The Internet Protocol Journal

September 1998 Volume 1, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

Missed the first issue of IPJ? Download your copy in PDF format from: www.cisco.com/ipj

From The Editor

We begin this issue with Part II of "What Is a VPN?" by Paul Ferguson and Geoff Huston. In Part I they introduced a definition of the term "Virtual Private Network" (VPN) and discussed the motivations behind the adoption of such networks. They outlined a framework for describing the various forms of VPNs, and examined numerous network-layer VPN structures, in particular, that of controlled route leakage and tunneling. In Part II the authors conclude their examination of VPNs by describing virtual private dial networks and network-layer encryption. They also examine link-layer VPNs, switching and encryption techniques, and issues concerning Quality of Service and non-IP VPNs.

IP Multicast is an emerging set of technologies and standards that allow many-to-many transmissions such as conferencing, or one-to-many transmissions such as live broadcasts of audio and video over the Internet. Kenneth Miller describes multicast in general, and reliable multicast protocols and applications in particular. Although multicast applications are primarily used in the research community today, this situation is likely to change as the demand for Internet multimedia applications increases and multicast technologies improve.

Successful deployment of networking technologies requires an understanding of a number of technology options ranging from wiring and transmissions systems via switches, routers, bridges and other pure networking components, to networked applications and services. *The Internet Protocol Journal* (IPJ) is designed to look at all aspects of these "building blocks." This time, Thayumanavan Sridhar details some of the issues in the evolution of Layer 2 and Layer 3 switches.

Interest in the first issue of IPJ has exceeded our expectations, and hard copies are almost gone. However, you can still view and print the issue in PDF format on our Web site at **www.cisco.com/ipj**. The current edition is also available on the Web. If you want to receive our next issue, please complete and return the enclosed card.

We welcome your comments, questions and suggestions regarding anything you read in this journal. We are also actively seeking authors for new articles. The Call for Papers and Author Guidelines can be found on our Web page. Please send your comments to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher

ole@cisco.com

What Is a VPN? — Part II

by Paul Ferguson, Cisco Systems
and Geoff Huston, Telstra

In Part I we introduced a working definition of the term "Virtual Private Network" (VPN), and discussed the motivations behind the adoption of such networks. We outlined a framework for describing the various forms of VPNs, and then examined numerous network-layer VPN structures, in particular, that of controlled route leakage and tunneling techniques. We begin Part II with examining other network-layer VPN techniques, and then look at issues that are concerned with non-IP VPNs and Quality-of-Service (QoS) considerations.

Types of VPNs

This section continues from Part I to look at the various types of VPNs using a taxonomy derived from the layered network architecture model. These types of VPNs segregate the VPN network at the network layer.

Network-Layer VPNs

A network can be segmented at the network layer to create an end-to-end VPN in numerous ways. In Part I we described a controlled route leakage approach that attempts to perform the segregation only at the edge of the network, using route advertisement control to ensure that each connected network received a view of the network (only peer networks). We pick up the description at this point in this second part of the article.

Tunneling

As outlined in Part I, the alternative to a model of segregation at the edge is to attempt segregation throughout the network, maintaining the integrity of the partitioning of the substrate network into VPN components through the network on a hop-by-hop basis. Part I examined numerous tunneling technologies that can achieve this functionality. Tunneling is also useful in servicing VPN requirements for dial access, and we will resume the description of tunnel-based VPNs at this point.

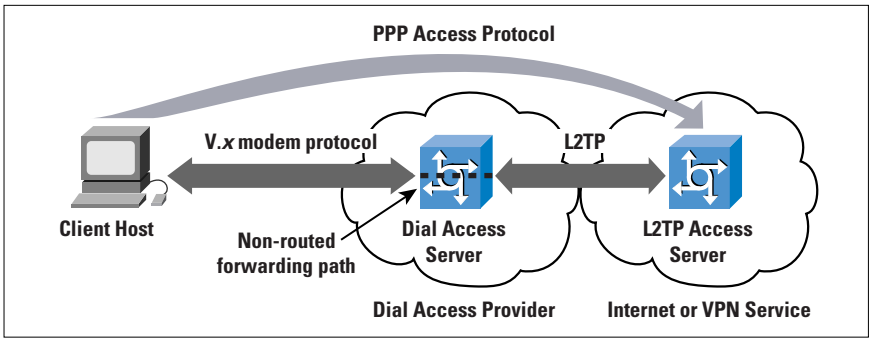
Virtual Private Dial Networks

Although several technologies (vendor-proprietary technologies as well as open, standards-based technologies) are available for constructing a *Virtual Private Dial Network* (VPDN), there are two principal methods of implementing a VPDN that appear to be increasing in popularity—*Layer 2 Tunneling Protocol* (L2TP) and *Point-to-Point Tunneling Protocol* (PPTP) tunnels. From an historical perspective, L2TP is the technical convergence of the earlier Layer 2 Forwarding (L2F)[1] protocol specification and the PPTP protocol. However, one might suggest that because PPTP is now being bundled into the desktop operating system of many of the world's personal computers, it stands to be quite popular within the market.

At this point it is worthwhile to distinguish the difference between "client-initiated" tunnels and "NAS-initiated" (Network Access Server, otherwise known as a Dial Access Server) tunnels. The former is commonly referred to as "voluntary" tunneling, whereas the latter is commonly referred to as "compulsory" tunneling. In voluntary tunneling, the tunnel is created at the request of the user for a specific purpose; in compulsory tunneling, the tunnel is created without any action from the user, and without allowing the user any choice in the matter.

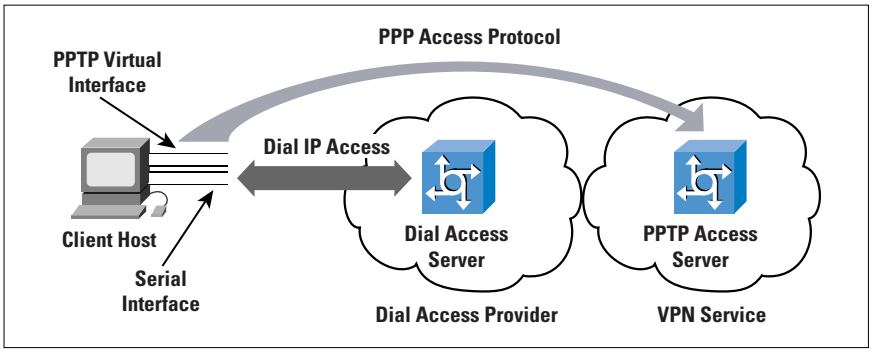
L2TP, as a compulsory tunneling model, is essentially a mechanism to "off-load" a dialup subscriber to another point in the network, or to another network altogether. In this scenario, a subscriber dials into a NAS, and based on a locally configured profile (or a NAS negotiation with a policy server) and successful authentication, a L2TP tunnel is dynamically established to a predetermined endpoint, where the subscriber's *Point-to-Point Protocol* (PPP) session is terminated (Figure 1).

Figure 1: PPP Tunnel Termination Model of L2TP



PPTP, as a voluntary tunneling model, on the other hand, allows end systems (for example, desktop computers) to configure and establish individual discrete point-to-point tunnels to arbitrarily located PPTP servers, without the intermediate NAS participating in the PPTP negotiation and subsequent tunnel establishment. In this scenario, a subscriber dials into a NAS, but the PPP session is terminated on the NAS, as in the traditional Internet access PPP model. The layered PPTP session is then established between the client end system and any upstream PPTP server that the client desires to connect to. The only caveats on PPTP connectivity are that the client can reach the PPTP server via conventional routing processes, and that the user has been granted the appropriate privileges on the PPTP server (Figure 2).

Figure 2: PPP Tunnel Termination Model of PPTP



Although L2TP and PPTP may sound extraordinarily similar, there are subtle differences that deserve further examination. The applicability of both protocols is very much dependent on what problem is being addressed. It is also about control—who has it, and why it is needed. It also depends heavily on how each protocol implementation is deployed—in either the voluntary or the compulsory tunneling models.

With PPTP in a voluntary tunneling implementation, the dial-in user can choose the PPTP tunnel destination (the PPTP server) after the initial PPP negotiation has completed. This feature is important if the tunnel destination changes frequently, because no modifications are needed to the client's view of the base PPP access when there is a change in the server and the transit path to the server. It is also a significant advantage that the PPTP tunnels are transparent to the service provider, and no advance configuration is required between the NAS operator and the overlay dial access VPN. In such a case, the service provider does not house the PPTP server, and simply passes the PPTP traffic along with the same processing and forwarding policies as all other IP traffic. In fact, this feature should be considered a significant benefit of this approach. The configuration and support of a tunneling mechanism within the service provider network would be one less parameter that the service provider has to operationally manage, and the PPTP tunnel can transparently span multiple service providers without any explicit service provider configuration. However, the economic downside to this feature for the service provider, of course, is that a "VPDN-enabled" network service can be marketed to yield an additional source of revenue. Where the client undertakes the VPDN connection, there is no direct service provider involvement and no consequent value added to the base access service.

From the subscriber's perspective, this is a "win-win" situation, because the user is not reliant on the upstream service provider to deliver the VPDN service—at least no more than any user is reliant for basic IP-level connectivity. The other "win" is that the subscriber does not have to pay a higher subscription fee for a VPN service. Of course, the situation changes when the service provider takes an active role in providing the VPDN, such as housing the PPTP servers, or if the subscriber resides within a subnetwork in which the parent organization wants the service provider's network to make the decision concerning where tunnels are terminated. The major characterization of PPTP-based VPDN is one of a roaming client base, where the clients of the VPDN use a local connection to the public Internet data network, and then overlay a private data tunnel from the client's system to the desired remote service point. Another perspective is to view this approach as "on-demand" VPDN virtual circuits.

With L2TP in a "compulsory" tunneling implementation, the service provider controls where the PPP session is terminated. This setup can be extremely important in situations where the service provider to whom

the subscriber is actually dialing into (let's call it the "modem pool provider" network) must transparently hand off the subscriber's PPP session to another network (let's call this network the "content provider"). To the subscriber, it appears as though the local system is directly attached to the content provider's network, when in fact the access path has been passed transparently through the modem pool provider's network to the subscribed content service. Very large content providers, for instance, may outsource the provisioning and maintenance of thousands of modem ports to a third-party access provider, who in turn agrees to transparently pass the subscribers' access sessions back to the content provider. This setup is generally called "wholesale dial." The major motivation for such L2TP-based wholesale dial lies in the typical architecture of the *Public Switched Telephone Network* (PSTN), where the use of wholesale dial facilities can create a more rational PSTN call load pattern with Internet access PSTN calls terminated in the local Central Office.

Of course, if all subscribers who connect to the modem pool provider's network are destined for the same content provider, then there are certainly easier ways to hand this traffic off to the content provider's network—such as simply aggregating all the traffic in the local Central Office and handing the content provider a "big fat pipe" of the aggregated session traffic streams. However, in situations where the modem pool provider is providing a wholesale dial service for multiple upstream "next-hop" networks, the methods of determining how each subscriber's traffic must be forwarded to his/her respective content provider are somewhat limited. Packet forwarding decisions could be made at the NAS, based on the source address of the dialup subscriber's computer. This scenario would allow for traffic to be forwarded along the appropriate path to its ultimate destination, in turn intrinsically providing a virtual connection. However, the use of assigning static IP addresses to dial-in subscribers is highly discouraged because of the inefficiencies in IP address utilization policies, and the critical success of the *Dynamic Host Configuration Protocol* (DHCP).

There are, however, some serious scaling concerns in deploying a large-scale L2TP network; these concerns revolve around the issue of whether large numbers of tunnels can actually be supported with little or no network performance impact. Since there have been no large-scale deployments of this technology to date, there is no empirical evidence to support or invalidate these concerns.

In some cases, however, appearances are everything—some content providers do not wish for their subscribers to know that when they connect to their service, they have instead been connected to another service provider's network, and then passed along ultimately to the service to which they have subscribed. In other cases, it is merely designed to be a matter of convenience, so that subscribers do not need to log into a device more than once.

Regrettably, the L2TP draft does not detail all possible implementations or deployment scenarios for the protocol. The basic deployment scenario is quite brief when compared to the rest of the document, and is arguably biased toward the compulsory tunneling model. Nonetheless, there are implementations of L2TP that follow the voluntary tunneling model. To the best of our knowledge, there has never been any intent to exclude this model of operation. In addition, at various recent interoperability workshops, several different implementations of a voluntary L2TP client have been modeled. Nothing in the L2F protocol would prohibit deploying it in a voluntary tunneling manner, but to date it has not been widely implemented. Further, PPTP has also been deployed using the compulsory model in a couple of specific vendor implementations.

In summary, consideration of whether PPTP or L2TP is more appropriate for deployment in a VPDN depends on whether control needs to lie with the service provider or with the subscriber. Indeed, the difference can be characterized with respect to the client of the VPN, where the L2TP model is one of a "wholesale" access provider who has numerous configured client service providers who appear as VPNs on the common dial access system, whereas the PPTP model is one of distributed private access where the client is an individual end user and the VPN structure is that of end-to-end tunnels. One might also suggest that the difference is also a matter of economics, because the L2TP model allows service providers to actually provide a "value-added" service, beyond basic IP-level connectivity, and charge their subscribers accordingly for the ability to access it, thus creating new revenue streams. By contrast, the PPTP model enables distributed reach of the VPN at a much more basic level, enabling corporate VPNs to extend access capabilities without the need for explicit service contracts with a multitude of network access providers.

Network-Layer Encryption

Encryption technologies are extremely effective in providing the segmentation and virtualization required for VPN connectivity, and they can be deployed at almost any layer of the protocol stack. The evolving standard for network-layer encryption in the Internet is *IP Security* (IPSec)[3, 4]. (IPSec is actually an architecture—a collection of protocols, authentication, and encryption mechanisms. The IPSec security architecture is described in detail in [3].)

While the *Internet Engineering Task Force* (IETF) is finalizing the architecture and the associated protocols of IPSec, there is relatively little network-layer encryption being done in the Internet today. However, some vendor proprietary solutions are currently in use.

