MBone – the Multicast Backbone

Hans Eriksson
Swedish Institute of Computer Science

Abstract

MBone – the Multicast Backbone is a virtual network on top of Internet and provides a multicasting facility to the Internet. MBone has been used to distribute audio and video from the IETF meetings, the Space Shuttle, seminars, just to mention a few items. MBone usage has also been the cause of severe Internet network meltdowns due to different reasons. This paper will discuss some experiences of MBone, the symptoms that has surfaced and how it has evolved during the last year into a kind of semi-maturity.

I. Introduction

The first thing many researchers like me do when they come to work is reading their electronic mail. The second thing on my list is to check what is on ”the MBone”. There might be video from the Space Shuttle, a seminar from Xerox, a plenary session from an interesting conference or a demo for the Swedish Prime minister.

It all started in March 1992 when the first audio cast on the Internet took place from the IETF (Internet Engineering Task Force) meeting in San Diego. At that event 20 sites listened to the audio cast. Two years later, at the IETF meeting in Seattle about ???[1] sites and ??? hosts tuned in to the two parallel broadcasting channels (audio and video) and also talked-back (audio) and joined the discussions! By now, everybody in the networking community takes it for granted that the IETF meetings will be distributed via MBone. MBone has also been used to distribute experimental data from a robot at the bottom of the Sea of Cortez (more later) as well as a late Saturday night feature movie – ”WAX or the discovery of television among the bees” by David Blair.

As soon as some crucial tools existed, the usage just exploded. Many people started using MBone for conferences, weather maps, research experiments, follow the Space Shuttle, etc. At SICS (Swedish Institute of Computer Science), we saw our contribution to SUNET (Swedish University Network) increase from 26GB/month in February-93 to 69GB/month in March-93. This was mainly due to multicast traffic as SICS at that time was the major connection point between US and Europe in MBone.

MBone has also (in)directly been the cause of severe problems in the NSFnet backbone, saturation of major international links rendering them useless as well as sites being completely disconnected due to ICMP responses flooding the networks. We will expand on this later in this paper.

II. Multicasting background

When we talk about MBone we sometimes mean the virtual network that implements multicasting, sometimes we refer to the applications that run on top of MBone (vat, nv, ivs, etc.), and often we mean everything. We will come back to the applications that are in use on MBone later in this paper, but for now we will concentrate on the MBone proper, the multicasting virtual network.

First let us define what we mean by the different types of ”casting”. The usual way packets are sent on the Internet is unicasting, i.e.. one host is sending to another specific single host. Broadcasting is when one host sends to all hosts on the same subnet. Normally the routers between one subnet and another subnet will not let broadcast packets pass through. Multicasting is when one host sends to a group of hosts.

On the link level (Ethernet, etc.) multicasting has been defined for some time. On the network level (IP) it started with the work of Steve Deering of Xerox PARC when he developed multicast at the IP level[1]. The IP address space is divided into different classes. An IP address is four bytes and the address classes A, B and C divide the addresses into a network part and a host part. The difference between the classes is the balance between bits designating network and hosts. Class A have one byte for the network and three for host, B has two bytes for each and class C has three bytes for the network and one for the host. To differentiate between the classes start with 0, 1 or 2 bits that are set followed by a zero bit. Class A addresses starts with binary ”0” and are in the range 0.0.0.0 to 127.255.255.255, B starts with ”10” with a range of 128.0.0.0 to 191.255.255.255 and finally class C starts with ”110” with a range of 192.0.0.0 to 223.255.255.255. However not all addresses are available for host addresses as some are defined for specific uses, e.g. broadcast addresses. Class D is indicated by ”1110” at the start giving an address range of 224.0.0.0 to 239.255.255.255. This class has been reserved for multicast addresses.

