Supporting Next-Generation Internet Applications Today

It's been a rotten day. In the morning staff meeting, the CEO got on his videoconference bandwagon again, but still won't spring for the ATM backbone you know will make it work. In the afternoon, the usual congestion from “that news service”—hogging bandwidth with multiple copies of its data—brought the network to a standstill. And just as you're ready to call it a day, e-mail brings news that the California subsidiary won a contract to retrofit a utility's electric meters, to meet deregulation demands. They want you to get a block of just 4.3 million IP addresses.

Feel trapped? You're not alone.

CANS AND A STRING

Your company has spent millions on a corporate intranet, but it might as well be two cans and a string when it comes to handling heavy traffic and sophisticated applications.

The problem is that your company, like most around the world, has built its intranet on the Internet Protocol (IP). The current standard, IPv4, has been around since 1978. IP-based networks are great for transporting traditional, delay-insensitive data from one point to another, but the times are changing.

Traffic volume is skyrocketing, threatening to bring the Internet and many intranets to their knees. At the same time, real-time applications like videoconferencing demand an even quicker network response. There's also an insatiable demand for IP addresses. IPv4 wasn't designed for any of these situations.

One alternative is to buy an ATM network, the only networking scheme designed from day one to support both traditional data and real-time, multimedia traffic. Unfortunately, if you want the elegance of ATM, you pay the price for a new or additional network infrastructure. You also pay to integrate ATM with the IP-based LAN that probably knits your desktops together. Not every CEO is willing to make that kind of investment.

TO UPGRADE OR NOT TO UPGRADE?
The next-generation Internet offers another alternative: Keep your existing architecture and solve some of your biggest headaches. IPv6, the proposed standard for the next-generation Internet, offers the most elegant solution. It specifically addresses many of the shortcomings of IPv4. IPv6 became a proposed Internet Engineering Task Force (IETF) standard in November 1994, and some router and network vendors are already introducing IPv6-compliant products.

But not everyone is upgrading to IPv6 right now. Many companies are working around at least some of IPv4’s problems with software patches. Some IT shops are also using two techniques—network address translation and classless inter-domain routing—that allow one IP address to serve multiple users. These techniques, however,
IT Pro January / February 1999

What IPv6 Promises

A bundant IP addresses. Every server or computer gains access to the Internet via an IP address. Just as phone companies need new area codes, so the Internet needs more and more addresses. IPv6 extends the address length from 32 to 128 bits, creating a vast number of available addresses.

Telling one packet from another. To guarantee QoS, routers need some way to tell which packets get priority. IPv6 incorporates a packet-labeling mechanism.

Support for multicasting. IPv4 repeatedly routes multiple copies of data to each and every receiver, creating obvious congestion problems. IPv6 introduces an anycast address to help with this problem. Anycast identifies nodes that can share packets, and routers use that information to send just one set of data to service several nodes.

Better support for security. IPv4 has trouble supporting security because an application can encode operations at only one length: 40 bytes. IPv6 permits encoding at variable lengths and at lengths greater than 40 bytes. Now applications can support authentication and security encapsulation. This means it would be possible to, say, mark packets with security information and have nodes permit or deny access to a segment of the network based on that information.

Multicasting with RTP

A lthough you can use RTP for real-time unicast transmission, its real strength lies in supporting multicast. With unicast, a source sends multiple copies of the same data to each receiver. If that data is, for example, a video clip going to 10 different receivers, the source feeds 10 copies of that video data into the network. Multicast conserves bandwidth and can speed up transmission rates by allowing a source to send a single copy of the data as far as possible along the network. But to do that, routers and switches need certain information; that’s what RTP helps provide.

To support multicast, each RTP packet includes a source identifier, identifying which group member generated the data, and a time stamp, so that the receiver can re-create the proper timing. It also identifies the payload format of the transmitted data.

RTP does not require its participants to use the same protocols or media formats. Instead, it defines two relay functions to handle conversions: mixers and translators. These intermediate components relay traffic between RTP entities whose traffic streams differ in some way (format,
A mixer collects inbound RTP streams from two or more sources, translates them, synchronizes and combines the streams, and forwards the mixed stream to one or more destinations. The mixer function directly supports multicast by either changing the data format or simply performing the mixing function. For example, a mixer might combine RTP streams from audio sources at different locations into a lower bandwidth mix for a listener with insufficient bandwidth to “hear” each individual source.

A mixer can also combine “on/off” sources—data signals that start and stop—such as audio. Mixing interleaves the data from several streams, essentially eliminating a single stream’s wasted “space”—those points at which a single participant is not sending data.

