The Internet Protocol Journal

December 1999 Volume 2, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In June 1992 when I was editor and publisher of *ConneXions—The Interoperability Report*, we published an article entitled "First IETF Internet Audiocast." Steve Casner and Steve Deering wrote: "The March Internet Engineering Task Force (IETF) meeting in San Diego was an exciting one for those interested in teleconferencing. In addition to several sessions on teleconferencing topics, we managed to pull off a 'wild idea' suggested by Allison Mankin from MITRE: live audio from the IETF site was 'audiocast' using IP multicast packet audio over the Internet to participants at 20 sites on three continents spanning 16 timezones."

Multicast has come a long way since 1992. Today, every IETF meeting features several live streams of not only audio but also video and slide presentations. Multicast continues to be developed in the IETF, as protocols and tools are being revised and refined. In two articles, Jon Crowcroft and Mark Handley describe the technologies behind multicast. The first article, included in this issue, looks at the current state of multicast. The second article, to appear in a future issue of IPJ, will look at the problems that need to be solved before multicast can become a truly scalable service for the Internet.

Research into new, high-speed networking technologies and applications is taking place in many parts of the world. One example of such a research effort can be found in the Internet2 Project. Larry Dunn describes some of the technology and application development being conducted by Internet2 members.

Interest in *IP Version 6* (IPv6) is growing as organizations contemplate a world where millions of devices such as cellphones, PDAs, cable TV set-top boxes and so on are "Internet Ready." The formation of the *IPv6 Forum* (**www.ipv6forum.com**) is some indication of this interest. We will look at a particular IPv4-to-IPv6 transition strategy in our next issue. In the meantime, Peter Salus takes a historical look at Internet addressing in our series "One Byte at a Time."

And so we reach the end of 1999 and the end of Volume 2 of *The Internet Protocol Journal*. We wish you a pleasant holiday season and an uneventful transition to Y2K.

—Ole J. Jacobsen, Editor and Publisher

ole@cisco.com

Internet Multicast Today

by Mark Handley, ACIRI and Jon Crowcroft, University College London

When you need to send data to many receivers simultaneously, you have two options: repeated transmission and broadcast. Repeated transmission may be acceptable if the cost is low enough and delivery can be spread out over time, as with junk mail or electronic mailing lists. Otherwise, a broadcast solution is required. With real-time multimedia, repeated delivery is feasible, but only at great expense to the sender, who must invest in large amounts of bandwidth. Similarly, traditional broadcast channels have been very expensive if they cover significant numbers of recipients or large geographic areas. However, the Internet offers an alternative solution: IP multicast effectively turns the Internet into a broadcast channel, but one that anyone can send to without having to spend huge amounts of money on transmitters and government licenses. It provides efficient, timely, and global many-to-many distribution of data, and as such may become the broadcast medium of choice in the future.

The Internet is a datagram network, meaning that anyone can send a packet to a destination without having to preestablish a path. Of course, the boxes along the way must have either precomputed a set of paths, or they must be relatively fast at calculating one as needed, and typically, the former approach is used. However, the sending host need not be aware of or participate in the complex route calculation; nor does it need to take part in a complex *signaling* or *call setup* protocol. It simply addresses the packet to the right place, and sends it. This procedure may be a more complex procedure if the sending or receiving systems need more than the default performance that a path or network might offer, but it is the *default* model.

Adding multicast to the Internet does not alter the basic model. A sending host can still simply send, but now there is a new form of address, the multicast or host group address. Unlike unicast addresses, hosts can dynamically subscribe to multicast addresses and by so doing cause multicast traffic to be delivered to them. Thus the IP multicast *service model* can be summarized:

- Senders send to a multicast address
- Receivers express an interest in a multicast address
- Routers conspire to deliver traffic from the senders to the receivers

Sending multicast traffic is no different from sending unicast traffic except that the destination address is slightly special. However, to receive multicast traffic, an interested host must tell its local router that it is interested in a particular multicast group address; the host accomplishes this task by using the *Internet Group Management Protocol* (IGMP).

Point-to-multipoint communication is nothing new. We are all used to the idea of broadcast TV and radio, where a shared medium (the radio frequency [RF] spectrum) is partitioned among users (transmitter or TV/radio station owners). It is a matter of regulation that there is typically only one unique sender of particular content on any given frequency, although other parts of the RF spectrum are given over to free use for multiparty communication (police radio, citizen band radio, and so on).

The Internet multicast *model*[3] is very similar. The idea is to convert the mesh wide-area network that is the Internet (whether the public Internet, a private enterprise net, or intranet makes no difference to the model), into a shared resource for senders to send to multiple participants, or groups.

To make this group communication work for large-scale systems—in the sense of a large number of recipients for a particular group, or in the sense of a large number of senders to a large number of recipients, or in the sense of a large number of different groups—it is necessary, both for senders and for the routing functions to support delivery, to have a system that can be largely independent of the particular recipients at any one time. In other words, just as a TV or radio station does *not know* who is listening when, an Internet multicast sender does not know who might receive packets it sends. If this scenario sends out alarm bells about security, it shouldn't. A unicast sender has no assurance about who receives its packets either. Assurances about disclosure (privacy) and authenticity of sender/recipient are largely separate matters from simple packet delivery models. Security is a topic of much research and the focus for the recently formed *Internet Research Task Force* (IRTF) research group, *Secure Multicast Group* (SMuG).

The Internet multicast model is an extension of the datagram model; it uses the fact that the datagram is a self-contained communications unit that not only conveys data from source to destination, but also conveys the source and destination address information. In other words, in some senses, datagrams *signal* their own path, both with a source and a destination address in every packet.

By adding a range of addresses dedicated for sending to groups, and providing independence between the address allocation and the rights to send to a group, the analogy between RF spectrum and the Internet multicast space is maintained. Some mechanism, as yet unspecified, is used to dynamically choose which address to send to. Suffice it to say that for now, the idea is that somehow, elsewhere, the address used for a multicast session or group communication activity is chosen so that it does not clash with other uses or users, and is advertised to potential senders and receivers.

Unlike the RF spectrum, an IP packet to be multicast carries a unique source identifier, in that such packets are sent with the normal unicast IP address of the interface of the sending host.

It is also worth noting that an address that is being used to signify a group of entities must surely be a logical address (or in some senses a name) rather than a topological or topographical identifier. We shall see that this means there must be some service that maps such a logical identifier to a specific set of locations in the same way that a local unicast address must be mapped (or bound) to a specific location. In the multicast case, this mapping is distributed. Note also that multicast Internet addresses are in some sense "host group" addresses, in that they indicate a set of hosts to deliver to. In the Internet model, there is a further level of multiplexing, that of transport-level ports, and there is room for some overlap of functionality, since a host may receive packets sent to multiple multicast addresses on the same port, or multiple ports on the same multicast address.

This model raises numerous questions about address and group management, such as how these addresses are allocated. The area requiring most change, though, is in the domain of the routing. Somehow the routers must be able to build a distribution tree from the senders to all the receivers for each multicast group. The senders don't know who the receivers are (they just send their data), and the receivers don't know who the senders are (they just ask for traffic destined for the group address), so the routers have to do something without help from the hosts. We will examine this scenario in detail in the section "Multicast Routing."

Roadmap

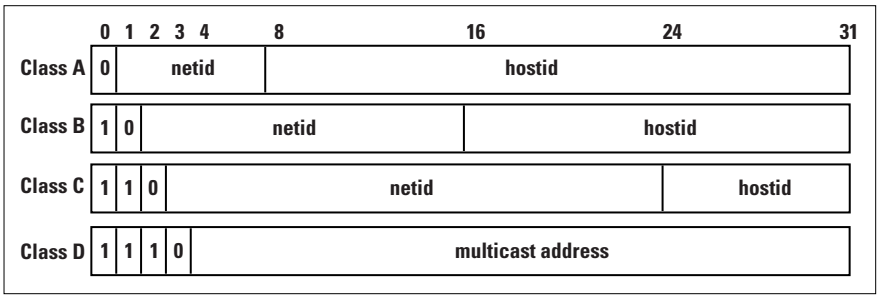
The functions that provide the Standard Internet Multicast Service can be separated into host and network components. The interface between these components is provided by IP multicast addressing and IGMP group membership functions, as well as standard IP packet transmission and reception. The network functions are principally concerned with multicast routing, while host functions also include higher-layer tasks such as the addition of reliability facilities in a transport-layer protocol. That's the order in which we cover each of these functions in the rest of this article. At the end of the article we list the current status of *Internet Engineering Task Force* (IETF) specification for the various components.

Host Functions

As we stated above, host functionality is extended through the use of the IGMP protocol. Hosts and routers, which we will look at later, must be able to deal with new forms of addresses. When IP Version 4 addressing was first designed, it was divided into classes as shown in Figure 1.

Figure 1: Internet Address Classes

A	1.0.0.0	to 126.255.255.255
B	128.0.0.0	to 191.255.255.255
C	192.0.0.0	to 223.255.255.255
D	224.0.0.0	to 239.255.255.255



Originally Class A was intended for large networks, B for midsize networks, and C for small networks. Class D was later allocated for multicast addresses. Since then, classless addressing has been introduced to solve Internet scaling problems, and the rules for Classes A, B, and C no longer hold, but Class D is still reserved for multicast, so all IPv4 multicast addresses start with the high-order 4-bit "nibble": 1110

In other words, from the 2^{32} possible addresses, 2^{28} are multicast, meaning that there can be up to about 270 million different groups, each with as many senders as can get unicast addresses! This number is many orders of magnitude more than the RF spectrum allows for typical analog frequency allocations.

For a host to support multicast, the host service interface to IP must be extended in three ways:

- A host must be able to join a group, meaning that it must be able to reprogram its network level, and possibly, consequentially, the lower levels, to be able to receive packets addressed to multicast group addresses.
- An application that has joined a multicast group and then sends to that group must be able to select whether it wants the host to loopback the packets it sent so that it receives its own packets.
- A host should be able to limit the *scope* with which multicast messages are sent. The Internet Protocol contains a *Time-To-Live* (TTL) field, used originally to limit the lifetime of packets on the network, both for safety of upper layers, and for prevention of traffic overload during temporary routing loops. It is used in multicast to limit how "far" a packet can go from the source. We will see below how scoping can interact with routing.

When an application tells the host networking software to join a group, the host software checks to see if the host is a member of the group. If not, it makes a note of the fact, and sends out an IGMP membership report message. It also maps the IP address to a lower-level address and reprograms its network interface to accept packets sent to that address. There is a refinement here: a host can join "on an interface;" that is, hosts that have more than one network card can decide which one (or more than one) they wish to receive multicast packets via. The implication of the multicast model is that it is "pervasive," so it is usually necessary to join on only one interface.

