Hans Eriksson

**MBONE: The Multicast Backbone**

The first thing many researchers like me do when they come to work is read their email. The second thing on my list is to check what is on the MBone—the Multicast Backbone, which is a virtual network on “top” of the Internet providing a multicasting facility to the Internet. There might be video from the Space Shuttle, a seminar from Xerox, a plenary session from an interesting conference or a software demonstration for the Swedish prime minister.

It all started in March 1992 when the first audiocast on the Internet took place from the Internet Engineering Task Force (IETF) meeting in San Diego. At that event 20 sites listened to the audiocast. Two years later, at the IETF meeting in Seattle about 567 hosts in 15 countries tuned in to the two parallel broadcasting channels (audio and video) and also talked back (audio) and joined the discussions! The networking community now 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 (as will be described 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, to follow the Space Shuttle, for example. At the Swedish Institute of Computer Science (SICS) we saw our contribution to the Swedish University Network SUNET, increase from 26GB per month in February 1993 to 69GB per month in March 1993. This was mainly due to multicast traffic as SICS at that time was the major connection point between the U.S. and Europe in MBone.

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

**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, for example), and often we mean everything. We will come back to the applications that are in use on MBone later in this article, but for now we will concentrate on the MBone proper, the multicasting virtual network.

First let us define what is meant by the different types of “casting.” The usual way packets are sent on the Internet is unicasting, that is, 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 (e.g., Ethernet) multicasting has been defined for some time. On the network level (Internet Protocol or IP) it started with the work of Steve Deering of Xerox PARC when he developed multicast at the IP level [3]. 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 addresses have one byte for the network and three for host, B addresses have two bytes for each, and class C addresses have 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 start with binary “0” and are in the range 0.0.0.0 to 127.255.255.255, class B starts with “10” with a range of 128.0.0.0 to 191.255.255.255, and class C starts with “110” with a range of 192.0.0.0 to 223.255.255.255. Not all addresses are available for host addresses, however, 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, that is, get packets with a specific multicast address, the host issues an Internet Group Management Protocol (IGMP) 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 local area network (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 utilizing MBone, the sender does not know who will receive the packets. The sender just sends to 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 drawback 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 ST-2 [6, 8].

**MBone Today**

As previously mentioned, 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 that is running the mrouted multicast routing demon. The mrouted’s are connected with one another via unicast tunnels.

In Figure 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 be sent 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 on which tunnels the packet ought to be placed. At the other end of the tunnel is another mrouted that 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 has subscribed to that group (multicast address) and if so, put it onto the subnet to be picked up by the client.

Tunnels
When sending the multicast packet through the tunnel, the multicast packets must be re-packaged. There are two methods of doing this, adding the Loose Source and Record Route (LSRR) IP option and encapsulation. The first implementations of mrouted used the LSRR IP option. 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 have been some problems with this approach (as will be described later) that 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 the Distance Vector Multicasting Routing Protocol (DVMRP). To implement the primary and backup tunnels in Figure 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. However, since DVMRP is slow on propagating changes in network topology, rapid changes will be a problem.

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 pass, the TTL will be decremented by 1. If a packet’s 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 the 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 [1, 5, 9]. Pruning as implemented in the MBone today works roughly like this: If a mrouted gets a multicast packet for which it has no receiving clients or 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 downstream mrouted gets a client that joins that pruned multicast group, it will signal its upstream neighbors 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.

Management
There is no “network provider” of the MBone. In the spirit of the Internet, MBone is loosely coordinated via a mailing list. When end users want 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 arranged to another point in MBone.

From time to time, there have 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 to 300Kb per second with spikes up to 500Kb per second.

Applications
Since MBone was set up, a number of wide-ranging applications have surfaced. We have seen the astronauts repairing the Hubble telescope, listened to seminars and seen cars come and go at the Bolt, Beranek, and Newman parking lot in Boston. I will give an overview of some of the events that have 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 vol (visual audio tool) by Steve McCanne and Van Jacobsen of Lawrence Berkeley Laboratory. The nesot (network voice terminal) by Henning Schulzrinne of AT&T/Bell Laboratories is another audio tool.
Video tools are Ira (inria videoconferencing system) by Thierry Thurletti of INRIA in Sophia Antipolis, France and mv (network video) by Ron Frederick of Xerox PARC.

