Wide-Area IP Multicast Traffic Characterization (Extended Version)

Robert Beverly, k claffy

Abstract— **IP multicast is gaining acceptance among service providers as the protocols and infrastructure mature. Yet characteristics of multicast traffic remain poorly understood. Using passive OC-12 monitors we observe multicast traffic on links connecting aggregated customers and peer networks to our native multicast backbone network. We first refine existing traffic flow profiling methodologies via an exploration of temporal differences in multicast packet trains. Based on this framework we collect multicast flow traces from four geographically dispersed nodes in the vBNS network over a one-month period. We present multicast-specific traffic characteristics including packet and flow size distributions, packet duplication and fragmentation, address accumulation and address space distributions. We analyze the distribution of sources per group and the implications on the applicability of emerging single-source protocols. Analysis reveals results contrary to prevailing wisdom, including: (i) a preponderance of single-packet flows; (ii) a highly variable packet size distribution, with many large packets and strong modes; (iii) the existence of fragmented multicast traffic; and (iv) an insignificant number of simultaneous multiple-source groups. Based on our analysis, we recommend policies for deployment and improvements to protocol implementations.**

I. Introduction

IP multicast is an increasingly popular technology that relies on the data network to provide packet replication for one-to-many or many-to-many communication. Collaborative and multimedia applications with large audiences, such as voice and video, are optimally suited to multicast distribution as it limits the traffic load imposed on the network. Despite the advantages of multicast, many service providers have historically not embraced the technology because of weak demand, unknown traffic and routing implications, undefined billing models and a strong motivation to maintain the stability of existing unicast customer connections [1]. However in the last decade multicast has evolved from an experimental sub-scalable technology into a mature network service that providers are now deploying.

Worldcom's very-high-performance Backbone Network Service (vBNS) [2], currently a nationwide OC-48c packet over SONET backbone, began offering IP multicast to the research and education community in 1995 under an NSF infrastructure grant. Following the expiration of the five-year grant, the vBNS transformed into a commercial service and has attracted commercial customers with unique requirements such as high-performance IP multicast. Typical customer multicast applications include satellite broadcast replacement, audio and video distribution, multimedia conferencing and distributed simulations. The breadth and scope of customers and applications, in combination with robust connectivity to other peer multicast-enabled backbones, provides an ideal network to investigate. Additional details on the vBNS network and its multicast service are given in [3].

Situated at each vBNS point-of-presence (POP) is an OC12MON passive traffic monitor. These monitors facilitate measurement and analysis of ingress and egress traffic by optically splitting the OC-12c ATM link between the core router in a POP and the edge aggregation switch to which customer circuits connect. Custom software on the monitors allows us to analyze multicast packets and flows.

We first define a methodology by which to characterize multicast flows and then collect multicast IP header traffic traces from four geographically dispersed nodes on the vBNS over a one-month period. We present multicast-specific traffic patterns and characteristics including: packet and flow size distributions, packet duplication, packet fragmentation, address accumulation and group address space utilization. In addition, we observe the quantity of multiple-source multicast and the applicability of emerging single-source protocols.

The paper begins with a summary of previous traffic studies and related work in Section II. Section III describes the traffic monitoring hardware and software. The monitoring points and node equipment are detailed in Section IV. Section V defines the multicast flow metrics and Section VI presents a detailed traffic analysis. The paper concludes with a summary of major findings and suggestions for future research.

An abridged version of this paper appears in *IEEE Network*, January/February 2003.

R. Beverly is with Worldcom Advanced Internet Technology, Ashburn, VA and MIT Lab for Computer Science, Cambridge, MA. E-Mail: rbeverly@mit.edu

k claffy is with CAIDA, San Diego Supercomputing Center, University of California, San Diego, CA. E-Mail: kc@caida.org

II. Previous work

In recent years a number of significant traffic studies have been performed; however, IP multicast traffic profiling has received little dedicated attention. The body of previous multicast measurements typically involves actively querying network devices such as routers [4] or requires the monitor to join the multicast group in order to pull traffic to the measurement host. Our method of passive measurement has neither of these dependencies. We summarize applicable work in this section as well as highlighting any differences in approach.

Mah captured several multicast-specific packet header traces from the UC Berkeley campus in 1993 [5]. His data provides a historical baseline, but represents sub-Megabit rates collected at the edge of the network over relatively short (less than a day) time periods. Additionally, the applications and protocols have changed considerably since this study.

Almeroth offers insight into the growth and usage of the MBone, the multicast backbone, by listening for Session Announcement Protocol (SAP) [6] advertisements and automatically joining the multicast group of each announced session and then capturing packets belonging to the session [7]. By monitoring captured Real Time Control Protocol (RTCP) [8] packets, Almeroth estimates end user participation and behavior. While we draw on this analysis for comparison, our measurement approach is broader. First, our monitoring does not depend on applications including RTCP support, or the successful network delivery of RTCP packets. Second, we monitor all multicast traffic rather than just traffic to explicitly announced sessions. The value of monitoring all traffic within a provider network is underscored by the fact that our one-month study captured only 25% fewer packets than were captured in this 4.5-year study.

Sarac presents a framework for managing multicast traffic and a survey of active measurement and SNMP-based multicast tools [9]. These tools are particularly useful for determining the quality or extent of multicast connectivity, but require active end host or router participation. SNMP provides a common interface for monitoring multicast protocols as well as coarse-grained statistics including packet and byte counts. However our analysis examines the traffic in much greater detail without any dependence on the routing hardware maintaining traffic statistics.

Thompson's 1997 analysis of the internetMCI backbone [10], now owned by Cable & Wireless, classified multicast as IP-in-IP (IP protocol type 4) packets, since multicast traffic within the backbone was encapsulated in unicast IP at that time. The IP-in-IP traffic exhibited no discernible daily pattern, although as much as 20% of the studied link's byte traffic and 10% of the packets were IP-in-IP during several five-minute sample intervals. Thompson's analysis artificially expired flows that were active for more than an hour. Because multicast applications can be long-lived, this expiration may cause unrepresentative traffic spikes in the distribution of multicast flow sizes. We examine flow timeouts in Section V.

A more recent long-term ten month report on NASA Ames Internet Exchange (AIX) traffic [11] did not specifically examine multicast, but found that only 0.06% of the packets and 0.09% of the bytes were IP-in-IP. This much lower level is likely due to the decline of tunneled multicast traffic in favor of modern protocols that transport multicast packets natively.

For our study we present a complete view of the network from a service provider's perspective as opposed to the limited view offered at the edge of the network. We capture all multicast packet headers from several high capacity links aggregating many commercial customers and dozens of peers to the vBNS multicast backbone. Our monitoring is non-intrusive, non-sampled, has no protocol dependencies and does not impose artificial flow timeouts.

III. Traffic monitoring hardware and software

This section describes the OC12MON[1] passive traffic monitor hardware and software used to gather the data for our study. The OC12MON is the evolutionary product of the original OC3MON [12] developed in an MCI-NLANR collaboration. The OC3MON evolved into CAIDA's[2] CoralReef project to support broader user requirements. Because of vBNS-specific design constraints and history, including a requirement to run multiple monitoring processes simultaneously, we did not use the CoralReef driver for this study. We did, however, use the efficient CoralReef hash code [13].

A. *OC12MON hardware*

Each OC12MON is a rack-mounted workstation with dual 600MHz Intel Pentium III processors, two Ultra-SCSI LVD 32GB drives, an Ethernet interface, two OC12 interface cards and a 66MHz 64-bit-wide PCI bus. The ATM interface cards are manufactured by Mindspeed Technologies, formerly Apptel [14]. Because the monitors are PC-based, they are inexpensive enough to facilitate widespread deployment in the network.

Optical splitters carry a fraction of the light from each fiber to the receive port of an ATM interface card on the

[1]While the backbone links are OC48, there was no working OC48 monitor at the time of our study. Because the OC12MON captures all ingress and egress customer traffic, it was sufficient for our analysis.

