in an on-going session. Hence RTP can be used for such applications as multipoint conferencing, building upon the other protocol services of the Integrated Service Internet.

When such protocols are widely deployed and applications are developed to use them, there will certainly be a demand to run such protocols in native mode over ATM. It would be pointless to obtain QoS support from the network layer, only to have LANE preclude that support from being mapped to their equivalents in the ATM network. There is clearly a very clear and natural mapping between the concepts and mechanisms of the Integrated Services Internet and ATM (flow IDs and flowspecs to ATM connections and traffic contracts, respectively).

Hence the Integrated Services Internet can be thought of as eventually providing the packet level control infrastructure for the physical network infrastructure of ATM, where the former provides application services and the latter realizes the requested QoS guarantees. In this way, the true value of ATM can be exploited, while preserving a network independent service infrastructure for application portability. In order to realize the vision, however, there must be native mode protocol support over ATM.

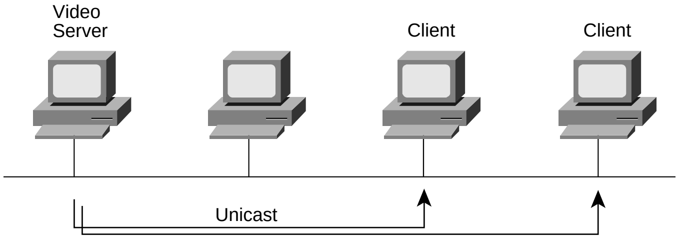
VI. Multicasting

Traditional network computing applications, including some of today's network multimedia applications, involve communication only between two computers. A two-user video conferencing session using Intel ProShare, for example, is a strictly unicast (see definition below) transaction. However, a new breed of network multimedia applications like LAN TV, desktop conferencing, corporate broadcasts, and collaborative computing environments require 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, does the application rely on unicast, broadcast or multicast transmission facilities.

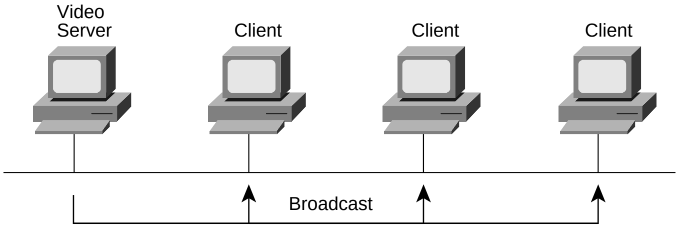
- Unicast (Figure 53). With a unicast design, applications send one copy of each packet to each client's Unicast address. Unicast transmission has significant scaling restrictions, especially if the group is large—because the same information has to be carried multiple times—even on shared links.

Figure 15.



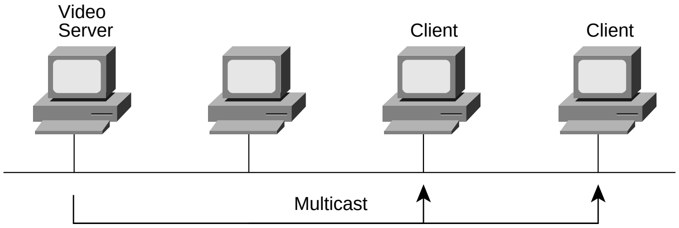
- Broadcast (Figure 54). In a broadcast design, applications send one copy of each packet and address it to a broadcast address that all devices listen to. This technique is even simpler than unicast for the application to implement. However, if this technique is used, the network must either stop broadcasts at the LAN boundary (a technique that is frequently used to prevent broadcast storms) or send the broadcast everywhere. Sending the broadcast everywhere can be inefficient if only a small group actually needs to see the packets.

Figure 16.



- Multicast (Figure 55). With a multicast design, applications can send one copy of each packet and address it to a group address. It rests on the client whether or not to listen to the multicast address. Multicasting is helpful in controlling network traffic while also curbing network and host processing by eliminating traffic redundancy.

Figure 17.



Many new network multimedia applications, like Insoft INTV! 3.0 and Apple QuickTime Conferencing 1.0, are implementing multicast transmission facilities because of the added efficiency that multicasting offers for both the network and 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 will require approximately 1.5Mbps per client viewer. In an unicast environment, the video server will send 1.5 x *n* (where *n*=number of client viewers) Mbps of traffic to the network. With a 10 Mbps pipe from the server, roughly 6–7 streams could be supported before the network runs out of bandwidth. In a multicast environment, the video server need only send 1 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 is implemented at both the data-link layer (layer 2) and the network layer (layer 3). Ethernet and FDDI, for example, support unicast, multicast, and broadcast addresses. An individual computer can listen 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 a data-link layer 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 a campus environment consisting of different media types, such as Ethernet, Token Ring, FDDI, ATM, Frame Relay, SMDS, and other networking technologies, it is best to implement multicast at the network layer.

There are several parameters that the network layer must define in order to support multicast communications:

- *Addressing.* There must be a network-layer 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 data-link layer 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 ability, 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 these packet distribution trees is to ensure that each packet exists only one time on any given network (that is, if there are multiple receivers on a given branch, there should only be one copy of the packets on that branch).

IP Multicast

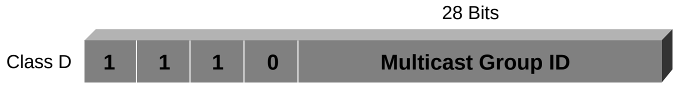
The Internet Engineering Task Force has been developing standards that address each of the issues described above.

- Addressing. The IP address space is divided into four pieces: Class A, Class B, Class C, and Class D. Classes A, B, and C are used for unicast traffic. Class D is reserved for multicast traffic. Class D addresses 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 available for routing IP Multicast traffic:
  - — RFC 1075 defines the Distance Vector Multicast Routing Protocol (DVMRP).
  - — RFC 1584 defines the Multicast Open Shortest Path First (MOSPF) protocol—an extension to OSPF that allows it to support IP Multicast.
  - — Two Internet standards-track drafts describe PIM—a multicast protocol that can be used in conjunction with all unicast IP routing protocols. These documents are entitled Protocol-Independent Multicast (PIM): Motivation and Architecture and Protocol-Independent Multicast (PIM): Protocol Specification.

Multicast Group Addressing

Figure 56 below shows the format of a class D IP multicast address.

Figure 18. Class D Address Format



Unlike Class A, B and C IP address, the last 28 bits of a Class D address have no further 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 octect are 0, this yields the 224 address.

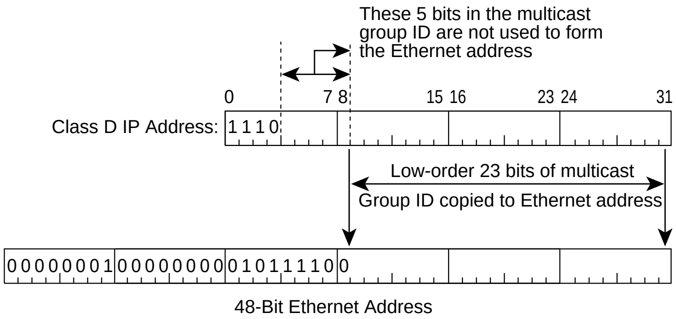
The set of hosts listening to a particular IP multicast address is called a host group. A host group can span multiple networks. Membership in a host is dynamic-hosts may join and leave host groups. Refer to the section below on IGMP for a detailed discussion if IP Multicast registration.

Some multicast group addresses are assigned as well-known addresses by the IANA (Internet Assigned Numbers Authority). These are called permanent host groups, similar in concept to the well-known TCP and UDP port numbers. These well-known multicast addresses are listed in the latest Assigned Numbers RFC. Address 224.0.0.1 means “all systems on this subnet,” and 224.0.0.2 means “all routers on this subnet.” The multicast address 224.0.1.1 is for NTP, network time protocol; 224.0.0.9 is for RIP-2 and 224.0.1.2 is for Silicon Graphics’ Dogfight application.

The IANA owns a block of Ethernet address which 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, this means 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 (see Figure 57 below). Since the upper 5 bits of the multicast address are ignored in this mapping, it is not unique. Thirty-two different multicast group IDs map to each Ethernet address.

