Measurements and Observations of IP Multicast Traffic

Bruce A. Mah
`bmah@CS.Berkeley.EDU`
The Tenet Group
University of California at Berkeley
and
International Computer Science Institute

ABSTRACT

In this report, we present measurements of IP multicast traffic taken at the University of California at Berkeley. We note that the volume and distribution of IP multicast traffic is highly variable, and can depend on a small number of active conversations. From examining many-way multimedia conferences, we note the need for some kind of conference control, either provided by the application or the users. We see that IP multicast conferencing traffic can exhibit characteristics very different from that of conventional wide-area data traffic. Finally, we show that scope control (controlling the extent of the propagation of data through the network) must be addressed by the network, because users have been seen to mismanage the scopes of their own transmissions.

KEYWORDS: Internet, IP, multicast, MBONE, multimedia, traffic measurements.

1 Introduction

Recent and current trends in networks are enabling various kinds of multimedia conferencing and distribution applications, which will most likely use some form of multicasting to transmit data across a network to several or many recipients. Recently, the first such applications have come into wide-spread use on the Internet, using the MBONE (Multicast Backbone) virtual network [Macedonia94]. These tools are noteworthy because they are the first of their kind to see production use on a wide scale (as exposed to experimental use in a more controlled environment). Their use of networks may be very different from that of conventional data applications. By examining the network traffic generated by these applications, we can derive some conclusions as to the usage, requirements, and implications for the Internet and future internetworks.

In this paper we present measurements of IP multicast traffic taken at the University of California at Berkeley. In Section 2 and Section 3, we briefly describe the IP multicast service and applications. In Section 4, we briefly explain our methodology. Section 5 presents our results obtained from measuring different types of IP multicast traffic. Our conclusions are in Section 6.

2 IP Multicast and the MBONE

Multicasting describes data transmission in which a source sends a single copy of data to multiple recipients simultaneously. [Deering89] defines a standard for multicasting IP datagrams across a broadcast-style IP subnet, such as an Ethernet. A host transmitting a multicast packet sends it to one of a special class of IP addresses; each address designates a particular *host group*. These IP addresses are then mapped to the multicast addresses of the underlying datalink layer. A host wishing to receive packets for a host group uses Internet Group Membership Protocol (IGMP) messages to join the group; it then simply receives packets with the appropriate datalink and IP addresses.

For multicast packets to be sent between subnets, they need to be forwarded by multicast routers, also known as *mrouters*. As commercial routers are only now beginning to support forwarding of IP multicast packets, many mrouters are general purpose computers, such as UNIX workstations. To support forwarding across non-multicast-capable routers, mrouters *tunnel* multicast packets encapsulated in IP datagrams addressed to other mrouters. To control the scope of multicasts, mrouters can impose thresholds on each of their attached subnets and tunnels. A multicast datagram will be forwarded over a subnet or tunnel only if it has an IP Time-To-Live (TTL) greater than the corresponding threshold. The multicast routers and the tunnels between them form a virtual network for supporting Internet-wide multicasts called the *MBONE*.

On UNIX mrouters, multicast packet forwarding is typically performed by the kernel; routing tables are managed by a user-level program called `mrouted` [Deering93]. Older versions of `mrouted` (version 1 and 2) use a truncated broadcast algorithm for distribution of multicast packets, in which packets meeting threshold requirements are always forwarded across tunnels but are only multicast to attached subnets when a receiving host has requested them using IGMP. The newest version of `mrouted` (version 3) uses a true multicast algorithm, which eliminates unnecessary forwarding via tunnels by "pruning" away unneeded branches of the multicast distribution trees. Currently, the Distance Vector Multicast Routing Protocol (DVMRP) is used to exchange routing information [Waitzman88].

3 Applications

The establishment of an Internet-wide multicast service has enabled a number of multimedia conferencing tools. They are generally designed for one-to-many or many-to-many conferences, using low-bandwidth, unreliable networks. Multiparty voice and audio conferencing facilities are provided by applications such as `vat` [Jacobson94b]. Such tools can be used for many-way conferences or one-to-many audio distribution. Typical desktop workstations can send and receive motion video using one of various video tools. `nv` [Frederick94], the most widely used of these at the time of this study, transmits and receives low bitrate, rate-controlled, compressed motion video. A shared whiteboard tool named `wb` [McCanne93], allows users to collaborate within a common drawing environment.

Various other applications, unrelated to video conferencing, have also been developed to use IP multicast. Low-bandwidth, still-image distribution is provided using `imm` [Dang93], which multicasts compressed images for display in a window or as a background picture. Some experiments using the MBONE for network news distribution were performed and documented in [Lidl94].

A session directory tool, `sd` [Jacobson93], presents a "viewer's guide" to currently advertised multicast sessions. `sd` processes exchange directory information among themselves using a well-known multicast address; for a given multicast session this indicates the IP host group address, name, description, programs, and any necessary parameters. Although not all sessions are advertised, `sd` is the preferred mechanism of disseminating the information needed for users to join a multicast session.