To illustrate, suppose several systems are members of an audio session and each generates its own RTP stream. Most of the time, only one source is active (sending data) although occasionally more than one source will send data at the same time. Several systems may join or leave a session, just as radios may be turned on and off during a broadcast.

A new system may try to join a session but not have sufficient network bandwidth to carry all the RTP streams. In this case the mixer could receive all the RTP streams, combine them into a single stream, and retransmit that stream to the new session member. In this way, RTP multicast transmissions consume less bandwidth and don’t clog your network.

A translator is a simpler component. It alters and passes RTP packets from inbound streams separately and transparently, without mixing packets or combining streams. Some translators operate between two domains with different lower level protocols. Others modify the format of the RTP packet in some way. For example, a receiver might use one translator to convert a protocol, encoding, encryption, and/or bandwidth.

Quality of service (QoS) is difficult to define but easy to recognize, particularly when it fails to meet your expectations. To support the quality of certain transmissions, a network must be able to measure and guarantee characteristics like bandwidth, error rate, maximum delay, and packet loss rate. This is the idea behind QoS.

The current Internet was built on a simpler principle, best effort. Best effort treats all packets equally, with no service levels, requirements, reservations, or guarantees. This worked fine until now: Users can tolerate some delay while Web browsing, but even slight congestion transforms an Internet phone call into gibberish. For delay-sensitive services, best effort only works when there are abundant network resources from one end of the network to the other.

A QoS mechanism translates a request for service (an Internet phone call, for example) into a set of traffic requirements for that service (throughput, delay, jitter, loss, and error rates). Some QoS mechanisms also try to obtain network resources (bandwidth or buffers) sufficient to meet the traffic requirements, and that is how various QoS mechanisms differ.
How Traffic Routing Will Change

Architects of the next-generation Internet are changing the way they think about packet transmission, and their thinking will show up in various ways in IPv6 and other protocols. When today’s Internet encounters congestion, it drops packets. This won’t do in a QoS network. Instead, look for techniques that emphasize the following:

- Admission control. The next Internet will be based on flow, not packets. If a set of routers collectively determine that resources are insufficient to guarantee a requested QoS, a data flow will not gain entry to the network, much like a telephone call.
- More sophisticated routing. Next-generation Internets will base routing decisions on a variety of QoS parameters, not just minimum delay.
- Better queuing discipline. A queuing policy determines which packet to transmit next if several packets are queued for the same output port. More effective queuing policies will account for the differing requirements of different flows.
- Discard policy. In the current Internet, packets are discarded primarily to relieve congestion, but the discarded packets are always re-sent. New applications tolerate missing packets well, so discard policies should exploit this characteristic to alleviate congestion.

Network performance is influenced by fundamental choices. If you understand those choices, it becomes easier to track technology trends and easier to defend your networking choices.

RESOURCE RESERVATION

A key change in network architecture involves the reservation of resources—the ability of senders to set aside enough bandwidth and other resources to support a given transmission. There are two ways to reserve network resources: implicit and explicit reservation schemes. The next-generation Internet will support both.

Implicit reservation schemes do not specifically reserve resources. Instead, they classify packets and depend on routers to translate the classifications into priorities. Delay-sensitive packets have the highest priority and so move to the front of a router’s queue. At the router, an intelligent queue scheduler prevents high-priority packets from being trapped behind a logjam of jumbo data frames. Implicit reservation schemes are simpler to implement than explicit ones, but they cannot make firm QoS guarantees.

Explicit resource reservations work much like phone calls. The source of a message first requests resources. If available, the network allocates resources, guarantees them for the session’s duration, and releases them when they are no longer needed. If resources are unavailable, the network might ask the requestor to consider a lower QoS level. (This is how ATM works.)

MULTICASTING WITH RSVP

Broadband multicast transmission requires explicit resource reservation. This is available today using RSVP, a signaling protocol designed to handle multicast traffic.

In essence, RSVP mitigates the bandwidth problems caused by multicast transmissions by supporting the sharing of resources. It works like this: Multicasting can generate an enormous number of packets if the network does not take advantage of the way receivers are clustered.

For example, let’s assume that 10 PCs in a dorm are on the same network branch, as shown in Figure 2. If the 10 students that own them all watch the same Internet TV program, a separate copy of the same data traverses the entire path from the sender to each of 10 receivers. RSVP changes that by allowing intermediate nodes to share resources. The network carries only one set of the data from the sender all the way to the dorm’s root node. In this case, the “shared” broadcast requires far fewer resources in the middle of the network.

RSVP also has some novel features. It reverses, for example, the way resources are reserved. Today’s QoS-enabled networks (basically ATM) are sender-oriented: Senders are responsible for reserving network resources. The originator of a message requests a specific QoS and has the network allocate the bandwidth that each receiver requires.