Figure 19. Multicast Address Mapping


These 5 bits in the multicast group ID are not used to form the Ethernet address
0
7 8
15 16
23 24
31
Class D IP Address:
1 1 1 0
Low-order 23 bits of multicast Group ID copied to Ethernet address
0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0
0 1 0 1 1 1 1 0
0
48-Bit Ethernet Address

Note that since the mapping is not unique, it implies that the device driver or the IP modules must perform filtering, since the interface card may receive multicast frames in which the host is really not interested.

Multicasting on a single physical network is simple. The sending process specifies a destination IP address that is a multicast address, 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 handling by joining a multicast group.

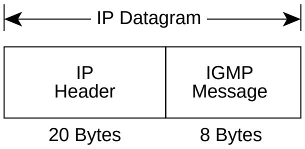
When a multicast datagram is received by a host, it must deliver a copy to all the processes that belong to that group. This is different from UDP where a single process receives an incoming unicast UDP datagram. With multicast it is possible for multiple processes on a given host to belong to the same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

Complications arise when extending multicasting beyond a single physical network and multicast packets pass through routers. A protocol is needed for multicast 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 (IGMP).

IGMP

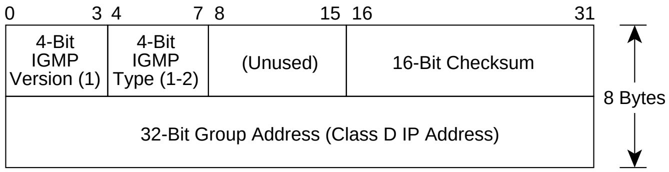
IGMP is considered part of the IP layer and uses IP datagrams to transmit datagrams. Unlike other protocols, IGMP has a fixed-size message within an IP datagram:

Figure 20. IP Datagram with IGMP Messages



IGMP messages are specified in the IP datagram with a protocol value of 2. Within the IP datagram, the IGMP message format looks as follows:

Figure 21. IGMP Message Format



The IGMP version is 1. An IGMP type of 1 is a query sent by a multicast router, and 2 is a response sent by a host. The checksum is calculated in the same manner 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.

Fundamental to multicasting is the concept of a process joining a multicast group on a given interface on a host. Membership in a multicast group on a given interface is dynamic (i.e. 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.

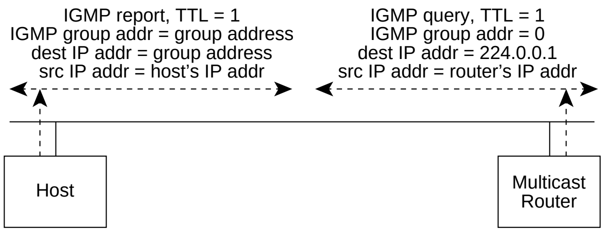
IGMP messages are used by multicast routers to keep track of group membership on each of the routers physically attached networks. The following rules apply:

1. A host sends an IGMP report when the first process joins a group. If multiple processes on a given host join the same group, only one report is sent, the first time a process joins the group. This report is sent out the same interface on which the process joined the group.
2. 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 won’t report the group.
3. A multicast router sends an IGMP query at regular intervals to see if any hosts still have processes belonging to any groups. The router must send one query out each interface. The group address in the query is 0 since the router expects one response from a host for every group that contains one or more members on host.
4. A host responds to an IGMP query by sending one IGMP report for each group that still contains at least one process.

Using these queries and reports, a multicast router keeps a table of which of its interfaces 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 link layer address) only out the interfaces that still have hosts with processes belonging to that group.

Figure 60 diagrams a simple IGMP interaction.

Figure 22. IGMP Reports and Queries


IGMP report, TTL = 1
IGMP group addr = group address
dest IP addr = group address
src IP addr = host's IP addr
IGMP query, TTL = 1
IGMP group addr = 0
dest IP addr = 224.0.0.1
src IP addr = router's IP addr
Host
Multicast Router

As the above diagram depicts, the TTL (Time to Live) field of the reports and queries is set to 1. This refers to the normal TTL field in the IP header. A multicast datagram with an initial TTL of 0 is restricted to the same host. By default, multicast datagrams are sent with a TTL set to 1. This restricts the datagram to the same subnet. Higher TTLs can be forwarded by multicast routers.

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, then 3, and so on. In this way the application locates the closest server, 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.

The last critical issue for delivering multicast traffic in a routed network is the multicast routing protocol. Currently there are three different multicast routing protocols, Distance Vector Multicast Routing Protocol (DVMRP), Multicast OSPF (MOSPF) and Protocol Independent Multicasting (PIM). The goal in each is establish paths in the network so that multicast traffic can effectively reach all group members.

DVMRP (RFC 1075)

DVMRP uses a technique known as Reverse Path Forwarding. When a router receives a packet, it floods the packet out of all paths except the one that leads back to the packet's source. Doing so allows a data stream to reach all LANs (possibly multiple times). If a router is attached to a set of LANs that do not want to receive a particular multicast group, the router can send a "prune" message back up the distribution tree to stop subsequent packets from traveling where there are no members.

DVMRP will periodically reflood in order to reach any new hosts that want to receive a particular group. There is a direct relationship between the time it takes for a new receiver to get the data stream and the frequency of flooding.

DVMRP implements its own unicast routing protocol in order to determine which interface leads back to the source of the data stream. This unicast routing protocol is very like RIP and is based purely on hop counts. As a result, the path that the multicast traffic follows may not be the same as the path that the unicast traffic follows.

DVMRP has significant scaling problems because of the necessity to flood frequently. This limitation is exacerbated by the fact that early implementations of DVMRP did not implement pruning. DVMRP typically uses tunneling to control flooding and in some cases the lack of 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. In the near future, the MBONE will move away from DVMRP opting to use PIM instead because of PIMs greater efficiency. Refer to the section on PIM below for a discussion of the benefits of PIM over DVMRP.

Multicast Extensions to OSPF (RFC 1584)

Multicast OSPF (MOSPF) was defined as an extension to the OSPF unicast routing protocol. 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. 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 on all routers, and trees must be recomputed when a link state change occurs or when the cache times out. This in turn can hinder multicast performance, depending upon the size of the network and the volatility of the multicast groups.

As expected, MOSPF works only in internetworks that are using OSPF.

MOSPF is best suited for environments that have relatively few source/group pairs active at any given time. It will work less well in environments that have many active sources or environments that have unstable links.

PIM (Internet Draft "Protocol-Independent Multicast [PIM]: Protocol Specification")

Unlike MOSPF which is OSPF-dependent, Protocol-Independent Multicast (PIM) works with all existing unicast routing protocols. And unlike DVMRP which has inherent scaling problems, PIM offers two different types of multipoint traffic distribution patterns to address multicast routing scalability: dense mode and sparse mode.

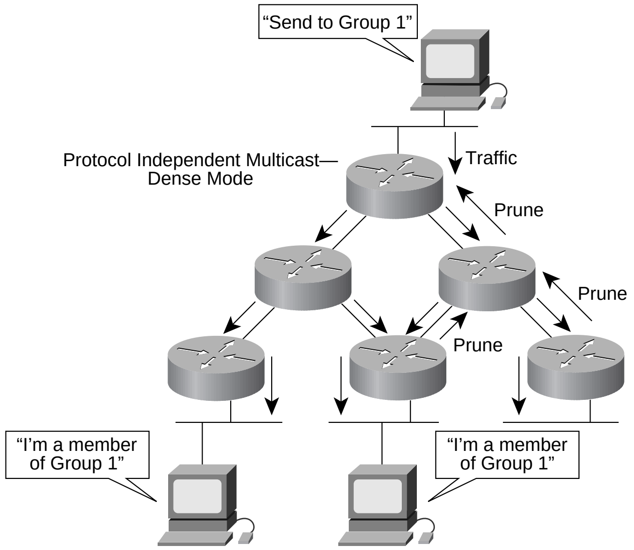
Dense mode is most useful when:

- 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 looks a lot like DVMRP. The most significant difference between DVMRP and dense-mode PIM is that PIM works with whatever unicast protocol is being used; PIM does not require any particular unicast protocol.