Taking a particular example to illustrate the IP-level to link-level mapping process, if a host joins an IP multicast group using an Ethernet interface, there is a mapping from the low 24 bits of the multicast address into the low 24 (out of 48) bits of the Ethernet address. Since this mapping is a many-to-one mapping, there may be multiple IP multicast groups occupying the same Ethernet address on a given wire, though it may be made unlikely by the address allocation scheme. An Ethernet LAN is a shared-medium network, thus local addressing of packets to an Ethernet group means that the packets are received by Ethernet hardware and delivered to the host software of *only* those hosts with members of the relevant IP group. Therefore, host software is generally saved the burden of filtering out irrelevant packets. Where there is an Ethernet address clash, software can filter the packets efficiently.

Operation of the IGMP protocol can be summarized as follows:

- When a host first joins a group, it programs its Ethernet interface to accept the relevant traffic, and it sends an IGMP Join message on its local network. This message informs any local routers that there is a receiver for this group now on this subnet.
- The local routers remember this information, and arrange for traffic destined for this address to be delivered to the subnet.
- After a while, the routers wonder if there is still any member on the subnet, and send an IGMP query message to the multicast group. If the host is still a member, it replies with a new message unless it hears someone else do so first. Multicast traffic continues to be delivered.
- Eventually the application finishes, and the host no longer wants the traffic. It reprograms its Ethernet interface to reject the traffic, but the packets are still sent until the router times the group out and sends a query to which no one responds. The router then stops delivering the traffic.

Thus joining a multicast group is quick, but leaving can be slow with IGMP Version 1. IGMP Version 2 reduces the leave latency by introducing a "Leave" message and a set of rules to prevent one receiver from disconnecting others when it leaves. IGMP Version 3 (not yet deployed) introduces the idea of *source-specific* joining and leaving, whereby a host can subscribe (or reject) traffic from individual senders rather than the group as a whole, at the expense of more complexity and extra state in routers.

Multicast Routing

Given the multicast service model described above, and the restrictions that senders and receivers don't know each others' location or anything about the topology, how do routers conspire to deliver traffic from the senders to the receivers?

We shall assume that if a sender and a receiver did know about each other, they could each send unicast packets to the other. In other words, there is a network with bidirectional paths and an underlying unicast routing mechanism already running. Given this network, there is a spectrum of possible solutions. At one extreme, we can flood data from the sender to all possible receivers and have the routers for networks where there are no receivers prune off their branches of the distribution tree. At the other extreme, we can communicate information in a multicast routing protocol conveying the location of all the receivers to the routers on the paths to all possible senders. Neither method is particularly desirable on a global scale, so the most interesting solutions tend to be hybrid solutions that lie between these extremes.

In the real world, there are many different multicast routing protocols, each with its own advantages and disadvantages. We shall explain each of the common ones briefly, because a working knowledge of their pros and cons helps us understand the practical limits to the uses of multicast.

Flood and Prune Protocols

Flood and Prune Protocols are more correctly known as *reverse-path multicast* algorithms. When a sender first starts sending, traffic is flooded out through the network. A router may receive the traffic along multiple paths on different interfaces, in which case it rejects any packet that arrives on any interface other than the one it would use to send a unicast packet back to the source. It then sends a copy of each packet out of each interface other than the one back to the source. In this way, each link in the whole network is traversed at most once in each direction, and the data is received by all routers in the network.

So far, this process describes *reverse-path broadcast*. Many parts of the network will be receiving traffic, even though there are no receivers there. These routers know they have no receivers (otherwise IGMP would have told them) and they can then send prune messages back toward the source to stop unnecessary traffic from flowing. Thus the delivery tree is pruned back to the minimal tree that reaches all the receivers. The final distribution tree is what would be formed by the union of shortest paths from each receiver to the sender, so this type of distribution tree is known as a *shortest-path tree* (strictly speaking, it's a reverse shortest path tree—typically the routers don't have enough information to build a true forward shortest-path tree).

Two commonly used multicast routing protocols fall in the class: the *Distance Vector Multicast Routing Protocol* (DVMRP)[4] and *Protocol Independent Multicast Dense-Mode* (PIM-DM)[5]. The primary difference between these protocols is that DVMRP computes its own routing table to determine the best path back to the source, whereas PIM Dense-Mode uses the routing table of the underlying unicast routing system, hence the term "Protocol Independent."

It should be fairly obvious that sending traffic *everywhere* and getting people to tell you what they don't want is not a particularly scalable mechanism. Sites get traffic they don't want (albeit very briefly), and routers not on the delivery tree need to store prune state. For example, if a group has one member in the UK and two in France, routers in Australia still get some of the packets, and they need to hold prune state to prevent more packets from arriving! However, for groups where most places actually do have receivers (receivers are "densely" distributed), this sort of protocol works well. So although these protocols are poor choices for a global scheme, they might be appropriate within some organizations.

MOSPF

Multicast Open Shortest Path first (MOSPF[12]) isn't really a category, but a specific instance of a protocol. MOSPF is the multicast extension to *Open Shortest Path First* (OSPF[11]), which is a unicast link-state routing protocol.

Link-state routing protocols work by having each router send a routing message periodically listing its neighbors and how far away they are. These routing messages are flooded throughout the entire network, so every router can build up a map of the network. This map is then used to build forwarding tables (using a Dijkstra algorithm) so that the router can decide quickly which is the correct next hop for a particular packet.

Extending this concept to multicast is achieved simply by having each router also list in a routing message the groups for which it has local receivers. Thus given the map and the locations of the receivers, a router can also build a multicast forwarding table for each group.

MOSPF also suffers from poor scaling. With flood-and-prune protocols, data traffic is an *implicit* message about where there are senders, so routers need to store unwanted state where there are no receivers. With MOSPF, there are *explicit* messages about where all the receivers are, so routers need to store unwanted state where there are no senders. However, both types of protocol build very efficient distribution trees.

Center-Based Trees

Rather than flooding the data everywhere, or flooding the membership information everywhere, algorithms in the center-based trees category map the multicast group address to a particular unicast address of a router, and they build explicit distribution trees centered around this particular router. Three main problems need to be solved to get this approach to work:

- How is the mapping from group address to center address performed?
- How is the center location chosen so that the distribution trees are efficient?
- How is the tree actually constructed given the center address?

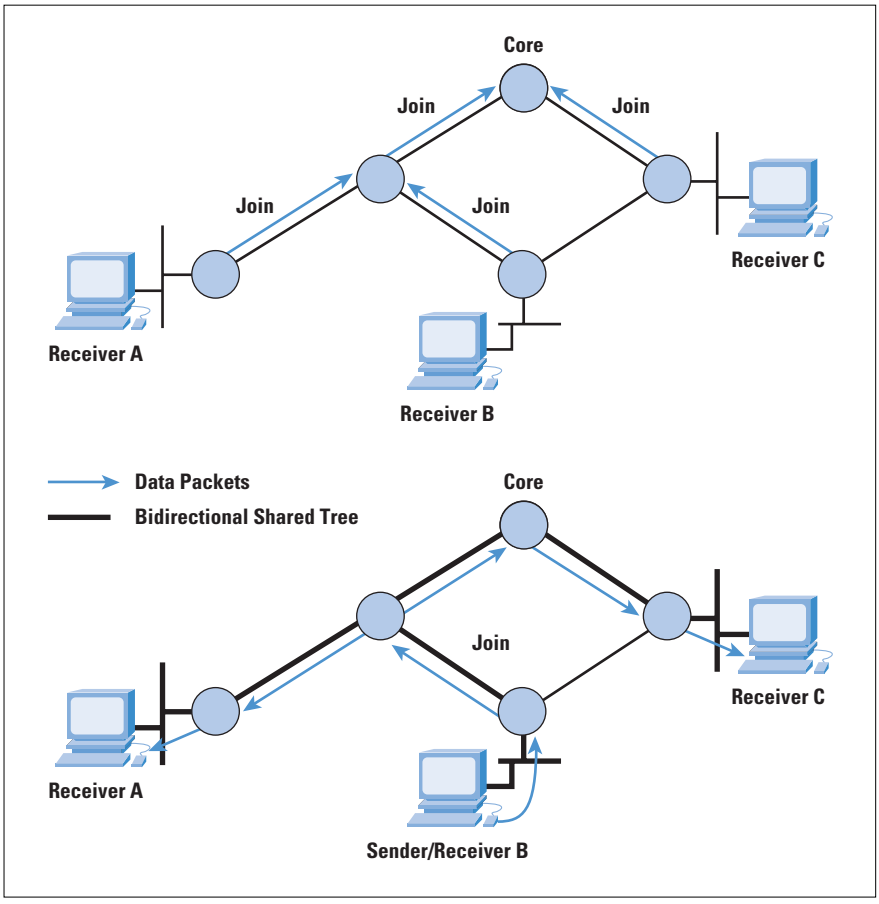
Different protocols have come up with different solutions to these problems. Three center-based tree protocols are worth exploring because they illustrate different approaches: *Core-Based Trees* (CBT), *PIM Sparse-Mode* (PIM-SM), and the *Border Gateway Multicast Protocol* (BGMP). However, we will leave discussion of BGMP until our second article because it is not currently deployed.

Core-Based Trees

Core-Based Trees (CBT[1]) was the earliest center-based tree protocol, and it is the simplest.

When a receiver joins a multicast group, its local CBT router looks up the multicast address and obtains the address of the Core router for the group. It then sends a Join message for the group toward the Core. At each router on the way to the Core, forwarding state is instantiated for the group, and an acknowledgment is sent back to the previous router. In this way, a multicast tree is built, as shown in Figure 2.

Figure 2: Formation of a CBT Bidirectional Shared Tree



If a sender (that is, a group member) sends data to the group, the packets reach its local router, which forwards them to any of its neighbors that are on the multicast tree. Each router that receives a packet forwards it out of all its interfaces that are on the tree except the one the packet came from. The style of tree CBT builds is called a "bidirectional shared tree," because the routing state is "bidirectional"—packets can

flow both up the tree toward the Core and down the tree away from the Core, depending on the location of the source, and packets are "shared" by all sources to the group. This scenario is in contrast to "unidirectional shared trees" built by PIM-SM as we shall see later.

IP multicast does not require senders to a group to be members of the group, so it is possible that a sender's local router is not on the tree. In this case, the packet is forwarded to the next hop toward the Core. Eventually the packet will either reach a router that is on the tree, or it will reach the Core, and it is then distributed along the multicast tree.

CBT also allows multiple Core routers to be specified, adding a little redundancy in case the Core becomes unreachable. CBT never properly solved the problem of how to map a group address to the address of a Core. In addition, good Core placement is a difficult problem. Without good Core placement, CBT trees can be quite inefficient, and so CBT is unlikely to be used as a global multicast routing protocol.

However, within a limited domain, CBT is very efficient in terms of the amount of state that routers need to keep. Only routers on the distribution tree for a group keep forwarding state for that group, and no router needs to keep information about any source; thus CBT scales much better than flood-and-prune protocols, especially for sparse groups where only a small proportion of subnetworks have members.