When a host wishes to join a multicast group, i.e. get packets with a specific multicast address, the host issues a IGMP (Internet Group Management Protocol) [request. The multicast router for that subnet will then inform the other routers so that such packets will get to this subnet and eventually be placed on the LAN where the host is connected. Frequently, the local router will poll the hosts on the LAN if they are still listening to

.

the multicast group. If not, no more such packets will be placed onto the LAN.

When doing multicasting á la MBone, the sender does not know who will receive the packets. The sender just sends on an address and it is up to the receivers to join that group (i.e.. multicast address). Another style of multicasting is where the sender specifies who should receive the multicast. This gives more control over the distribution, but one draw-back is that it does not scale well. Having thousands of receivers is almost impossible to handle this way. This second style of multicasting has been used in the ST (Stream Transport) [3][4].

III. MBone today

MBone is a virtual network running on top of the Internet. MBone is composed of networks (*islands*) that support multicast. On each of these islands, there is a host which is running the *mrouted* multicast routing demon. The mrouted's are connected with each other via unicast *tunnels.*

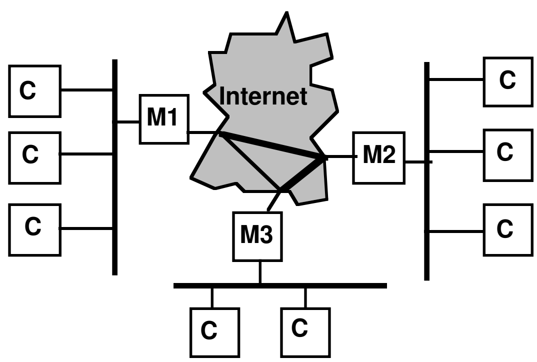

Fig 1: MBone topology – islands, tunnels, mrouted

In fig 1, we have three islands of MBone. Each island consists of a local network connecting a number of *client* hosts ("C") and one host running mrouted ("M"). The mrouted-hosts are linked with point-to-point tunnels. The thick tunnels are the primary feeds with the thin tunnel as a backup.

Basically, a multicast packet will start from one client who puts the packet on the local subnet. The packet will be picked up by the mrouted for that subnet. The mrouted will consult its routing tables and decide onto which tunnels the packet ought to be placed. At the other end of the tunnel is another mrouted who will receive the multicast packet. It will also examine its routing tables and decide if the packet should be forwarded onto any other tunnels. The mrouted will also check if there is any client on its subnet that have subscribed to that group (multicast address) and if so, put it onto the subnet to be picked up by the client.

III.A. Tunnels

When sending the multicast packet through the tunnel, we have to re-pack the multicast packets. There are two methods of doing this, adding the LSSRR IP option and encapsulation. The first implementations of mrouted used the IP Loose Source and Record Route option (LSRR). Mrouted modified the multicast datagram coming from a client by appending an IP LSRR option where the multicast address was placed. The IP destination address was set to the (unicast) address of the mrouted on the other side of the tunnel. There has been some problems with this approach (see below) which prompted the implementation of encapsulation. In this method the original multicast datagram will be put into the data part of a normal IP datagram that is addressed to the mrouted on the other side of the tunnel. The receiving mrouted will strip off the encapsulation and forward the datagram appropriately. Both these methods are available in the current implementations.

Each tunnel has a metric and a threshold. The metric is used for routing and the threshold to limit the distribution scope for multicast packets.

The *metric* specifies a routing cost that is used in Distance Vector Multicasting Routing Protocol (DVMRP). To implement the primary and backup tunnels in fig 1, the metrics could have been specified as 1 for the thick tunnels and 3 for the thin tunnel. When M1 gets a multicast packet from one of its clients, it will compute the cheapest path to each of the other M's. The tunnel M1-M3 has a cost of 3, whereas the cost via the other tunnels is (1+1) 2. hence, the tunnel M1-M3 is normally not used. However, if any of the other tunnels breaks, the backup M1-M3 will be used. DVMRP is a bit slow on propagating changes in network topology.