In dense mode, PIM will flood the network and prune back based on multicast group member information. In a LAN TV multicast environment, for instance, dense mode would be effective. This is because most likely there will be group member off of all subnets. Flooding the network, thus will be effective because little pruning would have been necessary. Cisco IOS supports PIM dense mode. Refer to the PIM configuration section below for configuring PIM dense mode.

Figure 23. PIM Dense-Mode Operation


"Send to Group 1"
Protocol Independent Multicast—Dense Mode
Traffic
Prune
Prune
Prune
"I'm a member of Group 1"
"I'm a member of Group 1"

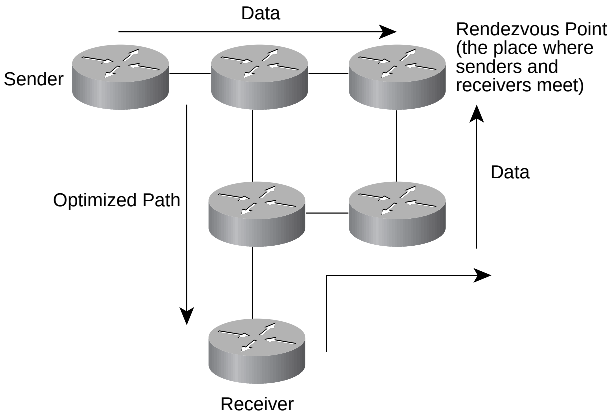
Sparse multicast is most useful when:

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

Sparse-mode PIM is optimized for environments where there are many multipoint data streams and each multicast stream goes to a relatively small number of the LANs in the internetwork. For these types of groups, Reverse Path Forwarding techniques make inefficient use of the network bandwidth. Sparse-mode PIM works by defining a Rendezvous Point. When a sender wants to send data, it first sends to the Rendezvous Point. When a receiver wants to receive data, it registers with the Rendezvous Point. Once the data stream begins to flow from sender to Rendezvous Point to receiver, the routers in the path will optimize the path automatically to remove any unnecessary hops. Sparse-mode PIM assumes that no hosts want the multicast traffic unless they specifically ask for it.

Cisco IOS supports PIM sparse mode. Refer to the PIM configuration section below for configuring PIM dense mode.

Figure 24. PIM Sparse-Mode Operation


Data
Rendezvous Point (the place where senders and receivers meet)
Sender
Data
Optimized Path
Receiver

Configuring PIM on Cisco Routers

The first step in configuring PIM on a Cisco router is to enable IP multicast routing. This is done using the following global configuration command:

ip multicast routing

Enabling PIM itself is done in interface configuration mode. Enabling PIM on an interface also enables IGMP. An interface can be configured to be in dense mode or sparse mode.

To enable dense-mode PIM on an interface use the following command:

ip pim dense-mode

To enable sparse-mode PIM on an interface use the following command:

ip pim sparse-mode

If the router is configured for sparse mode, one or more routers must be designated as a Rendezvous Point (RP). A router does not need to be configured as an RP, it will do this on its own. 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. A router can be configured so that packets for a single multicast group can use one or more RPs.

The IP address of the RP must be configured on leaf routers. Leaf routers are those routers that are directly connected either to a multicast group member or to a sender of multicast messages.

The RP address is used by first-hop routers to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop routers to send PIM join/prune messages to the RP to inform it about group membership. The RP does not need to know it is an RP.

To configure the address or an RP, perform the following global configuration command:

ip pim rp-address <ip-address> <access-list-number>

Other Multicast-related Cisco IOS Commands

The following section address multicast in general with Cisco IOS-based products.

Configure a Router to be a Member of a Group

Cisco routers can be configured to be members of multicast groups. This is useful for determining multicast reachability in a network. If a router is configured to be a group member and supports the protocol that is being transmitted to the group, it can respond. An example is ping. A router will respond to ICMP echo request packets addressed to a group for which it is a member.

To configure a router to join a multicast router group, enter the following interface configuration command:

ip igmp join-group <group-address>

Configure the Host-Query Message interval

Multicast routers send IGMP host-query messages to discover which multicast groups are present on attached networks. These messages are sent to the all-systems group address of 224.0.0.1 with a TTL of 1.

Multicast routers send host-query messages periodically to refresh their knowledge of memberships present on their network. If, after some number queries, the router discovers that no locals hosts are members of a multicast group, the router stops forwarding onto the local network multicast packets from remote origins for that group and sends a prune message upstream toward the source.

Multicast routers elect a PIM designated router for the LAN (subnet). This is the router with the highest IP address. The designated router is responsible for sending IGMP host-query messages to all hosts on the LAN. In sparse mode, the designated router also sends PIM register and PIM join messages towards the RP.

By default, the designated router sends IGMP host-query messages once a minute in order to keep the IGMP overhead on hosts and networks low.

To modify this interval, enter the following interface configuration command:

ip igmp query-interval <seconds>

Controlling Access to IP Multicast Groups

Multicast routers send IGMP host-query messages to determine which multicast groups have members off the routers attached local networks. The routers then forward to these group members all packets addressed to the multicast group. A filter can be placed on each interface to restrict the multicast groups that hosts on the subnet serviced by the interface can join.

To filter multicast groups, perform the following interface configuration command:

ip igmp access-group access-list-number

Modify PIM Message Timers

By default multicast routers send PIM router-query messages every 30 seconds. To modify this interval, perform the following interface configuration command:

ip pim query-interval <seconds>

Configure the TTL Threshold

The TTL value controls whether packets are forwarded out an interface. The TTL value is specified in hops. Any multicast packet with a TTL less that the interface TTL threshold is not forwarded on the interface. The default value is 0, which means that all multicast packets are forwarded on the interface.

To change the default threshold value, perform the following task in interface configuration mode.

ip multicast-threshold <ttl>

Configuring DVMRP Interoperability

Cisco multicast routers using PIM can interoperate with non-Cisco multicast routers using DVMRP. DVMRP interoperability is necessary when propagating MBONE traffic to a PIM-based Cisco infrastructure.

PIM routers dynamically discover DVMRP multicast routers on attached networks. Once a DVMRP neighbor has been discovered, the router periodically transmits DVMRP report messages advertising the unicast sources reachable in the PIM domain. By default, directly connected subnets and networks are not advertised. The router forwards multicast packets that have been forwarded by DVMRP routers and in turn forwards multicast packets to DVMRP routers.

Using the **ip dvmrp metric** command, the sources advertised and the metrics used can be configured. Additionally, all sources learned via a particular unicast routing process can be configured to be advertised into DVMRP.

It is necessary to use mrouted (pronounced M-ROUTE-D) version 2.2 (which implements a nonpruning version of DVMRP) or version 3.2 (which implements a pruning version of DVMRP) when Cisco routers are directly connected to DVMRP routers or interoperate with DVMRP routers over an MBONE tunnel. DVMRP advertisements produced by Cisco routers can caused older versions of mrouted to corrupt their routing tables and those of their neighbors.

To configure the sources that are advertised and the metrics that are used when transmitting DVMRP report messages, perform the following interface configuration command:

ip dvmrp metric <metric> <access-list-number> <protocol process id>

Advertise Network 0.0.0.0 to DVMRP Neighbors

The mrouted protocol is a public domain implementation of DVMRP. If the router is a neighbor to an mrouted version 3.4 machine, the router can be configured to advertise network 0.0.0.0 to the DVMRP neighbor. It must be specified whether only route 0.0.0.0 is advertised or if other routes can also be specified.

To advertise network 0.0.0.0 to DVMRP neighbors on an interface, perform the following task in interface configuration mode:

ip dvmrp default-information <originate|only>

Configure a DVMRP Tunnel

Cisco routers can support DVMRP tunnels to the MBONE. A DVMRP tunnel can be configured on a router if the other end is running DVMRP. The router then sends and receives multicast packets over the tunnel. This allows an IP domain to connect to the DVMRP router in the case where all routers on the path do not support multicast routing.

When a Cisco router runs DVMRP over a tunnel, it advertises sources in DVMRP report messages much as it does on real networks. In addition, DVMRP report messages received are cached on the router and are used as part of its Reverse Path Forwarding (RPF) calculation. This allows a multicast packets received over the tunnel to be forwarded by the router.