[2]Cooperative Association for Data Analysis, http://www.caida.org

monitor. Specifically, the fiber from the transmit port on the core router is connected to a splitter. The splitter takes this signal and sends 80% of the light to the receive port on the ATM switch and the remaining 20% to the receive port of the monitor. The fiber from the transmit port on the ATM switch is similarly split, thus two ATM interface cards and two splitters are required to monitor both directions of the link.

B. OC12MON software

The monitors run the Linux operating system which supports multiple processors, out-of-band management, a robust development environment and all common UNIX utilities. The OC12MON Linux Apptel driver provides a libpcap-compatible [15] interface to user applications. The card captures only the first cell of each packet on the link, delineated by ATM AAL5 trailers. Provided that the packet is not a fragment and contains no IP options, this captured cell will include the IP and transport layer headers. The monitor continually buffers cells to DMA memory from which multiple distinct user processes may then concurrently read. This flexible architecture supports specific real-time monitoring needs without interrupting other operational monitoring processes running in the background. For example, an intrusion detection application can run simultaneously with a traffic profiling application on the same monitor.

The collection program, written in C, uses the pcap library to gather multicast flows, packet size, byte and packet counts and IP fragment and option counts. We define a multicast flow in the next section. For each multicast packet, the monitor determines whether the flow is active by checking a hash table for existing flow state. If none exists, the monitor creates a flow entry and sets two flow timestamps, *first_seen* and *last_seen*. If there is existing state, the monitor increments the flow's packet and byte counts and updates the flow's *last_seen* timestamp. The byte count is incremented by the value of the IP total length field for non-fragmented packets. For fragmented UDP datagrams, identified by the 'more fragments' IP header bit being set and an IP fragmentation offset of zero, we use the value of the UDP length field in the first fragment which permits the program to reconstruct the size of the original packet as it left the originating host. This approach allows the monitor to characterize application behavior. For non-UDP fragments, where no UDP length field is available, the IP total length field is again used to represent the packet size. For the purpose of maintaining flow data, subsequent fragments of the same series are ignored.

Because multicast flows may be active for long periods, such as with long-lived audio or video conferencing, active flows are not artificially expired. However, a flow sweeper periodically checks for flows that are inactive for longer than the timeout period (i.e *last_seen* - *first_seen* > *timeout*) and expires them. By expiring inactive flows the monitor is not forced to maintain stale flow state. The collection software traps all UNIX signals and terminates gracefully by immediately expiring active flows. If a flow lasts for the entire duration of the collection period, it is expired once the program exits.

In addition to the aggregate per flow packet and byte counts, the monitor maintains packet size counts. The packet size distribution is stored in two arrays, one for each direction of the link, of 2^{16} elements corresponding to the size of the IP length field. For each multicast packet, the IP length field is used to increment the appropriate array element. We intentionally do not attempt to differentiate between fragmented and non-fragmented packets for the packet size distribution counts. In this way, application packet sizes are retained in the flow state, while the size of packets seen on the network are maintained separately. The monitor also records the number of IP fragments, IP datagrams with the 'don't fragment' (DF) bit set and packets with IP options. Every five minutes, the packet size arrays and fragment counts are written to disk and initialized to zero. Finally, the monitor maintains unicast and multicast packet and byte counters that are similarly written to disk every five minutes, so we can track trends in the relative proportion of multicast versus unicast traffic over time.[3]

We validated the performance and accuracy of the monitor using commercial traffic generation equipment [16]. In our test environment we synthesized 100 unique flows, 50 in each direction. Each flow consisted of small 100-byte packets running at 9,418 packets per second (28.3K cells per second) for a total of 941.8 Kpps (2.83 Mcps) on each card. This corresponds to 99.7% of full line rate on each interface. The individual flow counts were summed and showed no loss.

Flow and packet data from each monitor was copied to a central workstation for processing. We performed subsequent analysis on the data off-line using a collection of `perl` scripts.

IV. Description of the monitoring points

Each vBNS POP contains a backbone router with OC-48c packet over SONET connectivity to POPs in neighboring cities, an IPv6 aggregation router, an active performance host, an OC12MON and an ATM aggregation

[3]While packet and byte values can be derived from flow data, it is not possible to accurately determine the distribution of traffic over the flow's potentially long lifetime.

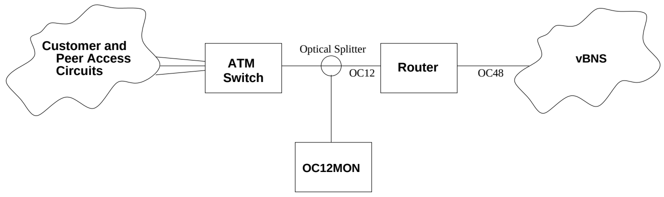


Fig. 1. Node Measurement Configuration

switch. Customer and peer circuits terminate on the ATM switch. The traffic monitor captures traffic by optically splitting the OC-12c ATM link between the backbone router and the ATM switch as shown in Figure 1. This study includes 29 days of data from Monday, July 2 20:50 Eastern Daylight Time (EDT) to Monday, July 30 2001 20:50 EDT. During this period, we collected data from four nodes in the United States: Chicago, Il, Houston, TX, Washington, DC and New York, NY, denoted as CHI, HOU, WAS and NYC respectively. We selected these nodes because of the large concentration of multicast customers and peer networks that connect at these particular POPs. Each POP supports a variety of commercial customers whose applications include streaming audio and video, distributed simulations and file distribution. The vBNS peers with dozens of large multicast-enabled networks including UUcast, Verio, Abilene, CANARIE, ESnet, NASA, TRANSPAC, SREN, etc. at the CHI node. The rendezvous point (RP) for public multicast groups in the vBNS is the CHI router. Because of the large and diverse set of customer and peer networks at CHI, it is the most interesting multicast monitoring point on the vBNS and we often use it for examples in this paper.

V. Multicast flow metrics

This section defines a multicast flow via an exploration of the relative effects of different flow timeout values.

In previous studies [17] various methods were used to define unicast flow beginning and end points, including monitoring TCP packets with SYN or FIN control flags. While such control packets unambiguously bound the flow, connectionless transport protocols, namely UDP, do not have any such mechanism. UDP is the most commonly used transport protocol for multicast applications. For this reason we used a temporal timeout approach to delineate begin and end points of multicast flows.

We define a multicast flow as a unidirectional IP traffic flow containing a class D destination IP address with a unique <*source-address, source-port, multicast-group-address, destination-port, IP-protocol*>. This definition correlates well with current sparse-mode multicast protocols such as Protocol Independent Multicast (PIM-SM) [18] that rely on shortest path distribution trees. In the PIM-SM model, state is established unidirectionally per unique source and group. While we note that PIM-SM can use the shared tree exclusively, the default behavior implemented by vendors is to switch to the shortest path tree immediately after the router directly attached to the receiver receives packets from the shared tree. Some router vendors do not provide a mechanism for changing this behavior. Previous flow definitions used 64-second expiration intervals [19], however because of the long-lived nature of multicast applications, we did not assume that this value was applicable to multicast measurement.

The monitors differentiate between an active flow and an expired flow by means of the timeout value. Any flow with a *last_seen* timestamp less than the difference between the current time and the timeout value is expired and the flow statistics are written to disk. To determine an appropriate timeout value, we collected a 24-hour multicast trace at the CHI node beginning Monday, July 9 2001 at 10:30 EDT. This trace included 55.96M packets and 55.49G bytes. Performing a similar analysis to [19], we applied four different flow timeout values: 4, 32, 256 and 2048 seconds to the same observed traffic.