Whereas IPSec has yet to be deployed in any significant volume, it is worthwhile to review the two methods in which network-layer encryption is predominantly implemented. The most secure method for network-

layer encryption to be implemented is end-to-end, between participating hosts. End-to-end encryption allows for the highest level of security. The alternative is more commonly referred to as "tunnel mode," in which the encryption is performed only between intermediate devices (routers), and traffic between the end system and the first-hop router is in plaintext. This setup is considerably less secure, because traffic intercepted in transit between the first-hop router and the end system could be compromised.

As a more general observation on this security vulnerability, where a VPN architecture is based on tunnels, the addition of encryption to the tunnel still leaves the tunnel ingress and egress points vulnerable, because these points are logically part of the host network as well as being part of the unencrypted VPN network. Any corruption of the operation, or interception of traffic in the clear, at these points will compromise the privacy of the private network.

In the end-to-end encryption scheme, VPN granularity is to the individual end-system level. In the tunnel mode scheme, the VPN granularity is to the subnetwork level. Traffic that transits the encrypted links between participating routers, however, is considered secure. Network-layer encryption, to include IPSec, is merely a subset of a VPN.

Link-Layer VPNs

One of the most straightforward methods of constructing VPNs is to use the transmission systems and networking platforms for the physical and link-layer connectivity, yet still be able to build discrete networks at the network layer. A link-layer VPN is intended to be a close (or preferably exact) functional analogy to a conventional private data network.

ATM and Frame Relay Virtual Connections

A conventional private data network uses a combination of dedicated circuits from a public carrier, together with an additional private communications infrastructure, to construct a network that is completely self-contained. Where the private data network exists within private premises, the network generally uses a dedicated private wiring plant to carry the VPN. Where the private data network extends outside the private boundary of the dedicated circuits, it is typically provisioned for a larger public communications infrastructure by using some form of time-division or frequency-division multiplexing to create the dedicated circuit. The essential characteristic of such circuits is the synchronization of the data clock, such that the sender and receiver pass data at a clocking rate that is fixed by the capacity of the dedicated circuit.

A link-layer VPN attempts to maintain the critical elements of this self-contained functionality, while achieving economies of scale and operation, by utilizing a common switched public network infrastructure. Thus, a collection of VPNs may share the same infrastructure for connectivity, and share the same switching elements within the interior of

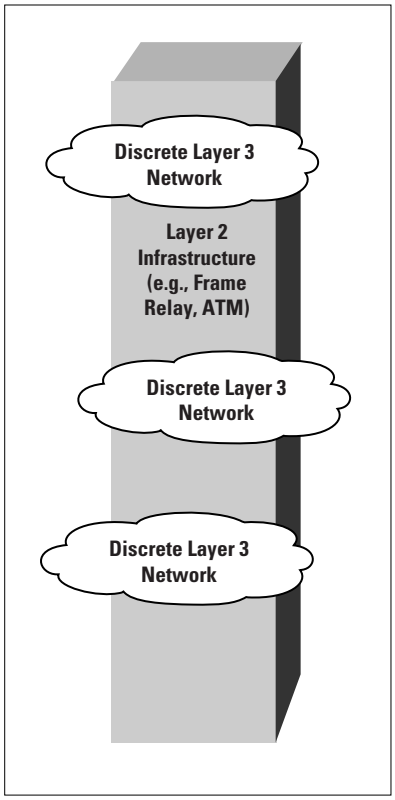


Figure 3: Conceptualization of Discrete Layer 3 Networks on a Common Layer 2 Infrastructure

the network, but explicitly must have no visibility, either direct or inferred, of one another. Generally, these "networks" operate at Layer 3 (the network layer) or higher in the OSI Reference Model, and the "infrastructure" itself commonly consists of either a *Frame Relay* or *Asynchronous Transfer Mode* (ATM) network (Figure 3). The essential difference here between this architecture of virtual circuits and that of dedicated circuits is that there is now no synchronized data clock shared by the sender and receiver, nor necessarily is there a dedicated transmission path that is assigned from the underlying common host network. The sender generally has no a priori knowledge of the available capacity of the virtual circuit, because the capacity varies in response to the total demand placed on it by other simultaneous transmission and switching activity. Instead, the sender and receiver can use adaptive clocking of data, where the sender can adjust the transmission rate to match the requirements of the application and any signaling received from the network and the receiver. It should be noted that a dedicated circuit system using synchronized clocking cannot be oversubscribed, whereas the virtual circuit architecture (where the sender does not have a synchronized end-to-end data clock) can indeed be oversubscribed. It is the behavior of the network when it transitions into this oversubscribed state that is of most interest here.

One of the nice things about a public switched wide-area network that provides virtual circuits is that it can be extraordinarily flexible. Most subscribers to Frame Relay services, for example, have subscribed to the service for economic reasons—it is cheap, and the service provider usually adds a *Service-Level Agreement* (SLA) that "guarantees" some percentage of frame delivery in the Frame Relay network itself.

The remarkable thing about this service offering is that the customer is generally completely unaware of whether the service provider can actually deliver the contracted service at all times and under all possible conditions. The Layer 2 technology is not a synchronized clock blocking technology in which each new service flow is accepted or denied based on the absolute ability to meet the associated resource demands. Each additional service flow is accepted into the network and carried on a best-effort basis. Admission functions provide the network with a simple two-level discard mechanism that allows a graduated response to instances of overload; however, when the point of saturated overload is reached within the network, all services will be affected.

This situation brings up several other important issues: The first concerns the engineering practices of the Frame Relay service provider. If the Frame Relay network is poorly engineered and is constantly congested, then obviously the service quality delivered to the subscribers will be affected. Frame Relay uses a notion of a per-virtual circuit *Committed Information Rate* (CIR), which is an ingress function associated with Frame Relay that checks the ingress traffic rate against the CIR.

Frames that exceed this base rate are still accepted by the Frame Relay network, but they are marked as *discard eligible* (DE). Because the network can be oversubscribed, the data rate within a switch will at times exceed both the egress transmission rate and the local buffer storage. When this situation occurs, the switch will begin to discard data frames, and will do so initially for frames with the DE marker present. This scenario is essentially a two-level discard precedence architecture. It is an administrative decision by the service provider as to the relative levels of provisioning of core transmission and switching capacity, and the ratio of network ingress capacity used by subscribers. The associated CIRs of the virtual circuits against this core capacity are critical determinants of the resultant deliverable quality of performance of the network and the layered VPNs.

For example, at least one successful (and popular) Frame Relay service provider provides an economically attractive Frame Relay service that permits a zero-rate CIR on PVCs, combined with an SLA that ensures that at least 99.8 percent of all frame-level traffic presented to the Frame Relay network will be delivered successfully. If this SLA is not met, then the subscriber's monthly service fee will be appropriately prorated the following month. The Frame Relay service provider provides frame level statistics to each subscriber every month, culled from the Frame Relay switches, to measure the effectiveness of this SLA "guarantee." This particular Frame Relay service provider is remarkably successful in honoring the SLAs because they conduct ongoing network capacity management on a weekly basis, provisioning new trunks between Frame Relay switches when trunk utilization exceeds 50 percent, and ensuring that trunk utilization never exceeds 75 percent. In this fashion, traffic on PVCs with a zero-rate CIR can generally avoid being discarded in the Frame Relay network.

Having said that, the flexibility of PVCs allows discrete VPNs to be constructed across a single Frame Relay network. And in many instances, this scenario lends itself to situations where the Frame Relay network provider also manages each discrete VPN via a telemetry PVC. Several service providers have *Managed Network Services* (MNS) that provide exactly this type of service.

Whereas the previous example revolves around the use of Frame Relay as a link-layer mechanism, essentially the same type of VPN mechanics hold true for ATM. As with Frame Relay, there is no data clock synchronization between the sender, the host network, and the receiver. In addition, the sender's traffic is passed into the ATM network via an ingress function, which can mark cells with a *Cell Loss Priority* (CLP) indication. And, as with Frame Relay, where a switch experiences congestion, the switch will attempt to discard marked (CLP) cells as the primary load shedding mechanism, but if this step is inadequate, the network must shed other cells that are not so marked. Once again, the quality of the service depends on proper capacity engineering of the network, and there is no guarantee of service quality inherently in the technology itself.

The generic observation is that the engineering of Frame Relay and ATM common carriage data networks is typically very conservative. The inherent capabilities of both of these link-layer architectures do not permit a wide set of selective responses to network overload, so that in order for the network to service the broadest spectrum of potential VPN clients, the network must provide high-quality carriage and very limited instances of any form of overload. In this way, such networks are typically positioned as a high-quality alternative to dedicated circuit private network architectures, which are intended to operate in a very similar manner (and, not surprisingly, are generally priced as a premium VPN offering). Technically, the architecture of link-layer VPNs is almost indistinguishable from the dedicated circuit private data network—the network can support multiple protocols, private addressing, and routing schemes, because the essential difference between a dedicated circuit and a virtual link-layer circuit is the absence of synchronized clocking between the sender and the receiver. In all other aspects, the networks are very similar.

These approaches to constructing VPNs certainly involve scaling concerns, especially with regard to configuration management of provisioning new *Virtual Connections* (VCs) and routing issues. Configuration management still tends to be one of the controversial points in VPN management—adding new subscribers and new VPNs to the network requires VC path construction and provisioning, a tedium that requires ongoing administrative attention by the VPN provider. Also, as already mentioned, full mesh networks encounter scaling problems, in turn resulting in construction of VPNs in which partial meshing is done to avoid certain scaling limitations. The liabilities in these cases need to be examined closely, because partial meshing of the underlying link-layer network may contribute to suboptimal routing (for example, extra hops caused by hub-and-spoke issues, or redirects).

These problems apply to all types of VPNs built on the "overlay" model—not just ATM and Frame Relay. Specifically, the problems also apply to *Generic Routing Encapsulation* (GRE) tunnels.

MPOA and the "Virtual Router" Concept

Another unique model of constructing VPNs is the use of *Multiprotocol over ATM* (MPOA)[5], which uses RFC 1483 encapsulation[6]. This VPN approach is similar to other "cut-through" mechanisms in which a particular switched link layer is used to enable all "Layer 3" egress points to be only a single hop away from one another.

In this model, the edge routers determine the forwarding path in the ATM switched network, because they have the ability to determine which egress point packets need to be forwarded to. After a network-layer reachability decision is made, the edge router forwards the packet onto a VC designated for a particular egress router. However, since the egress routers cannot use the *Address Resolution Protocol* (ARP) for destination address across the cloud, they must rely on an external server for address resolution (ATM address to IP address).

The first concern here is a sole reliance on ATM—this particular model does not encompass any other types of data link layer technologies, rendering the technology less than desirable in a hybrid network. Whereas this scenario may have some domain of applicability within a homogenous ATM environment, when looking at a broader VPN environment that may encompass numerous link-layer technologies, this approach offers little benefit to the VPN provider.

Secondly, there are serious scaling concerns regarding full mesh models of connectivity, where suboptimal network-layer routing may result because of cut-through. And the reliance on address resolution servers to support the ARP function within the dynamic circuit framework brings this model to the point of excessive complexity.

The advantage of the MPOA approach is the use of dynamic circuits rather than more cumbersome, statically configured models. The traditional approach to supporting private networks involves extensive manual design and operational support to ensure that the various configurations on each of the bearer switching elements are mutually consistent. The desire within the MPOA environment is to attempt to use MPOA to govern the creation of dynamically controlled, edge-to-edge ATM VCs. Although this setup may offer the carrier operator some advantages in reduced design and operational overhead, it does require the uniform availability of ATM, and in many heterogeneous environments this scenario is not present.

In summary, this model is another overlay model, with some serious concerns regarding the ability of the model to withstand scale.

"Peer" VPN models that allow the egress nodes to maintain separate routing tables have also been introduced—one for each VPN—effectively allowing separate forwarding decisions to be made within each node for each distinctive VPN. Although this is an interesting model, it introduces concerns about approaches in which each edge device runs a separate routing process and maintains a separate *Routing Information Base* (RIB, or routing table) process for each VPN community of interest. It also should be noted that the "virtual router" concept requires some form of packet labeling, either within the header or via some lightweight encapsulation mechanism, in order for the switch to be able to match the packet against the correct VPN routing table. If the label is global, the issue of operational integrity is a relevant concern, whereas if the label is local, the concept of label switching and maintenance of edge-to-edge label switching contexts is also a requirement.

Among the scaling concerns are issues regarding the number of supported VPNs in relation to the computational requirements, and stability of the routing system within each VPN (that is, instability in one VPN affecting the performance of other VPNs served by the same device). The aggregate scaling demands of this model are also significant. Given a change in the underlying physical or link-layer topology, the consequent

requirement to process the routing update on a per-VPN basis becomes a significant challenge. Use of distance vector protocols to manage the routing tables would cause a corresponding sudden surge in traffic load, and the surge grows in direct proportion to the number of supported VPNs. The use of link-state routing protocols would require the consequent link-state calculation to be repeated for each VPN, causing the router to be limited by available CPU capacity.