When a DVMRP tunnel is configured, an address for tunnel should be assigned for two reasons:

- To enable the sending of IP packets over the tunnel
- To indicate whether the Cisco IOS software should perform DVMRP summarization

An IP address can be assigned using the **ip address** interface configuration command. Alternatively, the **ip unnumbered** interface configuration command can be used to configure the tunnel to be unnumbered. Either of these two methods allows IP multicast packets to flow over the tunnel. The Cisco IOS software will not advertise subnets over the tunnel if the tunnel has a different network number from the subnet. In this case, the software advertises only the network number over the tunnel.

To configure a DVMRP tunnel perform the following tasks:

interface tunnel <number>
Specify a tunnel interface in global configuration mode. This puts the router into interface configuration mode.

tunnel source <ip address>
Set the tunnel interface’s source address. This is the IP address of the interface on the router.

t**unnel destination** <ip address>
Set the tunnel interface’s destination address. This is the IP address of the mrouted multicast router.

tunnel mode dvmrp
Configure a dvmrp tunnel

ip address <address mask>
Assign an IP address to the interface.

OR

ip unnumbered
Configure the interface as unnumbered

ip pim <dense-mode| sparse-mode>
Configure PIM on the interface

ip dvmrp accept-filter <access-list-number> <administrative-distance>
Configure an acceptance filter for incoming DVMRP reports

Monitoring and Maintaining IP Multicast Routing

To monitor IP multicast routing information and to clear IP multicast routing caches, perform one or more of the following task at the EXEC prompt.

Table 1.

clear ip igmp group <group-name | group-address | address>	Delete entries from the IGMP cache
clear ip mroute * | <group-name [source-address] | routing group-address [source-address]>	Delete entries from the IP multicast table
mbranch <group-address branch-address [ttl]>	Trace a branch of a multicast tree for a specific group

Table 1.

clear ip igmp group <group-name | group-address | address>	Delete entries from the IGMP cache
mrbranch <group-address branch-address [ttl]>	Trace a branch of a multicast tree for a group in the reverse direction
show ip dvmrp route [ip address]	Display the entries in the DVMRP routing table
show ip igmp groups [group-name | group-address | interface]	Display the multicast groups that are directly connected to the router and that were learned via IGMP
show ip igmp interface [type number]	Display multicast-related information about an interface

Table 1.

clear ip igmp group <group-name | group-address | address>	Delete entries from the IGMP cache
show ip mroute [group-name | group-address] routing **[summary] [count]** **show ip mroute** [group-name | [source-address] | group-address [source-address]]	Display the contents of the Ip multicast table

IP Multicasting Configuration Examples

Configure a router to operate in dense mode:

```
!
ip multicast-routing
!
interface ethernet 0
ip pim dense-mode
!
```

Configure a router to operate in sparse mode:

```
!
access-list 1 permit 224.2.0.1
!
ip multicast-routing
ip pim rp-address 10.8.0.20 1
interface ethernet 1
ip pim sparse-mode
!
```

Configure DVMRP Interoperability Examples:

The following example configures DVMRP interoperability for configurations when the PIM router and the DVMRP router are on the same network segment. In this example, access list 1 advertises the networks 198.92.35.0, 198.92.36.0, 198.92.37.0, 131.108.0.0 and 150.136.0.0 to the DVMRP router, and access list 2 is used to prevent all other networks from being advertised (**ip dvmrp** metric 0).

```
!
interface ethernet 0
ip address 131.119.244.244 255.255.255.0
ip pim dense-mode
ip dvmrp metric 1 1
ip dvmrp metric 0 2
!
access-list 1 permit 198.92.35.0 0.0.0.255
access-list 1 permit 198.92.36.0 0.0.0.255
access-list 1 permit 198.92.37.0 0.0.0.255
access-list 1 permit 131.108.0.0 0.0.0.255
access-list 1 permit 150.136.0.0 0.0.255.255
access-list 1 deny 0.0.0.0 255.255.255.255
access-list 2 permit 0.0.0.0 255.255.255.255
!
```

The following example configures DVMRP interoperability over a tunnel interface:

```
hostname staggerlee
!
ip multicast-routing
!
interface tunnel 0
no ip address
ip pim dense-mode
tunnel source Ethernet0
tunnel destination 192.70.92.133
tunnel mode dvmrp
!
interface ethernet 0
ip address 193.171.23.23 255.255.255.240 secondary
ip address 192.76.243.2 255.255.255.0
ip pim dense-mode
!
router igrp 11853
network 192.76.243.0
network 193.171.23.0
!
```

For more information about Multicast Routing Configuration, refer to CCO and Cisco Connection Documentation CD-ROM.

Apple Multicast: Simple Multicast Routing Protocol (SMRP)

While a considerable amount of work has been done to promote IP multicasting, a standard for multicast AppleTalk traffic has also been adopted—Simple Multicast Routing Protocol, or SMRP.

SMRP is a transport layer multicast protocol and is first offered with the 11.0 release of Cisco IOS. The SMRP specification dictates either IPX or AppleTalk for the underlying network layer protocol although the initial Cisco IOS release will only support AppleTalk.

On each local network segment, a router is elected as the primary node which will handle requests by local devices to create multicast groups on that segment. A device that wishes to multicast data requests assignment of a group address from the primary node by sending a Create Group Request packet. The primary node assigns an unused group address and returns the address to the requesting device using a Create Group Response packet.

Devices that want to receive multicast data from this group ask their local router to join the group using the Join Request packets. The local router then forwards the Join Request to the router that created the group. The creator router in turn responds with a Join Response.

Multicast data sent by the source is forwarded by the router downstream interfaces toward receivers. Receivers can join and leave a group at any time. A sender may delete the group at any time. And the routers will 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 polling the network for the presence of senders and receivers using Creator Query and Member Query packets. A router that detects the disappearance of a sender will delete the group A router that senses the disappearance of a receiver, if no other receivers exist on the segment, will inform its upstream neighbor to stop forwarding multicast data. Finally, each router periodically informs its neighbors of its presence by sending Hello packets.

To configure SMRP input the following commands:

smrp routing
Global configuration that enables SMRP routing on the router

smrp protocol appletalk
Interface configuration that specifies appletalk as the layer 3 protocol

SMRP configuration example:

```
!
smrp routing
!
interface ethernet 0
appletalk cable-range 10-10
appletalk zone CaseyJones
smrp protocol appletalk
!
```

SMRP Troubleshooting

The following commands are useful when troubleshooting SMRP.

Table 2.

show smrp route	Shows the SMRP routing table
show smrp group	Shows SMRP multicast groups
show smrp forward	Shows smrp forwarding tables
show smrp transactions	Shows transaction requests and responses
show smrp neighbors	Shows neighboring SMRP routers
debug smrp route	Debugs routing table changes
debug smrp group	Debugs the creation and deletion of groups
debug smrp forward	Debugs the creation and deletion of forwards

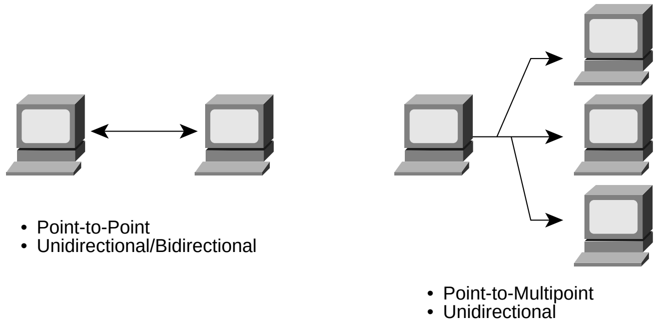
ATM Multicasting

Much of the discussion of ATM multicasting comes from Anthony Alles' *ATM Internetworking* paper.

There are two fundamental types of ATM connections:

- Point-to-point connections, which connect two ATM end-systems. Such connections can be unidirectional or bi-directional.
- Point-to-multipoint connections, which connects 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 points where the connection splits into two or more branches. Such connections are unidirectional, permitting the root to transmit to the leaves, but not the leaves to transmit to the root, or to each other, on the same connection. The reason why such connections are only unidirectional is described below.