The *threshold* is the minimum time-to-live (TTL) that a multicast datagram needs to be forwarded onto a given tunnel. When sent to the network by a client, each multicast packet is assigned a specific TTL. For each mrouted the packets passes, the TTL will be decremented by 1. If a packets remaining TTL is lower than the threshold of the tunnel that DVMRP wants to send the packet onto, the packet is dropped. With that mechanism we can limit the scope for a multicast transmission.

In the beginning there was no *pruning* of the multicast tree. That is, every multicast datagram is sent to every mrouted in MBone if it passes the threshold limit. The only pruning is done at the leaf subnets, where the local mrouted will put a datagram onto the local network only if there is a client host that has joined a particular multicast group/address. This is called truncated broadcast. As MBone grew, problems surfaced which we will discuss later. These problems prompted work on proper pruning of the multicast tree as well as work on other techniques for multicasting [5][6][7]. Pruning as implemented in MBone today works roughly like this: If an mrouted gets a multicast packet for which it has no receiving clients nor tunnels to forward it to, it will drop the packet but also send a signal upstream that it does not want packets with that address. The upstream mrouted will notice this and stop sending packets that way. If the down-stream mrouted gets a client that joins that pruned multicast group, it will signal its upstream neighbours that it wants these packets again. Regularly the information will be flushed and packets will flow to every corner of MBone until pushed back again.

III.B. Management

There is no "network provider" of MBone. In the spirit of Internet, MBone is loosely co-ordinated via a mailing list. When end-users wants to connect to MBone, they are encouraged to contact their network provider. If that network provider is not participating in MBone and for some reason does not want to, a tunnel can be set-up to another point in MBone.

From time to time, there has been major overhauls of the topology as MBone has grown. Usually this has been prompted by an upcoming IETF meeting. These meetings put a big strain on MBone. The IETF multicast traffic has been about 100-300Kb/s with spikes up to 500Kb/s.

IV. Applications

Since MBone was set up, a number of applications have surfaced that span quite a range. We have seen the astronauts repairing the Hubble telescope, listened to seminars and seen cars come and go at the BBN parking lot in Boston. I will give a overview of some of the events that has used MBone in some way. But first I will mention some of the most popular tools for using the MBone. This list is by no means complete as new applications appear regularly.

For audio we have *vat* (visual audio tool) by Steve McCanne and Van Jacobsen of Lawrence Berkeley Laboratory. *nevot* (network voice terminal) by Henning Schulzrinne of AT&T/Bell Laboratories is another audio tool.

Video tools are *ivs* (inria video-conferencing system) by Thierry Thurletti of INRIA in Sophia Antipolis, France and *nv* (network video) by Ron Frederick of Xerox PARC.

wb (white board) by McCanne&Jacobsen provides a shared drawing space and is specially useful for presentations over the MBone. wb can import slides in PostScript and the speaker can make small annotations during the lecture.

In fig 1 we have *sd* (session directory) by McCanne&Jacobsen.