PIM Sparse-Mode

The work on CBT encouraged others to try to improve on its limitations while keeping the good properties of shared trees, and *PIM Sparse-Mode*[7] was one result. The equivalent of a CBT Core is called a *Rendezvous Point* (RP) in PIM, but it largely serves the same purpose.

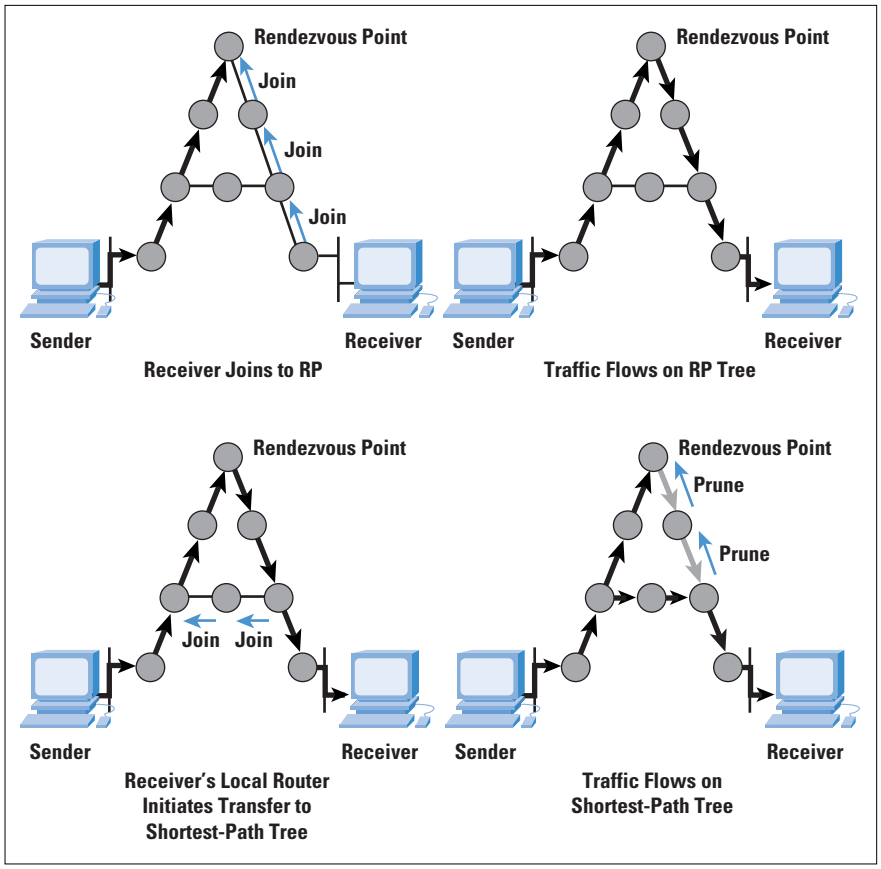
When a sender starts sending, whether it is a member or not, its local router receives the packets and maps the group address to the address of the RP. It then encapsulates each packet in another IP packet (imagine putting one letter inside another, differently addressed, envelope) and sends it unicast directly to the RP.

When a receiver joins the group, its local router initiates a Join message that travels hop-by-hop to the RP instantiating forwarding state for the group. However, this state is unidirectional state—it can be used only by packets flowing from the RP toward the receiver, and not for packets flowing back up the tree toward the RP. Data from senders is de-encapsulated at the RP and flows down the shared tree to all the receivers.

PIM-SM is an improvement on CBT in that discovery of senders and and tree building from senders to receivers are separate functions.

Thus PIM-SM unidirectional trees are not particularly good distribution trees, but they do start data flowing to the receivers. Once this data is flowing, the local router of a receiver can then initiate a transfer from the shared tree to a shortest-path tree by sending a source-specific Join message toward the source, as shown in Figure 3. When data starts to arrive along the shortest-path tree, a prune message can be sent back up the shared tree toward the source to avoid getting the traffic twice.

Figure 3: Formation of a PIM Sparse-Mode Tree



Unlike other shortest-path tree protocols such as DVMRP and PIM-DM, where prune state exists everywhere there are no receivers, with PIM-SM, source-specific state exists only on the shortest-path tree. Also, low-bandwidth sources such as those sending *Real-Time Control Protocol* (RTCP) receiver reports do not trigger the transfer to a shortest-path tree, a scenario that further helps scaling by eliminating unnecessary source-specific state.

Because PIM-SM can optimize its distribution trees after formation, it is less critically dependent on the RP location than CBT is on the Core location. Hence the primary requirement for choosing an RP is load balancing. To perform multicast-group-to-RP mapping, PIM-SM predistributes a list of candidates to be RPs to all routers. When a router needs to perform this mapping, it uses a special hash function to hash the group address into the list of candidate RPs to decide the actual RP to join.

Except in rare failure circumstances, all the routers within the domain will perform the same hash, and come up with the same choice of RP. The RP may or may not be in an optimal location, but this situation is offset by the ability to switch to a shortest-path tree.

The dependence on this hash function and the requirement to achieve convergence on a list of candidate RPs does, however, limit the scaling of PIM-SM. As a result, it is also best deployed within a domain, although the size of such a domain may be quite large.

Interdomain Multicast Routing

All the multicast routing schemes described so far suffer from scaling problems of one form or another:

- DVMRP and PIM-DM initially send data everywhere, and require routers to hold prune state to prevent this flooding from persisting.
- MOSPF requires all routers to know where all receivers are.
- PIM-SM needs predistribution of information about the set of RPs. Because traffic needs to flow to the RP, an RP cannot handle too many groups simultaneously, so many RPs are needed globally.

Thus each of these schemes is likely to be best deployed within a domain. How then does interdomain multicast routing take place?

Long-term solutions to this problem will be discussed in the second of these articles. In the meantime, the interim solution currently being deployed consists of multiprotocol extensions to the unicast *Border Gateway Protocol* (BGP) interdomain routing protocol, and a protocol called MSDP to glue PIM-SM domains together.

Multiprotocol BGP

For either technical or policy reasons, not all routers or peerings between Internet Service Providers (ISPs) are multicast capable. This situation complicates the use of PIM-SM for operation between domains because PIM assumes that the route obtained by unicast routing is good for multicast routing (strictly speaking, PIM assumes the reverse unicast path is good for forward-path multicast routing). If, in fact, the reverse unicast path is *not* good for forward-path multicast, then Join messages will often reach routers that do not support multicast, resulting in a lack of multicast connectivity. How then do we solve this problem?

BGP is the unicast interdomain routing protocol that is very widely used to connect unicast routing domains together. The multiprotocol extensions to BGP allow multiple routing tables to be maintained for different protocols. Thus with the *Multiprotocol Extensions for BGP-4* (MBGP)[2], you can build one routing table for unicast-capable routes and one for multicast-capable routes using the same protocol. PIM can then use the multicast-capable routes to forward Join messages and can, therefore, detour around parts of the network that don't support multicast.

Multicast Source Discovery Protocol

In addition to the problem of designing a scalable mechanism for mapping multicast groups to RPs, attempts to use PIM-SM as an interdomain protocol are hindered by ISPs' desire not to be dependent on other ISPs' facilities. For example, consider a multicast group consisting of senders and receivers in two domains, A and B, run by two different ISPs. If the RP is in domain A, and there is some problem in domain A, then senders and receivers in domain B might still be unable to communicate with each other using multicast, even though they are in the same domain, because initial PIM register messages must go via the RP. ISPs do not want to be dependent on other ISPs for connectivity within their own domain, so it appears that using PIM-SM as an interdomain protocol would be unacceptable, even if there were no scalability problems.

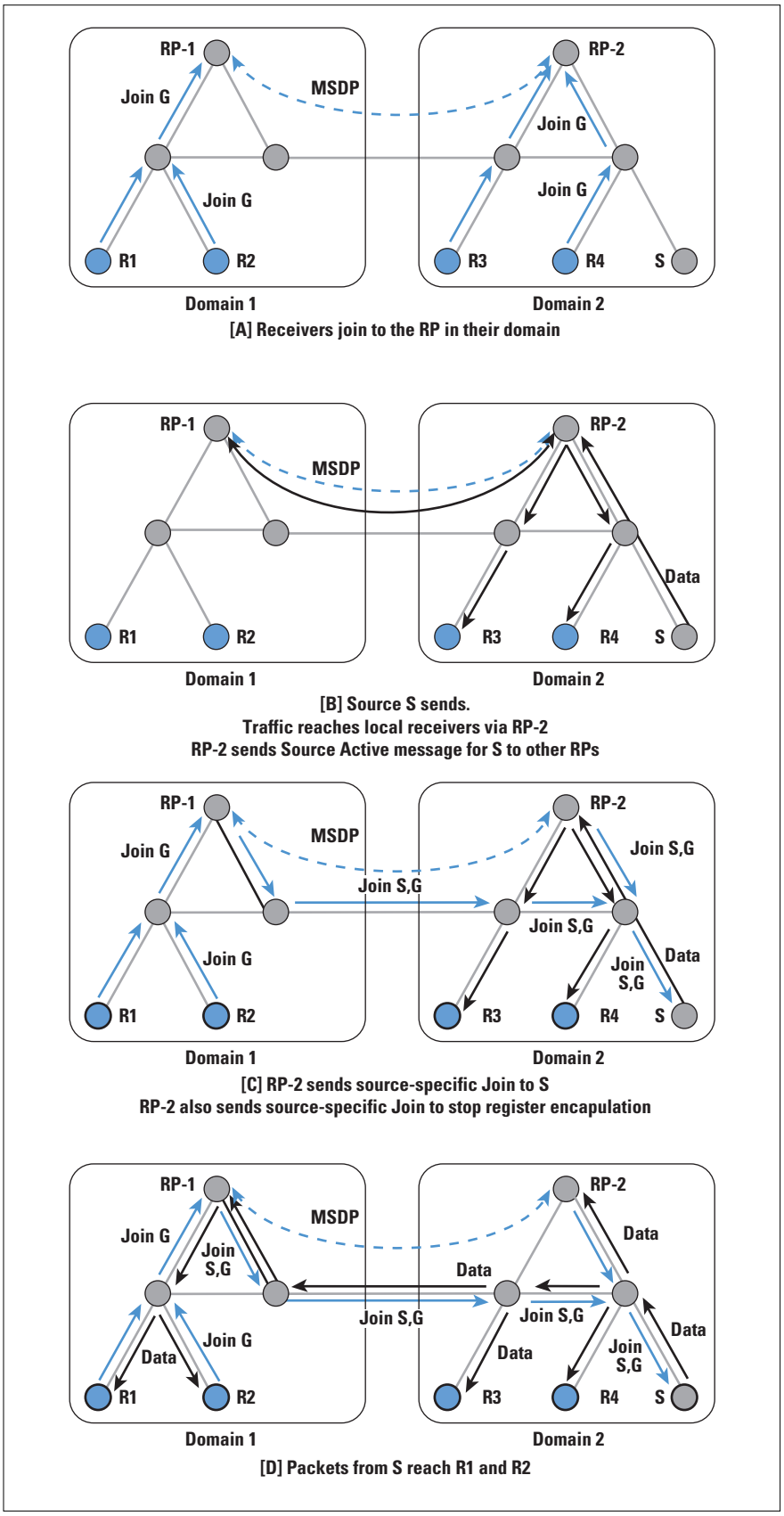
The *Multicast Source Discovery Protocol* (MSDP)[8] is an attempt to work around this problem. It does not provide a long-term scalable solution, but does provide a solution that solves the ISP interdependence problem.