4 Methodology

We were fortunate to have at our disposal a lightly-loaded FDDI network (shown in Figure 1), which served as a transit subnet between two mrouters running `mrouted` version 2. Because the FDDI network was lightly loaded, the total number of packets processed was small and hence the demands on our trace collection machine were low. The fact that neither mrouter was using a "pruning" `mrouted` meant that all of the traffic seen by either mrouter was forwarded across the link. In this way we were able to capture virtually all multicast traffic on the UC Berkeley campus network. This traffic includes MBONE-wide multicasts, locally-generated sessions, and multicasts across XUNET II, an experimental wide-area ATM network [Fraser92] [Lockwood93]. We note that the IP multicast data seen at Berkeley is not necessarily identical to that seen by other MBONE sites, due to the fact that the IP multicast tunnel thresholds may confine packets to a given campus, region, or continent.

To perform this study we collected measurements using `tcpdump` [Jacobson94a], a publically available program that collects network traces. We saved the packet headers for a variety of time periods, ranging from an hour to a day, at various times from September 1993 to May 1994. Our tracing machine was a DECstation 5000/240; it used `tcpdump` to store packet headers of all observed multicast traffic to a local disk, losing typically less than one packet in ten thousand. Our trace analysis was performed off-line, using a collection of locally-written `perl` and `awk` scripts.

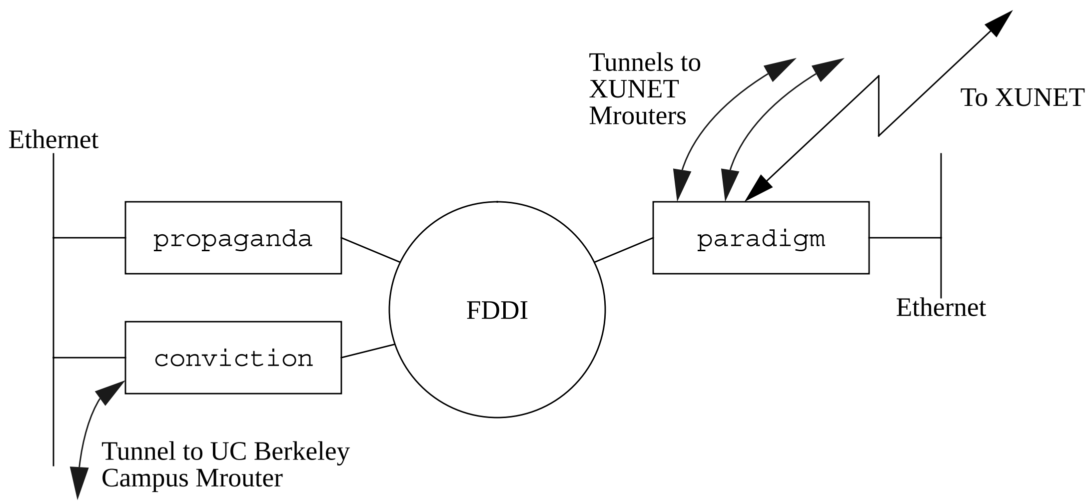

FIGURE 1. Network Environment. `conviction` and `paradigm` are MBONE routers; each forwards packets between its tunnel(s), its attached Ethernet, and the FDDI network. `propaganda`, the tracing host, sees all multicast traffic passing between the two MBONE routers across the FDDI network.

To perform conversation-level analysis, it is necessary to match packets with their sessions, using the information maintained by `sd`. Unfortunately, `sd` does not keep a log of updates to the list of active sessions; while this feature is generally unneeded in normal use, it is critical for tracing applications, as only the session directory updates contain the higher-level information necessary to determine the semantics of the packets going to a given IP multicast address.[1] We created a tool called `sdsnoop` to timestamp and log session directory update packets to a file; having this information allowed us to better record the active sessions during a trace.

5 IP Multicast Measurements

In this section, we present our measurements of IP multicast traffic. We examine both the total volume of data, as well as that generated by different types of applications, such as digital voice and video conferencing tools.

5.1 Traces Collected

Our traces were taken at various times from September 1993 to May 1994. They cover a variety of time periods, from slightly more than an hour to a full day. Table 1 summarizes the IP multicast traces taken and used in this study.

Date	Trace Length (hours: minutes)	Total Data	`tcpdump` Trace File Size
22 September 1993	1:32	620 MB	93 MB
20 January 1994	24:02	689 MB	181 MB
21 April 1994	4:41	357 MB	99 MB
25 May 1994	6:09	569 MB	169 MB

TABLE 1. IP Multicast Traffic Traces.

1. In some cases it may be possible to determine the type of data in a given IP multicast packet, but in general this is not feasible. The problem is similar to that of determining the semantic contents of a given TCP segment without information such as well-known ports.

The first trace was intended to capture the traffic generated by a large video conference. Quite by accident, the 20 January trace examined a time period during which little traffic was being sent via the MBONE. The other traffic traces captured varying periods of activity; in particular the 25 May trace was deliberately selected to capture data from several ongoing, advertised MBONE-wide multicasts.