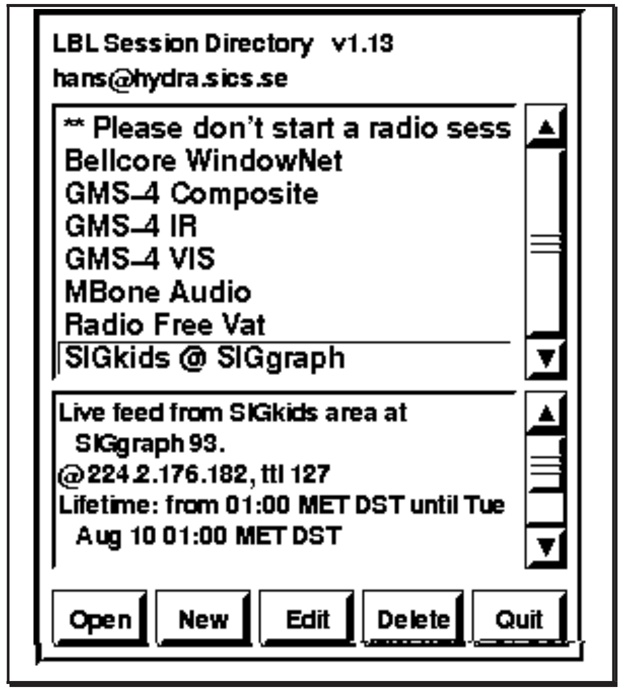

fig 1. sd – session directory

Sd offers a convenient way of announcing "sessions" that will take place on the MBone. When creating a session, you specify the multicast address (an unused address is suggested by sd) and the various tools that are used. Other people can then just click "Open" and sd will start all the necessary tools with appropriate parameters.

When this snapshot was taken, the SIGGRAPH conference was going on. As a special event at that conference, kids were invited to talk with people on the MBone. This event is highlighted in the sd snapshot. Going up in the list we have Radio Free Vat. This is the MBone "radio" station where anybody on the MBone can be the DJ! Next up is MBone Audio which is the common chat channel of the MBone. Everyone is free to join and start a discussion about whatever. As MBone spans about 16 time zones, not everybody is at their workstation when you ask "Is there anybody out there?"[8] but there is always someone out there! The GMS sessions is pictures from a satellite above Hawaii. The pictures (composite, infra read or visual spectra) are sent out using *imm* (Image Multicast Client) by Winston Dang of the University of Hawaii. Second to the top is the Bellcore WindowNet. If you tuned in to this session, you would see the outlook from a window from Bellcore. At the top we have not a session, but a plea. As audio and video consumes a fair amount of bandwidth and MBone is global, rebroadcasting your favourite local radio station onto MBone will put a hard strain on many networks. We will come back to this problem later on.

Not shown in this particular snapshot, but a frequent and very popular guest on MBone is the Space broadcast onto the MBone during the flights. The pictures of the astronauts travel a long way and traverses many different technologies before it hits the screen of your workstation. But it works!

Already mentioned in the introduction we have a different type of event, the 1993 JASON Project[9]. Woods Hole Oceanographic Institution provided software for Sun and SGI workstations so that anyone on the MBone could follow three underwater vehicles on their tours in the Sea of Cortez . Position data and some pictures were continuously distributed over the MBone. Beside being interesting for scientists in other fields, it was very valuable for oceanographic researchers to follow the experiments in real-time and give feedback immediately.

mmcc (multimedia conference control) by Eve Schooler of University of Southern California (USC)/Information Sciences Institute (ISI) is going beyond the simple support given by sd. More about this when discussing the MMUSIC protocol.

The popular Mosaic package from NCSA (National Center for Super computer Applications) is being enhanced by people at University of Oslo. The idea is to use Mosaic for lectures and let the speaker multicast control information to the mosaic programs run by the students.

We also have the media-on-demand server created by Anders Klemets of Royal Institute of Technology in Stockholm, Sweden, which offers unicasted replays of sessions that has been multicasted on the MBone.

This is just a snapshot of some of the developments that is taking place in the MBone community. New ideas surface often and implementations follow close behind.

V. Protocols

All traffic in MBone is using UDP (User Datagram Protocol) rather than usual TCP (Transport Control Protocol). TCP provides a point-to-point connection oriented reliable byte stream protocol, whereas UDP is just an transport level envelope around a IP packet with almost no control whatsoever. One reason for not using TCP is that the reliability and flow control mechanisms are not suitable to live audio casting for example. Occasional loss of an audio packet (as when using UDP) is usually acceptable whereas the delay for retransmission (when using TCP) is not acceptable in a interactive conference. Also, TCP does not easily lend itself to multicasting. One problem that has to be taken care of is that UDP packets may be duplicated and reordered ((beside being dropped) when transmitted over the Internet.