Using a smaller timeout value increases the total number of flows and creates flows with smaller byte and packet counts. A single bursty flow may be broken into multiple shorter flows if the timeout value is too low. Timeout values of 4, 32, 256 and 2048 seconds yielded approximately 213, 60, 44 and 25 thousand flows respectively. Increasing the flow timeout value from 4 to 32 seconds leads to a 71% reduction in the number of flows, however the flow reduction factor is decidedly non-linear as the timeout value increases. We find that even with a 2048 second timeout, 67% of the flows consisted of only a single packet. We discuss the predominance of single-packet flows in Section VI. A timeout value between 32 and 256 seconds is appropriate to minimize both the amount of stale state and the amount of new state creation. Since the 'hello' packet transmission interval for PIM defaults to 30 seconds with three retries, we considered 90 seconds a lower bound flow timeout so that PIM protocol traffic, which is itself multicast, would not generate unnecessary flow state if a 'hello' packet was lost. Because of the long-lived nature of multicast applications and the 90 to 256 second bounding condition, we selected 120 seconds as an appropriate timeout value for our study.

VI. Multicast traffic analysis

In this section we examine the multicast data collected in detail. Because the monitor maintains both individual packet data and stateful flow data, we divide our analysis

TABLE I
Summary of Collected Data

Site	Flows(k)	Pkts(M)	Bytes(G)	Mean Pkt(B)	Pkt StdDev(B)	Pkts/flow(k)	Bytes/flow(k)
CHI	3,411	5,918	2,998	507	439	1.7	879.1
HOU	0.13	3,962	5,520	1393	116	31,447.4	43,809,531.5
WAS	366	624	136	218	198	1.7	371.9
NYC	148	2,192	2,737	1249	452	14.8	18,423.4
Total	3,925	12,697	11,392	897	567	3.2	2,901.9

into two parts: non-flow-based and flow-based results.

A. Packet-based results

We first present non-flow-based results, including link traffic volume, packet size distribution and IP fragmentation statistics. We comment on time-of-day and day-of-week patterns and draw comparisons between the multicast and unicast traffic.

A.1 Link traffic volume

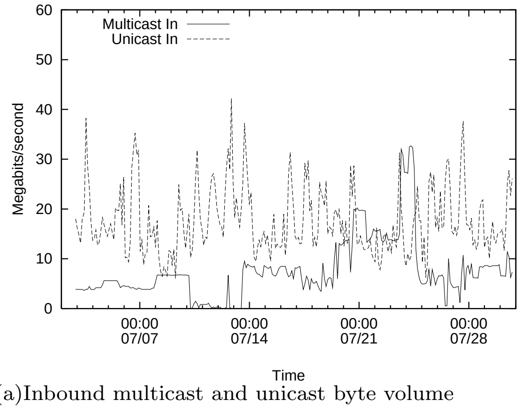
The four monitors collected a total of 11.39 terabytes (12.70G packets) over the month long collection period. Table I summarizes the collected data and provides average packet size, packet size standard deviation, average packets per flow and bytes per flow for each link. Each link has widely varying traffic patterns reflecting the variety of customers and applications at each site. Customers at HOU and NYC, for example, continually broadcast and receive high rate (12Mbps) video to a static group. Many of these applications were active for the entire duration of the collection period yielding extremely large flows. While the average (mean) packet size is rather large (897 bytes across all monitoring points), the standard deviations are also large (567 bytes). Note that mean values of packet size distributions carry little statistical importance due to strong modalities in the distribution. We examine packet size characteristics in detail in the next subsection.

We did not observe any correlation between unicast and multicast traffic levels. Unicast traffic on the links exhibited traditional diurnal patterns for both packet and byte volumes. In contrast, multicast traffic is relatively flat and invariant. For example, Figure 2 illustrates how the multicast traffic rate on the CHI link remains constant for multiple days. This characteristic is not unexpected since many multicast sources are automated or continuous, such as streaming video or periodic test traffic, and traffic on the link is not directly proportional to the number of receivers. Traffic will only increase when a new receiver joins a group from a point in the network with no other existing receivers. Similarly, traffic will only decrease when all receivers have disappeared from a particular link. Since many peer networks connect to CHI, individual fluctuations in user participation are not likely to affect traffic volumes. Multicast traffic peaked around 35, 18, 2 and 35 Mbps for the CHI, HOU, WAS and NYC sites respectively.

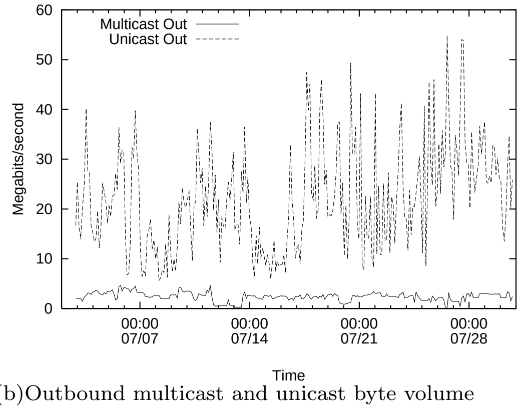
On a finer time granularity we do see intra-day and intra-week multicast traffic variability. For greater readability we focus on one week of traffic in detail, again from the CHI link. Figure 3 plots multicast bit rates for one week on the CHI link (an expansion of the right side of Figures 2(a) and 2(b)) beginning Monday July 23rd 2001. This graph shows a constant flat rate from 00:00 until approximately 08:00 most days. Each morning the rate jumps, particularly on the Monday morning shown in the graph. Intra-day byte traffic increased by as much as 300%. The outbound multicast rate shows less variability than the inbound rate, suggesting that additional receivers joining each morning contribute more to rate fluctuations. Another flat rate continues throughout the weekend, which began on Saturday 7/28. This pattern indicates that users may start and stop multicast applications during the day, but typically leave them running through the night and weekend. The fact that inbound and outbound rates are closely correlated suggests the presence of applications that employ receiver feedback mechanisms such as RTCP, have multiple sources or have senders and receivers on the same ATM circuit. We examine the effect of ATM as an access medium for multicast traffic in Section VI-B.4.

A.2 Packet size distribution

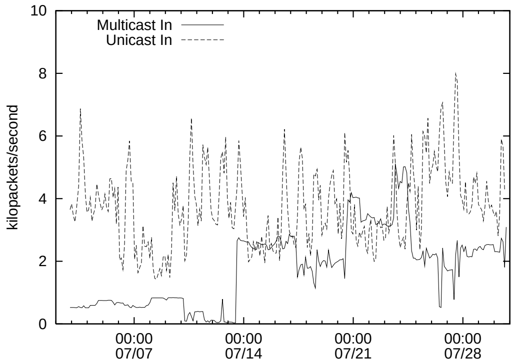
Router performance is typically bounded by packet rates rather than by bit rates. Multicast traffic is particularly problematic as the burden of packet replication is placed on the router. For comparison between unicast and multicast, we captured complete packet size data from the CHI link. Figure 4(a) shows a logarithmic scale histogram of unicast packet sizes seen (on both directions of the link) from 11:15 to 12:15 EDT on a Tuesday of the study. The unicast packet distribution exhibits strong modes at 40 and 1500 bytes with smaller modes at 90, 576 and 1420 bytes.



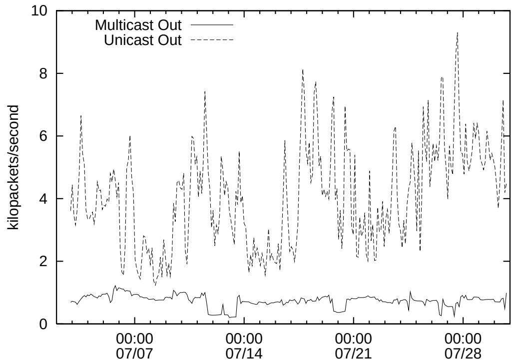
(a)Inbound multicast and unicast byte volume



(b)Outbound multicast and unicast byte volume



(c)Inbound multicast and unicast packet volume



(d)Outbound multicast and unicast packet volume

Fig. 2. Byte and Packet volume from CHI link (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