5.2 Aggregate Data

We first observe that the aggregate data on the MBONE depends quite a bit on single, “special” events, such as MBONE-wide multicasts or video-conferences across some subset of sites. Unlike wide-area data traffic on the Internet, MBONE traffic is typically the result of a small number (usually less than thirty) of conversations at any one time. The traffic produced by a single session can therefore have a large impact on the aggregate traffic profile.

The trace of traffic on 21 April 1994 captures qualitatively typical MBONE behavior. During most of the trace, between one and three audio sessions were active at once. No video data was observed during this trace. The average bitrate during this trace was 178 Kbps, with a peak during a one-second interval of 924 Kbps. Figure 2 shows the total throughput, averaged over one-second intervals. A similar, but slightly higher workload was seen in the 25 May 1994 trace; there were at least six video sessions and seven audio sessions active during this time (though not all at once). The average throughput seen during this six-hour interval was 216 Kbps, with a peak of 977 Kbps during a one-second interval.

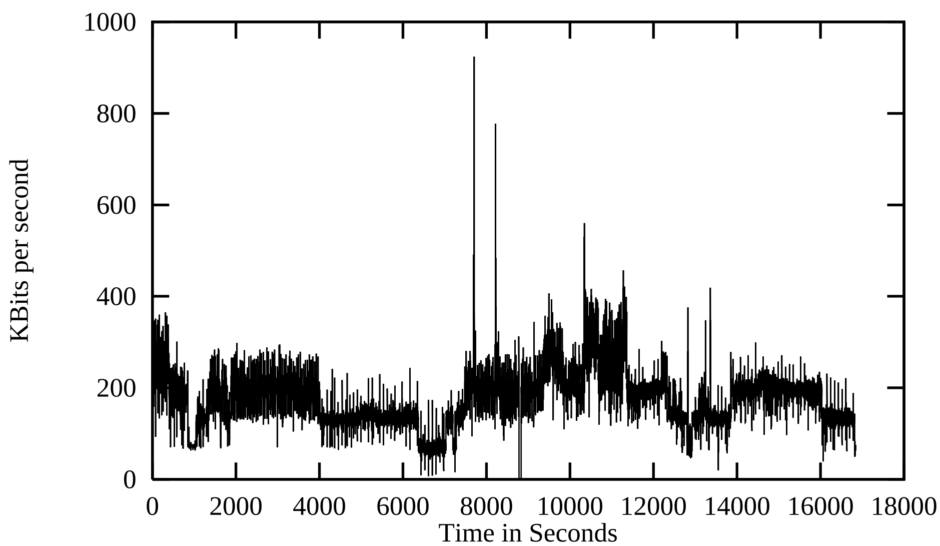

FIGURE 2. Total IP Multicast Bitrate, 21 April 1994.

The IP multicast traffic seen at Berkeley on 22 September 1993 was dominated by a project-wide videoconference across XUNET II (there was little other traffic; MBONE-wide multicasts were still a rarity at this time). This session involved up to nine simultaneous video sources and thirteen audio sources at six different sites across the United States, as documented in [Keshav94]. The average bitrate for this trace was 942 Kbps.[2] A plot of the total bitrate (again, averaged over one-second intervals) is shown in Figure 3. The gradual upward trend at the start of the trace shows the individual users joining the session and enabling their video transmissions. The more abrupt drop-off in bitrate at the end of the trace is caused by most of the participants “signing off” at the end of the video conference.

Finally, a “quiet day” on the MBONE is typified by the 24-hour trace taken on 20 January 1994. The only advertised MBONE-wide multicasts active were low-bitrate sessions. The mean observed data rate for this 24-hour trace was 65

2. This data was only seen across the XUNET links and directly attached campus networks; it was not routed over the main MBONE links!

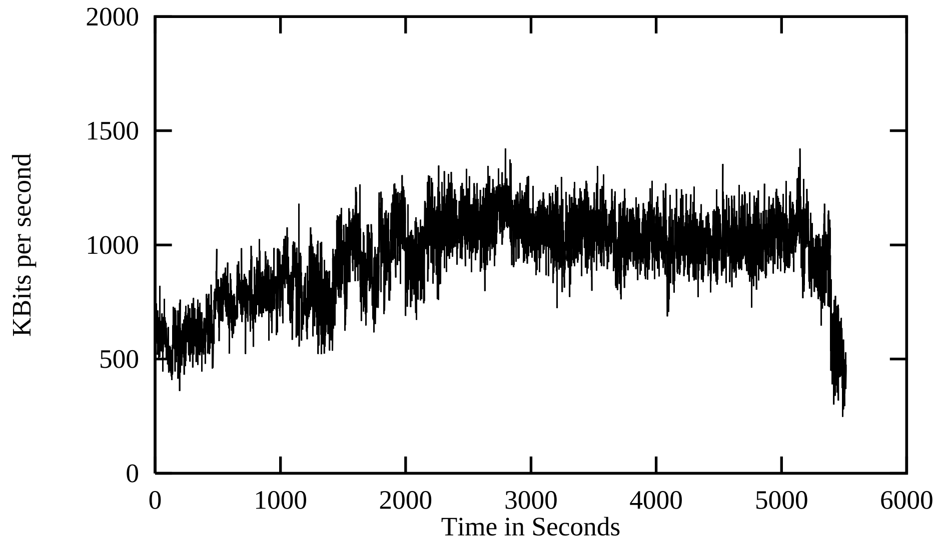

FIGURE 3. Total IP Multicast Bitrate During XUNET Videoconference, 22 September 1993.