On top of UDP most MBone applications use the RTP (Real Time Protocol) developed by the Audio-Video Transport Working Group within the IETF. Each RTP packet is stamped with timing and sequencing information. With appropriate buffering at the receiving hosts, this allows the applications to achieve continuos playback in spite of varying network delays.

Each media can be encoded and compressed in several ways. Audio is usually encoded in PCM (Pulse Code Modulation) at 8KHz with 8bit resolution giving 64kb/s bandwidth for audio. Including packet overhead it raises to about 75kb/s. By using GSM (Groupe Special Mobile, a cellular phone standard) one can get down to about 18kb/s including overhead.

Video is more demanding. The ivs tool uses the CCITT (Consultative Committee of International Telephone and Telegraph) standard H.261 [10] whereas the nv tool uses a unique compression scheme. It is possible to limit the amount of bandwidth that should be produced in both tools. The usual bandwidth setting is 128kb/s. How this translates into quality depends on the kind of scene that is captured.

VI. Experiences

During the life time of MBone a fair number of problems have been seen. Some are inherent to multicasting in general, and some are more specific to the current implementation of MBone.

In this section we will discuss a number of problems that has surfaced during operations of MBone. Some problems have a direct bearing on the MBone implementation, other problems have just been discovered by the use of MBone.

VI.A. Bandwidth

Currently there are three more or less permanent sessions going on in MBone. There is one audio and one video channel for free-for-all use and there is Radio Free Vat. Then we have the IETF meetings three times per year. Several major conferences and workshops are being transmitted onto the net, such as JENC-93 and some IETF WG meetings. We have also seen President Clinton and Vice-president Gore on MBone. We have already talked about the JASON project and the Space Shuttle. As MBone is today, it should be viewed as one single resource. Only in a few places can it handle more than one video channel together with audio. The IETF tries to make two video and four audio but do not always make that even if the best networkers in the Internet put in their best efforts. So far, we have not had any major collisions of major events. The collisions that have occurred have been resolved after some short discussions. Essentially it is a first-announce-first-serve scheduling. As MBone gets more and more popular, one can expect more collisions and the pressure for a particular slot will increase.

Some of the success of MBone is dependent on the ”courtesy” of TCP. When someone starts sending audio onto a fully loaded Internet link, it will cause packet losses for many of the connections that are running on that link. That is usually TCP connections and they will back off when packet losses occur. UDP-based audio does not have any such mechanism and will effectively take the bandwidth it needs.

On several occasions end-users have started a video session with a high ttl and subsequently swamped the net with a continuos stream of 300-500 kb/s. These users have not been malicious. Sometimes the program has just been started with "-ttl 116" instead of "-ttl 16". At other times, the users has not really been aware of what "256kb/s" really is netwise. Usually after the mistake has been pointed out, the users have stopped their transmissions. The problem is that with the new video and audio applications the mistakes have severe consequences and with multicasting in MBone, the consequences are spread globally. It will take some time before the user community get a feel for how much bandwidth video and audio takes. Existing applications like ftp can also use a lot of bandwidth but the back off mechanism of TCP ensures a fair split of resources, which UDP based application does not.

Lacking a fine grained resource allocation mechanism, a way to put a limit on the bandwidth usage of a tunnel could be very helpful. That would make many network providers a lot less nervous about letting multicast traffic loose.

VI.B. Thresholds on tunnels

Lacking multicast tree pruning, the only way to limit the scope of a multicast datagram is by using thresholds. If a datagram has a TTL (time-to-live) greater than the threshold, it will be forwarded onto the tunnel. Thresholds range between 0 and 255. The threshold levels chosen on the tunnels tries reflect both a geographic partitioning (e.g.. keeping an local conference local) and a choice of traffic (e.g.. restricting video more than audio). But expressing two dimensions of choices in one metric always introduces some trade-off.

