RC21512 (97046) July 21, 1999
Computer Science/Mathematics

IBM Research Report

A New Multicast Scheme for Small Groups

Rick Boivie
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

Research Division
Almaden – Austin – Beijing – Brazil – Cambridge – Dublin – Haifa – India – Kenya – Melbourne – T.J. Watson – Tokyo – Zurich

A New Multicast Scheme for Small Groups

Rick Boivie
T. J. Watson Research Center
Hawthorne, New York

Abstract

Multicast is a technology that allows an application program to send data to more than one destination. An application program that uses multicast sends a single copy of its data and the network delivers the data to a set of destinations on the applications's behalf. Multicast is important since it enables new classes of applications such as video distribution, audio and video conferencing, networked collaborative work environments, multiparty networked games etc. The Internet community has done a significant amount of work on IP multicast over the last decade [1-10] and as a result, there are a number of multicast applications that are used today on the Mbone, the multicast-capable virtual network that is layered on top of (portions of) the Internet [10]. Today's multicast schemes are scaleable in the sense that they can support very large multicast groups. But there are problems when a network needs to support a very large number of distinct multicast groups, such as a large number of small audio & video conferences, for example. In this paper, we describe a new scheme for multicast that complements the existing schemes. Whereas the existing schemes can support a limited number of very large multicast groups, the new scheme can support a very large number of small multicast groups for conferencing or other applications.

Introduction

The 1990's have been a time of explosive growth for the Internet and the Internet is becoming an increasingly important communications medium. Along with mail and chat, the Internet is increasingly used for applications like IP telephony and conferencing applications. It seems clear that multicast, the ability to send data to a group destinations, will be increasingly important for applications like IP telephony and conferencing as well as for other applications such as video distribution, collaborative work environments, multiparty networked games etc.

There seem to be two kinds of multicasts that are important: a broadcast-like multicast that sends data to a very large number of destinations and a "narrowcast" multicast that sends data to a fairly small group. An example of the first is the audio & video multicasting of a working group session from an IETF meeting to sites all around the world. An example of the second is a videoconference involving 3 or 4 parties. We believe, for reasons described below, that it makes sense to use different mechanisms for these two cases. As the recently chartered reliable multicast transport group said on their web site, 'it is believed that a "one size fits all" protocol will be unable to meet the requirements of all applications" [11].

Today's Multicast Schemes

Today's multicast schemes were designed to handle the case in which there is a limited number of potentially large multicast groups. These work well if one is trying to distribute broadcast-like channels all around the world but they have scaleability problems when there is a large number of groups.

In some of these schemes, the nodes in the network build a multicast distribution tree for each <source, multicast group> pair and they disseminate this multicast routing information to places where it isn't necessarily needed -- which leads to scaling problems if there are a large number of multicast groups.

Some other schemes try to limit the amount of multicast routing information that needs to be disseminated, processed and stored throughout the network. These schemes use a "shared distribution tree" that is shared by all the members of a multicast group and they try to limit the distribution of multicast routing information to just those nodes that "really need it". But these schemes also have problems. Because of the shared tree, they use less than optimal paths in routing packets to their destinations and they tend to concentrate traffic in small portions of a network. They also require that all of the routers in a multicast tree "signal", process and store multicast routing information. And they require that multicast routing information for the various multicast groups be communicated across inter-AS administrative boundaries. These requirements cause scaleability problems and increase administrative complexity if there are a large number of multicast groups.