In contrast, RSVP is receiver-initiated: Each receiver specifies the desired QoS and initiates the reservation process. The sender provides details about the traffic characteristics of a transmission to receivers (for example, the data rate of the TV broadcast) so that receivers can make an appropriate reservation request.

Multicast is the reason for this role reversal. A sender might have many receivers, and receivers typically have different equipment and bandwidth capabilities. It makes sense to match network resources with receivers’ QoS requirements, rather than what the sender is able to provide. For

high-speed video stream into a lower quality format with a reduced data rate. Translators thus help tailor transmissions to what a user’s system can handle.

RESOURCE RESERVATION

A key change in network architecture involves the reservation of resources—the ability of senders to set aside enough bandwidth and other resources to support a given transmission. There are two ways to reserve network resources: implicit and explicit reservation schemes. The next-generation Internet will support both.

Implicit reservation schemes do not specifically reserve resources. Instead, they classify packets and depend on routers to translate the classifications into priorities. Delay-sensitive packets have the highest priority and so move to the front of a router’s queue. At the router, an intelligent queue scheduler prevents high-priority packets from being trapped behind a logjam of jumbo data frames. Implicit reservation schemes are simpler to implement than explicit ones, but they cannot make firm QoS guarantees.

Explicit resource reservations work much like phone calls. The source of a message first requests resources. If available, the network allocates resources, guarantees them for the session’s duration, and releases them when they are no longer needed. If resources are unavailable, the network might ask the requestor to consider a lower QoS level. (This is how ATM works.)

MULTICASTING WITH RSVP

Broadband multicast transmission requires explicit resource reservation. This is available today using RSVP, a signaling protocol designed to handle multicast traffic.

In essence, RSVP mitigates the bandwidth problems caused by multicast transmissions by supporting the sharing of resources. It works like this: Multicasting can generate an enormous number of packets if the network does not take advantage of the way receivers are clustered.

For example, let’s assume that 10 PCs in a dorm are on the same network branch, as shown in Figure 2. If the 10 students that own them all watch the same Internet TV program, a separate copy of the same data traverses the entire path from the sender to each of 10 receivers. RSVP changes that by allowing intermediate nodes to share resources. The network carries only one set of the data from the sender all the way to the dorm’s root node. In this case, the “shared” broadcast requires far fewer resources in the middle of the network.

RSVP also has some novel features. It reverses, for example, the way resources are reserved. Today’s QoS-enabled networks (basically ATM) are sender-oriented: Senders are responsible for reserving network resources. The originator of a message requests a specific QoS and has the network allocate the bandwidth that each receiver requires.

In contrast, RSVP is receiver-initiated: Each receiver specifies the desired QoS and initiates the reservation process. The sender provides details about the traffic characteristics of a transmission to receivers (for example, the data rate of the TV broadcast) so that receivers can make an appropriate reservation request.

Multicast is the reason for this role reversal. A sender might have many receivers, and receivers typically have different equipment and bandwidth capabilities. It makes sense to match network resources with receivers’ QoS requirements, rather than what the sender is able to provide. For
example, one Internet TV receiver might want only the audio portion, while another might request reduced-quality video.

RSVP, therefore, uses a bottom-up approach to define the resources required for a session. Reservation messages originate at receivers, flow upstream toward the source (or sources), and are merged at branch points. When reservation messages merge, it is obvious that 10 receivers at the same location have requested the same feed. This backpropagation feature works for single senders and receivers as well.

RSVP also has a mechanism for providing routing information. Each receiver wishing to participate as a sender issues a message that propagates to every destination. Devices along the way create and maintain state information to be used for the routing. Receivers must update their states periodically so that the network can learn about changing network conditions and reservation requirements.

**COMPLEMENTARY APPROACHES FOR QOS**

Going forward, the IETF is developing two complementary initiatives that will add ATM-type QoS to IPv4. The integrated services (intserv) and differentiated services (diffserv) architectures approach QoS from different directions.

Intserv resembles ATM in some ways. For real-time applications, intserv makes a firm commitment to support a certain amount of delay and data loss (a guaranteed-delay service). It also provides a QoS comparable to best effort on a lightly loaded network with abundant resources (a controlled-load service). Although controlled service implies no firm guarantees, its delay and loss rates are expected to be low.

The diffserv initiative is still evolving, but some concepts are generally accepted. Simpler than intserv, it resembles a best-effort service with priority classes. Diffserv does not explicitly reserve resources, nor does it usually provide firm QoS guarantees. Instead, diffserv packets carry class markings that translate to priorities or weights. Switches and routers use packet markings to make decisions.