Kbps. The main traffic source was a local “radio” session, in which a user sent an audio multicast to the Berkeley campus.

In Table 2, we present a breakdown by traffic type of each of these four traces. As each trace covers different lengths of time, comparing the actual amount of data transferred is not necessarily useful. However, we note that for most of the time periods studied, audio (`vat`) traffic was the principle traffic source. The fraction of traffic attributed to each application, however, varies from trace to trace. In some cases a single event (for example, a videoconference) can be responsible for a large fraction of the bytes transferred. Examples of this phenomenon are the XUNET II video conference observed on 22 September and a local audio multicast observed on 20 January.

	22 September 1993 (1:32)		**20 January 1994 (24:02)**		**21 April 1994 (4:41)**		**25 May 1994 (6:09)**	
	MB	**%**	**MB**	**%**	**MB**	**%**	**MB**	**%**
`nv`	492.2	79.4	11.4	1.6	0.0	0.0	182.8	32.1
`vat`	127.0	20.5	605.1	87.7	212.3	59.5	245.7	43.2
`imm`	0.0	0.0	43.7	6.3	6.0	1.7	12.4	2.2
`wb`	0.6	0.0	0.0	0.0	2.6	0.7	33.0	5.8
`sd`	0.0	0.0	0.0	0.0	0.1	0.0	0.2	0.0
Unknown	0.0	0.0	28.7	4.2	136.0	38.1	94.5	16.6
Total	619.9		689.4		357.2		568.5	
Average Bitrate (Kbps)	942		65		178		216	
Peak Bitrate (Kbps)	1340		673		924		977	

TABLE 2. A Breakdown of IP Multicast Traffic, By Application. “Unknown” sources had no entries in the session directory, so their application type could not be determined.

Clearly the amount and type of IP multicast traffic is variable from day to day, and can be dependent on a small number of one-time, or non-regularly scheduled sessions. It is therefore difficult to construct a “typical” workload, although it may be possible to predict long-term trends.

5.3 Audio

Figure 4 shows the audio bitrate over time during the XUNET-wide video conference of 22 September 1993. This was a multi-party conference, with almost all of the thirteen participants transmitting audio data at some point during the session. The mean data rate for this `vat` session was 64 Kbps, the bitrate of the uncompressed audio encoding.

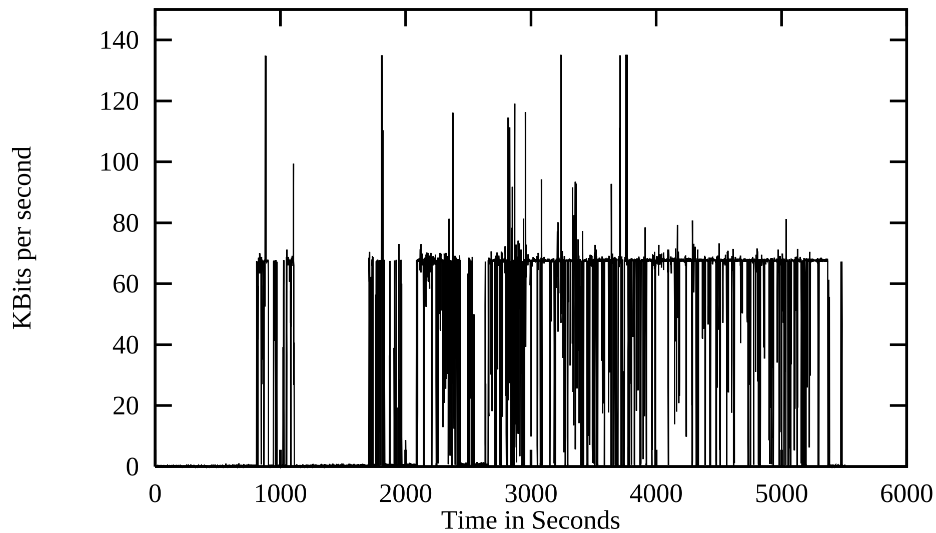

FIGURE 4. Audio (`vat`) Bitrate, XUNET Audio Session, 22 September 1993.

Because `vat` does not do mixing of audio samples, the result of two users speaking simultaneously is generally unintelligible; therefore a well-behaved audio conference will see only one speaker at a time. The bitrate of an audio conference is not likely to change appreciably with the number of participants.

Two interesting phenomena present themselves in Figure 4. First, the "spikes" in the bitrate were caused when multiple users attempted to speak at the same time, noticed they were conflicting with each other, and "backed off". The incidence of these "collisions" suggests that some sort of floor control, either at the application layer or the "user layer", is needed for multi-party audio conferences.