Figure 25. Types of ATM Connections



What is notably missing from these types of ATM connections is an analog to the multicasting or broadcasting capability common in many shared medium LAN technologies such as Ethernet or Token Ring. In such technologies, multicasting allows multiple end systems to both 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 (bi-directional) multipoint-to-multipoint connection. Unfortunately, this obvious solution cannot be implemented when using AAL5, the most common ATM Adaptation Layer (AAL) used to transmit data across ATM networks.

Unlike AAL 3/4, with its Message Identifier (MID) field, AAL 5 does not have any provision within its cell format for the interleaving of cells from different AAL5 packets on a single connection. This means that 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.

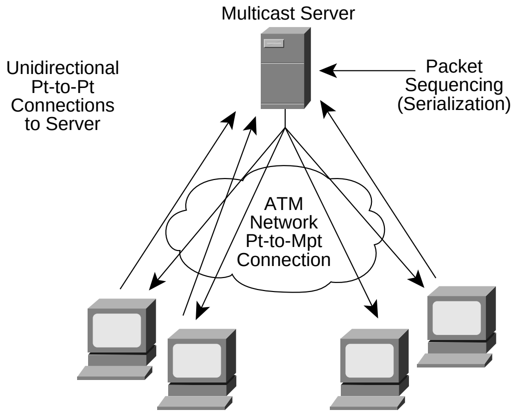
Despite the problems that AAL 5 has with multicast support, it is not really feasible to use AAL 3/4 for data transport instead. This is because AAL 3/4 is a much more complex protocol than AAL 5 and would lead to much more complex and expensive implementations; indeed, AAL 5 was developed specifically to replace AAL 3/4. In any case, while the MID field of AAL 3/4 could preclude cell interleaving problems, allowing for bi-directional, 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 space.

This is why ATM AAL 5 point-to-multipoint connections can only be unidirectional, for if a leaf node was to transmit an AAL 5 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 well 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 does require some form of multicast capability, since most existing protocols, being developed initially for LAN technologies, rely upon the existence of a low-level multicast/broadcast facility. Three methods have been proposed for solving this problem:

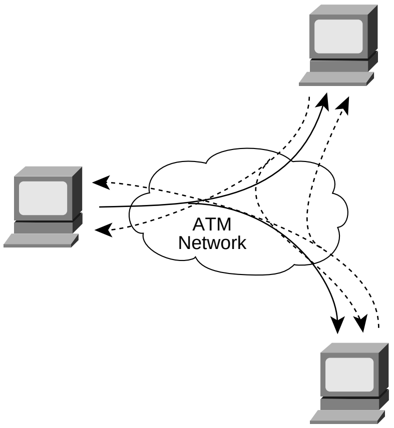
- VP-Multicasting: In this mechanism, a multipoint-to-multipoint VP links all nodes in the multicast group, and each node is given a unique VCI value within the VP. Interleaved packets can hence be identified by the unique VCI value of the source. Unfortunately, this mechanism requires a protocol to uniquely allocate VCI values to nodes; such a mechanism 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: In this mechanism, all nodes wishing 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 wishing to receive the multicast packets through a point-to-multipoint connection. The multicast server receives packets across the point-to-point connections, 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 26. Multicast Server Operation


Multicast Server
Unidirectional Pt-to-Pt Connections to Server
Packet Sequencing (Serialization)
ATM Network Pt-to-Mpt Connection

- Overlaid Point-to-Multipoint Connections: In this mechanism, all nodes in the multicast group establish a point-to-multipoint connection with each other node in the group, and, in turn, becomes a leaf in the equivalent connections of all other nodes. Hence, all nodes can both transmit to and receive from all other nodes.

Figure 27. Multicast Through Overlaid Point-to-Multipoint Connections



The last mechanism requires each node to maintain *n* connections for each group, where *n* is the total number of transmitting nodes within the group, while the multicast server mechanism requires only two connections. This mechanism also requires a registration process for telling nodes that join a group what the other nodes in the group are, so that it 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, as yet, no ideal solution within ATM for multicast. Higher layer protocols within ATM networks use both the latter two solutions for multicast. This is one example of why internetworking existing protocols with ATM is so complex. Most current protocols, particularly those developed for LANs, implicitly assume a network infrastructure very similar to existing LAN technologies—that is, a shared medium, connectionless technology with implicit broadcast mechanisms. As noted above, ATM violates all of these assumptions.

IP Multicasting with RFC 1577/Classical IP over ATM

Today, there is no specific support in the classical IP protocol for multicast operation. This has long been recognized as a critical weakness of RFC 1577, particularly in comparison to LANE. While RFC 1577 could be used to resolve a multicast IP address to an ATM address, this addresses neither the question of how nodes could register for membership within an IP multicast group, nor how an IP multicast group could be mapped to a form of ATM multicast.

Recently, however, some work has been done to define a mechanism for multicast in RFC 1577. This work attempts to support the IP multicast behavior described in RFC 1112, by a combination of multicast servers and overlaid point-to-multipoint connections. This work is currently at an early stage of definition, so only a brief overview of this work is presented here.

In development is the concept of a Multicast Address Resolution Server (MARS), which can be considered the analog of the ARP server in 1577. 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.

When an end-system wants to transmit to a particular multicast group, it opens a connection to the MARS, and issues a MARS_REQUEST message for that particular group. If any other node has not already registered to join that multicast address (that is, indicated a desire to receive traffic on that group address), the MARS then issues a MARS_NAK, informing the requesting node to "silently" drop the multicast packet. If the MARS has already registered one or more other nodes for that multicast address, however, the operation of the MARS is a function of whether the requested multicast address is configured to be served by a multicast server or by a multicast mesh.

In the multicast server case, the MARS returns a MARS_MULTI message that contains a "server map" of the one or more multicast servers serving the group. The requesting node then sets up a connection (point-to-point or point-to-multipoint, depending upon whether a single or multiple multicast server addresses are returned) to the set of multicast servers and transmits its multicast packets.

Note that in this case a node would receive back its own multicast packets; since many applications cannot tolerate receiving back their own data, devices—particularly routers—would need to filter out any multicast packets received from a multicast server containing its own source IP address. A number of mechanisms for facilitating this operation—including, possibly, changes to the RFC 1483 encapsulations—are under discussion.

In the case where the multicast address is served by a multipoint mesh, the MARS returns a MARS_MULTI message that contains a "host map" of addresses of other nodes already registered as members of that group, indicating a desire to receive traffic on the multicast address. In this case, the requesting node constructs a point-to-multipoint connection to that set of nodes and begins to transmit packets on that connection. In either case, mechanisms are used to ensure that the address list is transmitted to the requesting node in a reliable manner.

The more complex part of the protocol is how the list of nodes that wish membership in the multicast group is collected so as to receive data. In RFC 1112, a node that wishes membership within a multicast group must generate a Internet Group Management Protocol (IGMP) Report message and multicast this to the joining multicast group. The function of this message is to inform all multicast routers on the subnet of the existence of a node that wishes membership in a particular group on that subnet. The routers then use that indication to direct multicast traffic to that subnet, using a multicast routing protocol such as PIM. Note, therefore, that routers must listen "promiscuously" on all multicast groups.

Routers, however, also use a reserved multicast group, identified by the IP address 224.0.0.1, to monitor the status of multicast groups within a subnet. All multicast nodes must also be members of this group. Routers periodically send IGMP Queries for the particular multicast groups which they are currently forwarding to the reserved address. Any node on the subnet that is a member of that multicast group must respond with an IGMP Report message on the queried multicast address, unless some other node responds first. Also, all nodes that wish to participate in multicast operation must join the reserved multicast group in order to receive IGMP Queries.

MARS supports these RFC 1112 requirements by also using the MARS server as a multicast server to support two multicast groups for the reserved multicast group: the ServerControlVC, which links all multicast servers, and ClusterControlVC, which links all end systems (including routers) in the cluster.

Any multicast server that wishes to serve one or more particular multicast groups must first register itself with the MARS to indicate its intentions, using a MARS_MSERV message. The MARS uses such registration messages to construct the server map for each multicast address, which contains the ATM addresses of those servers that wish to serve the particular multicast group, to return it in any subsequent MARS_REQUEST message for the group. The MARS also adds any registering server to its ServerControlVC. Multicast servers obtain the list of nodes that wish to receive data on a particular address by sending a MARS_REQUEST to the MARS, just as with any other end system. The MARS, however, recognizes that the requester is a multicast server by noting its address in the server map, and returns the corresponding host map so that the server can construct its point-to-multipoint connection.