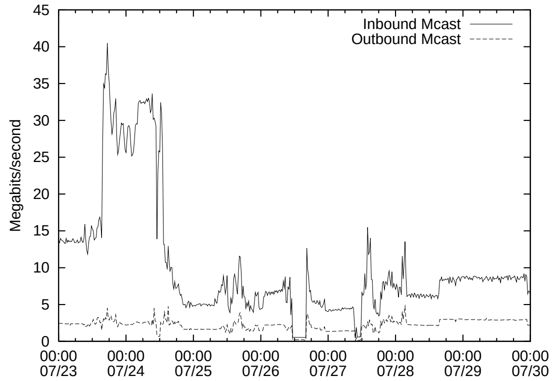
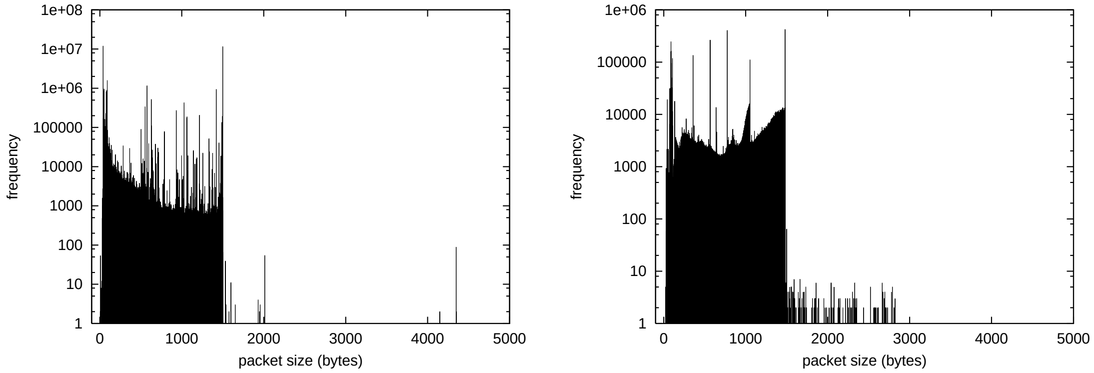


Fig. 3. Multicast traffic rate for the week of Monday, 23 July 2001

The modes are expected due to 40-byte TCP acknowledgment segments and the 1500-byte Maximum Transmission Unit (MTU) of Ethernet-attached hosts. The smaller 576-byte mode is attributable to TCP implementations using a default TCP Maximum Segment Size (MSS). We attribute the 90 and 1420-byte modes to specific applications running during the collection interval. Figure 4(b) is a logarithmic scale multicast packet histogram over the same period. Here we see the top four modes in descending frequency: 1480, 775, 566 and 88 bytes. The 1480-byte packets are likely from applications designed to avoid fragmentation in the face of network encapsulation (we discuss multicast packet fragmentation more in Section VI-A.3); we have not investigated likely sources of the other packet size modes.

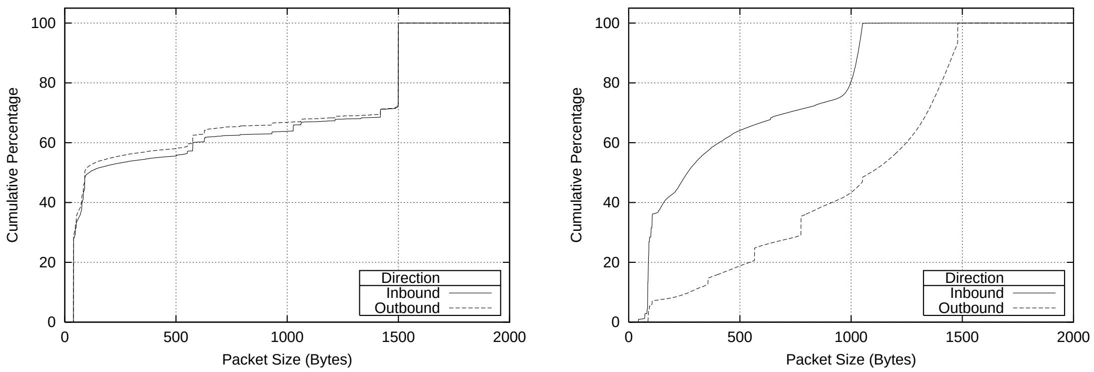
Further disparity between unicast and multicast packet size distributions is evident when also taking into account link directionality. Figure 5 depicts the cumulative packet size distributions, in each direction, for unicast and multicast traffic. While the unicast traffic packet size distribution is symmetrical, the multicast graph shows fewer small packets outbound than inbound.

Figure 6 plots the cumulative multicast packet size distribution for traffic in both directions at each site over the entire month collection period. These distributions belie the notion that multicast packets are always small. The strong mode at 1428 bytes seen both at HOU and NYC is from a commercial MPEG encoder [20] that defaults to



(a)Unicast packet size histogram (b)Multicast packet size histogram

Fig. 4. Logarithmic-scale packet size histograms from one-hour sample on CHI link (11:15 to 12:15 EDT, August 21, 2001)



(a)Unicast packet size histogram (b)Multicast packet size histogram

Fig. 5. Cumulative packet size histograms from one-hour sample on CHI link (11:15 to 12:15 EDT, August 21, 2001)

sending 1400 bytes of data, exclusive of IP and UDP headers. However, two other sites, CHI and WAS, show a much larger proportion of small packets. 62% of the packets at CHI are 500 bytes or smaller, while nearly 85% of the packets at WAS are 500 bytes or smaller. Almost all packets are 1500 bytes or smaller, suggesting that the source hosts are Ethernet-connected or that larger packets are fragmented by intermediate routers.

To understand how our empirical packet size results fit into the context of common multicast applications, we took a sample of 10,000 packets from four known multicast streams. Based on their SAP advertisements, we identified four different audio and video streams: H.261, MPEG-1 and two MPEG-2 streams. Figure 7 shows that while 80% of the H.261 stream's packets were 500 bytes or less, only 20% of the MPEG stream's packets, either MPEG-1 or MPEG-2, were less than 1000 bytes. We also noted that one of the MPEG-2 streams sent variable size packets, while the other MPEG-2 stream sent two fixed-size packets. The presence of these applications, or applications that behave similarly, is evident in the month-long results.

A.3 IP fragmentation

IP fragmentation occurs when a router must forward a datagram onto an interface with an MTU smaller than the size of the packet. Provided that the 'don't fragment' (DF) bit is not set in the IP header, the router will replicate the IP header, divide the IP data into (MTU minus IP header)-sized chunks and set the fragmentation offset for each new packet. When the packet arrives at its eventual destination, the end host must reassemble the packet. Fragmentation often hurts the performance of both routers, which must expend resources to split the packets, as well as end hosts, which must hold state waiting for all fragments and then reassemble the original packet after all fragments have arrived. If a single fragment of the fragment series is

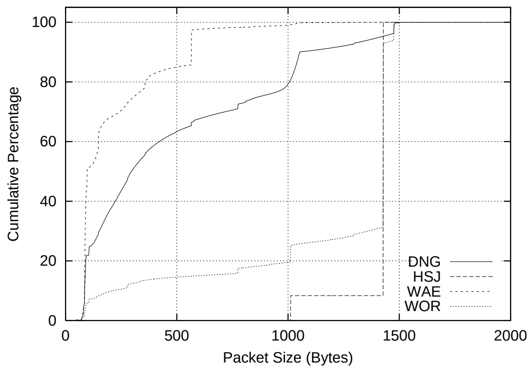


Fig. 6. Cumulative multicast packet size histograms (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

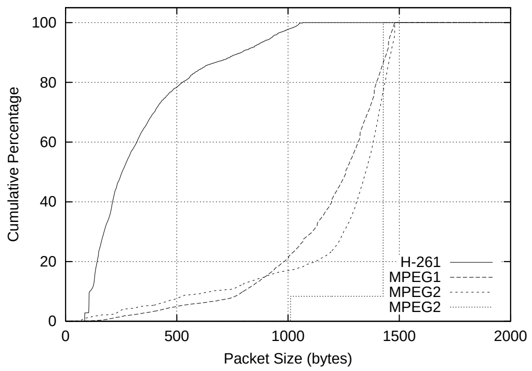


Fig. 7. Representative application packet size histograms (10,000 packet samples)

lost in transit, the other fragments are useless as the original packet cannot be reassembled. Packet loss is particularly problematic in streaming real-time applications such as video and audio, where no retransmission mechanisms are used since the conversation or video will have advanced past the point where the packet was lost.