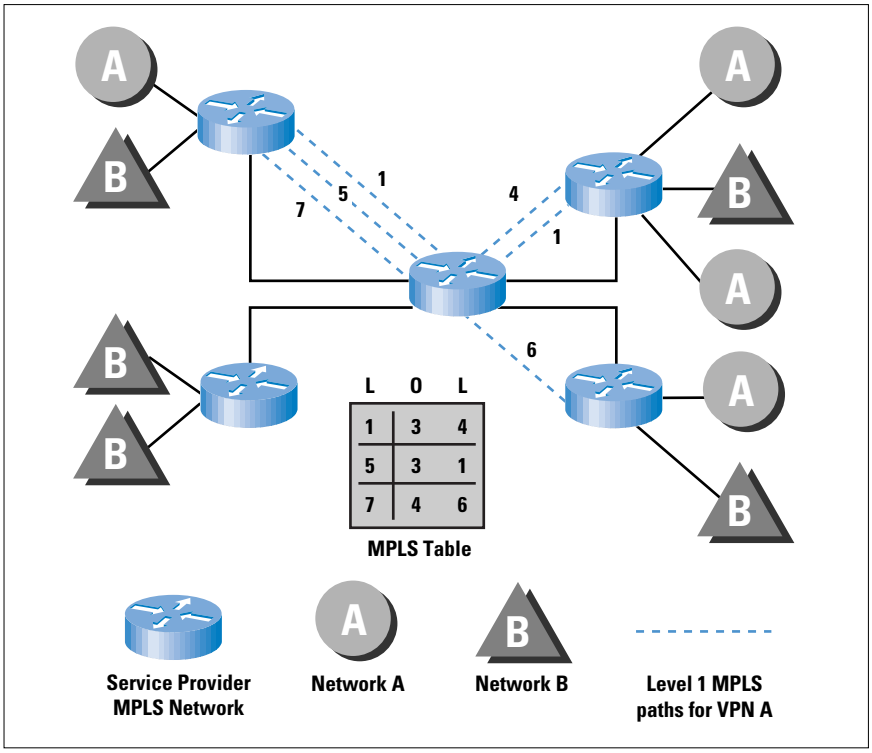
Multiprotocol Label Switching

One method of addressing these scaling issues is to use VPN labels within a single routing environment, in the same way that packet labels are necessary to activate the correct per-VPN routing table. The use of local label switching effectively recreates the architecture of a Multiprotocol Label Switching VPN. It is perhaps no surprise that when presented with two basic approaches to the architecture of the VPN—the use of network-layer routing structures and per-packet switching, and the use of link-layer circuits and per-flow switching—the industry would devise a hybrid architecture that attempts to combine aspects of these two approaches. This hybrid architecture is referred to as *Multiprotocol Label Switching* (MPLS)[7, 8].

The architectural concepts used by MPLS are generic enough to allow it to operate as a peer VPN model for switching technology for a variety of link-layer technologies, and in heterogeneous Layer 2 transmission and switching environments. MPLS requires protocol-based routing functionality in the intermediate devices, and operates by making the interswitch transport infrastructure visible to the routing. In the case of IP over ATM, each ATM bearer link becomes visible as an IP link, and the ATM switches are augmented with IP routing functionality. IP routing is used to select a transit path across the network, and these transit paths are marked with a sequence of labels that can be thought of as locally defined forwarding path indicators. MPLS itself is performed using a label swapping forwarding structure. Packets entering the MPLS environment are assigned a local label and an outbound interface based on a local forwarding decision. The local label is attached to the packet via a lightweight encapsulation mechanism. At the next MPLS switch, the forwarding decision is based on the incoming label value, where the incoming label determines the next hop interface and next hop label, using a local forwarding table indexed by label. This lookup table is generated by a combination of the locally used IP routing protocol, together with a label distribution protocol, which creates end-to-end transit paths through the network for each IP destination. It is not our intention to discuss the MPLS architecture in detail, apart from noting that each MPLS switch uses a label-indexed forwarding table, where the attached label of an incoming packet determines the next-hop interface and the corresponding outgoing label.

The major observation here is that this lightweight encapsulation, together with the associated notion of boundary-determined transit paths, provides many of the necessary mechanisms for the support of VPN structures[9]. MPLS VPNs have not one, but three key ingredients: (1) constrained distribution of routing information as a way to form VPNs and control inter-VPN connectivity; (2) the use of VPN-IDs, and specifically the concatenation of VPN-IDs with IP addresses to turn (potentially) nonunique addresses into unique ones; and (3) the use of label switching (MPLS) to provide forwarding along the routes constructed via (1) and (2). The generic architecture of deployment is that of a label-switched common host network and a collection of VPN environments that use label-defined virtual circuits on an edge-to-edge basis across the MPLS environment. An example is indicated in Figure 4, which shows how MPLS virtual circuits are constructed.

Figure 4: MPLS "Tunnels," or VPNs



Numerous approaches are possible to support VPNs within an MPLS environment. In the base MPLS architecture, the label applied to a packet on ingress to the MPLS environment effectively determines the selection of the egress router, as the sequence of label switches defines an edge-to-edge virtual path. The extension to the MPLS local label hop-by-hop architecture is the notion of a per-VPN global identifier (or *Closed User Group* (CUG) identifier, as defined in [5]), which is used effectively within an edge-to-edge context. This global identifier could be assigned on ingress, and is then used as an index into a per-VPN routing table to determine the initial switch label. On egress from the MPLS environment, the CUG identifier would be used again as an index into a per-VPN global identifier table to undertake next-hop selection.

Routing protocols in such an environment need to carry the CUG identifier to trigger per-VPN routing contexts, and a number of suggestions are noted in [5] as to how this could be achieved.

It should be stressed that MPLS itself, as well as the direction of VPN support using MPLS environments, is still within the area of active research, development, and subsequent standardization within the IETF, so this approach to VPN support is still somewhat speculative in nature.

Link-Layer Encryption

As mentioned previously, encryption technologies are extremely effective in providing the segmentation and virtualization required for VPN connectivity, and can be deployed at almost any layer of the protocol stack. Because there are no intrinsically accepted industry standards for link-layer encryption, all link-layer encryption solutions are generally vendor specific and require special encryption hardware.

Although this scenario can avoid the complexities of having to deal with encryption schemes at higher layers of the protocol stack, it can be economically prohibitive, depending on the solution adopted. In vendor proprietary solutions, multivendor interoperability is certainly a genuine concern.

Transport and Application-Layer VPNs

Although VPNs can certainly be implemented at the transport and application layers of the protocol stack, this setup is not very common. The most prevalent method of providing virtualization at these layers is to use encryption services at either layer; for example, encrypted e-mail transactions, or perhaps authenticated *Domain Name System* (DNS) zone transfers between different administrative name servers, as described in DNSSec (*Domain Name System Security*)[10].

Some interesting, and perhaps extremely significant, work is being done in the IETF to define a *Transport Layer Security* (TLS) protocol[11], which would provide privacy and data integrity between two communicating applications. The TLS protocol, when finalized and deployed, would allow applications to communicate in a fashion that is designed to prevent eavesdropping, tampering, or message forgery. It is unknown at this time, however, how long it may be before this work is finalized, or if it will be embraced by the networking community as a whole after the protocol specification is completed.

The significance of a "standard" transport-layer security protocol, however, is that when implemented, it could provide a highly granular method for virtualizing communications in TCP/IP networks, thus making VPNs a pervasive commodity, and native to all desktop computing platforms.

Non-IP VPNs

Although this article has focused on TCP/IP and VPNs, it is recognized that multiprotocol networks may also have requirements for VPNs. Most of the same techniques previously discussed can also be applied to multiprotocol networks, with a few obvious exceptions—many of the techniques described herein are solely and specifically tailored for TCP/IP protocols.

Controlled route leaking is not suitable for a heterogeneous VPN protocol environment, in that it is necessary to support all protocols within the common host network. GRE tunnels, on the other hand, are constructed at the network layer in the TCP/IP protocol stack, but most routable multiprotocol traffic can be transported across GRE tunnels (for example, IPX and AppleTalk). Similarly, the VPDN architectures of L2TP and PPTP both provide a PPP end-to-end transport mechanism that can allow per-VPN protocols to be supported, with the caveat that it is a PPP-supported protocol in the first place.

The reverse of heterogeneous VPN protocol support is also a VPN requirement in some cases, where a single VPN is to be layered above a heterogeneous collection of host networks. The most pervasive method of constructing VPNs in multiprotocol networks is to rely upon application-layer encryption, and the resulting VPNs are generally vendor proprietary, although some would contend that one of the most pervasive examples of this approach was the mainstay of the emergent Internet in the 1970s and 1980s—that of the UNIX-to-UNIX Copy Program (UUCP) network, which was (and remains) an open technology.

Quality-of-Service Considerations

In addition to creating a segregated address environment to allow private communications, the expectation that the VPN environment will be in a position to support a set of service levels also exists. Such per-VPN service levels may be specified either in terms of a defined service level that the VPN can rely upon at all times, or in terms of a level of differentiation that the VPN can draw upon the common platform resource with some level of priority of resource allocation.

Using dedicated leased circuits, a private network can establish fixed resource levels available to it under all conditions. Using a shared switched infrastructure, such as Frame Relay virtual circuits or ATM virtual connections, a quantified service level can be provided to the VPN through the characteristics of the virtual circuits used to implement the VPN.

When the VPN is moved away from such a circuit-based switching environment to that of a general Internet platform, is it possible for the Internet Service Provider to offer the VPN a comparable service level that attempts to quantify (and possibly guarantee) the level of resources that the VPN can draw upon from the underlying host Internet?

This area is evolving rapidly, and much of it remains within the realm of speculation rather than a more concrete discussion about the relative merits of various Internet QoS mechanisms. Efforts within the *Integrated Services Working Group* of the IETF have resulted in a set of specifications for the support of guaranteed and controlled load end-to-end traffic profiles using a mechanism that loads per-flow state into the switching elements of the network[12, 13]. There are numerous caveats regarding the use of these mechanisms, in particular relating to the ability to support the number of flows that will be encountered on the public Internet[14]. Such caveats tend to suggest that these mechanisms will not be the ones that are ultimately adopted to support service levels for VPNs in very large networking environments.

If the scale of the public Internet environment does not readily support the imposition of per-flow state to support guarantees of service levels for VPN traffic flows, the alternative query is whether this environment could support a more relaxed specification of a differentiated service level for overlay VPN traffic. Here, the story appears to offer more potential, given that differentiated service support does not necessarily imply the requirement for per-flow state, so stateless service differentiation mechanisms can be deployed that offer greater levels of support for scaling the differentiated service[15]. However, the precise nature of these differentiated service mechanisms, and their capability to be translated to specific service levels to support overlay VPN traffic flows, still remain in the area of future activity and research.

Conclusions

So what is a virtual private network? As we have discussed, a VPN can take several forms. A VPN can be between two end systems, or it can be between two or more networks. A VPN can be built using tunnels or encryption (at essentially any layer of the protocol stack), or both, or alternatively constructed using MPLS or one of the "virtual router" methods. A VPN can consist of networks connected to a service provider's network by leased lines, Frame Relay, or ATM, or a VPN can consist of dialup subscribers connecting to centralized services or other dialup subscribers.

The pertinent conclusion here is that although a VPN can take many forms, a VPN is built to solve some basic common problems, which can be listed as virtualization of services and segregation of communications to a closed community of interest, while simultaneously exploiting the financial opportunity of economies of scale of the underlying common host communications system.

To borrow a popular networking axiom, "When all you have is a hammer, everything looks like a nail." Every organization has its own problem that it must solve, and each of the tools mentioned in this article can be used to construct a certain type of VPN to address a particular set of functional objectives. More than a single "hammer" is

available to address these problems, and network engineers should be cognizant of the fact that VPNs are an area in which many people use the term generically—there is a broad problem set with equally as many possible solutions. Each solution has numerous strengths and also numerous weaknesses and vulnerabilities. No single mechanism for VPNs that will supplant all others in the months and years to come exists, but instead a diversity of technology choices in this area of VPN support will continue to emerge.

Acknowledgments

Thanks to Yakov Rekhter, Eric Rosen, and W. Mark Townsley, all of Cisco Systems, for their input and constructive criticism.

References

[1] Valencia, A., M. Littlewood, and T. Kolar. "Layer Two Forwarding (Protocol) 'L2F'." **draft-valencia-l2f-00.txt**, work in progress, October 1997.

[2] Droms, R. "Dynamic Host Configuration Protocol." RFC 2131, March 1997.

[3] Kent, S., and R. Atkinson. "Security Architecture for the Internet Protocol." **draft-ietf-ipsec-arch-sec-04.txt**, work in progress, March 1998.

[4] Additional information on IPSec can be found on the IETF IPSec home page, located at **http://www.ietf.org/html.charters/ipsec-charter.html**

[5] Heinanen, J. "Multiprotocol Encapsulation over ATM Adaptation Layer 5." RFC 1483, July 1993.

[6] The ATM Forum. "Multi-Protocol Over ATM Specification v1.0." **af-mpoa-0087.000**, July 1997.

[7] Callon, R., P. Doolan, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan. "A Framework for Multiprotocol Label Switching." **draft-ietf-mpls-framework-02.txt**, work in progress, November 1997.

[8] Rosen, E., A. Viswanathan, and R. Callon. "A Proposed Architecture for MPLS." **draft-ietf-mpls-arch-01.txt**, work in progress, March 1998.

[9] Heinanen, J. and E. Rosen. "VPN Support for MPLS." **draft-heinanen-mpls-vpn-01.txt**, work in progress, March 1998.

[10] Eastlake, D. and C. Kaufman. "Domain Name System Security Extensions." RFC 2065, January 1997. For further information regarding DNSSec, see: **http://www.ietf.org/html.charters/dnssec-charter.html**

[11] Dierks, T. and C. Allen. "The TLS Protocol—Version 1.0." **draft-ietf-tls-protocol-05.txt**, work in progress, November 1997. For more information on the IETF TLS working group, see **http://www.ietf.org/html.charters/tls-charter.html.** See also the article on SSL in the *Internet Protocol Journal*, Volume 1, No. 1, June 1998.

[12] Wroclawski, J. "Specification of the Controlled-Load Network Element Service." RFC 2211, September 1997.

[13] Shenker, S., C. Partridge, and R. Guerin. "Specification of Guaranteed Quality of Service." RFC 2212, September 1997.

[14] Mankin, A., F. Baker, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang. "Resource ReSerVation Protocol (RSVP) Version 1—Applicability Statement, Some Guidelines on Deployment." RFC 2208, September 1997.

[15] "Differentiated Services Operational Model and Definitions." **draft-nichols-dsopdef-00.txt**, work in progress, K. Nichols and S. Blake (editors), February 1998.