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

Figure 2 depicts the sd (session directory) by McCanne and Jacobsen. 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 taking place. As a special event at that conference, children 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 anyone on the MBone can be the "disk jockey." Next up is MBone Audio, which is the common chat channel of the MBone. Everyone is free to join and start a discussion about any subject. Because MBone spans about 16 time zones, not everyone is at their workstation when you ask "Is there anybody out there?" [7], but there is always someone out there! The Global Mapping Satellite (GMS) sessions are pictures from a satellite above Hawaii. The pictures (composite, infrared 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 consume a fair amount of bandwidth and MBone is global, rebroadcasting your favorite local radio station onto MBone will put a hard strain on many networks. We will come back to this problem later in this article.

** Please don't start a radio session
Bellcore WindowNet
GMS-4 Composite
GMS-4 IR
GMS-4 VIS
MBone Audio
Radio Free Vat
SIGkids @ SIGgraph

Live feed from SIGkids area at
SIGgraph 93.
@224.2.176.182, ttl 127
Lifetime: from 01:00 MET DST until Tue Aug 10 01:00 MET DST

Not shown in this particular snapshot, but a frequent and very popular guest on MBone are the Space Shuttle missions. The NASA select cable channel is broadcast on the MBone during the flights. The pictures of the astronauts travel a long way and traverse many different technologies.

Figure 1. MBone topology—islands, tunnels, mrouted
Figure 2. sd—session directory
before appearing on the screen of your workstation. But it works!

A different type of event was mentioned earlier, the 1993 JASON Project [4]. Woods Hole Oceanographic Institution provided software for Sun and Silicon Graphics workstations so 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.

The multimedia conference control (mmcc) by Eve Scholler of University of Southern California (USC)/Information Sciences Institute (ISI) goes beyond this simple support given by sd. We will include more about this when discussing the MMUSIC protocol.

The popular Mosaic package from the National Center for Supercomputer Applications (NCSA) 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 used 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 uncasted replays of sessions that have been multicast on the MBone.

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

Protocols

All traffic in MBone uses User Datagram Protocol (UDP) rather than the usual Transport Control Protocol (TCP). TCP provides a point-to-point connection-oriented reliable byte stream protocol, whereas UDP is just a transport-level envelope around an IP packet with almost no control whatsoever. One reason for not using TCP is that the reliability and flow control mechanisms are not suitable for 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 must be resolved 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 Real-Time Protocol (RTP) 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 continuous playback in spite of varying network delays.

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

Video is more demanding. The ivs tool uses the CCITT (Consultative Committee of International Telephone and Telegraph) standard H.261 [2] 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 per second. How this translates into quality depends on the kind of scene that is captured.

Experiences

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

A number of problems that have surfaced during operations of MBone will be discussed in this section. Some problems have a direct bearing on the MBone implementation, other problems have been discovered recently during the use of MBone.

Bandwidth

Currently there are three more-or-less permanent sessions going on in MBone. There is one audio and video channel for free-for-all use and there is Radio Free Vat. In addition to the IETF meetings, which are transmitted three times per year, several major conferences and workshops are being transmitted on the net, such as JENC-93 and some IETF working group meetings. We have also seen President Clinton and Vice President Gore on MBone, and we have already mentioned the JASON project and the Space Shuttle.

MBone in its present form 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 channels but does not always accomplish this, 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 brief discussions. Essentially it is a first-announce-first-serve scheduling. As MBone increases in popularity, 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. They are 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 time-to-live (TTL) and subsequently swamped the network with a continuous stream of 300 to 500kb per second. These users have not been malicious. Sometimes the program has just been started with “-t1 16” instead of “-t1 16” with the effect that it reaches most parts of the MBone instead of just the local part. At other times, the users have not really been aware of what “256kb per second” really is netwise. Very few links in the Internet can handle that load without severely disturbing normal traffic. 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 gets 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 a 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.

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 greater than the threshold, it will be forwarded onto the tunnel. Thresholds range between 0 and 255. The threshold levels chosen on the tunnel tries reflect both a geographic partitioning (e.g., keeping a 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-offs.

The guidelines establish 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 is shown in Table 1.

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

When true pruning gets widely deployed in MBone it will be possible to get only what you ask for.

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 4 ms. Looking at how much time it takes to forward a multicast packet we find that a SPARC 1+ needs ×1.0ms and a SPARC 10 needs ×0.0ms. 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 business. Eventually, the peers at the other ends of the tunnels will think that hydra is dead and hence they will re-route their packets. Soon after the load has gone down on 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 1994. With an IETF load of ≈500Kb per second it results in ≈6.5Mb per second 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. That is, if you have a SPARC 1+ as a mrouted host, it will not push more than 1,000 packets onto the net, probably much less. This is a dangerous practice unless you are the only entry point to a part of MBone. As stated previously, when your tunnel gets declared dead, MBone will choose another route if possible until your mrouted gets its breath back. This results in heavy route flapping, which becomes a global problem. If you are the only path to take, traffic will just stop for a while, which is a local problem only.