With MSDP, ISPs run PIM-SM within their own domain, and they have their own set of RPs for all groups within that domain. Additionally, the RPs within the domain are interconnected with each other and with RPs in neighboring domains using MSDP control connections to form a loose mesh.

The process is shown in Figure 4. Within domain 1, R1 and R2 send Join messages from group G to RP-1. Similarly, R3 and R4 send Join messages to RP-2. When S starts sending, its packets are encapsulated to RP-2 by its local router in the normal PIM-SM manner. RP-2 decapsulates the packets and forwards them down the group-shared tree within domain 2 to reach R3 and R4. In addition, it sends a *Source Active* message over the MSDP mesh to all other RPs. RPs like RP-1 that have active joiners for this group then send a source-specific Join back across the interdomain boundary toward S. Traffic is then delivered interdomain following the source-specific state laid down by the Join messages, and it is eventually delivered to R1 and R2.

MSDP uses the normal PIM-SM source-specific join mechanism interdomain following the MBGP multicast routes back to the source, but it sets up only a group-shared tree within each domain, avoiding the need to depend on remote RPs in different domains for the delivery of traffic between local members in a domain.

Figure 4: MSDP in Operation



As an interdomain routing protocol, however, MSDP has many shortcomings. In particular, every RP in every domain must be told about every source that starts sending, and a significant subset of the RPs must cache all this information so that receivers that join late can cause source-specific Joins to be sent by their local RP. Thus MSDP does not scale well if there are a large number of senders worldwide.

In addition, to ensure that the first few packets sent by a source do not get lost, they must be encapsulated and sent alongside the *Source Active* message to all the RPs that might possibly have receivers. If they are not encapsulated, then sources that send only a few packets every few minutes might never get any data through to receivers because the source-specific state has timed out after each time they send.

In summary, MSDP is not a scalable long-term solution to interdomain multicast routing. However, it does solve a real short-term problem faced by ISPs, and so it is currently seeing significant deployment.

Multicast Address Allocation

A local protocol for requesting multicast addresses from multicast address allocation servers has recently been standardized. This protocol is called *Multicast Address Dynamic Client Allocation Protocol*, or MADCAP[10]. It is a relatively simple request-response protocol loosely modeled after the *Dynamic Host Configuration Protocol* (DHCP)[6].

MADCAP is intended to be used with interdomain protocols that perform dynamic allocation of parts of the multicast address space between domains, but because these protocols are not yet deployed, they will be discussed in the second of these articles.

As an interim solution for interdomain address allocation, a simple static mechanism has been defined. This mechanism involves embedding the *Autonomous System* (AS) number of the domain as the middle 16 bits of a multicast address. Thus the domain with AS number 16007 would get multicast addresses in the range 233.64.7.0 to 233.64.7.255 (64 and 7 being the upper and lower bytes, respectively, of 16007). Known as *glop addressing*, this mechanism is experimental. It may be superseded by a dynamic mechanism in the longer term.

Multicast Scoping

When applications operate in the global Multicast backbone (MBone), it is clear that not all groups should have global scope. Not only is this constraint especially important for performance reasons with flood and prune multicast routing protocols, but it also is true with other routing protocols for application security reasons and because multicast addresses are a scarce resource. Being able to constrain the scope of a session allows the same multicast address to be in use at more than one place as long as the scopes of the sessions do not overlap. This is analogous to the same radio frequency being used by two radio stations operating far apart from one another—each will only be heard locally.

Multicast scoping can currently be performed in two ways, known as *TTL Scoping* and *Administrative Scoping*. Currently TTL scoping is most widely used, with only a very few sites making use of administrative scoping.

TTL Scoping

When an IP packet is sent, an IP header field called *Time To Live* (TTL) is set to a value between zero and 255. Every time a router forwards the packet, it decrements the TTL field in the packet header, and if the value reaches zero, the packet is dropped. The IP specification also states that the TTL should be decremented if a packet is queued for more than a certain amount of time, but this decrement is rarely implemented these days. With unicast, the TTL is normally set to a fixed value by the sending host (64 and 255 are commonly used) and is intended to prevent packets from looping forever.

With IP multicast, the TTL field can be used to constrain how far a multicast packet can travel across the MBone by carefully choosing the value put into packets as they are sent. However, because the relationship between hop count and suitable scope regions is poor at best, the basic TTL mechanism is supplemented by configured thresholds on multicast tunnels and multicast-capable links. Where such a threshold is configured, the router will decrement the TTL, as with unicast packets, but then will drop the packet if the TTL is less than the configured threshold. When these thresholds are chosen consistently at all of the borders to a region, they allow a host within that region to send traffic with a TTL less than the threshold, and to know that the traffic will not escape that region.

An example is the multicast tunnels and links to and from Europe, which are all configured with a TTL threshold of 64. Any site within Europe that wishes to send traffic that does not escape Europe can send with a TTL of less than 64 and be sure that its traffic does not escape.

However, there are also likely to be thresholds configured within a particular scope zone—for example, most European countries use a threshold of 48 on international links within Europe, and because TTL is still decremented each time the packet is forwarded, it is good practice to send European traffic with a TTL of 63, a scenario that allows the packet to travel 15 hops before it would fail to cross a European international link.

Administrative Scoping

In some circumstances it is difficult to consistently choose TTL thresholds to perform the desired scoping. In particular, it is impossible to configure overlapping scope regions as shown in Figure 5, and TTL scoping has numerous other problems, so more recently, administrative scoping has been added to the multicast forwarding code in *mrouted* and in most router implementations.

Administrative scoping allows the configuration of a boundary by specifying a range of multicast addresses that will not be forwarded across that boundary in either direction.

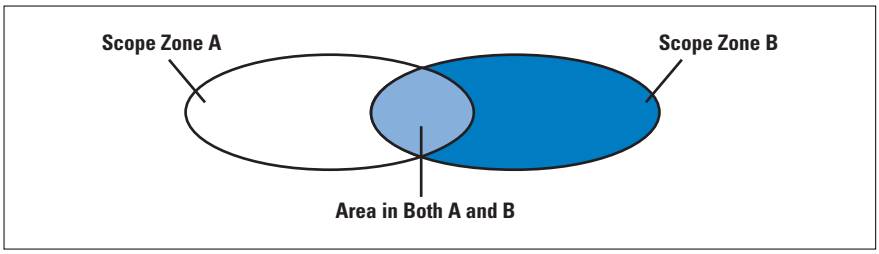


Figure 5: Overlapping Scope Zones possible with Administrative Scoping

Scoping Deployment

Administrative scoping is much more flexible than TTL scoping, but it has many disadvantages. In particular, it is not possible to tell from the address of a packet where it will go unless all the scope zones that the sender is within are known. Also, because administrative boundaries are bidirectional, one scope zone nested within or overlapping another must have totally separate address ranges. This makes address allocation difficult from an administrative point of view, because the ranges ought to be allocated on a top-down basis (largest zone first) in a network where there is no appropriate top-level allocation authority. Finally, it is easy to misconfigure a boundary by omitting or incorrectly configuring one of the routers. With TTL scoping it is likely that in many cases a more distant threshold will perform a similar task, lessening the consequences, but with administrative scoping, there is less likelihood that this scenario will occur.

For these reasons, administrative scoping has been viewed by many network administrators as a speciality solution to difficult configuration problems, rather than as a replacement for TTL scoping, and the Mbone still very much relies on TTL scoping. However, this situation is set to change as a protocol for automatically discovering scope zones (and scope zone misconfigurations) starts to be deployed. This protocol is called the *Multicast Zone Announcement Protocol* (MZAP)[9], and it will shortly become an IETF Proposed Standard. Eventually the use of configured TTL scopes to restrict traffic will cease to be used as a primary scoping mechanism.

Summary

In this article we have looked at the various routing systems that are used to devise delivery trees over which multimedia data can be sent for the purposes of group communication, and at address allocation and scoping mechanisms for this traffic.

After ten years of experimentation, IP multicast is not currently a ubiquitous service on the public Internet, but significant deployment has taken place on private intranets. The existing multicast routing and address allocation mechanisms work well at the scale of domains. However, as we have seen, there are still significant technical problems

concerning scaling to be overcome before multicast can be a ubiquitous interdomain service. In addition to the routing problems, we also still lack deployed congestion control mechanisms for multicast traffic, which are essential if multicast applications are to be safely deployed.

Despite these issues, IP multicast still shows great promise for many applications. Solutions have been devised to many of the remaining problems, although they have not yet been deployed. In the second of these articles, we will look at the proposed solutions for scalable interdomain routing and address allocation. We will also touch on multicast congestion control and the solutions that are currently emerging from the research community.

Document Status

A list of IETF specifications for the protocols discussed in this article is given below. We include the status for each document as of this writing (November 1999). For more information, check the IETF Web pages at **`www.ietf.org`**

Document	Status
IGMP v1	IETF Standard (RFC 1112)
IGMP v2	IETF Proposed Standard (RFC 2236)
IGMP v3	IETF work in progress
DVMRP	IETF Experimental Standard (RFC 1075)
PIM-Dense Mode	IETF work in progress
Multicast OSPF	IETF Proposed Standard (RFC 1584)
Core Based Trees	IETF Experimental Standard (RFC 2201)
PIM Sparse-Mode	IETF Experimental Standard (RFC 2362)
Multiprotocol BGP	IETF Proposed Standard (RFC 2283)
MSDP	IETF work in progress
MADCAP	IETF Proposed Standard (RFC 2730)
Glop Addressing	IETF work in progress

References

[1] Ballardie, A., "Core Based Trees (CBT version 2) Multicast Routing," RFC 2189, September 1997.

[2] Bates, T., Chandra, R., Katz, D., and Rekhter, Y., "Multiprotocol Extensions for BGP-4," RFC 2283, February 1998.

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

[4] Deering, S., Partridge, C., and Waitzman, D., "Distance Vector Multicast Routing Protocol," RFC 1075, November 1988.

[5] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Helmy, A., Meyer, D., and Wei, L., "Protocol Independent Multicast Version 2 Dense Mode Specification," Internet Draft, work in progress.

[6] Droms, R., "Dynamic Host Configuration Protocol," RFC 1531, October 1993.

[7] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., and Wei, L., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, June 1998.

[8] Farinacci D. et al. "Multicast Source Discovery Protocol (MSDP)," Internet Draft, work in progress, June 1998.

[9] Handley, M., Thaler, D., and Kermode, R., "Multicast-Scope Zone Announcement Protocol (MZAP)," Internet Draft, work in progress.