Table II shows the absolute and relative number of multicast packets at each measurement point with the IP DF flag set, as well as the number of multicast fragments. The number of multicast fragments in this context refers to the number of IP packets on the link with either a non-zero fragmentation offset or the 'more fragments' IP header field set. At each site we find between 0 and 23% of the packets had the DF bit set, suggesting that applications sending these packets are sensitive to fragmentation-induced loss or delay. Between 0 and 1% of the packets at each site were fragments. Among all sites, 3.2% of the packets were marked as DF and 0.5% were fragmented. Our fragmentation results are similar to those found by Shannon, *et al.* in a study of the prevalence of IP packet fragments on the Internet [21]. While the links monitored in Shannon's study carried very little if any multicast traffic, the study noted that UDP is the most prevalent protocol of fragmented traffic and the majority of UDP fragmentation was caused by streaming media applications, particularly video-conferencing.

TABLE II
MULTICAST IP FRAGMENTATION

Monitor	Total Pkts(M)	DF Pkts(M)	Frag Pkts(M)
CHI In	1,792.7	179.6(10.0%)	7.2(0.4%)
CHI Out	4,321.1	100.7(2.3%)	36.5(0.9%)
HOU In	972.6	0.0(0.0%)	0.0(0.0%)
HOU Out	2,989.7	0.0(0.0%)	0.0(0.0%)
WAS In	346.4	26.7(7.7%)	0.0(0.0%)
WAS Out	277.5	60.7(21.9%)	0.0(0.0%)
NYC In	2,107.2	0.1(0.0%)	25.0(1.2%)
NYC Out	209.7	47.4(22.6%)	0.1(0.0%)
Total	12,971.9	415.2(3.2%)	68.8(0.5%)

B. Single-packet multicast flows

In this section we turn our attention from packet to flow-level analysis. We first discuss our method of filtering spurious multicast flows to isolate representative application flows. Using this sanitized flow data we analyze flow volume and length, duplicate flows attributable to the underlying ATM link layer, group address space and source address space. The group space discussion includes an analysis and visualization of multicast address utilization. We also discuss properties of source addresses observed in our data, particularly in the context of single-source versus multiple-source multicast protocols.

B.1 Flow filtering

We use our flow-based methodology to analyze flow volume and length, group distribution and protocol distribution. However, as seen in Section V, a large proportion of flows consist of a single packet. Despite our intuitive assumption that long-duration conversations are more common in a multicast context than for traditional transaction-based applications such as web browsing or file transfer, 75.7% of the multicast flows observed carried only a single packet, i.e., about 3.0M of the 3.9M flows collected, with 21,072 unique destinations (group addresses). Our measurement monitor did not capture sufficient payload to determine definitely the source of one-packet flows, so inferences are necessary. In this section we discuss previous studies with sources of one-packet flows and whether our data is consistent with such sources.

Almeroth's MBone study [7] observed a significant number of one-packet flows: for more than 10% of the addresses they collected, only a single RTCP packet [8] was seen. RTCP is the control protocol for the Real Time Protocol (RTP), a protocol for data with real-time characteristics. RTP provides payload identification, sequencing and time stamping. In conjunction with RTP, RTCP monitors receiver quality and periodically sends this quality information to all participants. According to our uni-directional flow definition, the RTP and RTCP packets, flowing in opposite directions, will be treated as separate flows. A user who tries to join an MBone group but finds no content and leaves quickly will induce a single RTCP packet to be sent back to the source. RTP uses an even-numbered UDP port while RTCP uses an odd-numbered UDP port. 3.3M (85.1%) of the one-packet flows observed on all our monitored links were UDP, with 3.1M (93.0%) of those using an even UDP port number and the remainder using an odd UDP port number. Because RTCP does not appear to significantly account for the predominance of single-packet flows, we considered other possible causes.

Data from Almeroth's study identified approximately 20 group addresses used by multicast routing protocols, group management and debugging tools, which we collectively call protocol control traffic. However, we find insignificant percentages of the one-packet flows attributable to such protocol control traffic (in the 224.0.0.0/24 range) or other debugging tools such as *mtrace*. Indeed, the top 20 one-packet flow destinations, relative to total flow count, include many groups used by well-known applications. 50% of the one-packet flows are packets destined to the Access Grid Lobby (224.2.177.155) [22], the Beacon Server (233.2.171.1) [23], or the SAP group (224.2.127.254). Thus the majority of the one-packet flows are not representative of typical multicast applications but rather seem to be test and directory traffic.

Therefore to understand multicast application traffic behavior, we filtered out all one-packet flows and protocol control traffic including link-local (224.0.0.0/24), *mtrace* (224.0.1.32), and Cisco RP (224.0.1.39, 224.0.1.40) flows. For CHI, the filtered data represents 508,322 flows totaling 2.996T bytes (5.910G packets). Thus removing all one-packet flows leaves 99.9% of the original byte and packet counts but only 14% of the flows. Most significant is the effect that flow filtering has on the number of unique sources and destinations as shown for all monitoring points in Table III. The number of unique sources was reduced by a factor of 12 and the number of unique destinations was reduced by a factor of 18 for the CHI flow data. As expected given the connectionless nature of multicast, the filtered traffic consisted of almost all UDP. UDP accounted for 99.999% and 99.998% of the packets and bytes respectively. Many other transport protocols are represented; the top five after UDP are DCN-MEAS, ICMP, LEAF-1, VMTP and CHAOS (IP protocol numbers 19, 1, 25, 81 and 16). However none of these protocols represented more than 1/1000th of a percent of the total traffic.

TABLE III
EFFECT OF SINGLE-PACKET FLOW FILTERING

Monitor	Data Flows	Unique Sources	Unique Dests.
CHI Raw	3,410,769	37,028	20,714
CHI Sanitized	508,322	2,886	1,133
HOU Raw	126	14	11
HOU Sanitized	37	6	6
WAS Raw	366,270	488	52
WAS Sanitized	333,451	435	36
NYC Raw	148,588	962	295
NYC Sanitized	111,299	301	48
Total Raw	3,925,753	38,492	21,072
Total Sanitized	953,109	3,628	1,223

B.2 Flow volume

To provide multicast capability, currently adopted protocols, such as PIM[18], specify how routers should build distribution trees to optimize packet replication. Typically, the multicast source is the root of the distribution tree with islands of receivers connected via branches of the tree. As a result, each router in a multicast network must maintain forwarding state on a per-group, per-source basis. Unlike unicast, multicast forwarding state is dynamic and depends on the presence of sources and receivers. Each monitored flow represents a unique multicast source and group. Since routing protocol and tree construction events can be triggered by user data transmission, understanding flow rates in the backbone is important.

Figure 8 shows the number of active multicast flows at CHI as a function of time and corresponds directly to the amount of forwarding state theoretically required in the backbone router. To generate this plot we divided the month-long collection period into five-minute intervals. We used the flow timestamps to determine if the flow was active for any part of a given interval. For a recent snapshot on July 24, 2001, the vBNS CHI router had 932 multicast state entries, 289 of which were in a forwarding state, while the remainder were pruned. For comparison, an early 1999 snapshot from a core multicast router in another study [7] had forwarding state for 199 multicast group addresses.

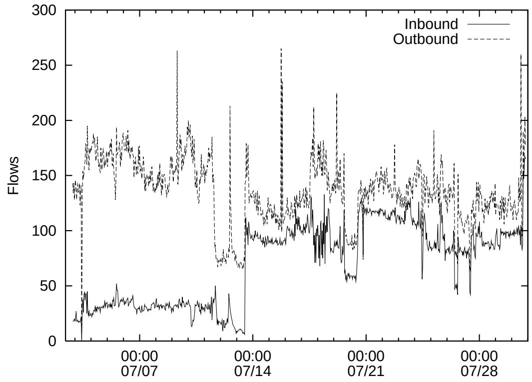


Fig. 8. Multicast flows over time, CHI link, (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

Such a small number of forwarding entries is easily manageable, but suggests a relatively small multicast user base. The number of active flows remains relatively constant, but unlike the multicast rate plots, the flow data shows no discernible time-of-day pattern.