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 does a rerouting, the tunnels stay and become less optimally placed. As an example, traffic from Sweden to the United Kingdom (UK) would normally go via the tunnel from Stockholm to Washington over a T1 link and then take the tunnel from Stockholm to London over another T1 link. When the first link malfunctioned, traffic was rerouted via Amsterdam and London and then to the U.S. The multicast traffic from Sweden to the UK ended up going Stockholm to London to 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 also incorporate multicasting [5].

MBone as a Bug Trigger
During the IETF meeting in Washington, D.C. in November 1992, there were problems with the NSFnet

<table>
<thead>
<tr>
<th>Traffic type</th>
<th>TTL</th>
<th>≈kb per second</th>
<th>threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM audio 1</td>
<td>255</td>
<td>15</td>
<td>224</td>
</tr>
<tr>
<td>GSM audio 2</td>
<td>293</td>
<td>15</td>
<td>192</td>
</tr>
<tr>
<td>PCM audio 1</td>
<td>191</td>
<td>75</td>
<td>160</td>
</tr>
<tr>
<td>PCM audio 2</td>
<td>159</td>
<td>75</td>
<td>128</td>
</tr>
<tr>
<td>Video 1</td>
<td>127</td>
<td>130</td>
<td>96</td>
</tr>
<tr>
<td>Video 2</td>
<td>95</td>
<td>130</td>
<td>64</td>
</tr>
<tr>
<td>local event audio</td>
<td>63</td>
<td>≥250</td>
<td>32</td>
</tr>
<tr>
<td>local event video</td>
<td>31</td>
<td>≥250</td>
<td>1</td>
</tr>
</tbody>
</table>
which were due to the multicast traffic coming from the IETF meeting. At that time the tunnels were all using the lose source route option (LSRR). In modern router technology, packets are handled by the interface cards as much 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 \( \approx 75 \text{Kb} \) per second and two video channels of \( \approx 130 \text{Kb} \) per second. This was about 400 packets per second that was sent from the IETF site to the mroute at Cornell. That murdered in turn, fed a number of tunnels so the traffic from Cornell onto ENSS133 at Ithaca was above 1,000 packets per second that had to be handled by the main CPU. Adding to that, a great number of ICMP unreachable messages were generated. The CPU was having a difficult time when regular routing updates were added. This lead to routing timeouts 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 were fixed, for example, EGP (External Gateway Protocol)-derived routes will now be aggregated into a single BGP (Border Gateway Protocol) update message. The most important change in MBone was the use of true encapsulation instead of LSRR option for the tunneling.

During the packet video workshop at MCNC, Van Jacobson observed a phenomenon in which it seemed that routing updates severely impacted the audio transmissions. The congestive loss rate was about 0.5% but every 30 seconds he observed huge losses (50% to 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 and traceroutes.

Many hosts and routers do not handle multicast traffic properly. Often they respond by sending an ICMP redirect or network unreachable. These responses are not in accordance with the IP specifications. This is usually not a problem until we have several such hosts reacting with ICMP 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 entire route back to the original sender.

**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 unintentional 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 issues are still difficult research issues, like resource control and real-time traffic control. Other work is directed toward better management hooks and tools and incorporating multicasting in the Internet routers. Maybe there are better technologies for multicasting than those currently used in the MBone? The IDMR (Inter-Domain Multicast Routing) working group in IETF is working on this.

MBone has enabled a lot of applications. One problem when starting the applications is the question of what addresses should be used. 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, or implicit knowledge of who is listening out there. A receiver can just “tune in,” like a radio. Some applications would want some kind of information about who is listening, for example by asking MBone which hosts are currently in a particular multicast group. There are mechanisms in some applications for end-to-end control of who is listening (i.e., encryption) but there is so far no common architecture for this.

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

We look forward to the next round of developments as the MBone continues to evolve.

**References**


7. Pink, F. Dark Side of the Moon.


**About the Author:**

HANS ERIKSSON is a researcher at the Swedish Institute of Computer Science (SICS), located in Kista, outside Stockholm. Current research interests include computer networking and distributed multimedia applications. **Author’s Present Address:** Swedish Institute of Computer Science, Box 1263, 164 28, Kista, Sweden; email: hans@sics.se

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.