[10] Hanna, S., Patel, M., and Shah, M., "Multicast Address Dynamic Client Allocation Protocol (MADCAP)," RFC 2730, December 1999.

[11] Moy, J., "OSPF Version 2," RFC 2328, April 1998.

[12] Moy, J., "Multicast Extensions to OSPF," RFC 1584, March 1994.

[13] Miller, C. K., "Reliable Multicast Protocols and Applications," *The Internet Protocol Journal*, Volume 1, No. 2, September 1998.

JON CROWCROFT is a professor of networked systems in the Department of Computer Science, University College London, where he is responsible for a number of European and U.S. funded research projects in Multi-media Communications. He has been working in these areas for over 18 years. He graduated in Physics from Trinity College, Cambridge University, in 1979, and gained his MSc in Computing in 1981, and PhD in 1993. He is a member of the ACM, the British Computer Society, and is a Fellow of the IEE and a senior member of the IEEE. He is a member of the Internet Architecture Board (IAB) and was general chair for the ACM SIGCOMM from 1995 to 1999. He is also on the editorial team for the ACM/IEEE *Transactions on Networks*. With Mark Handley, he is the co-author of *WWW: Beneath the Surf* (UCL Press); he also authored *Open Distributed Systems* (UCL Press/Artech House), and with Mark Handley and Ian Wakeman, a third book, *Internetworking Multimedia* (Morgan Kaufmann Publishers), published in October 1999.
E-mail: **J.Crowcroft@cs.ucl.ac.uk**

MARK HANDLEY received his BSc in Computer Science with Electrical Engineering from University College London in 1988 and his PhD from UCL in 1997. For his PhD he studied multicast-based multimedia conferencing systems, and was technical director of the European Union funded MICE and MERCI multimedia conferencing projects. After two years working for the University of Southern California's Information Sciences Institute, he moved to Berkeley to join the new AT&T Center for Internet Research at ICSI (ACIRI). Most of his work is in the areas of scalable multimedia conferencing systems, reliable multicast protocols, multicast routing and address allocation, and network simulation and visualisation. He is co-chair of the IETF Multiparty Multimedia Session Control working group and the IRTF Reliable Multicast Research Group.
E-mail: **mjh@aciri.org**

[This article is based in part on material in *Internetworking Multimedia* by Jon Crowcroft, Mark Handley, and Ian Wakeman, ISBN 1-55860-584-3, published by Morgan Kaufmann in 1999. Used with permission].

The Internet2 Project

by Larry Dunn, Cisco Systems

Communication, connectivity, education, entertainment, e-commerce—across a broad spectrum of activities, the commodity Internet has made a strong impact on the way we live, work, and play. Nevertheless, many classes of applications do not yet run well, and some don't run at all, over the commodity net. As new applications are developed in disciplines from medicine to engineering to the arts and sciences, their success increasingly depends on an ability to use networks effectively. In research and education collaborations all over the world, efforts are under way to make use of new network technologies and develop network services that will facilitate these advanced applications. One such effort in the United States is called the *Internet2* Project[1].

The Internet2 Project was started in 1996 by 34 U.S. research universities. It has since grown to over 140 universities, and includes several corporate members and international partners. This article examines network technology used in Internet2, and looks at some of the engineering challenges involved in facilitating applications being developed by Internet2 members.

Background

In 1995, the U.S. National Science Foundation (NSF) funded a program to create the *very-high-performance Backbone Network Service* (vBNS)[2]. The NSF provided funding to MCI, who interconnected five U.S. supercomputer centers and 3 *Network Access Points* (NAPs), where it was envisioned that supercomputer clients and other vBNS users would connect.

By 1996, congestion stemming from academic traffic to the commodity Internet had seriously congested the NAPs; it was accordingly recognized that clients of the supercomputer centers might be better served if the Research Universities, where Principal Investigators often resided, were themselves *directly* connected to the vBNS. So in 1996, the NSF accepted proposals as part of the *High-Performance Connections* (HPC) program[3]. Schools applying for an HPC grant might receive $350,000 over a 2-year period, provided their proposals met various criteria, including meritorious research that would benefit from the high-performance connection, a solid network plan, intention to investigate capabilities enabled by such a connection, commitment to share results with the community, matching funds from the University, and so on.

In October 1996, representatives from 34 universities met, and concluded that, while not all the schools had projects involving "meritorious research" that would meet the NSF criteria, they all *did* have a critical interest in deploying the kind of applications that such high-performance connections could enable.

Thus, to facilitate development and deployment of applications that would further the research and education mission of member universities, the Internet2 Project was formed.

From the beginning, the stated intention was to enable applications that could not run, or could not run well, on the "Commodity Internet." Networks would be utilized or constructed only so as to facilitate this applications-enabling goal, and results/methods would be applied to the broader community as rapidly as possible.

Applications Focus

The list of applications being used or developed by Internet2 members is extensive. Several fall in the category of "meritorious research" as mentioned in the NSF HPC criteria. Examples include: remote instrument control (for instance, telescopes, microscopes), high-performance distributed computation, and large-scale database navigation. Other applications that further the education mission of member universities include tools to facilitate multisite collaboration, and asynchronous learning. Many examples in areas from science, engineering, art, language, music, and more can be found at the Internet2 applications Web site[4]. In addition to individual applications, a couple of broad initiatives have a relationship with Internet2, including *The Internet2 Digital Video Initiative,* housed at the *International Center for Advanced Internet Research* (iCAIR)[5], and the *Internet2 Distributed Storage Infrastructure Initiative* (I2-DSI)[6].

The above applications share several challenging requirements, many of which translate to resource commitments that must be met by the network in an end-to-end fashion, including bandwidth and jitter. Additionally, the applications can become scalable only if more-mature middleware and control-plane infrastructure is developed. Necessary components include features such as *Authentication, Authorization, and Accounting* (AAA), scheduling, and coordination of resources managed by multiple administrative domains.

One compelling example of the network challenges present in a virtual collaborative environment is exemplified by a CAVE (*Cave Automated VR Environment*). See [7] for more details, but in brief, a single CAVE is a (10 x 10 x 10)-foot cube, with one wall removed. Users enter through the open wall, and using lightweight stereo-three-dimensional (3D) glasses, and a radio frequency (RF) mouse, can interact with an immersive environment created by rear-screen and direct projection on multiple walls and the floor. As an example, the interconnection of multiple CAVEs allows design teams in remote locations to jointly experience the operating "feel" of a new vehicle, and to dynamically adjust, design, or control parameters to see how the modified vehicle behaves.

The developers of CAVE software at Argonne National Labs have noted that the data flows in a CAVE consist of at least: control, text, audio, video, tracking, database, simulation, haptic, and rendering flows. Additionally, they have estimated the latency, jitter, and bandwidth requirements for these flows. Some of the flows represent a challenge in a single resource dimension, others have strict requirements in multiple resource dimensions.

Backbone Networks

At this time, Internet2 members may connect to either of two backbone networks, or both.

The vBNS is operated by MCI/Worldcom. It consists primarily of an IP-over-ATM network. Most schools connect at DS3 or OC-3c via ATM to a vBNS ATM switch. Interior vBNS links are OC-12c ATM. The schools peer with a Layer 3 router; a router is attached to each of the vBNS ATM switches. The vBNS routers are logically connected to each other via a full mesh of *Unspecified Bit Rate* (UBR) *Virtual Circuits* (VCs). The ATM switches are connected to each other via a second layer of ATM switches, which are part of MCI's commercial Hyperstream offering. While schools pass the vast majority of their traffic via peering at Layer 3 with the nearest vBNS border router, other services are available, including the option to establish *Variable Bit Rate* (VBR) VCs as needed, and the possibility to place some of the ATM-attached hosts of the school directly in a vBNS Classical IP *Logical IP Subnet* (LIS). This setup allows such hosts to send bytes directly to other ATM-attached hosts, bypassing the routers of both the school and the vBNS. The vBNS also carries native IP multicast traffic among members. In addition, the vBNS has a native IPv6 offering, which is achieved by deploying routers that run IPv6, and provisioning VCs to schools also running IPv6. The vBNS has also begun to offer an *IP-over-Synchronous Optical Network* (SONET) service, the first instance of which is an OC-48 *Packet-over-SONET* (POS) link from Northern to Southern California. Because the nominal partnership arrangement with the NSF expires in the year 2000, the vBNS has established a new network offering [called *Next Generation Network* (NGN)], to which schools and other entities may connect if the vBNS/NSF partnership is not renewed.

Measurement Tools in vBNS

One of the outcomes of the vBNS program has been the development of a variety of high-performance measurement tools. One such tool, called *OC-3mon* (and now, *OC-xMon*), was developed to allow passive capture (using optical splitters) of ATM cell and IP header information, to facilitate high-speed flow characterization. More detail is available at the vBNS Web site[2]. Recently, further development of OC-xMon has been undertaken by the *Cooperative Association for Internet Data Analysis* (CAIDA)[8]. CAIDA has perhaps the best collection of high-performance public-domain measurement and analysis tools in the world, and its Web site is definitely worth browsing.

The second backbone network to which Internet2 members can connect is called *Abilene*[9]. Abilene was constructed by the *University Corporation for Advanced Internet Development* (UCAID) in collaboration with three industrial partners and Indiana University (IU). Partner contributions include fiber capacity from Qwest, SONET gear from Nortel, and routers from Cisco. The Abilene Network Operations Center (NOC) is staffed and operated by Indiana University. The network uses OC-48c POS interior links that initially connect ten routers in a partial mesh (a few interior links started as OC-12c, but are being upgraded). Abilene participants can connect at OC-3c or OC-12c, using either POS or ATM. See [10] for details on the router hardware architecture. For an insightful look at a research project that shows how this architecture can scale, see the second link in[10] and also see Stanford Professor Nick McKeown's Tiny Tera homepage at[11].

Measurement Tools in Abilene

It's worth spending a bit of time at the Abilene NOC Web site[12]. One of the interesting tools developed there is the "Abilene Weather Map"[13]. Abilene NOC has indicated that it will make source code for this tool available to Internet2 schools.

Gigapop Technology Survey

Internet2 schools can connect to either vBNS or Abilene directly. However, it is also common for several schools to converge their links at a "gigapop." This gigapop then connects to Abilene and/or vBNS, and possibly to commodity Internet Service Providers (ISPs) (to carry the "Commodity Internet" traffic of the school). Additionally, non-Internet2 schools, libraries, K-12, and state government networks also often converge at gigapops. Non-Internet2 schools typically don't forward traffic over Abilene or vBNS. But the common meeting point allows local exchange of local traffic, often affords larger aggregate commodity Internet connectivity for the gigapop participants, and allows direct access to other services that might be offered at the gigapop (Web caching, and so on).

The connectivity architecture used at gigapops varies widely. Detailed documentation for several gigapops can be found at[14]. Some Gigapops are "Layer 2," meaning that each participant is responsible for exchanging routes and traffic among themselves directly. More often, gigapops are "Layer 3," meaning that the gigapop provides a router with which gigapop participants peer. The gigapop router then typically exchanges traffic with vBNS and/or Abilene, and possible commodity ISPs.