Secondly, we note that there were two periods of activity during the trace: a shorter one lasting for a few minutes (around the 1000-second mark of the trace), then a longer one lasting slightly more than an hour (from about 1800 seconds into the trace through 5500 seconds). The initial activity consisted of most of the conference participants enabling their microphones, setting and checking volume levels, and performing other "utility" functions. The second, longer, period of activity represented the main portion of the conference.

5.4 Video

`nv` was the only known source of motion video observed during any trace. Figure 5 shows the packets transmitted by a single `nv` source at Berkeley during a typical portion of the XUNET-wide video conference on 22 September 1993. Qualitatively, this video source is representative of all the video sources active during this conference. `nv` performs differencing between video frames; it also uses rate control to regulate its network usage. As can be seen `nv` transmits data in bursts, with each burst representing a video frame, divided into 1K packets. The periods between bursts correspond to the effect of rate control, which delays the transmission of the next frame in an attempt to maintain a user-specified average bitrate. The default bitrate for `nv` is 128 Kbps; this parameter is user-adjustable.

By examining the bitrate of the traffic over different timescales, we can see the effect of `nv`'s rate control algorithm. The average bitrate over the length of the trace (approximately ninety minutes) was 125 Kbps; thus in the long term,

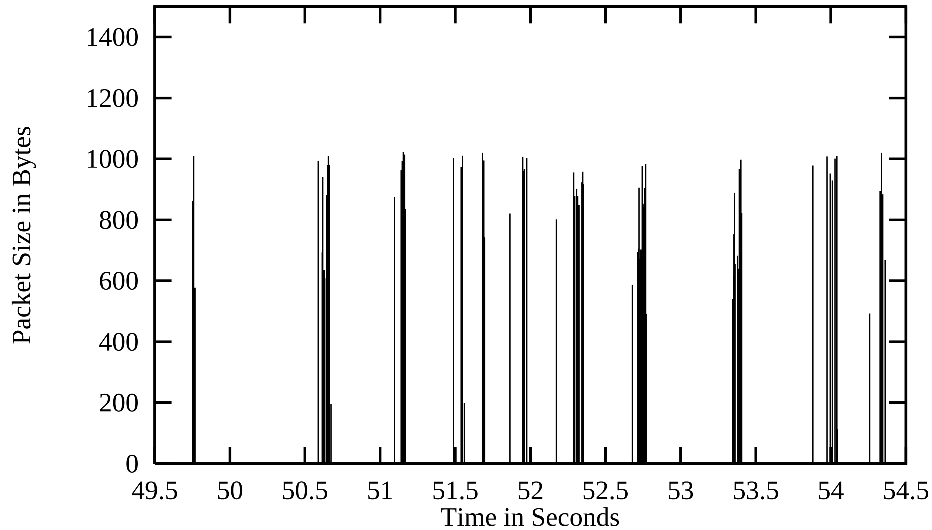

FIGURE 5. Packet Arrivals for a Single Source to XUNET Video (`nv`) Session, 2 September 1993.

`nv` is properly limiting its traffic. The peak bitrates over various intervals from one-tenth of a second to a minute are shown in Table 3.

Interval (seconds)	Peak Bitrate (Kbps)	Peak to Average Ratio
0.1	1735.8	13.83
0.5	502.0	4.00
1.0	270.4	2.15
5.0	168.2	1.34
10.0	144.8	1.15
30.0	132.8	1.06
60.0	130.9	1.04

TABLE 3. Peak Bandwidths for an `nv` Conversation Over Varying Intervals. The average bitrate for this connection was 125.5 Kbps.

The dramatically higher bandwidth at the shortest period (one-tenth of a second) is due to the fact that in this conversation, many frames were separated by relatively long intervals, due to the effect of rate control (`nv` sends out all the packets for an entire frame at once, then waits an interval appropriate to the user-selected bitrate and the amount of data contained in the frame). This resulted in bursts separated by periods during which no traffic was sent at all. This burstiness would be alleviated if `nv` were able to send the packets for a given video frame more evenly spaced.

By examining the cumulative distribution of packet sizes for this conversation (shown in Figure 6), we observe that `nv` sends fairly large packets (in this context, the term “packet size” refers to the UDP payload, excluding headers and before any fragmentation by IP). The minimum observed packet size was 12 bytes, with a maximum of 1796 bytes[3], a mean of 836 bytes, and a median of 905 bytes. We contrast this distribution with those of wide-area TCP conversations measured in [Cáceres92], in which 80% of `telnet` and `rlogin` packets carry less than 10 bytes of user data, while almost 90% of `ftp` packets are less than 512 bytes long.[4] Assuming that this trace is representative of `nv` conversations, we can conclude that `nv` generally sends larger packets than `ftp` and `telnet`/`rlogin`, the most prevalent applications seen across wide-area portions of the Internet.