Intserv and diffserv are likely to coexist, with intserv used by early adopters, by corporate intranets, and when applications require strict QoS guarantees. Diffserv is a simple, scalable alternative for the remainder of networks and could dominate the public Internet in a few short years. But it is unclear how well diffserv might work if 90 percent of traffic is marked delay-sensitive.

Hybrid QoS schemes like IP over ATM are also under study. The Internet Engineering Task Force is studying how intserv’s guaranteed delay and controlled-load services might map to ATM’s rich suite of service categories. The ATM Forum recently accepted work items to investigate ATM and diffserv interworking. Interworking initiatives are important because much of the core of today’s (and perhaps tomorrow’s) IP-based Internet is ATM.

**Challenges to Multicasting**

Steve Deering is one of the primary developers of multicast standards for the Internet Engineering Task Force. He says most Internet routers you can buy today include support for multicast, so the technology is available. Some ISPs are now offering commercial multicast service. But no one considers multicast to be essential. In the absence of a killer app, multicast’s adoption will be evolutionary, taking a few years.

Meanwhile, there is still a long-term scalability question. How many concurrent multicast groups can we support at the same time? We know we can support tens of thousands, but what if we have millions of three-person teleconferences where the members are distributed all over the globe?

Read more of IT Professional’s interview with Deering in the next issue.
New protocols are a response to the growing variety and volume of traffic on the Internet and intranets. The developers of IPv6 specifically refer to supporting a future in which low-power, handheld devices may tap into the Internet, as may refrigerators, soda machines, and electric meters. They acknowledge that any protocol developed will be viable only if it remains compatible with current standards and plans for incremental change—few companies can afford to change their systems all at once.

Protocols such as IPv6, RSVP, and RTP are attempting to take the Internet into the future while meeting the needs of users today.

William Stallings is a consultant, lecturer, and author of more than a dozen professional reference books and textbooks on data communications and computer networking. He has three times received the award for the best computer science textbook of the year from the Text and Academic Authors Association (1998, Operating Systems, 3rd ed.; 1997, Data and Computer Communications, 5th ed.; 1996, Computer Organization and Architecture, 4th ed.). He has a PhD from MIT in computer science. His home in cyberspace is http://www.shore.net/~ws.

Traffic on a network or Internet falls into two broad categories: elastic and inelastic. Each has unique requirements.

Elastic traffic can adjust to changes in delay and throughput across the Internet and still meet the needs of its applications. TCP/IP-based intranets were designed to support this type of traffic. With TCP, routers alleviate congestion by reducing the rate at which data feeds into the network.

E-mail, file transfer, remote logon, network management, and Web access are all examples of elastic traffic, but they have different requirements.

- E-mail is generally quite insensitive to changes in delay.
- When performing online file transfer, users expect the delay to be proportional to the file size and are sensitive to throughput changes.
- Delay is generally not a serious concern of network management applications. However, if failures in the Internet or an intranet cause congestion, network management messages must get through with minimum delay.
- Remote logon and Web access are interactive applications and hence quite delay sensitive.

So, even elastic traffic could benefit from an Internet that addresses QoS requirements.

Inelastic traffic does not easily adapt—if at all—to changes in delay and throughput across an Internet. The prime example is real-time traffic, such as voice and video. Inelastic traffic can require

- A minimum throughput value. Many inelastic applications—like voice and video—require a firm minimum throughput.
- Minimal delays. Applications like stock trading—in which receiving information late can have adverse financial consequences—are extremely delay-sensitive.
- Limits to delay variation. The larger the allowable delay, the longer the real delay in data delivery and the greater the delay buffer size required at receivers. Real-time interactive applications, such as videoconferencing, may require a reasonable upper bound on delay variation.
- No packet loss. Real-time applications vary in the amount of packet loss—if any—they can sustain.

These requirements are difficult to meet in an environment with variable queuing delays and congestion losses. So to accommodate inelastic traffic, an Internet architecture must meet two new requirements.

First, an architecture must provide some means to give preferential treatment to applications with more demanding requirements. Applications need to state their requirements—either ahead of time in some sort of service request function or on the fly—using fields in the IP packet header.

Second, an architecture must continue to support inelastic traffic. Inelastic applications typically do not back off and reduce demand when faced with congestion, in contrast to current TCP-based applications. Therefore, in times of congestion, inelastic traffic will continue to supply a high load, crowding elastic traffic off the Internet. A reservation protocol can help control this situation by denying service requests that would leave too few resources available to handle current elastic traffic.

For More Information

The Internet Engineering Task Force is responsible for several protocols, including IP, RTP, and RSVP. Its working group on the next-generation Internet offers a comprehensive overview of standards development in that area at http://playground.sun.com/pub/ipng/html/ipng-main.html.