Some gigapops are implemented at a single site (for instance, *Metropolitan Research and Education Network* [MREN], *Southern Crossroads* [SoX]), while others are "distributed gigapops," meaning gigapop equipment exists at multiple locations (for instance, the *California Research and Educational Network* [CalREN-2], and *The Great Plains Network* [GPN]). Following are a couple of specific gigapop examples.

MREN

The MREN[15] is built on a Layer 2 gigapop near Chicago that joins schools and research facilities from Illinois and several states in the Midwest. MREN members typically connect with OC-3c ATM links. Since MREN is a Layer 2 gigapop, the border router of each member peers directly with the border routers of other members. Additionally, each member's border router might peer with the Chicago-area vBNS or Abilene border router. vBNS and Abilene routers (as well as several other national research and international networks) peer here. Physically, the facility is built upon the Network Access Point (NAP) facility provided by Ameritech Advanced Data Services (AADS)[16]. Routers typically peer with each other via ATM UBR *Permanent Virtual Paths* (PVPs), although other arrangements are possible.

CENIC/CalREN-2

The Corporation for Education Network Initiatives in California (CENIC)[17] has constructed CalREN-2. The CalREN-2 distributed gigapop is interesting in several respects. First, as the name implies, it represents a distributed gigapop. In this case, three separate SONET ring facilities provide connectivity for Northern, Central (Los Angeles area), and Southern California schools. These three regions are linked to each other, and also to external networks.

Second, in each ring, there are two sets of OC-12c connections to each adjacent school. CalREN-2 has currently utilized these connections to construct both a ring of ATM connectivity, and a separate, parallel ring of POS connectivity. As a result, CalREN-2 is uniquely positioned to experiment simultaneously with both ATM and POS connectivity, performance, and QoS characteristics.

Third, to take the Northern schools as an example, the ring structure allows for a variety of Layer 3 topologies to be explored. For example, in a ring with these size and bandwidth characteristics, what are the tradeoffs on application-level performance of inducing more hops while keeping the per-hop bandwidth high, versus dividing the bandwidth into smaller slices but creating a partial mesh that reduces the average Layer 3 hop count?

Engineering Challenges

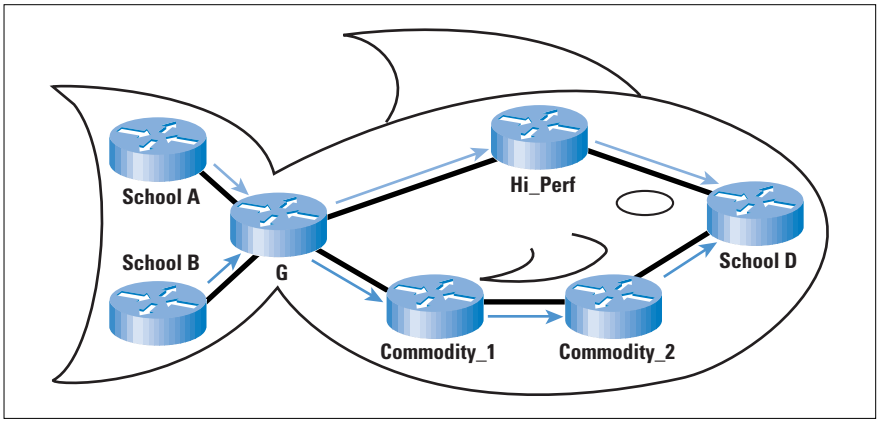
This section looks at some of the engineering challenges present in Internet2. They revolve around enabling applications with new network services, implementing appropriate policy, and doing all of this at high speed. Specifically, we'll look at Explicit Routing, Multicast, and Quality of Service.

Explicit Routing—The Fish Problem

The condition that several schools often converge at a gigapop, combined with the constraint that sometimes the funders of high-performance connectivity require that only the funded schools are allowed to use the high-performance connection, gives rise to a need for

"explicit routing" at the gigapop. The gigapop can forward packets through either a high-performance connection, or through the commodity Internet. Usually, for a single destination, traditional routing would have the gigapop use the "best" path to forward all packets to a particular destination. But when multiple policies must be implemented at the gigapop, the gigapop router must be able to "override" normal routing and forward packets on a path that's not the "best." A concrete example is shown in Figure 1.

Figure 1: The "Fish Problem"



Consider packets from schools A and B, both headed for destination D. Assume both schools are connected to gigapop G, and that G has two paths to D; one along *G-Hi_perf-D,* and the other along *G-Commodity_1-Commodity_2-D*. Further assume that A is allowed to use either path (and would perfer *G-Hi_perf-D*), but that B is prohibited from using *G-Hi_perf-D*. This scenario describes the "Explicit Routing Problem," and since it is often drawn in a shape resembling a fish, is also known as the "Fish Problem." The essence is that a routing decision at G must be made on some other criteria than just the destination IP address.

A couple of solutions to the fish problem have been used in the past, but they tend to have problems with either speed or scalability. For example, "policy routing," which usually includes a method to look at both source and destination address, has historically shown low performance. Inserting ATM switches and using virtual circuits has been used in some cases, but this solution has scaling problems and requires extra equipment. Today, many Internet2 gigapops use a separate router per policy. In the case of needing two policies in the example above, this means two routers. This solution is expensive, but does have high performance.

One promising idea is to implement enough of the "policy routing" process in hardware to allow high-speed *source+destination+other_bits* lookups. While straightforward in concept, some point out that even with line-rate source-address routing capability, the method is flawed because it requires significant manual configuration, and is prone to creating black holes for traffic upon link failure. Proponents suggest that these shortcomings can be overcome.

Another promising mechanism is becoming available as a result of work done to facilitate *Multiprotocol Label Switching* (MPLS) in routers and switches. The idea here is that one of the underlying pieces of technology required for MPLS is "multi-FIB" (multiple *Forwarding Information Bases*). Instead of the traditional "single-FIB," which always uses "the best" route to a destination, multi-FIB allows multiple forwarding tables to exist in a single router. This setup will allow a gigapop to implement multiple policies in one router, rather than the "one box per policy" that several gigapops have used previously. Note that in the case of a gigapop with a single router on which all members converge, multiple policies can be achieved with multi-FIB without actually using MPLS *Label-Switched Paths* (LSPs). For more complex gigapops, where members themselves may converge high-performance-eligible and ineligible traffic before forwarding on a single link to the gigapop, one might consider using simple LSPs to present the gigapop with traffic that is predifferentiated.

Multicast

Many of the applications in Internet2 schools use multicast. In addition to flows for videoconferencing or distance learning that use MPEG-1 (or slower) rates, a wide variety of applications require high-performance, scalable multicast. Examples include high-resolution immersive environments, collaborative real-time medical image diagnosis, and high-fidelity conferencing or distance learning (for instant, digital video camera rates of 30 mbps). When the Internet2 project began, many schools were on the *Multicast Backbone* (Mbone), and used *Distance Vector Multicast Routing Protocol* (DVMRP) tunnels to participate in multicast. Over the past year, one of the strong areas of collaboration between the Internet2 schools and the vendor community has been to develop and implement a migration strategy that allows Internet2 backbones, gigapops, and schools to move toward high-performance, scalable, native multicast support.

At the Internet2 conference in San Francisco in September 1998, the vBNS backbone was exposed to unprecedented levels of multicast stress. In a somewhat painful, but worthwhile, learning experience, it was concluded that *Protocol Independent Multicast-Dense Mode* (PIM-DM) did not scale well in highly meshed, high bitrate backbones. As a result, the vBNS has shifted to *PIM-Sparse Mode* (PIM-SM), and Abilene is being constructed with PIM-SM.

The current set of multicast components being applied in Internet2 (and leading ISPs) include: PIM-SM, *Multicast Border Gateway Protocol* (MBGP), and the *Multicast Source Discovery Protocol* (MSDP). MBGP allows distribution of routing information such that unicast and multicast routing can use noncongruent topologies.

MSDP allows independent domains to exchange information about multicast sources without creating interdomain Rendezvous Point (RP) dependencies. As they become standardized, it is expected that the *Border Gateway Multicast Protocol* (BGMP) and *Multicast Address Set Claim* (MASC) will be added to this infrastructure set.

Quality of Service

An area of broad interest in the Internet2 community centers on *Quality of Service* (QoS). The heart of QoS involves establishing strategies through which applications can be assured access to appropriate network resources when required. Typical examples of resources include end-to-end bandwidth, latency, or jitter. Of course, collateral issues and dimensions abound, including end-to-end vs. segment-only QoS; signaled vs. static provisioning; amount of state required by various approaches; level of granularity, precision, and strength of QoS "guarantee;" AAA issues; and reliability and recovery dynamics.

In an effort to start small, but make concrete progress, the Internet2 QoS working group[18] has launched an experiment called the *Qbone*[19]. Participants include backbone networks, gigapops, and individual schools and research labs worldwide. The Qbone will focus on deploying and using components developed by the Internet Engineering Task Force's (IETF) *Differentiated Services* working group (Diffserv)[20].

The initial Qbone plan is to deploy an approximation to the Expedited Forwarding (EF)[21] forwarding behavior. The Qbone will start by statically allocating a small amount of EF bandwidth across boundaries between Autonomous Systems (ASs) to allow small EF flows among arbitrary combinations of schools/labs. Large flows, in these early stages, will have to be handled manually (much as they are today). In later stages the plan is to use *Bandwidth Brokers* (BBs) currently under development[22] to aid in the automation of adjusting resource commitments between ASs (using interdomain BBs), and to aid in accepting application resource requests (using intradomain BBs, combined with policy servers and AAA mechanisms). The precise mechanics for BB interaction, trade-offs among signaling frequency, amount of state, scalability, and so on are certainly topics of research, but that's part of what makes Qbone participation fun!

Summary

There is no single application or technology that makes Internet2 unique or exciting. Rather, the effort required to enable new applications that have strong bandwidth, latency, jitter, and coordination requirements has resulted in an infusion of energy from a variety of disciplines. Internet2 requires stretching existing technologies (ATM, POS, multicast, measurement), nurturing developing technologies (Quality of Service, explicit routing, Dense Wave-Division Multiplexing [DWDM], mobility), and participating in the invention of new technologies (all-optical infrastructures, extending AAA, and other resource allocation and

scheduling middleware). Internet2 requires attention to maturing components in backbone, gigapop, and campus environments in order to deliver on the promise of speedy transference of lessons learned to the commodity Internet. The effort so far has resulted in demonstration of truly stunning, impactful, and useful applications. It is the convergence of effort and rapid rate of change that makes Internet2 a challenging and rewarding endeavor.