PAUL FERGUSON is a consulting engineer at Cisco Systems and an active participant in the Internet Engineering Task Force (IETF). His principal areas of expertise include large-scale network architecture and design, global routing, Quality of Service (QoS) issues, and Internet Service Providers. Prior to his current position at Cisco Systems, he worked in network engineering, analytical, and consulting capacities for Sprint, Computer Sciences Corporation (CSC), and NASA. He is coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, published by John Wiley & Sons, ISBN 0-471-24358-2, a collaboration with Geoff Huston. E-mail: **ferguson@cisco.com**

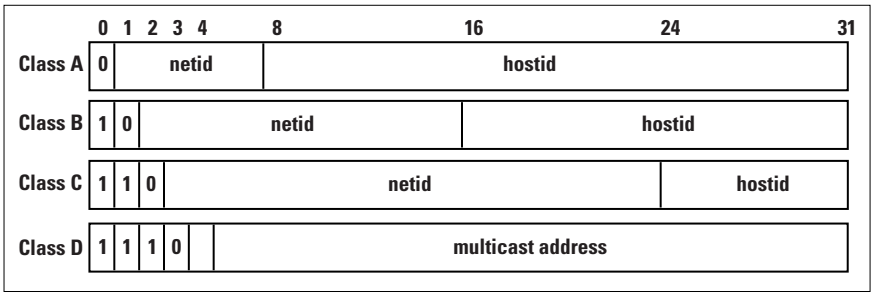
GEOFF HUSTON holds a B.Sc and a M.Sc from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and was an inaugural member of the Internet Society Board of Trustees. He is coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, published by John Wiley & Sons, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. E-mail: **gih@telstra.net**

Reliable Multicast Protocols and Applications

by C. Kenneth Miller, StarBurst Communications

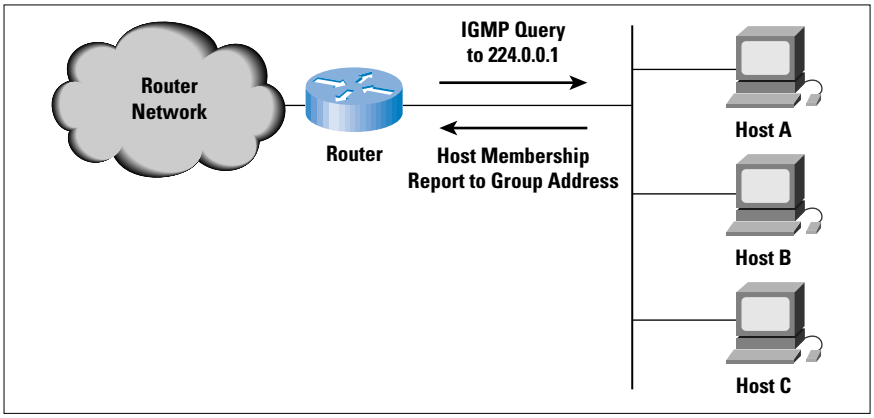
Multicast IP network services offer new opportunities to provide value-added applications that involve many-to-many transmission such as conferencing or network gaming, or one-to-many transmission such as multimedia events, tickertape feeds, and file transfer, where the many could be thousands or even conceivably millions. Multicast IP services use a different kind of IP address, called Class D. In contrast to individual host addresses (Classes A–C), which include a host and a network component and usually are semipermanent, Class D multicast addresses may by design be used only for a particular session, or can be semipermanent, as multicast groups may be set up and torn down relatively quickly, on the order of seconds. The IP address structure is shown in Figure 1.

Figure 1: IP Address Types



Hosts join groups at the receiver's initiation using the *Internet Group Management Protocol* (IGMP). When a host joins a group, it notifies the nearest multicast subnet router of its presence in the group, as shown in Figure 2. First defined in RFC 1112[1], IGMPv1 is still the version of IGMP most widely supported. IGMPv2 has recently been documented as an official RFC (RFC 2236[2]). The main feature that IGMPv2 brings is reduced latency for leaving groups. In IGMPv1, the designated multicast router for the subnet polls for multicast group members; no response between polls indicates that all hosts in a particular multicast group have left the group, and that the routers can prune back the multicast routing tree.

Figure 2: IGMPv1 Dialog



Network infrastructure devices, for example, routers, need to provide a routing protocol to forward multicast packets to group members, in a fashion similar to that performed for unicast routing. Multicast IP packet forwarding is best effort, just as it is with unicast packet forwarding. However, most unicast applications use TCP as a transport layer to provide guaranteed packet ordering and delivery. Some examples of applications that use TCP are the *File Transfer Protocol* (FTP) for file transfer and the *Hypertext Transfer Protocol* (HTTP) for Web access.

However, TCP is a unicast (point-to-point) only transport protocol. Thus, all multicast applications must run on top of the *User Datagram Protocol* (UDP) or alternatively, interface directly to IP via "raw" sockets and provide their own customized transport layer, as shown in Figure 3. UDP provides only minimal transport-layer services, error detection, and port multiplexing. Thus, if any errors or packet loss due to congestion occur, packets are simply lost to the application, and they are not recoverable. Thus, *all multicast applications must have a specific transport-layer service to support that particular application*. When that transport layer operates over UDP, it operates in the application layer with the application. When it interfaces directly to IP using "raw" sockets, the specialized transport layer operates at the transport layer, but is specialized to the particular application that uses it.

It should be noted that TCP supports only data reliability; it is not suited for transport of multimedia streams, which require consistent time delivery at the receiver and only need to be semireliable. Thus, multimedia streaming applications need a specialized transport layer such as the *Real-Time Transport Protocol* (RTP)[3] for unicast as well as multicast transmissions.

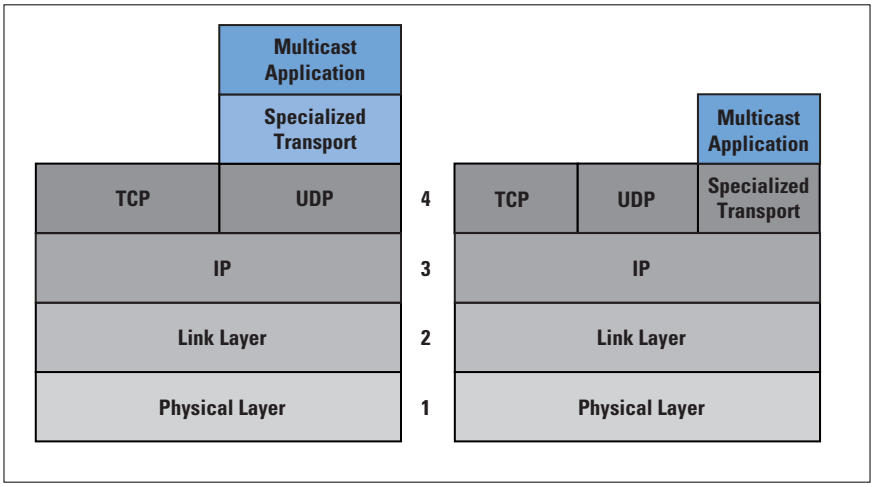


Figure 3: Specialized Multicast Transport Protocols Operate over UDP or IP

Many equate multicast with multimedia, thinking that the Internet and private intranets will become an alternative entertainment media to television by using multicast IP network services and multimedia streaming technology. However, numerous other multicast applications require reliability rather than timeliness; they are multicast applications that are similar to those unicast applications that operate over TCP, except that delivery is to many recipients rather than just one.

Reliable Multicast Application Categories and Requirements

Reliable multicast applications come in three basic categories with differing requirements, as shown in Figure 4.

Figure 4: Reliable Multicast Application Categories

Application Type	Latency Req.	Reliability	Scalability
Collaborative	Low	Semi/Strict	<100
Message Str.	Low/Medium	Semi/Strict	to Millions
Bulk Data	Not Real Time	Strict	to Millions

Collaborative applications such as data conferences (whiteboarding) and network-based games are many-to-many applications with modest scaling requirements of less than 100 participants. This kind of application requires low latency of less than 400 msec so that responses do not cause discomfort to the human participants. Transmission does not always need strict reliability; for example, refresh of background information for a network game could wait for the next refresh.

Message streaming applications such as tickertape and news feeds also often require low latency. Tickertape feeds to brokerage houses need to be very timely because the information loses value greatly with time. Time is very much money in this application, and there is also a need for strict reliability.

Tickertape feeds to consumers are purposely delayed by minutes because they are usually transmitted without charge, but they cannot be so stale as to be viewed as "old" information. This data does not have a strict reliability requirement because the next trade of a particular security refreshes the data. News feeds likewise have only a moderate latency requirement. If the news feeds are sent in a carousel fashion, that is, each news story is repeated, strict reliability may not be needed because it is refreshed in the next transmission of the same story.

Bulk data delivery has no specific latency requirement. Often there is a desire to schedule delivery during the night, when there is less network traffic. At other times, the desire is to receive the data almost

immediately. However, at all times the entire "file" or piece of data needs to be received to be complete. Strict reliability is the rule; for example, if any bit of a software image is lost, the data is worthless.

Message streaming and bulk data application scaling requirements span the gamut from tens to possibly even millions.

Reliable multicast transport protocols, in contrast to multimedia streaming transport protocols, have not yet been standardized. However, numerous reliable multicast protocols exist; some have been used only for research, while others have been commercialized.

The *Reliable Multicast Research Group* (RMRG) in the *Internet Research Task Force* (IRTF) is now studying reliable multicast. It is chartered to recommend techniques for a working group in the *Internet Engineering Task Force* (IETF) to create a set of reliable multicast standards.

Standardization Effort

The standardization effort has been started in an IRTF research group to study the problems and possible solutions by Internet researchers. This effort was first placed in the hands of researchers because the problems were considered very difficult to solve in the global Internet. Some of the concerns about reliable multicast were discussed in an expired Internet Draft published in November 1996 by the Transport Area Directors of IETF.

These concerns formed the basis for the work of the RMRG, which was formed in early 1997. The concerns from that document follow:

"A particular concern for the IETF (and a dominant concern for the Transport Services Area) is the impact of reliable multicast traffic on other traffic in the Internet in times of congestion (more specifically, the effect of reliable multicast traffic on competing TCP traffic). The success of the Internet relies on the fact that best-effort traffic responds to congestion on a link (as currently indicated by packet drops) by reducing the load presented on that link. Congestion collapse in today's Internet is prevented only by the congestion control mechanism in TCP.

There are a number of reasons to be particularly attentive to the congestion-related issues raised by reliable multicast proposals. Multicast applications in general have the potential to do more congestion-related damage to the Internet than do unicast applications. This is because a single multicast flow can be distributed along a large, global multicast tree reaching throughout the entire Internet.

> Further, reliable multicast applications have the potential to do more congestion-related damage than do unreliable multicast applications. First, unreliable multicast applications such as audio and video are, at the moment, usually accompanied by a person at the receiving end, and people typically unsubscribe from a multicast group if congestion is so heavy that the audio or video stream is unintelligible. Reliable multicast applications such as group file transfer applications, on the other hand, are likely to be between computers, with no humans in attendance monitoring congestion levels.
>
> In addition, reliable multicast applications do not necessarily have the natural time limitations typical of current unreliable multicast applications. For a file transfer application, for example, the data transfer might continue until all of the data is transferred to all of the intended receivers, resulting in a potentially-unlimited duration for an individual flow. Reliable multicast applications also have to contend with a potential explosion of control traffic (e.g., ACKs, NAKs, status messages), and with control traffic issues in general that may be more complex than for unreliable multicast traffic.
>
> The design of congestion control mechanisms for reliable multicast for large multicast groups is currently an area of active research. The challenge to the IETF is to encourage research and implementations of reliable multicast, and to enable the needs of applications for reliable multicast to be met as expeditiously as possible, while at the same time protecting the Internet from the congestion disaster or collapse that could result from the widespread use of applications with inappropriate reliable multicast mechanisms. Because of the setbacks and costs that could result from the widespread deployment of reliable multicast with inadequate congestion control, the IETF must exercise care in the standardization of a reliable multicast protocol that might see widespread use."

One of the statements in this document is very specious:

> "First, unreliable multicast applications such as audio and video are, at the moment, usually accompanied by a person at the receiving end, and people typically unsubscribe from a multicast group if congestion is so heavy that the audio or video stream is unintelligible. Reliable multicast applications such as group file transfer applications, on the other hand, are likely to be between computers, with no humans in attendance monitoring congestion levels."

This statement is a very weak argument; it is not reliable to depend on a human to turn off a nonfunctioning event. Do we typically turn off the television when we leave the house? Or leave the room to do something else?

In contrast, some of the reliable multicast protocols such as the *Multicast File Transfer Protocol* (MFTP) have the sense of a finite session, and automatically time out and leave a group, even if all group members did not receive all the content.

Essentially what is desired is a reliable multicast protocol that behaves like TCP in that it backs off in the face of congestion approximately the same way as TCP and shares the bandwidth with TCP traffic "fairly." This feature is of prime importance to Internet researchers who wish to specify protocols that can scale to the global Internet and not cause harm to the traffic already present.

Two additional significant problems need to be solved: scalability and the ability to operate with scalability over many different network infrastructures.

