The MBONE

The Internet's Other Backbone

by Jay A. Kreibich

The ``MBONE'' is the common name given to the Internet's IP Multicast Backbone. It is a virtual network that allows IP multicast messages to be sent across the Internet. The MBONE is commonly used to send live video and audio from major computer conferences, NASA shuttle missions, rock concerts, and many other events of interest to the Internet community. Anyone with a workstation and the proper network connections can listen in.

The MBONE allows many people to collaborate and view data simultaneously by using multicast network technology. Multicast network technology is not a new thing, but it has traditionally lacked an important role in the networking industry. Increasing interest in multimedia and collaborative applications has changed this and caused a recent increase in the use of multicast networks.

The most common way to move data around a network is *unicast*. Unicast network messages are sent to one and only one other computer. It is easy to compare a unicast network connection with a traditional phone call: each phone call is a communication between only two people. Although this scheme works very well for most traffic, unicast doesn't scale in a collaborative multimedia environment in which large amounts of data must be sent to many computers.

In order to understand the problem with using unicast transport schemes, suppose you wanted to use a networked video application to talk with a group of friends. For this example, we will assume all the computers are on a single link-level network (e.g. a single Ethernet) so that we do not have to take routers into account. Let's say that our collaborative video application is sending data at a constant rate of 100 kilobits (KB) each second, and that our computers are linked by a 1000 KB/sec network (roughly Ethernet speeds). If unicast network messages are being used to send the data, each packet needs to be sent to every computer. Since the network can only carry 1000 KB/sec, and you need to send 100KB/sec to every other person, only 10 other friends can watch you.

A collaborative system that is limited to 11 people is fair, but not great. The problem is that you still can't see them. If you want to truly collaborate, so that everyone can see everyone else, everybody needs to send data to everybody else. These conditions result in only three people being able to participate. With three people, each person is sending to two others, requiring 600KB/sec. Four people would require 1200KB/sec, and five people would require 2000KB/sec. It is simple to see that unicast simply does not scale past very small groups.

The other common transport scheme is *broadcast*. Each broadcast message is received by every computer on the network. Broadcast looks like an attractive alternative to unicast because the data only needs to cross the network once, but broadcast has its own share of problems. When using broadcast every computer connected to the network has to take the time and CPU cycles to collect and analyze the received data to determine whether it's useful. Computers that are not running the collaborate application can waste a lot of processor time. Keeping with the phone example, using a broadcast message is like having every phone in a whole office building on the same line. When someone calls, everyone has to take the time to pick up their phone to see if the phone call is for them. Broadcast networks are also severely limited in size because they depend on the physical link-level network supporting the ability to send broadcast messages. For single link-level networks (such as a single Ethernet) this works well but if you need to cross physical networks the protocols fall apart. Almost all networks forbid broadcast messages to cross subnets, for example.

The compromise between these two transport schemes is *multicast*. Multicast allows a single message to be sent to multiple computers, but unlike broadcast only those computers that are ``tuned in'' to the multicast group will receive the data. Multicast networks can be thought of as a phone ``party line'' or conference call for computers. A large group of computers can communicate together, but only those that ask to be part of the group are included.

Multicast works much like broadcast on a single subnet. If a host wishes to communicate to other computers, a single message is sent across the network. Many shared media networks such as Ethernet and FDDI use hardware *Media Access Control* (MAC) addresses for flow control of the link layer packet. The networking hardware uses the MAC addresses to determine whether to buffer or ignore a packet. MAC addresses allow the networking hardware to easily and quickly sort network packets without involving the main machine's CPU(s). Normally the networking hardware only accepts packets with MAC addresses that match the local address or the broadcast MAC address. When a computer becomes a member of a multicast group, it gives the networking hardware additional MAC addresses for which to watch. There is a specific mapping between most types of MAC addresses and IP group addresses that allows most of the IP group filtering to happen at the hardware level. Unlike broadcasting, this allows different multicast groups to take place on the same subnet without a drop in overall computer performance.

When transmitting within a single subnet, most multicast IP implementations are dependent on the physical link-layer network, just as broadcast messages are. This has no effect on local subnets, but if you want to send multicast IP packets between machines that are on different subnets, the messages have to be routed between all the subnets. In order to do this, the various routers that are serving the effected subnets need to communicate to each other the desire to transmit multicast packets.

In the same sense that it is very inefficient to send a broadcast message to an entire LAN, it is inefficient to send a single multicast message to every subnet in a LAN (this also defeats the point of using multicast unless someone on every subnet of the LAN wants the data). If every subnet of the LAN is thought of as a node in a graph, and edges represent router connections, the ideal solution to this

problem is simply a minimum spanning tree connecting all of the subnets using a specific multicast group. The spanning tree will keep all of the multicast subnets connected with a minimum number of jumps across non-multicasting subnets; this results in maximum efficiency. Every time a subnet is added or dropped from a group the spanning tree for that group is adjusted to maintain efficiency.