Any end node that wishes to join and transmit to any multicast group—for instance, as triggered by an IGMP Report—must first register with the MARS server, using a MARS_JOIN message for the IP address 0.0.0.0. The MARS then adds the node as a leaf of its ClusterControlVC.

The node can the issue another MARS_JOIN message to request membership in any IP multicast group. The MARS server then stores the address of the requesting node in the host list that is associated with that group, so it can be returned in any subsequent MARS_REQUEST message for the group. The MARS then adds any node that sends a MARS_REQUEST for the group to this VC.

Note that all nodes in the cluster, regardless of whether or not they wish to transmit data to a group, must also send a MARS_JOIN to be added to the multicast group for the reserved address. The subsequent operation of the MARS is then a function of whether the group is being served by a multicast mesh or by multicast servers.

In the former case, where multicast meshes are used, the MARS forwards the MARS_JOIN message on the ClusterControlVC to inform any nodes that may already be members of the requested multicast group of the existence of a new member. All nodes transmitting to the group over existing point-to-multipoint connections then add the new requesting node to their connections using add-leaf messages.

Similarly, any node that wishes to leave a multicast mesh multicast group sends a MARS_LEAVE request to the MARS Server. This removes the node's ATM address from the list of ATM addresses registered with the IP multicast address and then forwards the message on its ClusterControlVC. This allows transmitting end systems to remove the leaving node from their point-to-multipoint connection. Transmitting nodes use timers and other mechanisms to clear inactive connections and conserve connection resources.

In the case of a group served by multicast servers, the MARS forwards any MARS_JOIN or MARS_LEAVE request to the registered multicast servers using the ServerControlVC. This allows the relevant multicast servers, which serve the group in concern, to either add or delete the requesting node from their own point-to-multipoint connections.

Multicast routers form a special case of end systems since they must, as per RFC 1112, receive IGMP Reports on any and all multicast group addresses. They must promiscuously join all groups by sending a block join message to the MARS for all addresses. Any node that sends a MARS_REQUEST subsequently ends up also transmitting to the router, either through a multicast server, or through its own point-to-multipoint connection. Note, however, that while routers must register to join all multicast groups, they do not need to allocate connections to any groups that do not have transmitting nodes. Also proposed is are mechanisms to allow routers to register and to promiscuously listen to only a subset of multicast connections. Routers must also register to transmit to the reserved group by sending a MARS_REQUEST for the reserved address.

Routers then use the reserved multicast group to transmit IGMP messages. Since all nodes that are members of multicast groups are also members of this reserved group, they monitor such IGMP Queries and respond to the corresponding multicast groups. The routers serving these groups then receive the IGMP Responses.

VII. Design Considerations for Networked Multimedia Environments

With Cisco's wide range of switching, routing, and ATM platforms, there are a variety of different network designs that will support the deployment of networked multimedia applications.

The best starting point for designing an infrastructure for networked multimedia applications is with an accurate profile of each of the multimedia applications selected. It is particularly important that the following questions be asked of each networked 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 (QoS) parameters?

The first question is extremely important to ask. Many of today's networked multimedia applications are packet-based video/audio applications. As such, these applications are transmitted using the traditional layer 3 protocols: IP, IPX, or AppleTalk.

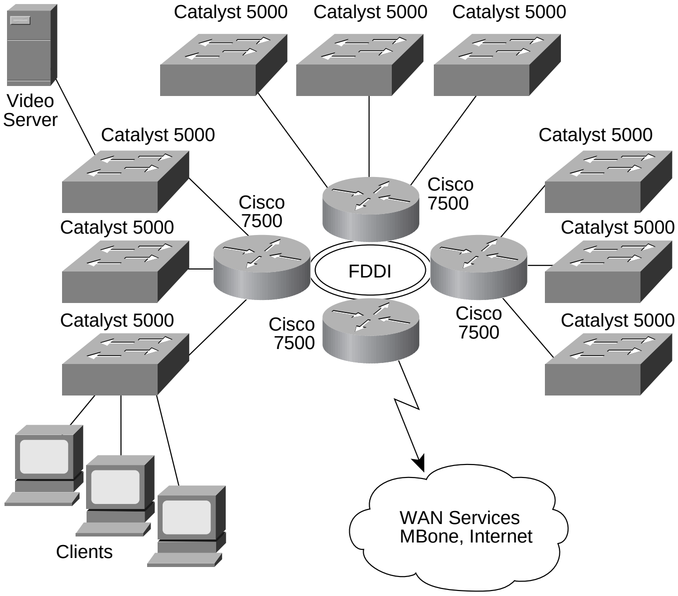
Stream-based applications, on the other hand, are best exemplified in ATM environments where video is captured and converted directly into ATM cells and transmitted "natively" using ATM through the ATM switch fabric. Typically these multimedia applications are CBR (refer to ATM QoS section for definition) and thus use AAL1 and circuit emulation for transmission.

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, 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 RFC 1577, Classical IP over ATM. However, if the ATM network needs to support IP multicast or additional LAN protocols the ATM network would run LAN emulation (LANE) instead.

The combination of Cisco switches and routers interconnected using a high-speed backbone technology (Fast Ethernet, FDDI, or ATM) will provide sufficient bandwidth for most networked multimedia applications in the campus environment. In fact if the networked multimedia applications were strictly point-to-point (i.e., unicast traffic) as in Figure 66, a micro-segmented switch/router network would be sufficient:

Figure 28. Switch/Router Campus Design



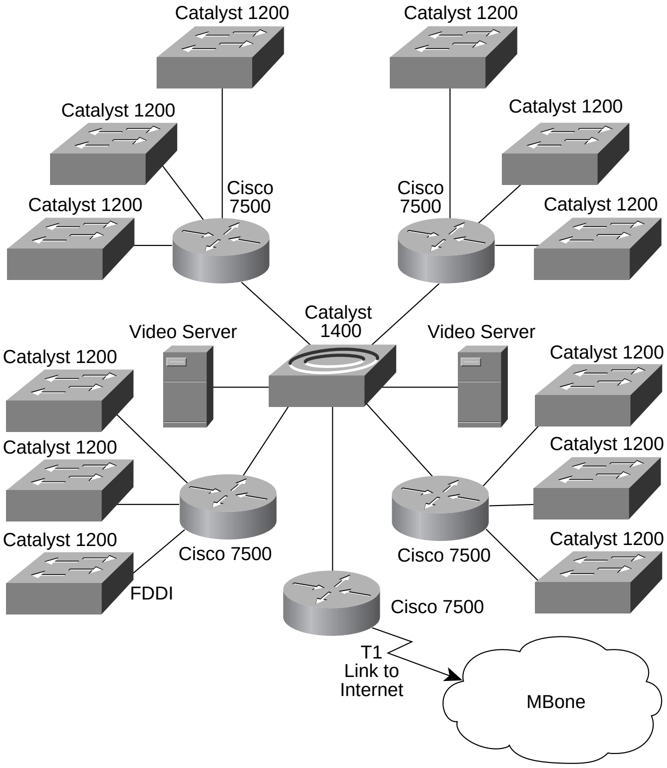
While the above network design succeeds for unicast applications that only impose bandwidth demands on the network, it will require further tailoring to run multicast applications. From a switch's perspective, layer 2 multicast frames are transmitted to all ports like regular layer 2 broadcast frames. For example, if a client accesses a multicast video stream on a server, the multicast transmission will be forwarded to all switch ports. This essentially undermines the performance benefits of switching.

Fortunately, there are three primary strategies for controlling multicast transmission in the campus LAN and WAN. These are discussed in the following sections.

Catalyst 1200/Cisco Router Campus Design

The first design, targeted for low port density switched Ethernet environments, relies on Catalyst 1200 switches for client and server access and Cisco routers for core connectivity. As depicted in Figure 67 below, this strategy controls multicast traffic by deploying IGMP at the switch port. This in turn allows multicast traffic to only be sent to ports that have registered an IGMP Join.