As audiences grow and more of the Internet becomes multicast enabled, flow state in the network will grow proportionately. Emerging applications that use hundreds or thousands of group addresses in order to, for instance, provide granular receiver rate control, would cause an order of magnitude increase in the current forwarding state. Large fluctuations in the number of flows would trigger high rates of PIM register, join and prune messages as well as inter-AS (autonomous system) protocol messages. The processor cycles and memory space required on routers to process these messages has destabilized portions of the current multicast Internet in the past.

B.3 Flow length

Flow length provides insight into multicast session length and, for continuous sources, receiver interest. Figure 9(a) shows the inverse cumulative distribution of multicast flow durations, measured in seconds, for flows from all four sites. Figure 9(b) shows the same data, grouped into nine bins of different durations. Figure 10(a) plots the flow packet volume distribution, i.e., the number of packets seen for each flow. The flow packet volume is skewed toward low packet counts. Figure 10(b) shows the multicast flow byte volume distribution centered around 1000 to 10,000 bytes. Even after filtering out single-packet flows, the majority of remaining flows are relatively short-lived despite our intuitive notion.

The large number of short flows has definite implications on protocol implementations. For example, vendors typically configure their routers to build shortest path tree (SPT) state to the source immediately once the directly connected router knows the source's IP address. Our findings suggest the potential value in an alternative mechanism (packet count or rate based) before performing the shared tree to SPT switch over. The ability to further control the cutover on a per group basis would provide even greater flexibility.

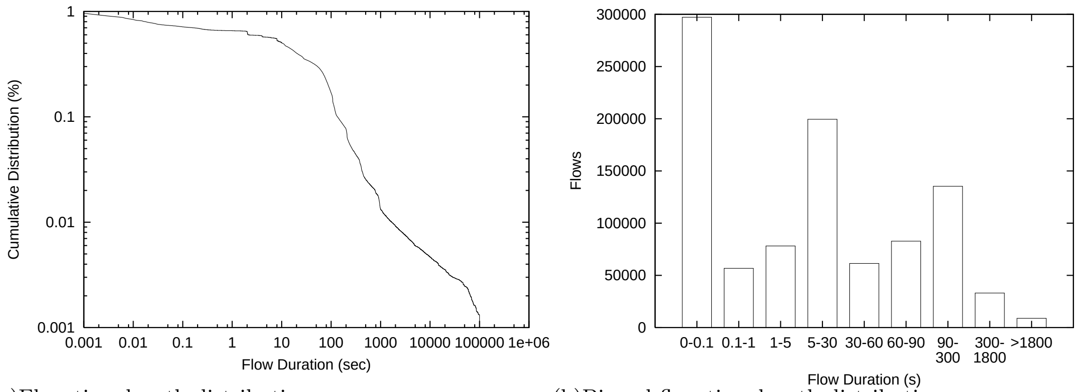
In addition, the large number of small flows has implications on the Multicast Source Discovery Protocol (MSDP) [24]. MSDP is used between rendezvous points, typically to establish inter-domain multicast state between autonomous systems. Each new unique source and group causes an MSDP 'source active' (SA) message to be flooded to all peers. This behavior has caused inadvertent denial of service attacks and network failure on routers running MSDP when malicious or non-malicious programs scan the entire 32-bit IP address range rapidly. Based on our findings, an MSDP mechanism that floods SA messages only for flows of a certain duration or rate is highly desirable and offers greater infrastructure protection than currently available.

B.4 Duplicate flows

Our data revealed unexpected results including duplicate packets and flows in two different directions with the the same source. Fenner notes several factors that contribute to duplicate multicast packets, including Ethernet bridges that re-forward packets, PIM timeouts and forwarding loops [25]. Further analysis found that, in addition to these reasons, the ATM substrate was responsible for many of the anomalous flows as we describe next.

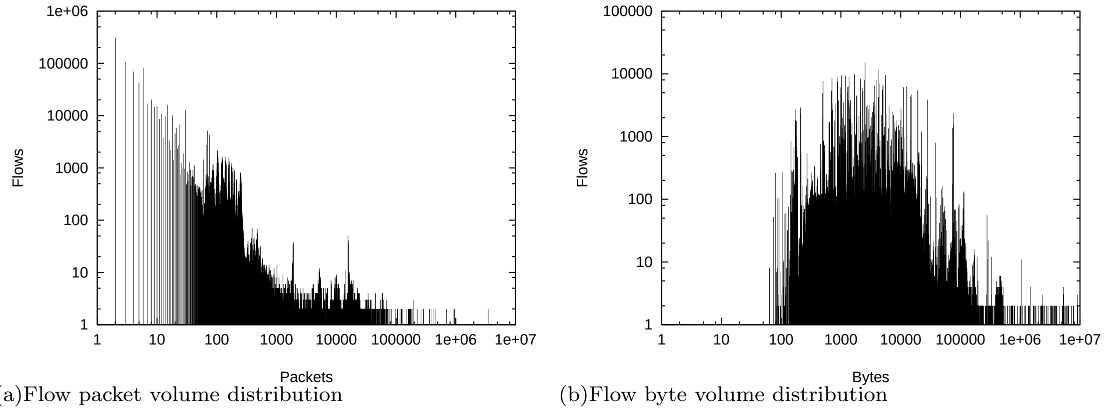
Using ATM as a data link layer exposes potential inefficiencies due to its decoupling from the IP layer. Multiple customers' traffic may arrive at the router on the same physical ATM interface, but on different logical interfaces. Each logical interface has a unique ATM permanent virtual circuit (PVC) and is treated as a separate interface for the purpose of building multicast distribution trees. Several anomalies may occur as a result. A multicast flow may be observed in both directions of the ATM link when the upstream and downstream interfaces are on the same physical interface. If there are multiple downstream logical interfaces on the same physical interface, the multicast flow byte and packet counts will be a fanout-based multiple of the number of bytes and packets sent to the group.

To analyze this phenomenon, we found identical flows that were seen in both directions of the same physical link and then determined the downstream fanout. In 3.4M total multicast flows seen on the CHI link, 19,913 (0.58%) of the flows had a downstream receiver on the same ATM physical interface as the upstream. 55,953 flows had multiple downstream neighbors on the same ATM physical interface



(a)Flow time length distribution (b)Binned flow time length distribution

Fig. 9. Flow time length distributions, all sites (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)



(a)Flow packet volume distribution (b)Flow byte volume distribution

Fig. 10. Flow packet and byte volume distributions, all sites (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

as the upstream. This accounts for 800.4K extra packets (0.014% of the total packet count) and 297.6M extra bytes (0.010% of the total byte count) due to the ATM overlay.

Unfortunately, this analysis method cannot detect the case where the upstream interface is the wide area OC48c circuit facing another backbone router but the downstream fanout includes two or more logical ATM interfaces on the same physical interface. To analyze this situation, we wrote a specialized flow monitoring application that detected packets seen belonging to the same flow with duplicate IP identification fields (ID) in a 10-second interval. Within a one-hour monitoring interval, the monitor saw 0.5M of 9.9M packets (4.7%) and 148M of 6,986M bytes (2.1%) with duplicate IP IDs. 3,550 duplicate packets had IP ID zero, attributable to machines with IP stacks that force IP ID to zero and set the DF bit to zero. 14,482 flows had packets with duplicate IP IDs.

The most efficient remedy for this unnecessary packet duplication is to use PIM to open ATM multicast virtual circuits. In this manner the replication occurs on the ATM switches at the optimal point where the physical circuits diverge. However on our network the number of duplicates does not justify the additional complexity of employing multicast ATM virtual circuits.