The guidelines establishes that traffic within one site should be sent with a ttl of 16, within one "community" 32 and global traffic should have 127. The IETF transmission plan says the following:

Traffic type	*TTL*	*threshold*
GSM audio 1	255	224
GSM audio 2	223	192
PCM audio 1	191	160
PCM audio 2	159	128
Video 1	127	96
Video 2	95	64
local event audio	63	32
local event video	31	1

Tab 1: ttl & thresholds from IETF

The table says that if you only want to get audio channel 1 with the GSM compression (18kb/s), your tunnel should have a threshold of 224.

Thresholds is a very coarse method to limit traffic. With the current plan, there is no way you can use your 256kb/s link to get one PCM audio channel (≈75kb/s) and one video (≈130kb/s). To get PCM you will open up for GSM audio (≈18kb/s each) and to get video you will also get the other PCM channel. To achieve that, true multicast tree pruning has to be implemented.

VI.C. Tunnel fan-out

Some mrouted hosts have a fair number of tunnels. The top ones had 11 tunnels (early June 1993). However, not all of them are primary tunnels. For example, hydra.sics.se (a SPARC 1+) has 10 tunnels but only 5 of them are primary. The aggregate traffic from an IETF meeting roughly generates one packet every 4ms. Looking at how much time it takes to forward an multicast packet we find that a SPARC 1+ needs ≈1.0ms and a SPARC 10 needs ≈0.6ms. This suggests that hydra.sics.se is saturated during IETF sessions. This will show up not only as dropped packets, but also as dropped tunnels. As hydra will be busy in kernel forwarding of multicast packets, the mrouted will not get any cycles to run its businesses. Eventually the peers at the other ends of the tunnels will think that hydra is dead and hence they will reroute their packets. Soon after the load has gone away from hydra, mrouted will get some cpu-cycles and talk to its peers and the overloading will start again.

A related problem is saturation of the local ethernets. FIX-West had 15 tunnels during the IETF meeting in March. With an IETF load of ≈500kb/s it results in ≈6.5Mb/s pushed over the FIX-West ethernet. Even without the MBone traffic, that ethernet is already busy. Some mrouted operators use the cpu overload as a means to limit the impact on the local network. I.e., if you have an SPARC 1+ as a mrouted host, it will not push more than 1000 packets onto the net, probably much less. This is dangerous practice unless you are the only entry-point to a part of MBone. As stated above, when your tunnel gets declared dead, MBone will choose another route if possible until your mrouted gets its breath back. This resulting in heavy route flapping which becomes a global problem. If you are to only path to take, traffic will just stop for a while which is a local problem only.

VI.D. MBone tunnels vs. Internet links

Tunnels are set up along the links of the underlying real network. But when a link fails and the underlying network makes a rerouting, the tunnels stays and become less optimally placed. As an example, traffic from Sweden to UK would normally go via the tunnel Stockholm - Washington over a T1 link and then take the tunnel Washington to London over another T1 link. When the first link malfunctioned, traffic was re-routed via Amsterdam and London and then to the US. The multicast traffic Sweden to UK ended up going Stockholm - London - Washington and then back to London.. Eventually we made a manual reroute of tunnels. Putting multicast routing into the routers enables the multicast routing to follow the unicast routing. There is a proposal for an extension to the OSPF (Open Shortest Path First) routing protocol to incorporate also multicasting [7].[