3. Packet sizes greater than 1024 bytes are the result of a bug in some versions of `nv`.

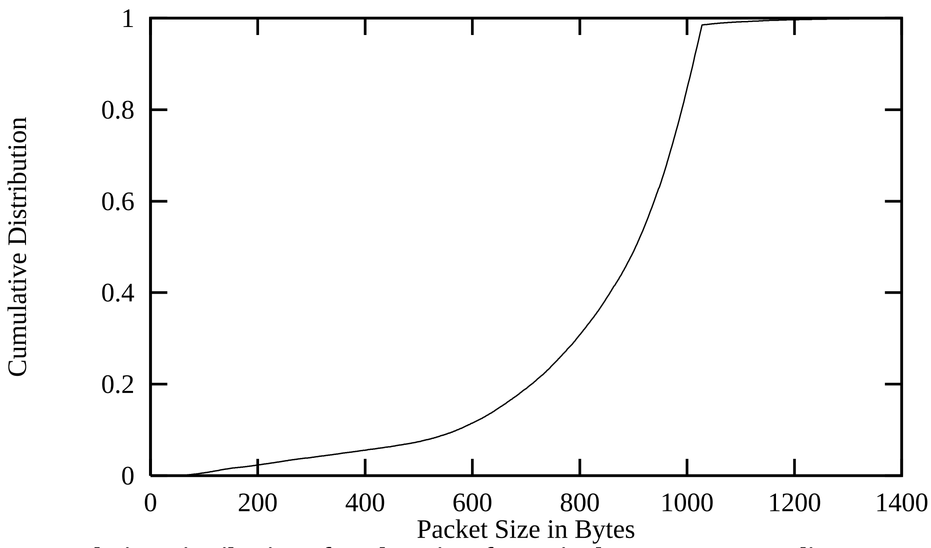

FIGURE 6. Cumulative Distribution of Packet Sizes for a Single nv Source, Sending to XUNET Video, 22 September 1993.

It is also useful to consider the distribution of packet interarrivals for this conversation, and contrast it with the more conventional data applications. Figure 7 shows the cumulative distribution of interarrival times. Packets were collected on the same subnet as the sending host, in order to eliminate any artifacts caused by queueing in gateways or mrouters. The average interarrival time was 46 ms, with a median of 6 ms. These interarrivals tend to be shorter than those found in conventional data applications; the median interarrivals for `ftp` conversations were seen to be in the tens of milliseconds[5] and in the hundreds of milliseconds for `telnet`/`rlogin`. We note, however, that the interarrivals of packets in any given `nv` conversation are somewhat dependent on the sending rate selected by the user.

5.5 Image Distribution

We next examine the video traffic generated by the still image distribution program `imm`. `imm` is designed for the reliable multicast of compressed graphics images (typically tens or hundreds of kilobytes) which change slowly (for example, once or twice an hour). The bitrate over time for a typical `imm` session is shown in Figure 8, in which a single source was multicasting JPEG-compressed satellite images from the GOES-7 weather satellite, once every half-hour. These images usually are about 50-100 KB in size (sources for some other sessions send images about 200-300 KB in size). A day-long traffic trace of this session showed an average throughput of 270 bps (0.27 Kbps) and a one-second peak throughput of 8.8 Kbps. This is a small amount of data compared to most other network services.

Multicast data were either being transmitted by the image server program or by the receiving clients. The former data consist of the image, sent in 1 KB chunks approximately one second apart (these are the tall “spikes” in Figure 8). The server and clients use a negative acknowledgement protocol, in which clients request portions of the image they know to be missing; these negative acknowledgements account for the remainder of the traffic. For many of the one-second intervals observed, there was no data at all transmitted to this session.

We suggest that multicast-based services such as `imm` are appropriate for precisely these types of data: files that are reliably transmitted over long intervals, which will be used by a large number of people. Naive users wishing to

4. [Cáceres92] notes that the packets sizes of “bulk transfer” applications such as `ftp` are heavily dependent on the Maximum Transfer Unit (MTU) of a network, so such statistics should not necessarily be used to model these applications. Typical `telnet`/`rlogin` packets are much smaller than any network’s MTU, and the `nv` measurements discount the effects of the MTU.

5. The interarrival times of `ftp` packets depend on the effects of line speed and TCP flow control, so this comparison is not particularly useful.