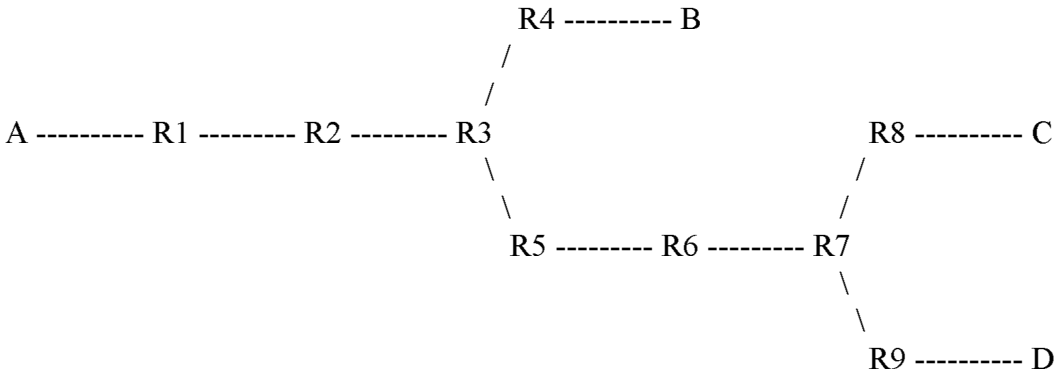
Small Group Multicast -- Introduction

The multicast scheme proposed here attempts to eliminate these problems for the case of small groups. It's very scaleable in that it can handle a very large number of these groups since the nodes in the network do not need to disseminate or store any multicast routing information for these groups. And since it doesn't use any multicast routing protocol, there are no inter-AS multicast routing "peering" issues to contend with. The proposed scheme has the additional benefit that packets always take the "right" path as determined by the ordinary unicast route protocols. Unlike the "shared tree" schemes, the new scheme minimizes network latency and maximizes network efficiency. In our view, this scheme removes some important obstacles that have, to this point, prevented the widespread acceptance and adoption of multicast and we believe this scheme can make multicast practical for very large numbers of small groups -- which as suggested above is a very important case. (The scheme described here is not appropriate for "broadcast" channels, e.g. broadcasting an IETF meeting all across the Internet -- but solutions already exist for that problem.)

The scheme proposed here takes advantage of one of the fundamental tenets of the Internet "philosophy", namely that one should move complexity to the edges of the network and keep the middle of the network simple. This is the principle that guided the design of IP and TCP and it's the principle that has made the incredible growth of the Internet possible. The reason that the Internet has been able to scale so well is that the routers in the core of the network deal with large CIDR blocks as opposed to individual hosts or individual "connections". The routers in the core don't need to keep track of the individual TCP connections that are passing through them. And the IETF's diffserv effort is based on the idea that the routers shouldn't have to keep track of a large number of individual RSVP flows that might be passing through them. It's our belief that the routers in the core shouldn't have to keep track of a large number of individual multicast flows either.

Small Group Multicast -- Details

The idea here is to let the source node keep track of the destinations that it wants to send packets to and eliminate the need for the routers to store any state for the various multicast groups. For example, let's suppose that A is trying to get his packets distributed to B, C & D in the figure below.



This can be accomplished as follows. A can send a new type of packet to its default router, R1, that includes the list of destinations for the packet. The new packet type, let's call it a "small group multicast" packet, is a level 3 packet which is to say that it's at the same level as IP in the protocol stack. In fact it has pretty much the same function as an IP packet, except for the fact that it is addressed to a list of destinations as opposed to a single destination. Ignoring some details, the packet that A sends to R1 might look like this:

Level 2 header: < dest = level 2 address of R1 >
< src = level 2 address of A >
< protocol = small group multicast > (ie a new level 3 packet type)

Level 3 header: < dest = B C D >
< src = A >

followed by the payload that A wants delivered to B, C and D.

When R1 receives this packet it needs to properly process the multicast. The processing that a router does on receiving one of these "small group multicast" packets is as follows:

- Perform a route table lookup to determine the "next hop" for each of the destinations listed in the packet.
- Partition the set of destinations based on their next hops.
- Replicate the packet so that there's one copy of the packet for each of the next hops found in the previous steps.
- Modify the list of destinations in each of the copies so that the list in the copy for a given next hop includes just the destinations that ought to be routed through that next hop.
- Send the modified copies of the packet on to the next hops.

So, in the example above, R1 will send a single packet on to R2 with a destination list of < B C D > and R2 will send a single packet to R3 with the same destination list.

When R3 receives the packet, it will, by the algorithm above, send one copy of the packet to R4 with a destination list of < B > and 1 copy of the packet to R5 with a destination list of < C D >. R4 will then forward a single packet on to B. And R5 will forward the packet that it receives on to R6 which will pass it on to R7. When the packet reaches R7, R7 will send a packet on to R8, with a destination list of < C >, and a packet on to R9 with a destination list of < D >. (The packets sent to R8 and R9 could be "small group multicast" packets with a single address in the destination list or they could be ordinary unicast packets addressed to C & D respectively[1].) R8 and R9 will then forward appropriate packets on to C and D respectively.