VI.E. MBone as a bug trigger

During the IETF meeting in Washington DC November 1992, there were problems with the NSFnet which was due to the multicast traffic coming from the IETF meeting. At that time the tunnels were all using the loose source route option (LSRR). In modern router technology, packets are handled by the interface cards to as large extent as possible. Packets with IP options, however, are usually forwarded to the main cpu for handling. At that time, the IETF meeting was generating two audio channels of ≈75Kb/s and two video channels of ≈130Kb/s. This was about 400 packets per second that was sent from the IETF site to the mrouted at Cornell. That mrouted in turn, fed a number of tunnels so that the traffic from Cornell onto ENSS133 at Ithaca was above a thousand packets per second that had to be handled by the main cpu. Adding to that was a great number of ICMP unreachable messages generated. The cpu was having a hard time when to regular routing updating occurred also. This lead to routing time outs when other networks had problems due to excessive MBone traffic. A number of actions were taken to fix these problems. One of the immediate actions was the disabling of one video channel to lessen the load. In the aftermath, some inefficiencies of routing updates was fixed, e.g. EGP derived routes will now be aggregated into a single IBGP update. The major impact in MBone was the encouraging of using true encapsulation instead of LSRR option for the tunnelling.

During the Packet Video workshop at MCNC, Van Jacobson observed a phenomena where it seemed that routing updates severely impacted the audio transmissions. The congestive loss rate was about 0.5% but every 30 second he observed huge losses (50-85%) for about 3 seconds. Jacobson concluded that it was due to the LSRR option processing competing with routing updates. Not only does this affect MBone traffic, but also other traffic such as pings, traceroutes etc.

Many hosts and routers does not handle multicast traffic properly. Often they respond by sending an ICMP redirect or network unreachable. These responses are not according to the IP specifications. This is usually not a problem until we have several such host reacting with ICMPs to a number of audio streams of about 50 packets per second. Then any network tends to get flooded with ICMPs. It has happened that a site was disconnected from MBone due to a ”screaming” router. Over time, this problem has diminished as router vendors update their software. Also, with the new encapsulation tunnels, the ICMPs will be sent to the last tunnel endpoint, not the whole way back to the original sender.

VII. Conclusions

Multicasting can be a dangerous beast, but it also carries the promise of very useful applications. As an indication of this MBone usage is increasing very rapidly. As a probably not-intended side-effect, it has also brought out some bugs in some routers and hosts in the Internet. Before MBone can be provided as a regular widespread service, some issues have to be addressed.

Some of these are still hard research issues, like resource control and real-time traffic control. Other work is directed towards better management hooks and tools and incorporating multicasting in the Internet routers. Maybe there are better technologies for multicasting than what us used currently in the MBone? The IDMR (Inter-Domain Multicast Routing) WG in IETF is working on this.

MBone has enabled a lot of applications out there. One problem when starting the applications is what addresses should they use. Picking one randomly will be fine for quite a while, but eventually when MBone gets more crowded some mechanism has to be put in place for allocation of multicast addresses and port numbers.

As MBone is today, the sender has no control, nor implicit knowledge of who is listening out there. A receiver can just ”tune in”, like on a radio. Some applications would want some kind of information about who is listening.

When the going gets rough and a lot of packets are dropped, some applications would be help by some feedback on the actual performance of the network. A video application could for example, stop sending raw HDTV data when only 2% makes it to the receivers and instead start sending slow scan, heavily compressed pictures.

We look forward to the next round of developments on MBone.

References

[1] S. Deering, Host Extensions for IP Multicasting, RFC112, August 1989.

[2] IGMP ref…

[3] Gotta find the spec for Stream Protocol…

[4] Craig´s and Steve´s work on ST-II…

[5] Core Based Trees…

[6] RSVP, get IETF proc…

[7] Moy, John, Multicast Extensions to OSPF, IETF draft, July 1993

[8] Pink Floyd, The Wall.

[9] Maffei, Andy; Remote Science over the MBone during the 1993 JASON project, Woods Hole Oceanographic Institution.

[10] CCITT Recommendation H.261…

Author Information

Hans Eriksson is a researcher at SICS – Swedish Institute of Computer Science in Kista outside Stockholm. His interests has always been computer networking and for the last years he has been focusing on distributed multimedia applications. He is also engaged in SNUS – Swedish Networks Users Society and promoting Internetting in Sweden. He may be reached at <hans@sics.se>.