B.5 Group space

The class D IP address space, denoted by setting the high-order four bits of the IP address to `1110`, is reserved for multicast use. Unlike unicast, multicast addresses correspond to neither a particular user nor a particular location. The majority of the class D address space of 2^{28} addresses is as yet unallocated. The few multicast addresses that have been allocated, either individually or in blocks, are assigned by IANA (Internet Assigned Numbers Authority) to a particular purpose or application rather than to an entity [26]. One assigned block, 239/8, is dedicated to private multicast traffic. Traffic to this group should remain bounded within an organization's AS[27]. The vBNS allocates address space from this private block to its own customers with private multicast applications, for example video content that customers do not wish to broadcast to the global Internet. IANA allocates another 2^{24} addresses to the GLOP [28] and EGLOP[4] [29] 233/8 range. GLOP provides a block of 256 globally unique multicast addresses per AS by permuting the AS number into part of the group address. Using GLOP, each AS is guaranteed a unique block of Internet multicast addresses for its own use on the public Internet.

Some multicast applications consult with a central directory to lease a group address that has some assurance of being unique. However the success of this mechanism is limited to the extent which other applications use the same directory [30]. An example of this technique is *sdr* [31], the MBone session directory tool. The dynamic, per-session nature of group addresses and lack of consistent addressing policy underscores the difficulty in determining multicast application usage. Whereas many unicast applications may be identified by their TCP or UDP port numbers, such mapping is not possible with multicast. A handful of multicast applications use arbitrarily assigned, non-IANA static addresses. For example, we identify Access Grid Lobby (224.2.177.155) [22] and Norton Ghost (229.55.150.208) by their group addresses. While these applications can be identified by group address, the majority of multicast traffic cannot be classified into applications, neither via address nor port number. Further work is needed to classify multicast applications by other deductive methods, perhaps by identifying them on the basis of packet inter-arrival times or other behavior characteristics.

We observed a total of 21,072 unique multicast groups over our one-month measurement interval. The accumulation rate of group destinations was steady over time as shown in Figure 11. The initial spike in the graph is due to the large number of groups already present when monitoring began.

To determine coarse-level group distribution, we examined the proportion of data from all four monitoring points that contributed to each of the 16 blocks of 2^{24} multicast addresses. Figure 12 shows the relative flow, packet and byte percentages for each /8 multicast address block that contributed at least 1/10th of a percent to the group address distribution. The 224/8 group range accounts for 84% of the flows, but only 38% and 25% of the packets and bytes. Because of long-lived customer video applications assigned to private multicast address space, we see that the 239/8 group accounts for 67% of the total bytes and 43% of the total packets, but only 0.002% of the total flows. While we saw traffic present for each of the 16 blocks of multicast addresses, the most popular blocks were the public 224/8, the GLOP 233/8 group and the administratively scoped private 239/8.

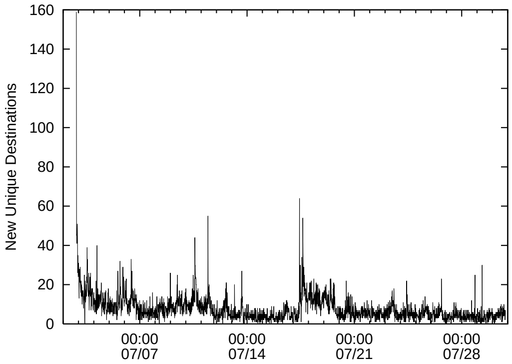


Fig. 11. Accumulation of unique IP destination addresses on CHI link (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

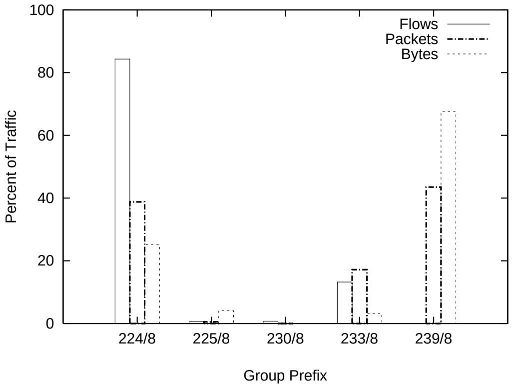


Fig. 12. Multicast group distributions, all locations (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

To measure the observed multicast address space utilization in detail, we used a visualization method similar to Braun's [32], later used by McCreary [33]. Figure 13 is a visualization of the observed utilization of the 224/8 block of 2^{24} total addresses. To generate a meaningful image, we divided the block into 2^{16} clusters by ignoring the least significant octet. The y-axis represents the second most significant octet and the x-axis reflects the third most significant octet in the IP address. The black dots depict segments of the address block that appear in the destination address field of collected flows.

Address space use in the 224/8 block is concentrated at the top of the image, indicating smaller second octets. This characteristic is attributable to user and developer psycho-

[4]GLOP is not an acronym but simply a humorous name invented by the RFC's authors.

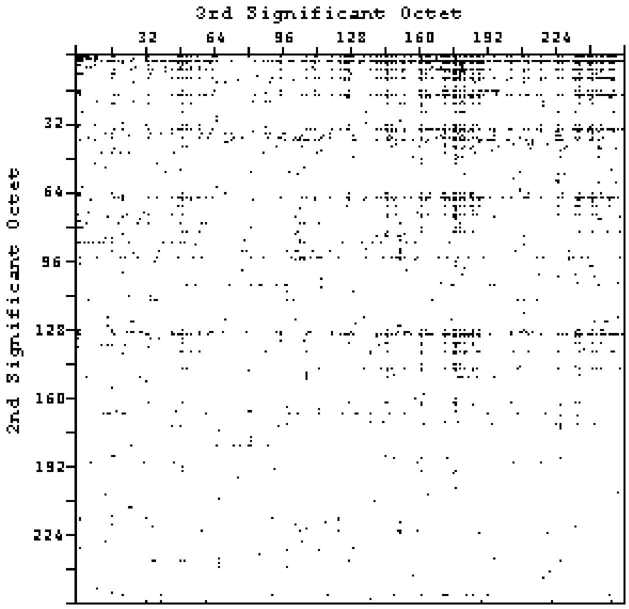


Fig. 13. 224/8 address space utilization, all locations (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

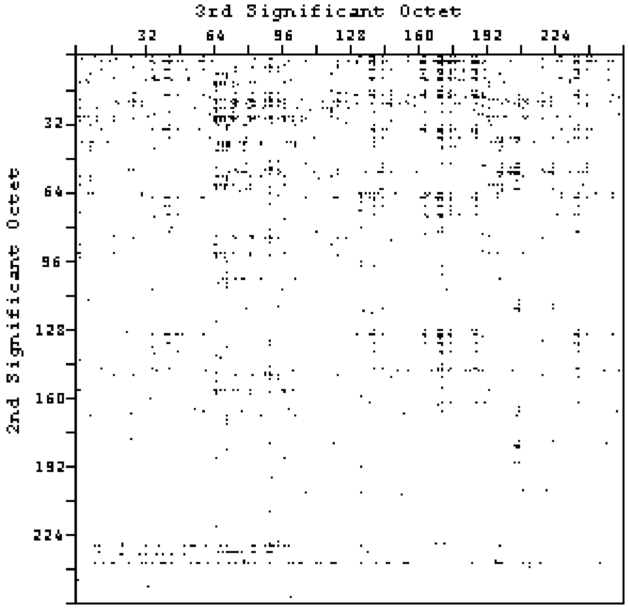


Fig. 14. 233/8 Address space utilization, all locations (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

logical predisposition to choose addresses at the beginning of the available block. As expected we see heavy use in the SAP addresses block, 224.2/16, the range used by dynamic sessions. We also see that while IANA reserves 224.3.0.64 through 224.251.255.255 [26], there are multicast groups in use throughout the 224/8 address space. A pattern emerges around the third octet range of 160 to 192 suggesting a popular third octet address for many different second octets. Figure 14 shows a similar plot for the 233/8 address space. We note that the 233 GLOP space should correspond directly to the AS numbers in use in the Internet. Because portions of the AS space are reserved, the space below a second octet of 128 in the graph should be blank. However, again we see much of the address space being used indiscriminately without formal permission.