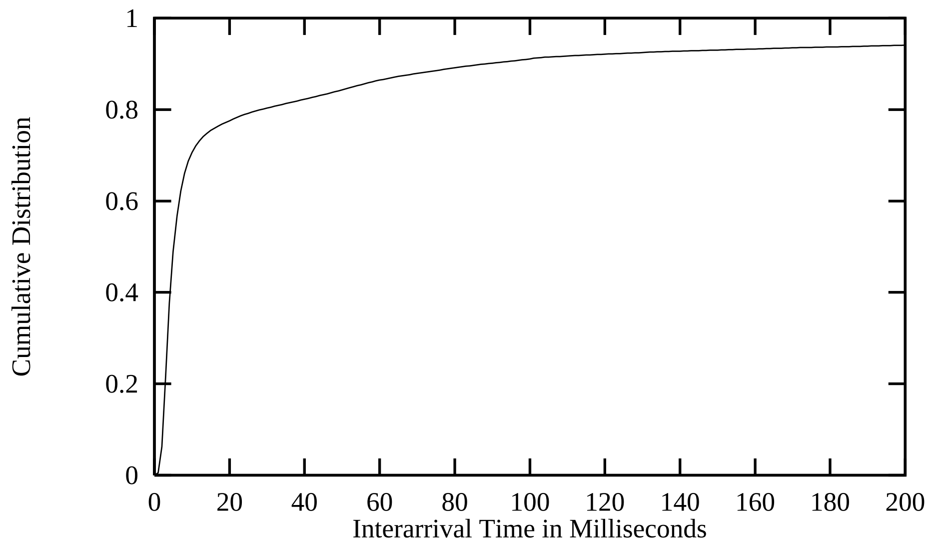

FIGURE 7. Cumulative Distribution of Packet Interarrival Times for a Single `nv` Source, Sending to XUNET Video, 22 September 1993.

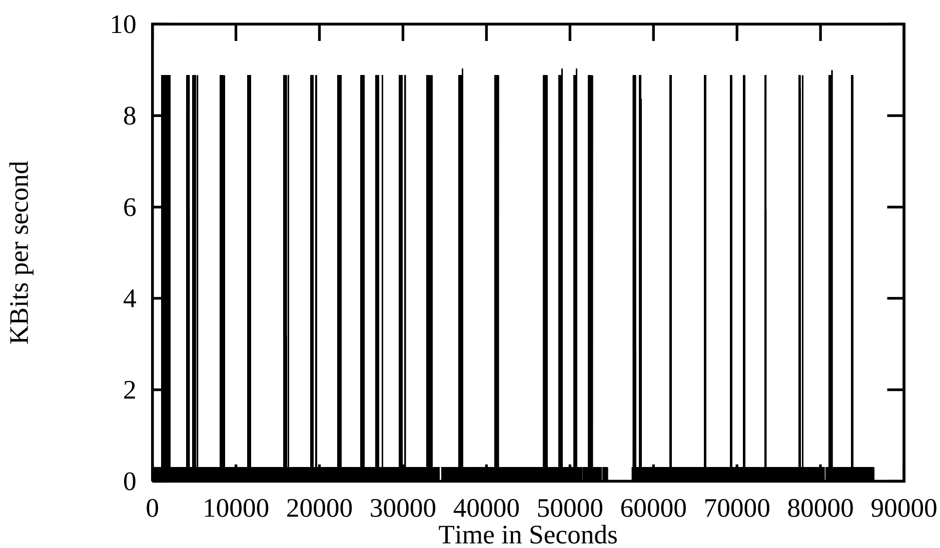

FIGURE 8. Image Distribution (`imm`) Bitrate, GOES-7 VIS Session, 20 January 1994.

retrieve such data (weather maps, for example, for background pictures) have been known to write automated scripts to fetch files from remote archive servers; multiple users retrieving the same files from an off-site server obviously is inefficient and can have a noticeable effect on the network traffic to and from an Internet site. For example, [Paxson94] documents that for some month-long intervals, periodic transfers of this type constituted up to 61% of all off-site FTP connections at the Lawrence Berkeley Laboratory.

5.6 Anomalous Traffic

Examination of the 20 January 1994 trace revealed a rather curious fact. Although there were no active audio or video sessions advertised in `sd`, IP multicast traffic was sent to a total of 65 destinations (we define a destination as a unique IP multicast address and port number pair). We excluded a local audio session from further consideration, in order to concentrate on traffic coming into Berkeley from off-site. A breakdown of the off-site traffic seen at Berkeley is shown in Table 4.

Traffic Type	Destinations	Data (MB)	Fraction (%)
imm (advertised)	7	45.8	33.3
nv (advertised)	3	11.3	8.2
vat (advertised)	12	42.8	31.1
wb (advertised)	2	0.0	0.0
Unadvertised but known	7	7.6	5.5
Unknown	33	30.0	21.8
Total	64	137.5	100.0

TABLE 4. Off-Site IP Multicast Traffic, 20 January 1994.

Further analysis of the 33 unknown destinations implies that almost all of them were either data or control for `vat` sessions (`vat` uses one destination for audio data and another to maintain the list of all participants in the session). Most of these sessions lasted only a few minutes, with zero to two sources sending audio data. Frequently all participants were at the same site. One session, however, had ten participants, transmitting at various times throughout the length of the 24-hour trace. We evaluate this network traffic as being due to new or inexperienced MBONE users experimenting with the tools either within their workgroups or in small, private conferences. Although this type of traffic was not of use to the MBONE community at large, we note that the TTLs of these transmissions were sufficiently high that a majority of (if not all) MBONE routers received them.

A later analysis of the four-hour-long trace of 21 April 1994 showed an even more dramatic incidence of similar, unknown traffic sessions; 38.1% of all IP multicast bytes seen at Berkeley (142.6 MB out of 374.6 MB) were from sessions unadvertised in `sd`. These bytes corresponded to 38 of the 63 destinations seen.

We believe that the wide extent of propagation was likely due to the fact that the versions of `vat` in use at the time had high default TTL values. Beginning with version 3.3, this parameter has been reduced to more properly limit the scope of transmissions where no TTL value is specified by the user.