Figure 29. Catalyst 1200/Cisco Router Design



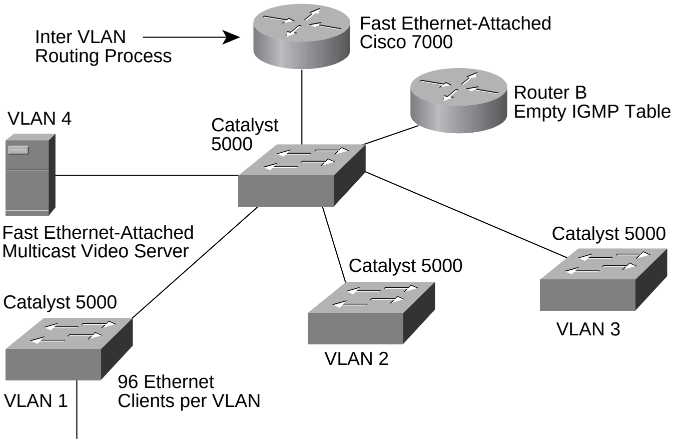
(8) 10 Mb Ports per Switch: For Client or Hub Access

Campus VLAN Designs

For high port densities for Ethernet or Token Ring, a combination of Catalyst 3000s, Catalyst 1600s, or Catalyst 5000s (Kalpana ProStacks) and Cisco routers is more effective. Typically in these designs, there is one switch or a group of switches per router interface. To address multicast traffic, these products employ VLAN technology. The 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 68 is an example of a campus VLAN design for use with a single network TV multicast application.

Figure 30. Network TV Multicast Design



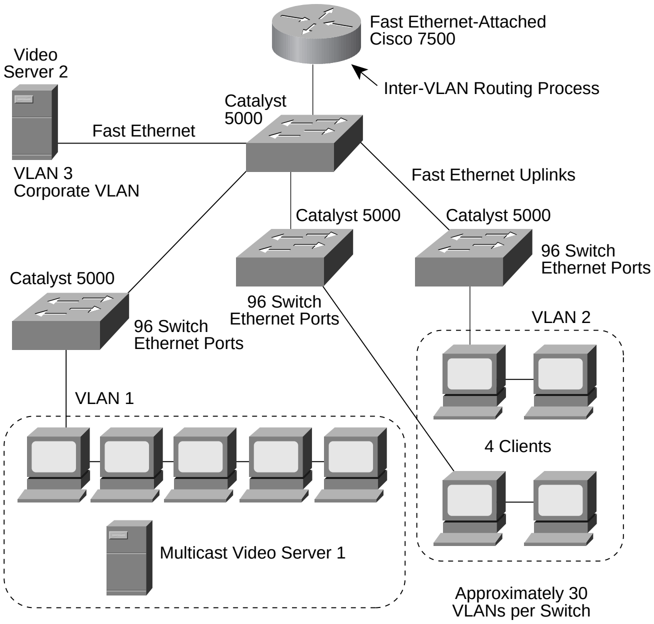
In Figure 68, there is only one VLAN per switch, resulting in a large number of clients per VLAN. The video source resides on the high-speed backbone and is in its own VLAN. During the multicast transmission, the video source sends a video stream on to the high-speed connection. The routers receive the video stream and send it out its high-speed link to the Catalyst 5000's VLANs.

Remember that when a given VLAN receives a multicast stream from the router, the transmission is forwarded to all members in that VLAN. Hence, this design is ideal in environments where every client in the VLAN tunes into the network TV transmission.

Remember also that the routers support IGMP, which will help to limit multicast traffic to only interfaces that have registered IGMP Joins from clients. In Figure 68, Router B has no IGMP receivers in its table and therefore multicast traffic is not forwarded out any of its interfaces.

To impose even greater control over multicast transmission, a micro-VLAN strategy can be employed. In this scenario, a switch will contain multiple VLANs, in turn limiting the multicast traffic to fewer clients (ports). Micro-VLANs are best utilized in multipoint videoconferencing environments and environments where there are multiple concurrent multicast video sources. In these environments there is the potential for many different multicast transmission to occur simultaneously. This in turn can impose some scalability issues unless the multicast traffic can be contained. Figure 69 shows a micro-VLAN design. Note that the VLANs are aligned based on multicast demands. VLAN 1, for example, contains clients that primarily receive video from multicast video server 1. VLAN 1 also receives from video server 2 in VLAN 3, the corporate broadcast service.

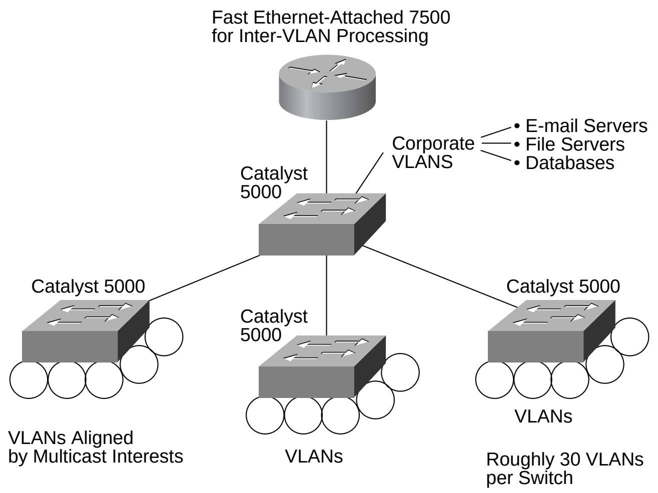
Figure 31. Micro-VLAN Design



As Figure 69 illustrates, the micro-VLAN approach minimizes the effects of multicast traffic by creating many small broadcast domains using VLANs.

One issue to keep in mind with the micro-VLAN design is that it may violate the 80/20 rule for designing VLANs. VLAN design is optimized when 80 percent of the traffic is intra-VLAN and 20 percent is inter-VLAN. 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 the e-mail server, for instance, would be an inter-VLAN process (see Figure 70). And since inter-VLAN communication must be handled by a router, it follows that as inter-VLAN communication increases, so to does route processing. Ultimately, the number of VLANs per router port should be determined by the multicast applications in use and their respective bandwidth requirements. It follows that high-bandwidth multicast applications will restrict the number of VLANs on a given router interface more than on bandwidth multicast applications.

Figure 32. Router Operation in Micro-VLAN Design



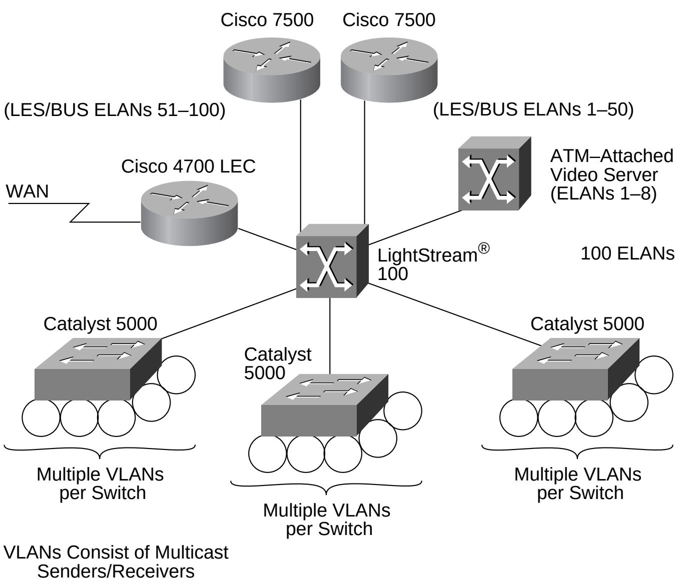
ATM LAN Emulation Design Considerations

As mentioned earlier in this document, VLAN communication is supported over Fast Ethernet using the ISL protocol; over FDDI using the 802.10 protocol; and over ATM using LAN emulation (LANE). While ISL and 802.10 implementations are relatively straightforward, LANE designs are a little more complex. Remember LANE's principal components:

- LEC
- LES/BUS
- LECS

Typically when designing an enterprise network using LANE technology, the primary issues center on LES/BUS scalability. Currently, all multicast transmission rely on the BUS (Broadcast Unknown Server) for delivery to all LECs within a given ELAN (emulated LAN). In a Cisco ATM network, the router operates as the BUS for a given ELAN. If the router supports multiple ELANs it follows that it runs multiple BUS processes. It also follows that router performance is a function of the number of ELANs it is a member of and the number of BUS processes that it executes. In environments where there are a large number of ELANs, it is recommended that additional routers be deployed to handle BUS functionality for each of the ELANs. Essentially, BUS functionality is distributed across a set of routers in the ATM network (see Figure 71 below).