Other Initiatives

Although this article has focused on aspects of Internet2 in the United States, there are many advanced Internet activities around the world. A partial list includes:

http://www.dante.net/ten-155.html (Europe)
http://www.ukerna.ac.uk (UK)
http://www.dfn.de (Germany)
http://www.renater.fr (France)
http://www.surfnet.nl (The Netherlands)
http://apan.or.kr (Asia/Pacific)
http://www.singaren.net.sg (Singapore)
http://www.canet3.net (Canada)
http://www.cudi.edu.mx (Mexico)
http://www.reuna.cl (Chile)
http://www.ngi.gov (U.S. Federal)
http://www.startap.net (International peering)

A more complete list of advanced Internet initiatives is maintained at:
http://www.cisco.com/aii

References

[1] **http://www.internet2.edu**

[2] **http://www.vbns.net**

[3] See latest press release at:
http://www.nsf.gov/od/lpa/news/press/99/pr9915.htm
...and updated program announcement at:
http://www.nsf.gov/pubs/1998/nsf98102/nsf98102.txt

[4] **http://apps.internet2.edu**

[5] **http://i2dv.nwu.icair.org/** and **http://www.icair.org/**

[6] **http://dsi.internet2.edu/**

[7] **http://evlweb.eecs.uic.edu/pape/CAVE**
...has a great introduction to CAVE technology.
Also see the *Electronic Visualization Laboratory* homepage at:
http://www.evl.uic.edu/EVL/index.html

[8] **http://www.caida.org**

[9] **http://www.ucaid.org**, and Abilene specifics at:
http://www.internet2.edu/abilene

Abilene router details are at:
[10] **http://www.cisco.com/warp/public/cc/cisco/mkt/core/12000/index.shtml**
...and Nick McKeown's paper is at:
http://www.cisco.com/warp/public/cc/cisco/mkt/core/12000/tech/fasts_wp.pdf

[11] **http://tiny-tera.stanford.edu/tiny-tera/index.html**

[12] **http://www.abilene.iu.edu**

[13] **http://hydra.uits.iu.edu/~abilene/traffic**

[14] Following are several gigapop sites:
California's CENIC/CalREN2: **http://www.cenic.org,**
The Pacific/Northwest gigapop: **http://www.pnw-gigapop.net**
The Great Plains Network: **http://www.greatplains.net**
The Southern Crossroads, with members from Southeastern Universities Research Association: **http://www.sox.net**
MidAtlantic Crossroads: **http://www.networkvirginia.net/MAX**
MREN: **http://www.mren.org**
WestNet: **http://www.scd.ucar.edu/nets/Projects/Westnet**
North Carolina Gigapop: **http://www.ncni.net**
The Texas Gigapop: **http://noc.gigapop.gen.tx.us**
Northern Crossroads: **http://www.nox.org**
Philadelphia area Magpi: **http://www.magpi.net**
Pittsburgh-based NCNE: **http://www.ncne.net**
New York: **http://www.nysernet.org**

[15] **http://www.mren.org**

[16] **http://www.aads.net,** and **http://nap.aads.net/main.html**

[17] **http://www.cenic.org**

[18] **http://www.internet2.edu/qos/wg**

[19] **http://www.internet2.edu/qos/qbone**

[20] **http://www.ietf.org/html.charters/diffserv-charter.html**

[21] **http://www.ietf.org/rfc/rfc2598.txt**

[22] **http://www.merit.edu/working.groups/i2-qbone-bb**

LARRY DUNN is the Technology Development Manager in the Advanced Internet Initiatives Division at Cisco Systems. He serves on the Internet2 Quality of Service and Routing working groups. After receiving his PhD from the University of Minnesota (Electrical Engineering '92), he served as Director of Networking there, and subsequently as Director of Strategic Markets and Applications (Education) for FORE Systems. He periodically teaches Advanced Networking courses at the University of Minnesota. Research interests include test vector generation for combinational logic, network design and analysis, and Quality of Service techniques and deployment strategies.
E-mail: **ldunn@cisco.com**

One Byte at a Time: Internet Addressing

by Peter H. Salus

The source of all knowledge where the Internet is concerned is the set of *Requests for Comments* (RFCs). Because there are now well over 2,700 RFCs, however, only a few people track history, evolution, and outright paradigm shift.

Each node on the Internet—router or end system (often called "host" or "server")—has a unique identifier attached to it; this identifier is its *address*. Any packet sent between nodes must use the destination address to tell the intervening routers where it should go.

In RFC 1 (April 1969), Steve Crocker laid out a scheme that allotted five bits to address space: enough for 32 addresses. By September 1969, when *Interface Message Processor* (IMP) No. 1 was installed in Kleinrock's lab at UCLA, this number had grown to six bits (63 addresses). By 1972, it had become apparent that this number would be insufficient, and the address space was enlarged to eight bits (255 addresses). In fact, the *Advanced Research Projects Agency Network* (ARPANET) hit only 63 hosts in January 1976. This number was, however, already a lot in terms of the `HOSTS.TXT` tables that were distributed to every site. By August 1983, there were 213 hosts, and the eight-bit address barrier was being pushed.

Cerf's original version of TCP (RFC 675; December 1974) and Postel's of IP (RFC 760; January 1980) increased this "address space" to 32 bits, but the structure of the ARPANET was "flat," that is, the hierarchical distributed name-to-address database we are familiar with only came about with Mills' conceptualization of the *Domain Name System* (DNS) (RFC 799; September 1981), and its implementation by Paul Mockapetris (RFCs 882 and 883; November 1983).

Address Classes

The Internet Protocol uses a 32-bit addressing scheme and originally four classes of networks: A, B, C, D. (See Figure 1 on page 5). There are only 128 Class A networks, but each can have 16,777,216 unique host identifiers. Next, there are 16,384 Class B networks, with 65,535 unique identifiers; 2,097,192 Class C networks, with 255 hosts; and over 268 million Class D multicast groups. (A fifth class, Class E, is reserved and not available for general use).

Address Depletion

Using the 32-bit IP addressing scheme allowed for about 4 billion hosts on 16.7 million networks. Although this number of various kinds of addresses seemed like a lot, the expansion of the use of the Internet over the past decade has been explosive, and the original address classes did not allow for a flexible address assignment based on an organization's particular need.

In August 1990 during the Vancouver *Internet Engineering Task Force* (IETF) meeting, Frank Solensky, Phill Gross, and Sue Hares projected that the current rate of assignment would exhaust the Class B space by March 1994.

CIDR

Classless Inter-Domain Routing or CIDR (RFCs 1518 and 1519; September 1993) was introduced to improve both routing scalability and address space utilization in the Internet. By eliminating the notion of "network classes," CIDR allows for a better match between address requirements and address allocation. This results in expansion of the scope of hierarchical routing, which in turn improves scaling properties of the Internet routing system. CIDR has proven to be the palliative that has enabled the Internet to continue functioning while growth continues.

Even with this palliative, it was predicted in 1994 that, using the current allocation statistics, the Internet will exhaust the IPv4 address space between 2005 and 2011. With five more years of experience, which has also brought greater uncertainty as to gross numbers, we can push these dates out a bit, but exhaustion will come eventually.

Another factor that has slowed down the address depletion rate is the use of *Network Address Translation* (NAT). NAT devices allows an organization to have one external ("public") address and many private (net 10 is often used) addresses internally. Since the internal addresses are not "seen" from the outside, they do not need to be globally unique. This approach has downsides (some protocols weren't designed with NATs in mind), but from the address depletion point of view, it is a win. RFC 1597 describes "Address Allocation for Private Internets."

If you are interested in current Internet addressing, an excellent book is available: *TCP/IP Addressing,* by Buck Graham, AP Professional, 1997. Graham does an excellent job on addressing, routing, and the various bizarries involved in optimal routing, efficient use of address space, and making network management less onerous. This book is, however, not intended to be for elementary instruction; Graham primarily speaks to the professional market.

IPng aka IPv6

In the summer of 1994, the IETF set up an Internet Protocol next generation (IPng) task force, cochaired by Scott Bradner and Allison Mankin. (IPng later became known as IPv6 for "IP version 6"). Recommendations from that task force were released in October 1994 for discussion at the December 1994 IETF meeting. The basic goal was to have something in place before 2000, so that the time limit would not be pushed.

Unfortunately, as Bradner and Mankin stated in their recommendation: "Some people pointed out that this type of projection makes an assumption of no paradigm shifts in IP usage. If someone were to develop a new 'killer application,' (for example, cable-TV set top boxes), the resultant rise in the demand for IP addresses could make this an over-estimate of the time available."

IPv6 provides for 128-bit addressing. This number is gigantic: larger than the estimated total number of molecules in the universe.

Books

Two noteworthy books are available on IPv6 itself: Christian Huitema's *IPv6: The New Internet Protocol* (ISBN 0-13-241936-X, Prentice Hall, 1996) and Scott Bradner and Allison Mankin's anthology *IPng* (ISBN 0-201-63395-7, Addison-Wesley, 1996), which provides an explanation of the task force's process and explicates the services that are provided for (as, for example, ATM support). These books are both dated, but they are the best available now. Keeping up with what's going on is easy, thanks to the IETF's Web site **`http://www.ietf.org`**.

An excellent business and technical case for IPv6 is found in the Internet Architecture Board draft by Steve King and several colleagues (**`draft-iab-case-for-ipv6-05.txt`**). Other works in progress deal with the adjustments to Open Shortest Path First (OSPF), multicasting, mobility, and so on.

Transition

The period from 1981 through 1983—the time of conversion to DNS—was painful to all concerned. Over the past 15 years we have learned a lot, but the switch from IPv4 to IPv6 may be yet more painful. The drafts tell the tale of those who are striving to make things easier.

There has been much discussion about various kinds of transition mechanisms, and some of these may be less painful (more automated) than we might at first think. Remember, this pain is not because of the innate difficulty, but veering a ship that carries fewer than 250 passengers is far easier than veering a ship that carries 60 million. Some members of the community think that the pain may not justify the gain. The author is not one of them. It has been nearly 20 years since TCP/IP was made official, yet there are still UUCP networks.

In the author's opinion, IPv6 will be here in a few years, if not sooner.

Reference

[1] Fink, R., "IPv6—What and Where It Is," *The Internet Protocol Journal*, Volume 2, No. 1, March 1999.

PETER H. SALUS is the author of *A Quarter Century of UNIX* (1994) and *Casting the Net: From ARPANET to Internet and Beyond* (1995). He is the Editor in Chief of *The Handbook of Programming Languages* (1998). His e-mail address is:
`peter@pedant.com`

Book Review

An Engineering Approach to Computer Networking

An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network, Srinivasan Keshav, ISBN 0-201-63442-2, Addison-Wesley, 1997, **http://www.awl.com/cseng/titles/0-201-63442-2/**

The rapid convergence of telephone and data networks brings with it a collision of two diverse approaches to fundamental network design. This "New World," as it is often called, requires us to understand both the analog-to-digital evolution of the voice network, with its redundant search for faultless reliability, and the persistent tolerance of the data network. Mirroring the industry trend, this book explores the three major networking technologies: ATM, the Internet, and telephone networks, with the idea that the design of any modern network requires consideration of the influence of at least two of the three technologies.