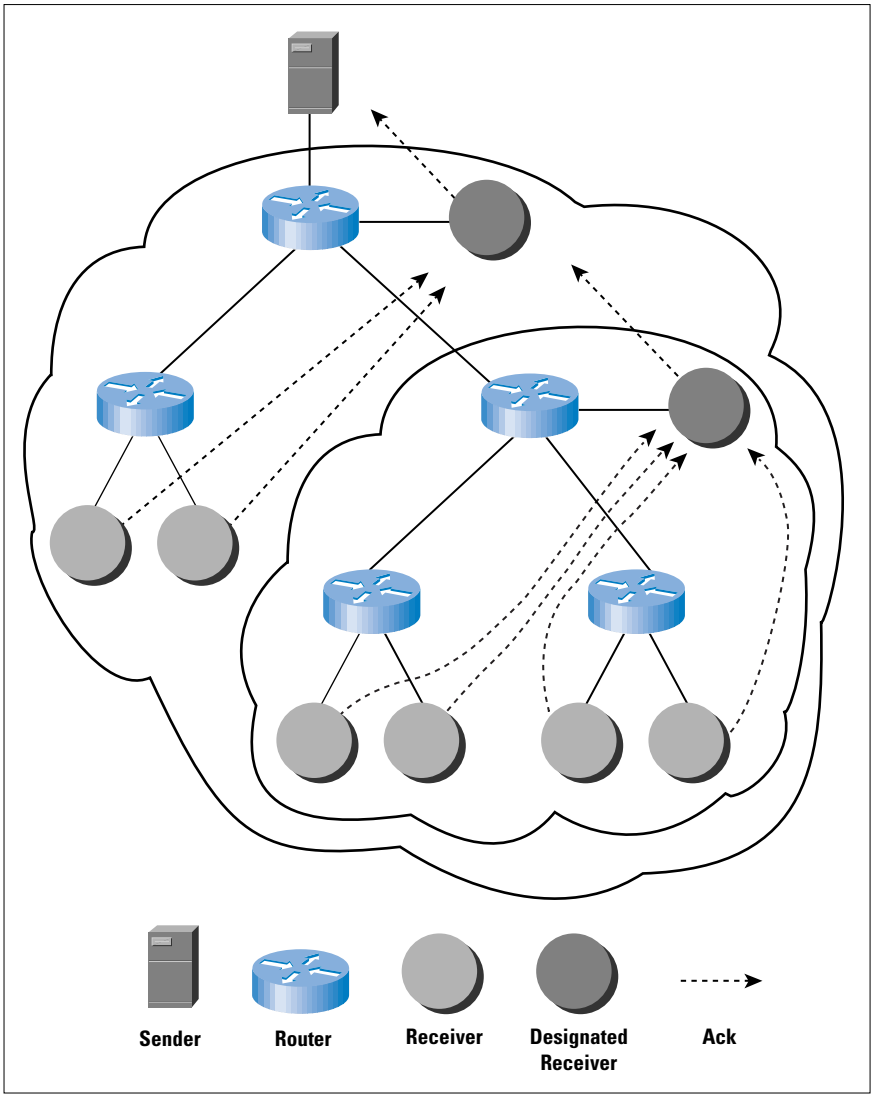
Scaling Issues and How Current Reliable Multicast Protocols Solve Them

Two primary issues are related to scaling, that is, the ability to handle large groups. The first and most significant is widely known as acknowledgment/negative acknowledgment (ACK/NAK) implosion. As the number of receivers grows, the amount of back traffic to the sender eventually overwhelms its capacity to handle them. Additionally, the network at the sender site becomes congested from the cumulative back traffic from the receivers.

The second issue is one of retransmissions (often referred to as "repairs"). If the packet loss is uncorrelated at the receivers, retransmissions grow, so the data may need to be sent multiple times to satisfy all the receivers. Measurements of the *Multicast backbone* (Mbone) have shown that loss consists of both correlated and uncorrelated parts[4]. Satellite networks will also exhibit mostly uncorrelated loss, unless receivers are geographically close.

Various methods have been used to achieve scaling by reducing the amount of ACK/NAK administrative traffic while still retaining reliability. A straightforward approach is to simply deploy repeaters/aggregators in the network, as shown in Figure 5. This approach is provided by the *Reliable Multicast Transport Protocol* (RMTP)[5]. RMTP provides for *designated receivers* (DRs) that collect status messages from nodes in a local RMTP domain and provide repairs (retransmissions of missing data), if available. Receivers direct the administrative messages to the DR by unicast. Thus, the DR provides both local recovery and consolidation of control traffic to the next DR in the hierarchy if the data requested is not available.

Figure 5:
RMTP Designated
Receivers



A second approach is to allow any receiver to provide the repair, biasing the request to the nearest receiver that has the requested data. This approach, called *Scalable Reliable Multicast* (SRM)[6], depends on the concept of repair by any receiver that has the data to gain scalability in reducing administrative back traffic to the source, putting the onus of responsibility on receivers to ensure that they get missed data.

Group members in SRM send low-frequency *session* messages to the group so that their neighbors can learn their status, measure the delay among group members and learn group membership, and detect the last packet in a burst. Session messages are designed to take only about five percent of the traffic in the session.

Receivers with missing data wait a random time period before issuing repair requests, allowing suppression of duplicate requests similar to the mechanism that IGMP uses on its subnet. A similar process occurs for making the actual repairs. The random backoff time for both repair requests made by receivers and repairs made by senders is a

function of "closeness" to the sender and requesting receiver. Thus, those closest to each other time out first and make the repair request or the actual repair in an attempt to keep repairs as local as possible. A receiver that sees the first request and determines that it is the same request that it would have made simply stays silent, reducing potential redundant requests. The requester continues to send repair requests until the repair is received.

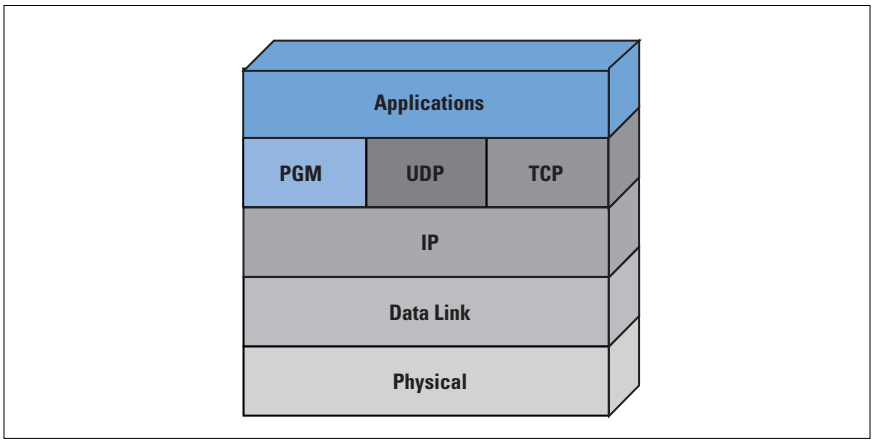
Any receiver may satisfy the repair request, because all receivers are required to cache previously sent data. Any receiver that can satisfy the request is prepared to do so; a random backoff timer is used before a repair is sent, and if it sees the repair being sent by another group member, it stays silent to reduce the probability of sending duplicate repairs.

SRM was first developed to be the reliable multicast protocol to operate with the *wb* whiteboard data conferencing tool developed by Lawrence Berkeley Labs (LBL) researchers, SRM is currently operational over the Mbone, the experimental multicast network of the Internet.

A third approach is to have the network infrastructure, that is, routers, help in providing scaling. This approach, called *Pretty Good Multicast* (PGM)[7], is a new proposal that was first publicly presented to the RMRG meeting held in February 1998.

One design goal of the creators of PGM was simplicity and the ability to optimally leverage routers in the network to provide scalability. PGM is an example of a protocol that bypasses UDP and interfaces directly to IP via "raw" sockets, as shown in Figure 6.

Figure 6: PGM Interfaces Directly to IP



PGM provides no notion of group membership; it simply provides reliability within a source's transmit window from the time a receiver joins a group until it departs.

PGM has only a few data packets that are defined:

ODATA: original content data

NAK: selective negative acknowledgment

NCF: NAK confirmation

RDATA: retransmission (repair)

SPM: source path message

Each PGM packet contains a *Transport Session Identifier* (TSI) to identify the session and source of that data, so multiple sessions may be easily identified by PGM-aware routers and receivers. ODATA, NCF, RDATA, and SPM packets flow downstream in the distribution tree, and NAK packets flow upstream toward the source.

PGM is designed for scalability as well as the ability to serve real-time applications. Thus there is a need for timeliness. This need is handled by the *transmit window*, which defines a sliding window of data such that if no NAKs are received by the sender or a designated local retransmitter by the time the window is up, the data is simply not available for repairs.

PGM is totally NAK based, so the scaling issue is to reduce the number of NAKs sent back to the source, while at the same time protecting against lost NAKs. Enter here the router assist, as shown in Figure 7.

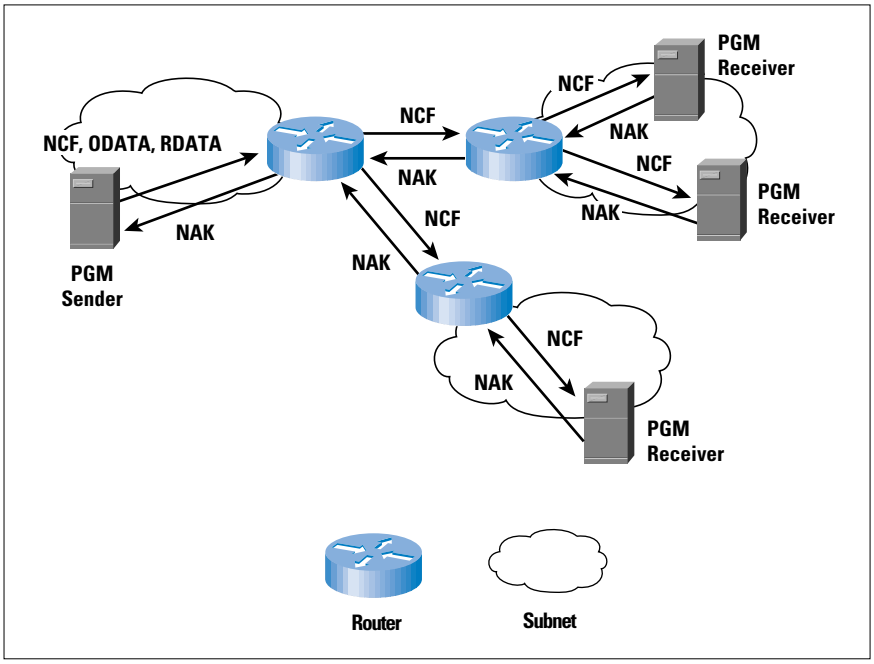


Figure 7: PGM NAK/NCF Dialog

NAKs are unicast from PGM-router to PGM-router, initiated by the receiver that lost data sending a NAK to its nearest PGM-aware router. Each PGM-aware router keeps forwarding NAKs until it sees an NCF or RDATA, which indicates that a repair is being sent. NAK *suppres-*

sion is provided by a receiver's subnet PGM-aware router, and all PGM-aware routers *eliminate* duplicate NAKs all the way upstream to the source.

The unicast path back to the source must be the same path as the downstream multicast tree. SPMs are sent downstream interleaved with ODATA packets to establish a source path state for a given source and session. PGM-aware routers use this information to determine the unicast path back to the source for forwarding NAKs. SPMs also alert receivers that the oldest data in the transmit window is about to be retired from the window and will thus no longer be available for repairs from the source. SPMs are sent by a source at a rate that is at least the rate at which the transmit window is advanced. This rate provokes "last call" NAKs from receivers and updates the receive window state at receivers.

PGM-aware routers also keep state on where the NAKs come from in the distribution tree so that they may constrain the forwarding of RDATA repairs to only those ports from which NAKs requesting that repair were received. This scenario eliminates the transmission of repair data to parts of the distribution tree where the repair is not needed.

The PGM feature can also optionally redirect NAKs to a *designated local retransmitter* (DLR) rather than the source. A DLR announces its presence to provoke the redirection of NAKs for that session and source.

A fourth approach is to not have a low-latency requirement (that is, only serve "bulk data" delivery applications) and use this feature to advantage to gain scalability. MFTP was first published as an Internet Draft in February 1997, and an update was submitted in April 1998[8].

MFTP also has a provision for sender-based group creation, with different group models, and the group setup protocol to notify receivers to join the group. Group creation is discussed later in this article.

The basic MFTP protocol breaks the data entity to be sent into maximum size "blocks," where a block by default consists of thousands or tens of thousands of packets, depending on packet size used. This setup is shown in Figure 8.

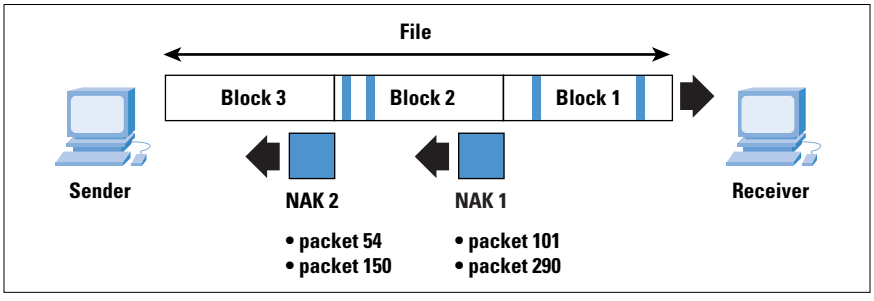


Figure 8: MFTP Blocks

MFTP is a "NAK-only" protocol; that is, if data is received correctly in a block, nothing is sent back to the sender. If one or more packets are in error or missing in a block, receivers respond with a NAK that consists of a bit map of the bad packets in the block. It is thus a *selective reject* mechanism. In this respect, MFTP is similar to RMTP; the main difference is that MFTP explicitly attempts to make the block as large as possible for scaling purposes.

NAKs are normally sent unicast back to the source, unless aggregation to improve scaling using enabled network routers is used. In this case, the NAKs are sent multicast to a special administrative traffic group address.

MFTP does not repair after each block, however; it takes advantage of the non-real time nature of the application for benefit. The data entity, such as a file, is sent initially in its entirety in a *first pass*. The sender collects the NAK packets for a block from all the receivers. One NAK packet from a receiver can represent thousands or even tens of thousands of bad packets, reducing NAK implosion by orders of magnitudes. The collection of NAKs received by the sender from all the receivers is logically OR-ed together to represent the collective need for repairs for the receiving group. These repairs are sent by the sender in a *second pass* to the group. If certain receivers already have the repair, it is simply ignored. This scenario is repeated, if necessary, until all repairs are received by all receivers or until a configurable timeout occurs.