There are four major protocols that are used to route multicast IP. The first is the *Internet Group Management Protocol* (IGMP [RFC 1112]). Computers on a subnet use IGMP to inform applicable routers that they would like to use a specific IP multicast group to communicate with other hosts outside of the local subnet. The routers also need to communicate with each other to establish the spanning tree between all the routers servicing multicast traffic. The two most common protocols to do this are *Multicast OSPF* (M-OSPF [RFC 1584]), and *Protocol Independent Multicast* (PIM [IETF-IDMR Draft]).

M-OSPF attempts to keep an optimum minimum spanning tree at all times. This is the most efficient way to run a network, but also creates a lot of overhead for the routers in a very dynamic network. PIM has two basic modes knows as ``Dense Mode" and ``Sparse Mode." PIM does not attempt to keep an optimal spanning tree, but instead only recomputes effected subsections of the tree. The algorithm used to recompute the tree depends on the active mode. This does not always lead to the most efficient routing scheme, but greatly reduces the amount of overhead calculations the routers must perform.

Both M-OSPF and PIM work very well across a large LAN (a campus or large business, for example), but don't scale well beyond a few hundred routers. If all the routers on the Internet were to talk M-OSPF or PIM to each other, each would need to hold in memory the spanning tree for all the routers connected to the Internet. If this wasn't bad enough, every time a large number of hosts entered or left a multicast group the entire spanning tree of thousands of nodes would need to be recomputed -- routers would spend all of their time running Dijkstra's algorithm to recompute the spanning tree.

The answer to our problem, as it so often is in computer science, is another layer of abstraction. The original problem with multicast was that subnets could multicast within themselves, but couldn't talk to each other. This problem was fixed by having the routers set up spanning trees via M-OSPF or PIM. If all the routers of a LAN are talking M-OSPF or PIM the problem becomes a series of LANs that can multicast within themselves, but can't talk to each other.

The purpose of the MBONE is to connect all the smaller multicast aware LANs into one huge world wide virtual multicast network. This is done in the same fashion that the lower level protocols tie together the local subnets of each LAN. Each LAN becomes a node in the MBONE tree and the various sites communicate with each other using the *Distance Vector Multicast Routing Protocol* (DVMRP [RFC 1075]) to establish efficient communication pathways. DVMRP does not attempt to recompute a minimum spanning tree for the entire MBONE each time a node changes group membership, but it does monitor traffic flows and prunes those parts of the network tree that do not require specific data.

The term ``virtual network" is used to describe the MBONE because the shape and bandwidth of the

MBONE is defined by a series of permanent high-level networking ``tunnels'' between major sites on the Internet, not the physical connections that these software tunnels use. This distinction is important. When talking about the MBONE's shape, it is the software configuration of the tunnels, not the physical connections between the sites, that is important. This extra layer of abstraction effects not only network links, but also bandwidth. The bandwidth of a MBONE link is software limited to only a small fraction of the total available bandwidth. This is done to prevent breakdown of the physical networks the MBONE is using. When the MBONE is being used to send live video, it can easily eat up very large amounts of bandwidth. Limiting the link bandwidth prevents the MBONE from taking bandwidth away from other Internet users.

A typical configuration for connecting a LAN to the MBONE would involve configuring all the routers within the LAN to speak M-OSPF or PIM. This allows all the subnets of the LAN to transmit multicast to each other. In order to connect the LAN to the MBONE, a gateway machine needs to be configured to run *mrountd*. This configuration includes the assignment of DVMRP tunnels to other sites. These other sites will also have to configure their gateway machines to connect back to the LAN. The gateway machine will watch all the multicast groups that the LAN is using, and see if any other sites on the MBONE are members of those groups. If there are other sites requesting the data, mrouted encapsulates the multicast data into unicast messages, and sends them off to the sites on the other end of the tunnels. Each node of the MBONE will pass the data through it's various tunnels until the data gets where it is needed. At this point the multicast data is stripped out of the unicast message by the remote LAN's gateway machine and injected into that site's LAN for the normal routers to feed to the proper subnet and host.

By linking together various multicast networks, the MBONE creates a continuous multicast network. Assuming your campus network has a connection to the MBONE and has multicast savvy routers, multicasting around the world is as easy as down the hall. If you want to multicast something around the world, it is as easy as hooking up a camera to a workstation, picking a multicast group, and telling the software to go.

The various multicast ``groups'' are really just Class D IP addresses. Class D addresses are defined as any address that has 0xE as it's first nibble, or the addresses 224.x.x.x thru 239.x.x.x. This provides over 250 million multicast groups to choose from. Most of these addresses are reserved however, so user data that is sent across the MBONE should be restricted to the groups 224.2.x.x.