This book is a textbook. Keshav himself declares in the preface that "textbooks, almost by definition, tend to be boring," and the reader will recall this subtle warning shortly into Chapter 2. This is definitely a book for those who have at least an intermediate knowledge of data networking and a need to understand the component parts of network implementations. Keshav takes a true engineering approach, in that he attempts to teach the building blocks of the major networking technologies—and this approach is what makes the book one of my all-time favorites. By examining the component parts and why they are required, Keshav leaves you prepared to engineer a network that meets any number of diverse criteria.

Organization

The book is organized into three sections. Section 1 gives an introduction to the future of data and voice networks and then introduces three of the major networking technologies. This section also gives an overview of the historic construction of networks, along with some fundamental definitions of some of the engineering principles by which networks function. As early as Chapter 1, Keshav explores the engineering philosophy behind common network technologies, illustrating the theories that underlie their design. My favorite example is his suggestion that the telephone network was engineered to be intelligent because its endpoints, the telephones, are simply dumb. While this sounds obvious, it provides a fundamental perspective on the design of the system that proves invaluable to understanding the origin of the various "components" of the network.

Section 2 begins with a short but requisite review of protocol layering and, after a brief discussion of common design constraints, begins to dissect the major components required of almost any network implementation. Chapter 8 is a fairly comprehensive review of switching and, as the book's title suggests, the chapter is full of comparative anatomy. Read this chapter for its valuable insight into why various switching mechanisms have emerged and for its comparison of how various switching functions are handled on three major networking technologies. Chapter 9 deals with scheduling network resources, with an excellent comparison of the variety of scheduling mechanisms and their effect on connections and packets. It covers policy considerations that are also required of scheduling disciplines, giving the reader a set of strategies for network design. Chapter 11 covers routing of packets as well as routing in the telephone network. In my opinion, this discussion alone makes this book a required part of any networking professional's library. Admittedly, there are books that better explain routing in both of these environments, but because of the proximity of the topics, this presentation helps the reader to understand the mechanics of both systems in a way that provides insight into the inherent issues posed by both technologies.

Section 3 pulls together the various component functions discussed in Section 2 and explains some of their implementation in the form of protocols. Section 3 is a short section, probably not intended as a thorough survey of networking protocols. Keshav documents an excellent set of references for Section 3, however, and leaves it up to the reader to pursue those that are relevant to his or her professional development.

Required Reading

An Engineering Approach to Computer Networking is definitely an A+ book, and should be required reading for anyone interested in the inner workings of data and voice networks. Although the author expects the reader to absorb quite a bit in every chapter, the time spent is well invested. The book is a refreshing alternative in that it provides an answer to the question of "why" the network works rather than being another treatise on "how" the network works.

—Jim LeValley, Cisco Press
levalley@cisco.com

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

Internet Policy Institute Launched

On November 9th, 1999 a group of distinguished Internet visionaries and scholars announced the creation of the *Internet Policy Institute,* the nation's first independent, nonpartisan think tank devoted exclusively to providing research and hard data on the Internet and society. The group also announced its first research project and an initiative aimed at educating the presidential contenders.

The creation of the new think tank was announced by Jim Barksdale, former CEO of Netscape, Vint Cerf, Senior Vice President of Internet Architecture of MCI WorldCom, Esther Dyson, author and Chairman of EDventure Holdings, Inc., Mario Morino, Chairman of The Morino Institute, and Kimberly Jenkins, President of the Internet Policy Institute.

The new, nonprofit think tank will employ well-known experts and scholars to research subjects ranging from the role of the Internet in privacy to the Internet's impact on taxation and health care.

"The Internet is surrounded by noise, hype, rumors, marketing, IPOs and the hopes of starry-eyed start-ups, but there is very little hard data on which policymakers can base critical decisions that will determine the future of the new medium and how it affects society," said Barksdale, co-chairman of the Internet Policy Institute's Board of Directors. Wayne Clough, President of Georgia Tech, is his co-chairman.

"The speed at which society has adopted the Internet is unprecedented," said Cerf, who was Chairman and founding president of the Internet Society, as well as one of the designers of the TCP/IP protocol. "If, as we expect, half the world will be online within the next four years, we must make sure that the policy decisions we make now are based on solid, well-researched data."

The Institute announced its first research project, to be undertaken in collaboration with The Brookings Institution, on "The Economic Payoff from the Internet Revolution." The research will be led by Alice Rivlin, former vice chair of the Federal Reserve System's Board of Directors and former Office of Management and Budget director, now with the Brookings Institution, and Robert E. Litan, Vice President and Director of Economic Studies at The Brookings Institution and former associate director of the Office of Management and Budget. The research will produce the first comprehensive, systematic economic study by an independent research group of the subject.

The nature and extent of the impact is of special importance to macroeconomic policy—specifically monetary policy—to the extent that the Net is having or will have a material and sustained impact on the growth rate of productivity. The impact the Net has on specific industries, and the way it affects barriers to entry, has important implications for antitrust and regulatory policy.

Exactly one year before the next presidential election, the Internet Policy Institute also announced its first publications project, "Briefing the President: What the Next President of the United States Needs to Know About the Internet and Its Transformative Impact on Society." The Institute also released the introduction to the project by Barksdale, while Cerf outlined the contents of the next paper, "What is the Internet (and What Makes It Work)" that will be released December 1. Over the course of the coming months, the Institute will release 13 papers to be presented in briefings to all the leading presidential contenders and later compiled into a book.

"We didn't know five years ago the direction that the Internet would take," Barksdale said. "I'll bet that five years from now, we'll be surprised by its new directions. We need to assure that an honest, objective approach is taken on Internet issues, to prevent decision making that hinders the potential of this amazing medium," he said. For more information see: **http://www.internetpolicy.org**

APRICOT 2000

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in at the Intercontinental Hotel in Seoul, Korea from February 28th to March 2nd, 2000. APRICOT provides a forum for key Internet builders in the region to learn from their peers and other leaders in the Internet community from around the world. The week-long summit consists of seminars, workshops, tutorials, conference sessions, and birds-of-a-feather sessions—all with the goal of spreading and sharing the knowledge required to operate the Internet within the Asia Pacific region. For more information see:
http://www.apricot.net

More on Web Caching

If you enjoyed the article on Web Caching in our September 1999 issue, you might find the following paper of interest: "A Survey of Web Caching Schemes for the Internet," by Jia Wang. You can find this article in the October 1999 issue of ACM SIGCOMM's *Computer Communications Review* (Volume 29, Number 5). The paper is also available on line in either PostScript or PDF format:
http://www.acm.org/sigcomm/ccr/archive/1999/oct99/ccr9910-jia-wang.html

ICANN Update

On September 28, 1999, the United States Department of Commerce, Network Solutions, Inc. (NSI), and The Internet Corporation for Assigned Names and Numbers (ICANN) announced a series of agreements they had tentatively reached to resolve outstanding differences among the three parties. On November 4, 1999, based on public comment in writing and at a public forum held at the 1999 ICANN annual meeting, the ICANN Board approved revised versions of these agreements. The agreements were signed by the three parties on November 10, 1999. The full text of the agreements can be found on the ICANN Web site at **`www.icann.org`**. Here we include some highlights:

- NSI will operate the registry for the **`.com`**, **`.net`**, and **`.org`** top-level domains according to requirements stated in the agreement and developed in the future through the ICANN consensus-based process. All accredited registrars will have equal access to this registry.
- A revised registrar accreditation agreement between ICANN and registrars was adopted. To continue to register names with the **`.com`**, **`.net`**, and **`.org`** registry operated by NSI after November 30, 1999, registrars must have entered a new Registrar License and Agreement with NSI and the revised ICANN accreditation agreement.
- A revised NSI-Registrar License and Agreement was created under which competitive ICANN-accredited registrars are permitted to place and renew registrations in the registry.
- An amendment was made to Cooperative Agreement #NCR 92-18742 originally entered between NSI and the National Science Foundation (NSF) in 1992. On October 7, 1998, NSI and the United States Department of Commerce (which by then had assumed the NSF's role as lead agency of the U.S. Government) entered an Amendment 11 to that Cooperative Agreement under which NSI agreed to implement a shared registration system in which competitive registrars would enter registrations into the **`.com`**, **`.net`**, and **`.org`** registry on an equitable basis. Amendment 19 solidifies those arrangements and provides that in operating the registry NSI will abide by consensus policies adopted in the ICANN process.

At the annual meeting in early November, nine new directors joined the ICANN Board of Directors. They are Robert Blokzij, Ken Fockler and Pindar Wong named by the The Address Supporting Organization (ASO); Amadeu Abril i Abril, Jonathan Cohen and Alejandro Pisanty named by the Domain Name Supporting Organization (DNSO); Jean-François Abramatic, Vinton G. Cerf and Philip Davidson named by the Protocol Supporting Organization (PSO).

The newly expanded ICANN Board will take on a major challenge in 2000 in its consideration of contending proposals for the future of Top Level Domains. After years of vociferous argument, the DNS community is no closer than it ever has been to a consensus on whether new name registries should be created, and if so, with what structure and registration rules.

Interplanetary Internet Special Interest Group Formed

The Internet Society (ISOC) recently announced the formation of the Interplanetary Internet Special Interest Group (IPNSIG). The IPNSIG exists to allow public participation in the evolution of the Interplanetary Internet. The technical research into how the Earth's Internet may be extended into interplanetary space has been underway for several years as part of an international communications standardization body known as the Consultative Committee on Space Data Systems (CCSDS). (See **http://www.ccsds.org/**)

The CCSDS organization is primarily concerned with communications standardization for scientific satellites, with a primary focus on the needs of near-term missions. In order to extend this horizon out several decades, and to begin to involve the terrestrial internet research and engineering communities, a special Interplanetary Internet Study was proposed and subsequently funded in the United States.

The Interplanetary Internet Study is funded by the Defense Advanced Research Projects Agency's Next Generation Internet Initiative, and presently consists of a core team of researchers from the NASA Jet Propulsion Laboratory, MITRE Corporation, SPARTA, Global Science & Technology and consulting researchers from The University of Southern California Information Sciences Institute, University of California Los Angeles and the California Institute of Technology. The primary goal of the study is to investigate how terrestrial internet protocols and techniques may be extended and/or used as-is in the exploration of deep space. The study team has also founded the IPNSIG and has formed the core of an Interplanetary Internet Research Group under the sponsorship of the Internet Research Task Force (IRTF).

The NASA IPN Study Team will act as liaison between the satellite and space communities and the ISOC/IRTF communities. The NASA IPN Study Team will assist with requirements and understanding of the deep space environment and missions, while the primary research on new or modified protocols will be conducted by the IRTF. In addition, the NASA Study Team will also act as liaison with the CCSDS.

The NASA Study Team will also enable simulated and actual opportunities to test protocols and the use of internet techniques in the space environment. For more information, visit: **ipn.jpl.nasa.gov/**

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-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

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