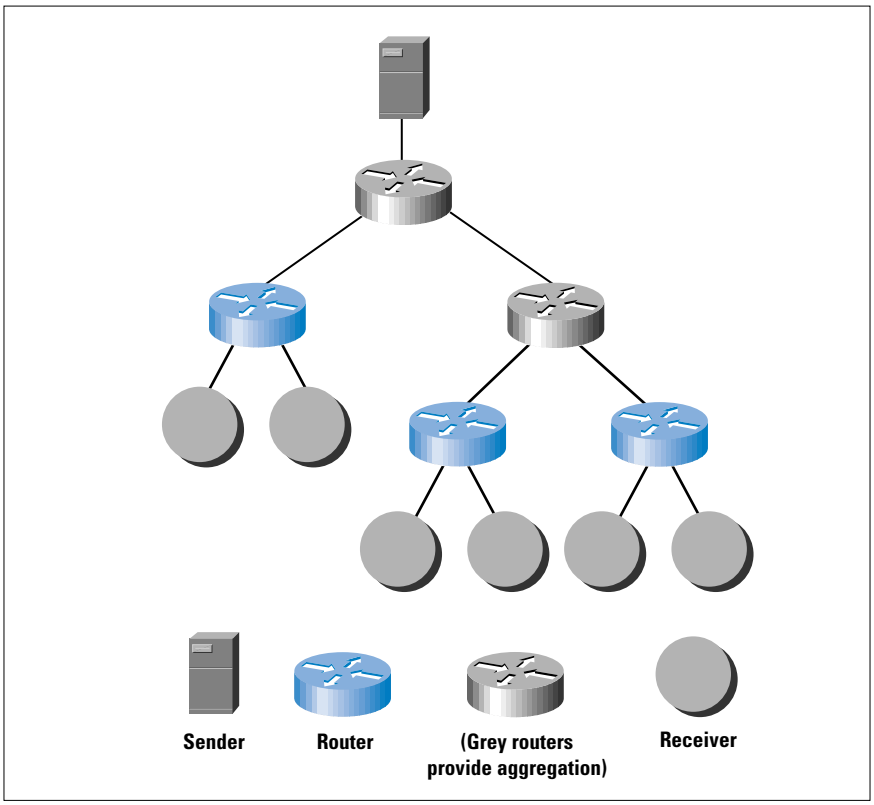
Thus, packet ordering services are not provided, and holes in the data caused by dropped packets or packets in error are filled in as they are received.

The sender is *rate based;* in other words, it transmits at a data rate set by the operator to be less than or equal to what the network can handle. The protocol is thus very efficient with high-latency networks such as satellites, and it is impervious to network asymmetry. It also attempts to be as scalable as possible on one-hop networks such as satellite networks, and it provides for extensions so that network elements may aggregate downstream responses to increase scalability further, depending on the network configuration.

This aggregation capability is shown in Figure 9. The network element, which can be a router, collects MFTP administrative back traffic routers are members. These routers aggregate back traffic from all nodes downstream in the multicast tree from the source, including registrations, NAKs, and dones. Registration and done messages are used by MFTP's group setup protocol, and they are described later in this article.

Depending on the network configuration, this aggregation capability can further improve the scalability of MFTP by orders of magnitudes.

Figure 9: Routers as Network Aggregators



The upper limit to scalability with no network aggregation of administrative traffic is in the tens of thousands of receivers. For example, for a *Maximum Transmission Unit* (MTU) of 1500 bytes (the Ethernet maximum), the default block size is over 11,000 packets. If the number of receivers is 10,000 and each receiver has at least one bad packet per block, then there will be a total of 10,000 NAK packets coming back to the sender from the group about that block, approximately the same number of packets as were sent in the forward direction in that block. MFTP provides for a NAK backoff timer to spread the NAKs out in time to the sender to avoid bursts. If the bandwidth is symmetric at the sender, the sender should be able to handle this maximum NAK. In many situations, the amount of back traffic could exceed forward traffic.

MFTP also has provision for a crude congestion control mechanism. The sender at the beginning of a session sends *announce* messages. These messages are used for many functions, including the setting up of groups. Additionally, it conveys a packet loss parameter to all receivers. This packet loss threshold parameter may be used by receivers to leave the group if the packet loss exceeds the threshold. Leaving the group prunes the distribution tree, relieving the congestion in that section of the tree.

Commercial Usage

The reliable multicast protocols previously discussed are the most prominent ones on the market today. RMTP has been deployed in its message streaming version for a billing record distribution application within a very large telecommunications carrier, but it has had generally limited deployment. It also does not scale over satellite networks, where most of the early multicast deployments reside.

SRM has been used by the research community only over the Mbone, and it is still being refined. Another problem with SRM is that in its current incarnation, it supports neither asymmetric nor satellite networks. Some early Internet Service Provider (ISP) multicast implementations, offer multicast support in only one direction; SRM requires total multicast support.

PGM is new and offers promise, but there is no deployment yet, and it likely will not occur until early 1999. PGM also requires router support in a terrestrial land-line network to gain scaling.

MFTP has the limitation that it supports only bulk transfer applications. However, one trade-off is that it can support all network infrastructures, including satellite infrastructures with scaling. MFTP has also been available commercially in products with the longest application support, dating back to 1995. Thus, MFTP-based products have the largest installed base of any reliable multicast-based product being used over WANs. The largest commercial installation of over 8,500 remote sites in the group is the General Motors[9] dealer network. Several other commercial installations of MFTP-based applications number over 1,000 group members.

Advanced Research Topics Discussed in Reliable Multicast Research Group

A promising technique to reduce the amount of repair data that needs to be retransmitted is called *erasure correction*. This technique can significantly reduce the amount of repairs that need to be resent if the packet loss is largely uncorrelated at the receivers. It uses a *forward error correction* (FEC) code to generate parity packets to be used for repairs only. This setup provides benefit if errors at receivers are uncorrelated. For example, suppose 16 receivers each have one missing packet, but they are all different. Rather than send all 16 original data packets, one FEC packet could be sent that could correct the one missing packet at all 16 receivers, requiring retransmission of only one packet rather than 16.

If the loss is correlated, then many of the receivers lose the same data, and erasure correction is of no benefit. However, there is also no penalty, except for the need for computing power at both the sender and the receivers to perform the FEC correction calculations. Simulations have show[10] that there is a greater than 2:1 reduction in the number of repairs needed to be sent with our example of 10,000 receivers. This benefit will be even larger when group sizes become larger than tens of thousands.

Perhaps a more significant application for FEC is a congestion control technique known as *layering*[11,12]. With layering, numerous groups are set up by the sender, all with different rates. Receivers that can receive at the highest rate join all the "layer" groups. Those receivers that cannot receive at the highest rate simply leave "layers" until congestion is relieved, and they take longer to receive the data. For this to work without sending data redundantly, the number of parity packets created must be very large compared to the number of data packets.

There are some further issues that have been pointed out by the researchers with the Other issues with the layering approaches have been pointed out by the researchers, however. For layering to be effective, the routing tree should be identical for the different groups; otherwise congestion will not be relieved on a part of the tree. This may not always be the case, especially in sparse mode routing protocols, where selection of the rendezvous point or core is based on group address.

Even if the same distribution tree is used for the different layers, it has been pointed out[12] that leaves of hosts downstream from a congested link should be coordinated; otherwise the action of less than all of them has no effect on congestion. Additionally, a receiver could cause congestion by adding a layer that another receiver could interpret as congestion, causing it to drop a layer with no effect.

Thus, layering using FEC techniques is an interesting technique that shows promise for use in congestion control. However, there are issues associated with this type of layering that researchers still need to address.

Another technique that has been proposed for congestion control is bulk feedback to the sender[13]. If the sender receives an excessive number of NAKs from receivers, it drops the sender's transmission rate with an algorithm that attempts to emulate the behavior of TCP. This approach is an obvious one because it is an extension of the process in which TCP falls back in the face of congestion.

This approach, however, has two basic problems. The first is that there is delay, because the sender needs to get feedback from the multitude of receivers before it acts. This delay can be considerably longer than in the case of TCP, which needs feedback from only one receiver.

The second flaw is that one errant receiver can effectively penalize the whole group, because the sender reduces the rate to the total group.

This approach is not viewed as a viable solution for these reasons. In fact, the general consensus is that congestion control decision making will be required at the multiple receivers rather than at the sender for both scaling and timeliness reasons.

Another idea that is now receiving intense study by researchers is that of "subcasting"[14,15,16]. The key idea in subcasting is to optimize local repair to be a retransmitter that may be just above a link congestion point, as shown in Figure 10. The problem is to gain knowledge of the network topology so as to locate a receiving host that is willing to retransmit and that has the repair data.

Then the repairs need to be contained within only the region of the network that lost the original transmission, that is, the "subcast" region.

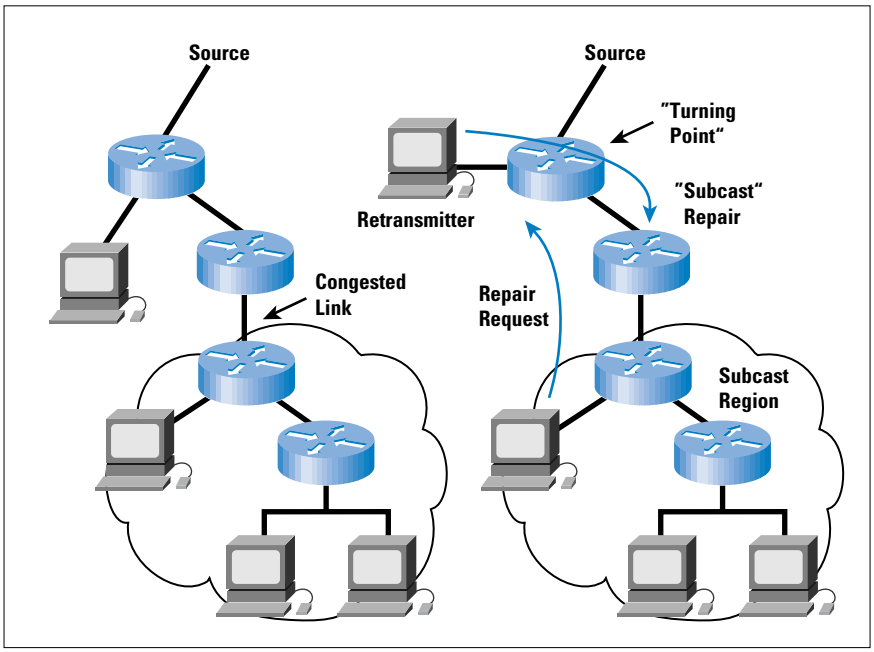


Figure 10: Optimized Local Repair

One proposal is to ask for assistance from the network routers. They know the topology and could be used to find the closest willing retransmitter that has the repair. The router could also direct the repair to only the affected region: the *subcast*.

This technique can be viewed as an extension of concepts originally proposed in SRM to provide local recovery. It assumes that most loss is caused by congested links, and that uncorrelated loss is caused by a series of mildly congested links with few group members. This model is probably the right one for many land-line routed networks; it is problematical with other network infrastructures.

Nevertheless, it is an interesting proposal that merits further research effort. Local repair is destined to be an important tool to meet the goal of improved scalability with minimal traffic overhead.

Group Creation and Destruction

The process of joining a group and leaving a group in IP multicast is left to a potential group member that uses IGMP to notify the nearest multicast router of its membership state. However, mechanisms need to be in place to allow potential members of a group to gain the information needed to decide to join the group.

There are two basic ways to accomplish this scenario for one-to-many sessions. The first and most common is the "broadcast TV" model. The *Multiparty Multimedia Session Control* (MMUSIC) working group of the IETF has developed some protocols that can be used to advertise content. The *Session Announcement Protocol* (SAP)[17] provides the mechanism to send a stream on a "well-known" multicast address to announce content to any potential listeners who may be interested. It uses the *Session Description Protocol* (SDP)[18] to describe the contents that are announced. These two protocols together have been used to create a session directory tool that is available on the Mbone. This setup creates essentially the equivalent of a "preview channel" such as is often available on cable television systems.

SDP is also used to post content on Web sites, which advertise that content to anyone who wishes to receive it.

Although these protocols were originally developed primarily to advertise multimedia streaming applications, they are also applicable for data. They provide a useful tool for "push" vendors to advertise multicast "channels" based on content that any consumer can "tune in" to.

Internet researchers describe this model as providing "loosely coupled" sessions, because the sender does not know who is listening, much like radio or TV broadcasters do not know who tunes in to their stations.

MFTP also includes a group setup protocol. The "closed group" option in MFTP provides a mechanism to create a "tightly coupled" session that is very useful to organizations that wish to deliver critical information from a central site to many remote branch offices. The closed group provides a means for the sender to define a group list centrally and direct those members so defined to join the group. This scenario is somewhat similar to e-mail, except more robust.

These instructions are sent in an "announce" message on a special multicast group address that the superset of possible candidate receivers always listens to. Hosts so directed to join the group notify their designated multicast router of their membership directed to join the group notify their designated multicast router of their membership using IGMP and "register" back to the sender of their presence. Thus, the sender knows group membership before transmission commences, and the sender can then also positively confirm delivery.

This approach has proven very desirable for organizations that have many branches where information is desired to be sent at the discretion and time determined by the and desire to send information at the discretion and time determined by the sender, and usually the information is delivered to a branch office server. Several deployments of applications that use MFTP and the closed group model with group members approaching 10,000 exist.

The MMUSIC group has also created the *Session Invitation Protocol* (SIP)[19], which is used to invite members to a conference of some sort, including possibly a data conference. This protocol is appropriate for use with whiteboard applications, for example.

Summary and Conclusions

Although multicast has often been viewed as synonymous with multimedia, there is a wide spectrum of reliable multicast applications that involve the transfer of data to multiple group members. Because this wide spectrum of applications has many different requirements, as shown in Figure 4, no one reliable multicast protocol can handle all applications and network infrastructures. The result is that numerous reliable multicast protocols are likely to become standardized, and today numerous reliable multicast protocols are either in commercial products/toolkits or due to be available soon.

The reliable multicast standardization effort now resides in the IRTF, because Internet researchers are concerned about congestion control and fairness to TCP for any protocols that might become standardized for general Internet use. This problem is difficult to solve, given the disparate requirements placed on protocols by the wide variety of applications and different network infrastructures.