Note that it's important that the packet that is sent to a given next hop only includes destinations for which that next hop is the next hop listed in the route table. If the list of destinations in the packet sent to R4, for example, also included C and D, R4 would send "extra packets" on to those nodes on a less than optimum path. This could waste a lot of bandwidth if one is multicasting a videoconference, say. And this could cause serious problems when route loops occur since a multicast packet could "spray" large numbers of packets in a number of different directions as it travels around a loop. Since the packet that is sent to a given next hop only includes the destinations that are supposed to be reached through that next hop, these problems are eliminated.

Note that when routing topology changes, the routing for a multicast flow will automatically adapt to the new topology since the path a multicast packet takes to a given destination always follows the ordinary, unicast routing for that destination.

Interoperability with Today's Routers

One disadvantage of the proposed scheme is that all the routers between the source and the various destinations need to be able to properly process the new multicast packets. But, the scheme can be modified slightly to workaround routers that don't understand the new scheme. In the modified scheme, the packet that A sends to R1 in the example above would look like this:

Level 2 header:	< dest = level 2 address of R1 >
	< src = level 2 address of A >
	< protocol = IP >
Level 3 header:	< dest = R1 >
	< src = A >
	< protocol = small group multicast > (ie a new protocol type)
Level "3.5" header:	< dest = B C D >

[1] One advantage of using an ordinary unicast for the last hop is that this allows hosts with "standard" TCP/IP stacks to receive the new multicast transmissions in a way that doesn't require any modifications in the host TCP/IP stacks. (If a source sends packets to a multicast "exploder", source nodes can also use "standard" TCP/IP stacks.)

< src = A >

followed by the payload that A wants delivered to B, C and D.

Note that a router that doesn't understand this new protocol will, upon receiving this kind of packet, send an icmp message back to the source if the router adheres to RFC1812, "Requirements for IP Version 4 Routers" -- as all routers should. Section 5.2.7.1 of RFC1812 says that a router should send an ICMP Destination Unreachable message with code of 2, signifying Protocol Unreachable, if the transport protocol designated in a datagram is not supported in the transport layer of the final destination.

So a router that doesn't understand this new protocol should send an icmp "destination unreachable, protocol unreachable" message back to the source. (If the icmp message is lost for some reason, a subsequent "small group multicast" packet will cause another icmp to be generated.) So the source will know when a router doesn't understand the new protocol. Furthermore, since the icmp message will include the initial part of the original packet, the source will also know the destinations that are not reachable via "small group multicast", so the source can use unicast packets to reach those destinations. When routing topology changes, additional icmp "destination unreachable, protocol unreachable" messages may be generated and the source may use unicasts for additional destinations. The source can also periodically send a "small group multicast" for the destinations that are on its "unicast list" ie the list of nodes that it is reaching via unicast. Destinations that become reachable via "small group multicast" (ie those do not appear in subsequent icmp "destination unreachable, protocol unreachable" messages) can then be removed from the unicast list[2].

Thus, the "small group multicast" scheme can perform some multicasting in an environment that includes "legacy" routers that do not understand the new scheme. It won't work particularly well if there are many routers that don't understand the new scheme but this backwards compatibility may be important since it makes some of the benefits of multicast possible before all the routers in a network have been upgraded which can be very useful since it may take some time to upgrade all the routers in a large network.

Reliable Multicast

One additional advantage of the small group multicast scheme is that it can be easily adapted to provide a "reliable multicast". The multicast scheme that we've been discussing to this point provides for an "unreliable" multicast in which there is no provision for retransmitting packets that are lost due to network congestion or because they were garbled during transmission due to line-noise, say. This kind of "unreliable" transmission is useful in many applications in which the "timeliness" of packet delivery is more important than getting an "old" packet that was lost re-transmitted. IP telephony and video conferencing are applications in which the timeliness of packet delivery is important and the retransmission of lost packets is not useful.

---

[2] Another possibility is to send a multicast "ping" periodically to the set of destinations and then use unicast to reach those destinations that don't respond to the ping.

In other applications, it makes sense to re-transmit a lost packet even if the re-transmitted packet is going to be 1 or 2 hundred msecs "late". For example, in a conferencing application, a reliable multicast could be used to reliably and efficiently transmit a foil presentation or the contents of a whiteboard or a shared document to multiple conference participants.