Figure 33. Distributed LES/BUS Design

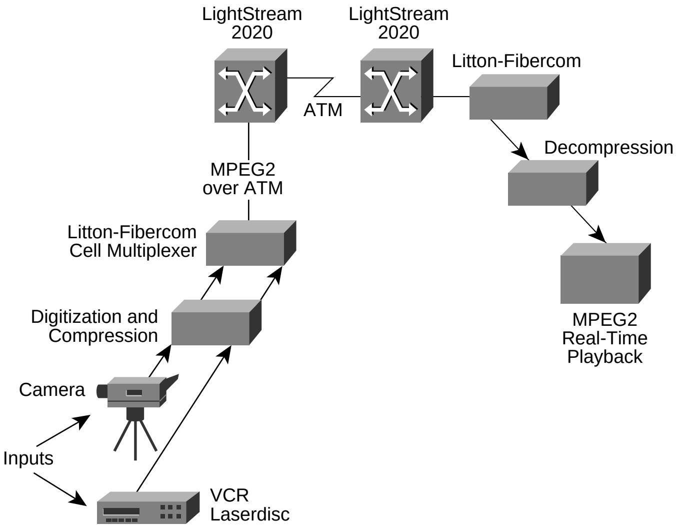


Currently LANE is the only method for addressing multicast packet-based video. As mentioned earlier in this document, RFC 1577 currently has no provision for resolving layer 2 multicast addresses into ATM addresses.

Native-Mode ATM Network Design

As mentioned earlier, native-mode applications bypass traditional layer 3 packetization and run directly at layer 2 (ATM layer). And as mentioned earlier, LANE is best suited for "best effort" traffic or in ATM-speak, ABR traffic. Consequently, LANE is not the best environment for applications that require more predictable network service, particularly CBR and VBR multimedia applications. For these applications, it is best to run natively using ATM and bypass LANE all together. In a native-mode 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 an available native-mode 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 72 shows a diagram of MPEG2 running over ATM.

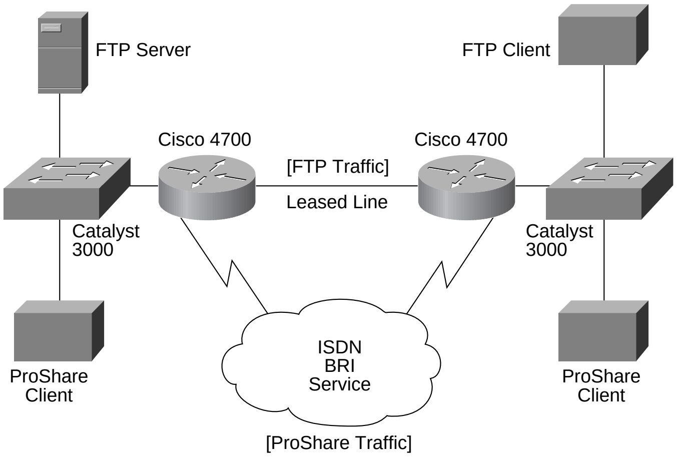
Figure 34. MPEG2 over ATM



WAN Design Considerations

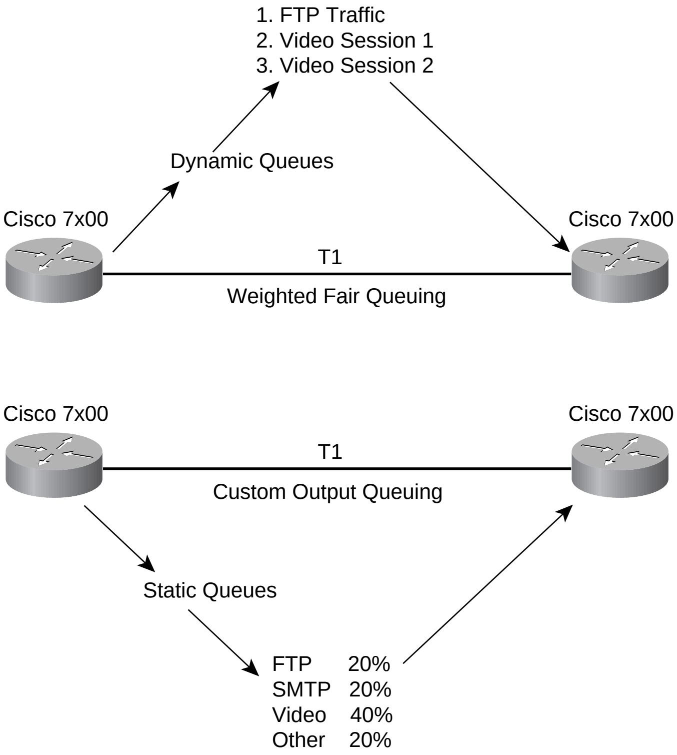
Much of the strategy for WAN design for networked multimedia was discussed earlier in this document. To summarize, consider circuit-switched technologies for running networked multimedia. This in turn will limit WAN charges to only the time the multimedia application is in use. Use the Cisco IOS policy-based routing feature to restrict multimedia traffic to the circuit-switched connection (see Figure 73).

Figure 35. Policy-Based Routing for Networked Multimedia



Additionally, take advantage of the Cisco IOS traffic queuing methods—Priority Output Queuing, Custom Queuing, and Weighted Fair Queuing (WFQ)—to optimize WAN traffic patterns. For example, set up a queue for a particular multicast session or use WFQ to dynamically queue the multicast stream (see Figure 74).

Figure 36. WAN Queuing Techniques



VIII. Conclusion and References

As this document details, networked multimedia is rapidly being deployed in Campus LAN and WAN environments. Cisco acknowledges this burgeoning trend by focusing its attention on delivering enterprise solutions that provide:

- Scalable bandwidth
- Quality of service
- Effective multicasting

From a bandwidth perspective, the wide-range of Cisco routing, switching and ATM platforms provide a multitude of network design possibilities that can successfully deliver bandwidth where it is most need.

Cisco IOS helps to further enhance the Cisco network multimedia solution by providing effective QoS and multicasting features.

Together, Cisco hardware and Cisco IOS provides a solid infrastructure for addressing both the needs of today’s network multimedia applications as well as tomorrow’s.

Network Multimedia References

For additional information on network multimedia both from a design and implementation standpoint, please refer to the following.

Cisco Documents

- *ATM Internetworking*, Anthony Alles
- *Designing Switched LANs*, Harbrinder Kang
- *Video over ATM and Existing Networks*, Mordechai Fester
- *ISDN Design and Implementation Guide*, Jeff Baher

World Wide Web Sites

- Multimedia General Web Site:
  - http://www.yahoo.com/Computers_and_Internet/Multimedia/
- Desktop Video Conferencing Applications:
  - http://www2.ncsu.edu/eos/service/ece/project/succeed_info/dtvc_survey/products.html
- Desktop Video Conferencing Applications by Platform or Standard:
  - http://www2.ncsu.edu/eos/service/ece/project/succeed_info/dtvc_survey/features.html
- Desktop Video Conferencing Standards:
  - http://www2.ncsu.edu/eos/service/ece/project/succeed_info/dtvc_survey/std.html
- Multimedia Standards:
  - http://viswiz.gmd.de/MultimediaInfo/#Standards
- Related Multimedia Information (these sites point to a variety of other web sites):
  - http://www2.ncsu.edu/eos/service/ece/project/succeed_info/dtvc_survey/other.html
  - http://viswiz.gmd.de:80/MultimediaInfo
- MPEG Companies:
  - http://www.crs4.it/~luigi/MPEG/mpegcompanies.html

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Cisco Systems has over 100 sales offices worldwide. Call the company's corporate headquarters (California, USA) at 408 526-4000 to contact your local account representative or, in North America, call 800 553-NETS (6387).

1/96