Nevertheless, a significant number of reliable multicast-based product deployments have already occurred over private networks. These have been shown to save organizations much money and to help create new business opportunities for them.

Stay tuned; reliable multicast-based applications are ready to be mainstreamed. Together with multimedia multicast applications, multicast applications of all forms will become common soon, first in private intranets and extranets and then in the Internet as a whole.

References

[1] Deering, S. "Host Extensions for IP Multicasting." RFC 1112, August 1989.

[2] Fenner, W. "Internet Group Management Protocol, Version 2." RFC 2236, November 1997.

[3] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. "RTP: A Transport Protocol for Real-Time Applications." RFC 1889, January 1996.

[4] Handley, M. "An Examination of Mbone Performance." ISI Report, January 10, 1997.

[5] Paul, S., Sabnani, K. K., Lin, J. C., and Bhattacharyya, S. "Reliable Multicast Transport Protocol (RMTP)." IEEE *Journal on Selected Areas in Communications*, April 1997.

[6] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing." ACM *Transactions on Networking*, November 1996.

[7] Farinacci, D., Lin, A., Speakman, T., and Tweedly, A. "PGM Reliable Transport Protocol Specification." Work in progress, Internet Draft, **`draft-speakman-pgm-spec-01.txt`**, January 29, 1998.

[8] Miller, K., Robertson, K., Tweedly, A., and White, M. "StarBurst Multicast File Transfer Protocol (MFTP) Specification." Work in progress, Internet Draft, **`draft-miller-mftp-spec-03.txt`**, April 1998.

[9] Miller, K. "Reliable Multicast Protocols: A Practical View." 22nd Conference on Local Computer Networks, November 1997.

[10] Kasera, S. K., Kurose, J., Towsley, D., "Scalable Reliable Multicast Using Multiple Multicast Groups." CMPSCI Technical Report TR 96-73, October 1996.

[11] Nonnenmacher, J., and Biersack, E. W. "Asynchronous Multicast Push: AMP." Proceedings of ICCC '97 International Conference on Computer Communications, Cannes, France, November 1997.

[12] Crowcroft, J., Rizzo, L., and Vicisano, L. "TCP-Like Congestion Control for Layered Multicast Data Transfer." Submitted to INFOCOM '98, August 1997.

[13] Sano, T., Yamanouchi, N., et al. "Flow and Congestion Control for Bulk Reliable Multicast Protocols—toward coexistence with TCP." Submitted to INFOCOM '98, presented at RMRG meeting in Cannes, France, September 1997.

[14] Hofmann, M. "Enabling Group Communication in Global Networks." Proceedings of Global Networking '97, June 1997.

[15] Papadopoulos, C., Parulkar, G., and Varghese, G. "An Error Control Scheme for Large-Scale Multicast Applications." Submitted to INFOCOM '98, presented at RMRG meeting in Cannes, France, September 1997.

[16] Levine, B. N., Paul, S., and Garcia-Luna-Aceves, J. J. "Deterministic Organization of Multicast Receivers Based on Packet Loss Correlation." Presented at RMRG meeting in Orlando, Fla., February 1998, submitted for publication.

[17] Handley, M. "SAP: Session Announcement Protocol." Work in progress, Internet Draft, **draft-ietf-mmusic-sap-00.txt**, November 1996.

[18] Handley, M., and Jacobson, V. "SDP: Session Description Protocol." Work in progress, Internet Draft, **draft-ietf-mmusic-sdp-07.txt**, April 1998.

[19] Handley, M., Schulzrinne, H., and Schooler, E. "SIP: Session Invitation Protocol." Work in progress, Internet Draft, **draft-ietf-mmusic-sip-04.txt**, November 1997.

(This article is based in part on material in the book *Multicast Networking and Applications* written by C. Kenneth Miller to be published by Addison Wesley Longman, Inc. in 1998.
ISBN 0-201-30979-3. Used with permission.)

C. KENNETH MILLER is the founder, Chairman, and Chief Technology Officer of StarBurst Communications. StarBurst Communications provides reliable multicast solutions for commercial applications with such corporate customers as GM, Ford, Chrysler, Toys 'R Us, Thomson Financial, and many others. Miller has been in the data communications industry since 1972. He founded Concord Data Systems in late 1980 and served as its President and CEO until 1986. Concord Data Systems produced high-speed dial modems. He was the author of the IEEE 802.4 LAN standard, which became the lower layer for the Manufacturing Automation Protocol (MAP) factory LAN standard. Miller was a regular columnist in *Data Communications Magazine* from 1992 to 1994. He has also published numerous articles and participated in many panels at trade show and other industry events. He is now writing a book entitled *Multicast Networking and Applications*, to be published in 1998 by Addison-Wesley. Miller received a BEE degree from Rensselaer Polytechnic Institute and a MSEE degree from the University of Pennsylvania, specializing in communications. Miller can be reached at **miller@starburstcom.com**

Layer 2 and Layer 3 Switch Evolution

by Thayumanavan Sridhar, Future Communications Software

Layer 2 switches are frequently installed in the enterprise for high-speed connectivity between end stations at the data link layer. Layer 3 switches are a relatively new phenomenon, made popular by (among others) the trade press. This article details some of the issues in the evolution of Layer 2 and Layer 3 switches. We hypothesize that that the technology is evolutionary and has its origins in earlier products.

Layer 2 Switches

Bridging technology has been around since the 1980s (and maybe even earlier). Bridging involves segmentation of local-area networks (LANs) at the Layer 2 level. A multiport bridge typically learns about the *Media Access Control* (MAC) addresses on each of its ports and transparently passes MAC frames destined to those ports. These bridges also ensure that frames destined for MAC addresses that lie on the same port as the originating station are not forwarded to the other ports. For the sake of this discussion, we consider only Ethernet LANs.

Layer 2 switches effectively provide the same functionality. They are similar to multiport bridges in that they learn and forward frames on each port. The major difference is the involvement of hardware that ensures that multiple switching *paths* inside the switch can be active at the same time. For example, consider Figure 1, which details a four-port switch with stations A on port 1, B on port 2, C on port 3 and D on port 4. Assume that A desires to communicate with B, and C desires to communicate with D. In a single CPU bridge, this forwarding would typically be done in software, where the CPU would pick up frames from each of the ports sequentially and forward them to appropriate output ports. This process is highly inefficient in a scenario like the one indicated previously, where the traffic between A and B has no relation to the traffic between C and D.

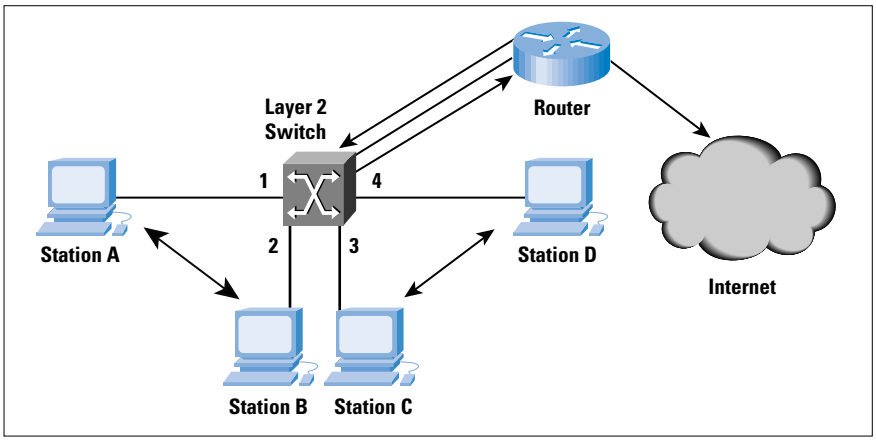


Figure 1: Layer 2 switch with External Router for Inter-VLAN traffic and connecting to the Internet

Enter hardware-based Layer 2 switching. Layer 2 switches with their hardware support are able to forward such frames in parallel so that A and B and C and D can have simultaneous conversations. The parallelism has many advantages. Assume that A and B are NetBIOS stations, while C and D are Internet Protocol (IP) stations. There may be no reason for the communication between A and C and A and D. Layer 2 switching allows this coexistence without sacrificing efficiency.

Virtual LANs

In reality, however, LANs are rarely so *clean*. Assume a situation where A,B,C, and D are all IP stations. A and B belong to the same IP subnet, while C and D belong to a different subnet. Layer 2 switching is fine, as long as only A and B or C and D communicate. If A and C, which are on two different IP subnets, need to communicate, Layer 2 switching is inadequate—the communication requires an IP router. A corollary of this is that A and B and C and D belong to different broadcast domains—that is, A and B should not "see" the MAC layer broadcasts from C and D, and vice versa. However, a Layer 2 switch cannot distinguish between these broadcasts—bridging technology involves forwarding broadcasts to all other ports, and it cannot tell when a broadcast is restricted to the same IP subnet.

Virtual LANs (VLANs) apply in this situation. In short, Layer 2 VLANs are Layer 2 broadcast domains. MAC broadcasts are restricted to the VLANs that stations are configured into. How can the Layer 2 switch make this distinction? By configuration. VLANs involve configuration of ports or MAC addresses. Port-based VLANs indicate that all frames that originate from a port belong to the same VLAN, while MAC address-based VLANs use MAC addresses to determine VLAN membership. In Figure 1, ports 1 and 2 belong to the same VLAN, while ports 3 and 4 belong to a different VLAN. Note that there is an implicit relationship between the VLANs and the IP subnets—however, configuration of Layer 2 VLANs does not involve specifying Layer 3 parameters.

We indicated earlier that stations on two different VLANs can communicate only via a router. The router is typically connected to one of the switch ports (Figure 1). This router is sometimes referred to as a *one-armed router* since it receives and forwards traffic on to the same port. In reality, of course, such routers connect to other switches or to wide-area networks (WANs). Some Layer 2 switches provide this Layer 3 routing functionality within the same box to avoid an external router and to free another switch port. This scenario is reminiscent of the large multiprotocol routers of the early '90s, which offered routing and bridging functions.

A popular classification of Layer 2 switches is "cut-through" versus "store-and-forward." Cut-through switches make the forwarding decision as the frame is being received by just looking at the header of the frame. Store-and-forward switches receive the entire Layer 2 frame

before making the forwarding decision. Hybrid adaptable switches which adapt from cut-through to store-and-forward based on the error rate in the MAC frames are very popular.

Characteristics

Layer 2 switches themselves act as IP end nodes for *Simple Network Management Protocol* (SNMP) management, Telnet, and Web based management. Such management functionality involves the presence of an IP stack on the router along with *User Datagram Protocol* (UDP), *Transmission Control Protocol* (TCP), Telnet, and SNMP functions. The switches themselves have a MAC address so that they can be addressed as a Layer 2 end node while also providing transparent switch functions. Layer 2 switching does not, in general, involve changing the MAC frame. However, there are situations when switches change the MAC frame. The IEEE 802.1Q Committee is working on a VLAN standard that involves "tagging" a MAC frame with the VLAN it belongs to; this tagging process involves changing the MAC frame. Bridging technology also involves the *Spanning-Tree Protocol*. This is required in a multibridge network to avoid loops. The same principles also apply towards Layer 2 switches, and most commercial Layer 2 switches support the Spanning-Tree Protocol.

The previous discussion provides an outline of Layer 2 switching functions. Layer 2 switching is MAC frame based, does not involve altering the MAC frame, in general, and provides transparent switching in parallel with MAC frames. Since these switches operate at Layer 2, they are protocol independent. However, Layer 2 switching does not scale well because of broadcasts. Although VLANs alleviate this problem to some extent, there is definitely a need for machines on different VLANs to communicate. One example is the situation where an organization has multiple intranet servers on separate subnets (and hence VLANs), causing a lot of intersubnet traffic. In such cases, use of a router is unavoidable; Layer 3 switches enter at this point.

Layer 3 Switches

Layer 3 switching is a relatively new term, which has been "extended" by a numerous vendors to describe their products. For example, one school uses this term to describe fast IP routing via hardware, while another school uses it to describe *Multi Protocol Over ATM* (MPOA). For the purpose of this discussion, Layer 3 switches are superfast routers that do Layer 3 forwarding in hardware. In this article, we will mainly discuss Layer 3 switching in the context of fast IP routing, with a brief discussion of the other areas of application.

Evolution

Consider the Layer 2 switching context shown in Figure 1. Layer 2 switches operate well when there is very little traffic between VLANs. Such VLAN traffic would entail a router—either "hanging off" one of the ports as a one-armed router or present internally within the switch. To augment Layer 2 functionality, we need a router—which

leads to loss of performance since routers are typically slower than switches. This scenario leads to the question: Why not implement a router in the switch itself, as discussed in the previous section, and do the forwarding in hardware?

Although this setup is possible, it has one limitation: Layer 2 switches need to operate only on the Ethernet MAC frame. This scenario in turn leads to a well-defined forwarding algorithm which can be implemented in hardware. The algorithm cannot be extended easily to Layer 3 protocols because there are multiple Layer 3 routable protocols such as IP, IPX, AppleTalk, and so on; and second, the forwarding decision in such protocols is typically more complicated than Layer 2 forwarding decisions.

What is the engineering compromise? Because IP is the most common among all Layer 3 protocols today, most of the Layer 3 switches today perform IP switching at the hardware level and forward the other protocols at Layer 2 (that is, bridge them). The second issue of complicated Layer 3 forwarding decisions is best illustrated by IP option processing, which typically causes the length of the IP header to vary, complicating the building of a hardware forwarding engine. However, a large number of IP packets do not include IP options—so, it may be overkill to design this processing into silicon. The compromise is that the most common (fast path) forwarding decision is designed into silicon, whereas the others are handled typically by a CPU on the Layer 3 switch.