Although the mere existence of unadvertised sessions is by itself not harmful, the nature of many of these sessions (few participants, frequently at one site), the volume of data transmitted and the scope of propagation suggest that some stronger protection against naive users is required. A more widespread deployment of `mrouted` version 3, which performs pruning of multicast forwarding trees, would alleviate this problem, as would implementations of the Protocol Independent Multicast (PIM) [Deering94]. Data from these “unknown” sessions, regardless of TTL, would be confined to tunnels and subnets leading to the session participants. Such pruning, however, could potentially make misbehaving users harder to locate [Jacobson94c].

We note that since the time of these measurements, a number of messages on the `mbone` mailing list have documented incidents in which users have sent high bitrate video with high TTLs, creating a significant disruption to MBONE performance.

6 Conclusions

In this paper we presented the results of measurements of IP multicast traffic seen on the MBONE. From these data, we conclude that the aggregate volume of IP multicast traffic is highly dependent on a small number of “special” events, such as Internet-wide multicasts of seminars. It is therefore difficult to determine what exactly constitutes a “typical” workload for IP multicast traffic. From observing packet traces of many-party audio conferences, we conclude a need for some sort of “floor control”, either in applications or between users. We have examined the traffic characteristics of the predominant video conferencing tool in use, and have seen that they are quite different from those of conventional data traffic in a wide-area environment. We note the use of multicasting to distribute still images, and contrast its network usage with more naive approaches to distribution. Finally, we note that inexperienced users sending unnecessary traffic have been seen to account for a significant fraction of MBONE traffic, indicating the need for stronger scope control.

7 Acknowledgements

The author gratefully acknowledges the comments and feedback of Professor Domenico Ferrari, Mark Moran, Hoofar Razavi, and Yvonne Sang, all of the University of California at Berkeley, and S. Keshav of AT&T Bell Laboratories. Special thanks to Kimberly Keeton of the University of California at Berkeley for her ideas and helpful suggestions. The author was supported by an NSF Graduate Fellowship and by Department of Energy grant DE-FG03-92ER25135. Additional support was provided by AT&T Bell Laboratories, Pacific Bell, and Digital Equipment Corporation.

8 References

[Cáceres92] R. Cáceres. “Multiplexing Traffic at the Entrance to Wide-Area Networks”, PhD Thesis, University of California, Berkeley, CA, December 1992.

[Dang93] W. Dang. `imm`, available by anonymous FTP from `ftp.hawaii.edu`, 1993.

[Deering89] S. Deering. “Host Extensions for IP Multicasting”, *Internet Request for Comments 1112*, August 1989.

[Deering93] S. Deering. `mrouted` (a component of the IP multicast distribution), available by anonymous FTP from `gregorio.stanford.edu`, 1993.

[Deering94] S.Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. “An Architecture for Wide-Area Multicast Routing”, *Proceedings of SIGCOMM ’94*, London, England, August 1994.

[Fraser92] A. Fraser, C. Kalmanek, A. Kaplan, W. Marshall, and R. Restrick. “XUNET-2: A Nationwide Testbed in High-Speed Networking”, *Proceedings of INFOCOM 1992*, Firenze, Italy, May 1992.

[Frederick94] R. Frederick. `nv`, available by anonymous FTP from `parcftp.xerox.com`, 1994.

[Jacobson93] V. Jacobson. `sd`, available by anonymous FTP from `ftp.ee.lbl.gov`, 1993.

[Jacobson94a] V. Jacobson, C. Leres, and S. McCanne. `tcpdump`, available by anonymous FTP from `ftp.ee.lbl.gov`, 1994.

[Jacobson94b] V. Jacobson and S. McCanne. `vat`, available by anonymous FTP from `ftp.ee.lbl.gov`, 1994.

[Jacobson94c] V. Jacobson. Message `<9407291733.AA11443@rx7.ee.lbl.gov>` to the `mbone` mailing list, July 1994.

[Keshav94] S. Keshav. “Experiences with Large Videoconferences on XUNET II”, *Proceedings of INET ‘94*, Prague, Czech Republic, June 1994.

[Lidl94] K. Lidl, J. Osborne, and J. Malcolm. “Drinking from the Firehose: Multicast USENET News”, *Proceedings of the USENIX 1994 Winter Conference*, San Francisco, CA, January 1994.

[Lockwood93] J. Lockwood. “Proposed Mbone/Xunet Extensions”, Revision 1.7.7, University of Illinois at Urbana-Champaign, October 1993.

[Macedonia94] M. Macedonia and D. Brutzman. “Mbone Provides Audio and Video Across the Internet”, *IEEE Computer*, April 1994.

[McCanne93] S. McCanne and V. Jacobson. `wb`, available by anonymous FTP from `ftp.ee.lbl.gov`, 1993.

[Paxson94] V. Paxson. “Growth Trends in Wide-Area TCP Connections”, to appear, *IEEE Network*, 1994.

[Waitzman88] D. Waitzman, C. Partridge, and S. Deering. “Distance Vector Multicast Routing Protocol”, *Internet Request for Comments 1075*, November 1988.