A reliable multicast scheme can be built by extending the multicast scheme that we've been discussing. The scheme would work as follows:

- An additional header which would include a sequence number and a checksum, similar to TCP, would be used to keep track of the bytes that have been sent and the bytes that have been successfully received by each of the receivers.
- Each of the receivers would send acknowledgments or ACKs to inform the sender of the bytes that have been successfully received. The checksum would be used to determine if a packet has been received error-free. As in TCP, the ACK includes a sequence number to indicate the last byte that has been successfully received. (The sequence number could be that of the last byte successfully received or the first byte that has not yet been successfully received.)
- As in TCP, the sender concludes that a receiver has not successfully received a packet when it doesn't receive an appropriate ACK within a certain period of time. As in TCP, the sender re-transmits a packet that has not been successfully received. But unlike TCP, which re-transmits a packet to a single receiver, the reliable multicast scheme may need to re-transmit a packet to more than one receiver.
- When the sender needs to re-transmit a packet, it uses a multicast for the re-transmission. Since the sender knows the receivers that it needs to re-send to, it can re-send to just those receivers. Thus the re-transmissions are optimized and bandwidth is not wasted re-transmitting information to nodes that have already successfully received that information. (If the sender needs to re-transmit a packet to a single receiver, the "multicast" will, of course, be to a "tree" with a single leaf.)

Summary

In summary, the disadvantages of the "small group multicast" scheme are:

- the extra bytes that are sent in a multicast packet for the list of destinations
- the need to use unicast packets in some cases to reach destinations that are behind "legacy" routers
- the need for a new IP stack in sending hosts and routers
- the fact that the scheme is not suitable for huge "broadcast-like" multicasts. It's targeted for "small" conferences.

The key advantages are:

- it's very scaleable. It can handle a very large number of small groups.
- the work involved is limited to just the nodes that are on the multicast tree
- no per flow state information is stored on the routers.
- no multicast route protocol messages are communicated or processed. No intra-AS or inter-AS route protocols.

- Minimum administrative complexity. No need for complicated inter-AS peering agreements. It's just as easy for a network administrator to support multicast as it is to support unicast. And it will be just as easy to support multicast across the Internet as it is to support unicast.
- traffic follows the correct paths. Traffic is not concentrated in a small part of the network. Minimum network latency. Maximum network efficiency.
- no need for class D addresses which means
  - no need for a server that hands out class D addresses which can be a bottleneck or a point of failure
  - no one can join the class D group and "eavesdrop" on the class D address. The source knows who he's sending to.
- the scheme can be easily adapted to provide a "reliable multicast"

The advantages of the small group multicast (or SGM) scheme suggest that this scheme can be a very useful complement to the existing multicast schemes. Whereas the existing schemes can support a limited number of very large multicast groups, the SGM scheme can support a huge number (ie virtually an unlimited number) of small multicast groups and thus can play an important role in supporting applications such as conferencing applications on the Internet.

Status

An initial implementation of small group multicast has been implemented at IBM's T. J. Watson Research Center in Hawthorne, New York. The initial implementation includes a simple application program that runs on AIX called mchat (for multicast chat), an SGM layer that is linked with a multicast application, and a simple SGM router stub that runs on AIX.

Acknowledgment

I'd like to thank Brian Carpenter, chairman of the Internet Architecture Board, for several discussions and for his feedback and ideas on the "Small Group Multicast"scheme.

References

[1] RFC 1075, Distance Vector Multicast Routing Protocol, D. Waitzman, C. Partridge, S.E. Deering, Nov. 1988

[2] S.E. Deering. Multicast Routing in a Datagram Internetwork. PhD thesis, Electrical Engineering Dept., Stanford University, Dec. 1991.

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

[4] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The Pim Architecture for Wide-area Multicast Routing, ACM Transactions on Networks, April 1996.

[5] RFC 2189, Core Based Trees (CBT version 2) Multicast Routing -- Protocol Specification, A. Ballardie, Sept., 1997

[6] RFC 2201, Core Based Trees (CBT) Multicast Routing Architecture, A. Ballardie, Sept. 1997

[7] RFC 2236, Internet Group Management Protocol, Version 2, W. Fenner, Nov. 1997

[8] RFC 2362, Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, D. Estrin et al, June 1998

[9] D. Estrin, D. Farinacci, V. Jacobson, C. Liu, L. Wei, P. Sharma, and A. Helmy, "Protocol Independent Multicast-dense Mode (pim-dm): Protocol Specification", Work in Progress.

[10] Frequently Asked Questions (FAQ) on the Multicast Backbone (MBONE), ftp://venera.isi.edu/mbone/faq.txt

[11] Reliable Multicast Transport Working Group web site, http://www.ietf.org/html.charters/rmt-charter.html, June 15, 1999