To summarize, Layer 3 switches are routers with fast forwarding done via hardware. IP forwarding typically involves a route lookup, decrementing the *Time To Live* (TTL) count and recalculating the checksum, and forwarding the frame with the appropriate MAC header to the correct output port. Lookups can be done in hardware, as can the decrementing of the TTL and the recalculation of the checksum. The routers run routing protocols such as *Open Shortest Path First* (OSPF) or *Routing Information Protocol* (RIP) to communicate with other Layer 3 switches or routers and build their routing tables. These routing tables are looked up to determine the route for an incoming packet.

Combined Layer 2/Layer 3 Switches

We have implicitly assumed that Layer 3 switches also provide Layer 2 switching functionality, but this assumption does not always hold true. Layer 3 switches can act like traditional routers hanging off multiple Layer 2 switches and provide inter-VLAN connectivity. In such cases, there is no Layer 2 functionality required in these switches. This concept can be illustrated by extending the topology in Figure 1—consider placing a pure Layer 3 switch between the Layer 2 Switch and the router. The Layer 3 Switch would off-load the router from inter-VLAN processing.

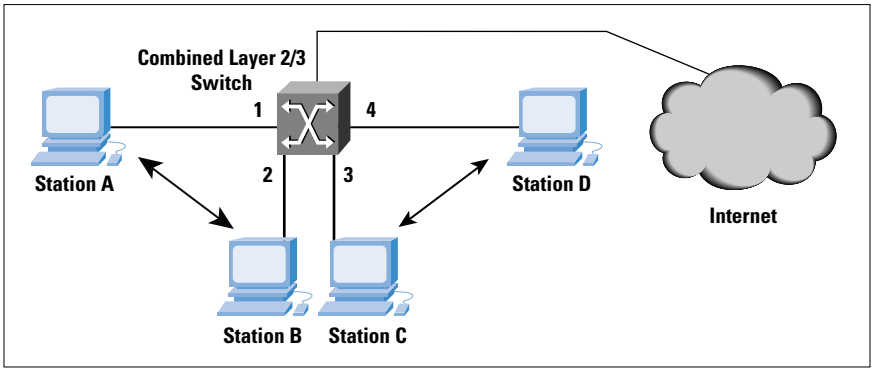


Figure 2: Combined Layer2/ Layer3 Switch connecting directly to the Internet

Figure 2 illustrates the combined Layer 2/Layer 3 switching functionality. The combined Layer 2/Layer 3 switch replaces the traditional router also. A and B belong to IP subnet 1, while C and D belong to IP subnet 2. Since the switch in consideration is a Layer 2 switch also, it switches traffic between A and B at Layer 2. Now consider the situation when A wishes to communicate with C. A sends the IP packet addressed to the MAC address of the Layer 3 switch, but with an IP destination address equal to C's IP address. The Layer 3 switch strips out the MAC header and switches the frame to C after performing the lookup, decrementing the TTL, recalculating the checksum and inserting C's MAC address in the destination MAC address field. All of these steps are done in hardware at very high speeds.

Now how does the switch know that C's IP destination address is Port 3? When it performs learning at Layer 2, it only knows C's MAC address. There are multiple ways to solve this problem. The switch can perform an *Address Resolution Protocol* (ARP) lookup on all the IP subnet 2 ports for C's MAC address and determine C's IP-to-MAC mapping and the port on which C lies. The other method is for the switch to determine C's IP-to-MAC mapping by snooping into the IP header on reception of a MAC frame.

Characteristics

Configuration of the Layer 3 switches is an important issue. When the Layer 3 switches also perform Layer 2 switching, they learn the MAC addresses on the ports—the only configuration required is the VLAN configuration. For Layer 3 switching, the switches can be configured with the ports corresponding to each of the subnets or they can perform IP address learning. This process involves snooping into the IP header of the MAC frames and determining the subnet on that port from the source IP address. When the Layer 3 switch acts like a one-armed router for a Layer 2 switch, the same port may consist of multiple IP subnets.

Management of the Layer 3 switches is typically done via SNMP. Layer 3 switches also have MAC addresses for their ports—this setup can be one per port, or all ports can use the same MAC address. The Layer 3 switches typically use this MAC address for SNMP, Telnet, and Web management communication.

Conceptually, the ATM Forum's *LAN Emulation* (LANE) specification is closer to the Layer 2 switching model, while MPOA is closer to the Layer 3 switching model. Numerous Layer 2 switches are equipped with ATM interfaces and provide a LANE client function on that ATM interface. This scenario allows the bridging of MAC frames across an ATM network from switch to switch. The MPOA is closer to combined Layer2/Layer 3 switching, though the MPOA client does not have any routing protocols running on it. (Routing is left to the MPOA server under the Virtual Router model.)

Do Layer 3 switches completely eliminate need for the traditional router ? No, routers are still needed, especially where connections to the wide area are required. Layer 3 switches may still connect to such routers to learn their tables and route packets to them when these packets need to be sent over the WAN. The switches will be very effective on the workgroup and the backbone within an enterprise, but most likely will not replace the router at the edge of the WAN (read Internet in many cases). Routers perform numerous other functions like filtering with access lists, inter-Autonomous System (AS) routing with protocols such as the *Border Gateway Protocol* (BGP), and so on. Some Layer 3 switches may completely replace the need for a router if they can provide all these functions (see Figure 2).

References

[1] *Computer Networks*, 3rd Edition, Andrew S. Tanenbaum, ISBN 0-13-349945-6, Prentice-Hall, 1996.

[2] *Interconnections: Bridges and Routers*, Radia Perlman, ISBN 0-201-56332-0, Addison-Wesley, 1992.

[3] "MAC Bridges," ISO/IEC 10038, ANSI/IEEE Standard 802.1 D-1993.

[4] "Draft Standard for Virtual Bridged Local Area Networks," IEEE P802.1Q/D6, May 1997.

[5] "Internet Protocol," Jon Postel, RFC 791, 1981.

[6] "Requirements for IP Version 4 Routers," Fred Baker, RFC 1812, June 1995.

[7] "LAN Emulation over ATM Version 1.0," **af-lane-0021.000**, The ATM Forum, January 1995.

[8] "Multiprotocol over ATM (MPOA) Specication Version 1.0" **af-mpoa-0087.000**, The ATM Forum, July 1997.

THAYUMANAVAN SRIDHAR is Director of Engineering at Future Communications Software in Santa Clara, CA. He received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at **sridhar@futsoft.com**

Book Review

Gigabit Ethernet

Gigabit Ethernet: Technology and Applications for High-Speed LANs, by Rich Seifert, ISBN 0-201-18553-9, Addison-Wesley, 1998, **http://www.awl.com/cseng/titles/0-201-18553-9**.

Gigabit Ethernet is storming its way onto the high-speed LAN scene. From a concept in 1984 to an emerging commercial reality in 1998, Gigabit Ethernet promises to give other high-speed LAN technologies, especially ATM, a serious run for their money. Capitalizing on the basic ease of use and deployment that has made other forms of Ethernet the most popular LAN technology of all, Gigabit Ethernet promises to add major bandwidth to such networks in a straightforward, completely compatible, and relatively affordable way. This book performs an excellent survey of the technologies, algorithms, and design principles that make Gigabit Ethernet possible, and also explains where the tremendous appeal of Gigabit Ethernet really lies. Much of the book is devoted to explaining Ethernet principles and operation in general, as well as exploring recent developments that have enabled gigabit technologies to emerge.

Organization

The book is divided into three parts. Part I explores the foundations that underpin Gigabit Ethernet, starting with a brief but cogent exploration of Ethernet before gigabit versions loomed on the horizon. The rest of Part I covers the trends in LAN usage in general, and Ethernet in particular, that laid the groundwork for Gigabit Ethernet. These trends include the move from shared media to dedicated media on many LANs, and likewise from shared LANs to dedicated LANs, and the concomitant deployment of full-duplex technologies to support bidirectional, high-bandwidth communications. Seifert, an original member of the DIX (Digital-Intel-Xerox) team that developed Ethernet, writes clearly and compellingly about complex issues, such as flow control, medium independence, and automatic configuration, as he explains what made Gigabit Ethernet possible, if not inevitable.

In Part II, Seifert turns his focus onto Gigabit Ethernet itself, beginning with an overview. In the rest of Part II, he explains how *Media Access Control* (MAC) works for half-duplex and full-duplex versions of Gigabit Ethernet, and makes a strong case for the essential irrelevancy of shared-media and half-duplex operation for Gigabit Ethernet. Along the way, Seifert also covers how Gigabit Ethernet networking devices, such as repeaters and switching and routing hubs, must be designed and how they work, and covers the behavior and operation of the physical layer at gigabit speeds.

He concludes this section of the book with a brief overview of the current IEEE Draft 802.3z specification that governs current Gigabit Ethernet operations, and mentions ongoing work in the 802.3ab subcommittee to define a workable implementation for Gigabit Ethernet on twisted-pair media (1000BaseT, as it will probably be known).

In Part III, Seifert tackles some of the most interesting material in this book. He begins with a discussion of how LANs and computers change roles over time in acting as the bottleneck for network use. The point here is that because of its extremely high bandwidth relative to the demands of most applications and end-user requirements, Gigabit Ethernet is likely to remain a backbone or clustering technology for the foreseeable future. He also explores the performance considerations for both networks and applications involved when extreme speeds or excessive bandwidths are available, to point out how bandwidth aggregation is presently Gigabit's most immediate and compelling contribution to networking.

Finally, he explores how Gigabit Ethernet compares to other high-speed networking technologies, including Fast Ethernet, *Fiber Distributed Data Interface* (FDDI), *High-Performance Parallel Interface* (HIPPI), *Fibre Channel*, and ATM. His discussion of why both ATM and Gigabit Ethernet are necessary, and why neither can fully supplant the other, represents a humorous and insightful analysis of why connection-oriented and connectionless communications and applications are both good, and why the two can never truly converge.

An Outstanding Contribution

A rundown of Seifert's layout and content, however, fails to do complete justice to this book. For one thing, Seifert's work includes the funniest and most ingenious footnotes I've seen in recent publications, including some truly horrendous puns and some downright howlers. For example, when discussing how repeaters work, he comments that "A jabbering station causes carrier sense to be continuously asserted and blocks all use of a shared LAN. A repeater looks for this condition and isolates the offending station." To this last sentence, he appends the following footnote: "Research is underway to determine if this mechanism can be extended for use on politicians and university lecturers." And this is just one of dozens of such gems that help to relieve the dryness that deeply technical material can sometimes manifest.

This book is also masterful simply because the author understands his material so well, and does such an outstanding job of explaining and exploring even the most abstruse networking concepts. Although I've been working with Ethernet for 15 years, I learned a great deal of new material from Part I of the book because old concepts were explained in new ways that improved my understanding. I suspect other readers will have one or two "Aha!" experiences from this tome as well.

But it's when making the case for full-duplex Gigabit Ethernet and exploring the requirements for switching and routing behaviors in Gigabit Ethernet networking devices that this material really shines.

Without a doubt, this book is among the very best of any of the literature available on high-speed networking today. I give it an A+ rating, not only because of the breadth and depth of its technical coverage and its compilation of essential concepts and information, but also because the author's deep understanding of networking protocols and communications needs enlivens all of his discussions of matters technical, business, and political. If you want to understand Gigabit Ethernet, this book is the obvious place to begin (and for many, to end) your search for enlightenment.

But even if all you want is a good read about expensive, exotic, and high-performance technology, Seifert's book offers the opportunity for outright enjoyment of the prose, and shared delight at untangling the technical dilemmas that any good design engineer must unravel on the road between a set of requirements and working implementation thereof.

—Ed Tittel
LANWrights, Inc.
etittel@lanw.com

More Book Reviews

We have more book reviews awaiting publication:

- *Internet Cryptography*, by Richard E. Smith, ISBN 0-201-92480-3, Addison-Wesley, 1998. Reviewed by Fred Avolio.
- *Web Security: A Step-by-Step Reference Guide*, by Lincoln D. Stein, ISBN 0-201-63489-9, Addison-Wesley, December 1997. Reviewed by Richard Perlman
- *IP Multicasting: The Complete Guide to Interactive Corporate Networks*, by Dave Kosiur ISBN 0-471-24359-0, Wiley Computer Publishing, 1998. Reviewed by Neophytos Iacovou.

So, make sure you receive the next issue of *The Internet Protocol Journal* due out in December 1998.

Fragments

More on The Future of the Domain Name System (DNS)

Shortly after our first issue went to press, the US Government issued a so-called White Paper as a follow on to the Green Paper. The White Paper, entitled "Management of Internet Names and Addresses," can be found at:

http://www.ntia.doc.gov/ntiahome/domainname/domainhome.htm

In early July, *The International Forum on The White Paper* (IFWP) was formed. The IFWP is "an ad hoc coalition of professional, trade and educational associations representing a diversity of Internet stakeholder groups." The IFWP held a series of meetings in Reston, Brussels, Geneva, Singapore and Buenos Aires to discuss the White Paper, specifically the incorporation of the *Internet Assigned Numbers Authority* (IANA). For more information on the IFWP process, see: **http://www.ifwp.org**

The IANA has posted draft bylaws for its incorporation on the IANA web site at: **http://www.iana.org**, and asked for community input. By the time you read this, the incorporation should already have taken place. We will provide an update in our next issue.

IETF Wins Award

The *Computer Professionals for Social Responsibility* (CPSR) has chosen the *Internet Engineering Task Force* (IETF) to be honored with the Norbert Wiener award for the group's influential role in the evolution of the Internet. In its 12-year history, this is only the second time the CPSR has recognized an organization rather than an individual. The IETF will accept the award at CPSR's annual conference, on Saturday evening, October 10, 1998, in Boston. The IETF is noted for its highly open and democratic processes that have affected the development of the Internet. The CPSR believes that such open processes are both extremely important and seriously threatened, and have accordingly made Internet governance the focus of its 1998 program year. The Norbert Wiener award was established in 1987 by the CPSR in memory of the originator of the field of cybernetics, whose pioneering work was one of the pillars on which the computer technology was created. See: **http://www.cpsr.org** and **http://www.ietf.org**

Send us your comments!

We look forward to hearing your comments and suggestions regarding anything you read in this publication. Send us e-mail at: **ipj@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-J4
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.