The basic concept behind groups is simple. To send data to a multicast group the data is simply addressed to the group address and then sent across the network. The routers and other machines will recognize the multicast addresses and route the packets to wherever they are needed.

The use of different group addresses help segment data. To keep subnets from getting multicast packets that are not relevant, every multicast group has it's own spanning tree. To keep efficiency high, it is standard procedure to keep different media data streams (sound, video, images, etc) on different groups. This is done so that hosts can selectively join only the groups they wish to. It is important that the

various streams use different groups, instead of simply different ports on the same group because the multicast routing algorithms segment data on the group level, not the port level. This is a very subtle effect, and is best shown with an example.

For our example we will have a site wishing to broadcast a high-bandwidth video stream and a low-bandwidth audio stream. If both streams are being broadcast to the same group, but different ports, there is no way for the routers to distinguish between the two streams. This becomes an obvious problem if there is a host on a low-bandwidth network that wishes to listen in on the broadcast. Our listener has enough bandwidth to handle the audio and not the video, so they only run the audio application. The audio application joins the multicast group which causes the routers to forward both the audio and video multicast packets in that group to the low-bandwidth subnet. This usually has the effect of flooding the network with so many video packets that enough audio packets are dropped to make it data unusable.

Now if we look at a similar example with the video and audio broadcast on different groups, things work out much better. If the streams are on different groups, when the low-bandwidth user runs the audio application their host joins the group associated with the audio data but not the video group. This causes the routers to forward the audio data to their network and host, but the subnet never gets any of the video data. Thus no data is transmitted that is not used, and efficiency is kept high. This is not to say that the use of UDP/IP port numbers is totally worthless. Using the same group but different ports for related data streams is a very useful programming technique to distinguish between related data streams. Vat (audio application), for example, sends the actual audio data on one port, and session membership information on a different port. By using different ports, the application is not required to multiplex data through a single connection, but since either data stream is somewhat meaningless without the other, there is no need to use different groups.

Although using different multicast groups helps reduce traffic, the MBONE is still very sensitive to bandwidth. As discussed earlier, the software tunnels that control the structure of the MBONE cap the bandwidth to avoid flooding the underlying networks. It only takes two or three high quality video signals to completely saturate the MBONE's bandwidth, so broadcasts are typically announced well in advance. The most important part of hosting a high-bandwidth MBONE broadcast (such as one that is going to use video) is reserving the network time and making sure your broadcast times do not conflict with other people who wish to use the MBONE.

Most traffic on the MBONE falls into a small number of specific categories. The standard video application is called *nv*, and was developed at Xerox PARC. nv understands many different video formats, and can send both video images and screen grabs. For audio there is *vat*, which was developed at Lawrence Berkeley National Laboratory. LBL also developed a collaborative white board tool known as *wb*, which allows users to draw and doodle on in a shared space. Other popular tools include *IMM* (low bandwidth still images), *IVS* (video), and *nevot* (audio). There are also many new tools in development such as *MUMBLE*, which transmits simple text, and *WebCast*, which can send WWW pages to many different viewers at the same time.

The best starting point for users wishing to use the MBONE is the Session Director, or simply *sd*. Developed at LBL, sd tracks all the current sessions on the MBONE and allows you to see at a glance all the public sessions that are currently being sent across the MBONE. For each session sd shows the various media types (video, audio, wb, etc.) and a short description of what the session is about. If you see a session that looks interesting sd can launch and configure all the appropriate viewers with a simple click.

Common channels in sd include audio from the US House of Representatives and the US Senate. NASA sends a continuous broadcast of shuttle communications whenever someone is in orbit, and most major computer conferences such as SigGraph, InterOp, and IETF meetings also send video and audio of what is going on. For something on the lighter side, check out Radio Free Vat, the Internet's very own all music radio station-- anyone can take a turn as DJ of the only world-wide music station. On Wednesday evenings it isn't unusual for *Severe Tire Damage* (the most technically hip garage band in the Bay Area, http://www.ubiq.com/std/band.html) to show up on sd. As if this wasn't pushing technology far enough, listeners can control the video camera and audio mixer through a series of WWW pages.

For more information about the MBONE, have a look at the MBONE Home Page: http://www.eit.com/techinfo/mbone/mbone.html . This page has links to all kinds of good information about the MBONE including the MBONE FAQ and a copy of *Dan's Quick and Dirty Guide to Getting Connected to the MBONE*, which explains how to get your network connected. There are also links telling you where to get all of the major MBONE software packages.

To see what's currently being sent across the MBONE, visit http://www.cmf.nrl.navy.mil/sd/ .

To look through the MBONE session agenda or book time on the MBONE, visit http://www.cilea.it/MBone/agenda.html or http://www.msri.org/mbone/ . If you are going to be sending a large amount of data (such as video) it is customary to book time at least two weeks in advance, and sooner if you can.