As multicast use grows, the dynamic nature of group address may be problematic, particularly since many groups use technically reserved address space. While authors of the multicast address space allocation routines [26], [27], [28] can see that their work is being employed to some degree, scalability of global multicast may require more concerted, enforceable efforts in address usage.

B.6 Source space

A group collision occurs when two sources unintentionally select the same dynamic multicast group. Receivers may then receive the content of both sources when in fact they are only interested in one of the sources. Levine, *et al.* [30] examine the problems arising from group collisions and the probability of group collision as the number of active multicast groups increases. Commercial multicast customers want to ensure that their content is uniquely addressable on the Internet and will not be disrupted by other activity. The ability to deliver globally unique multicast group addresses is essential for providers to support commercial multicast customers.

We observed a total of 38,492 unique IP addresses of multicast sources over our one-month collection interval. Figure 15 shows that the accumulation rate of sources was steady over time. To quantify the potential for group collisions, we calculated the number of unique sources per multicast group seen over the entire monitoring period. Figure 16 shows the frequency of sources per group seen in CHI.

In CHI's one-month of traffic, 717 groups (63%) had a single source, 233 groups (21%) had two sources, 41 groups (4%) had three sources and 142 groups (12%) saw four or more sources. Three well known groups saw a large number of unique sources across the monitored link: 625 sources sent to the Beacon Server group (233.2.171.1); 463 sources sent to the Access Grid Lobby (224.2.177.155); and 303 sources sent to the SAPv1 group (224.2.127.254). Each of the other monitors saw similar distributions of multicast address usage.

We next sought to understand how many of these multicast groups had not only multiple sources, but multiple

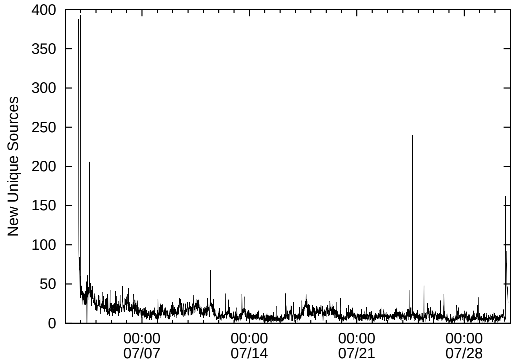


Fig. 15. Accumulation of unique IP source addresses on CHI link (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

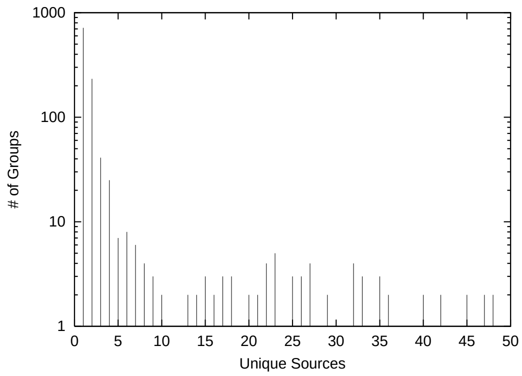


Fig. 16. Unique sources per multicast group, CHI link (20:50 EDT July 2, 2001 to 20:50 EDT July 30, 2001)

simultaneous sources. Simultaneous sources are indicative of groups with multiple active participants or possibly the result of address collisions. Address collisions typically go unnoticed as higher-layer transport protocols on the unsuspecting receiver discard unwanted traffic.

Proposals to simplify multicast protocols by assuming that the majority of multicast groups have only a single source have fueled debate over the validity of the single-source assumption. Source-Specific Multicast (SSM) [34], is one such model; SSM assumes that each multicast group has only a single source, eliminating the shared tree and rendezvous point complexity. We analyzed our multicast flow data to find flows belonging to the same multicast group, that coincided in time, but with legitimately different sources. We observed a total of 13 groups with sources overlapping at the same point in time, excluding the SAP group. The maximum number of simultaneous sources seen for a group at any given point in time was larger than expected. The Access Grid Lobby group experienced a maximum of 33 overlapping sources at one point in CHI. The Beacon Server group saw as many as 28 different overlapping sources. Nine simultaneous sources appeared in NYC for the 'Places all over the world' and three for 'Places all over the world (128k)' MBone sessions. Other non-SSM groups in addition to the Access Grid included: 224.0.1.85, 224.2.142.155, 224.2.153.71, 224.2.177.153, 225.1.2.3, 233.145.0.39 and 233.145.0.40.

Thus, over the month-long monitoring period, we find only 1.1% of the groups observed ever had multiple, simultaneous sources. Considering that the overwhelming majority of groups are single-source, and that one of the factors impeding wide spread adoption of multicast is the complexity incurred supporting multiple sources, we recommend SSM as the de facto standard.

Specifically, we are referring to the complexity required of the network when receivers have no prior knowledge of active sources for the group that they join. The network compensates for the suboptimal performance of the requisite rendezvous point and shared tree by maintaining a dual-plane routing architecture supporting also shortest path trees.

The RP mechanism introduces a number of complications and performance problems. First, an RP must be configured in each multicast domain and all multicast routers must know the address of the RP either through static configuration or some other dynamic protocol. Secondly, RPs must support decapsulation of IP-in-IP packets and all other routers in the domain must perform encapsulation on ingress multicast packets so that they can be forwarded via unicast to the RP. Encapsulation and decapsulation is inefficient and requires router resources that limit performance. Thirdly, to eliminate the single point of failure an RP poses, providers typically configure multiple RPs and use anycast addressing for redundancy. Next, supporting inter-domain multicast requires yet another protocol (MSDP) to coordinate source advertisements among RPs in different domains. Finally, as the network switches from a shared-tree to shortest-path tree, packet loss or duplication invariably occurs.

Debugging an any-source multicast network is substantially more difficult than an SSM network. A typical multicast problem requires understanding and debugging encapsulation and decapsulation, unicast and multicast routing, shared and shortest path state, anycast addressing and RPs. Inter-domain problems require debugging all of the aforementioned components in coordinated manner among providers, a daunting proposition.

SSM eliminates the complexity and simplifies the debugging of multicast networks. The vision of an all SSM network still faces several obstacles though, including operating system and network device IGMPv3 support and application-level support for multiple group multicast where needed. We believe the advantages of SSM, particularly with respect to advancing deployment efforts, far outweigh the obstacles to SSM adoption.

VII. Conclusions

In this paper we presented results of a month-long multicast traffic measurement study. Using passive OC12MON traffic monitors, we first collected multicast-specific IP packets from strategically selected nodes in the vBNS network. We found that a flow timeout between 90 and 256 seconds offers a balance between minimizing stale state and new state creation. From this flow profiling methodology we collected packet and flow data from four nodes on a major backbone giving visibility into many aggregated commercial customers and peer multicast networks.

We found that multicast traffic is time-of-day and day-of-week dependent, and exhibits a constant baseline rate. Contrary to prevailing wisdom, we saw highly variable packet size distributions, often with large packets and strong modes. Table IV enumerates the packet-based results of this study. Only 0.5% of the multicast traffic was fragmented while 3.2% of the traffic was marked 'don't fragment'. In contrast to common assumption, we found that the majority of flows, 76%, are short-lived and do not contribute significantly to multicast byte and packet volumes. Based on the predominance of short flows, we recommended serious consideration of changes to network protocols that initiate control events based on multicast application traffic, i.e. existing PIM and MSDP implementations. Our analysis of duplicate packets attributable to building multicast trees over incongruous link and network layers, such as ATM, found little gain from the complexity of ATM multipoint virtual circuits. Using a method to visualize multicast group address utilization, we saw that as a result of the dynamic nature of multicast addresses and loose address delegation, even IANA reserved blocks of multicast address space were indiscriminately used. Finally, an analysis of unique sources within a group revealed only a small number of groups with different simultaneous sources, but often with many different sources participating in those few groups. Table V lists the key flow-based results of this study.

This study focused on characterizing multicast traffic, however a number of areas warrant further investigation. Profiling multicast application traffic remains a difficult task given the dynamic nature of group addresses and the typically random transport layer port selection. One inferential approach to investigate is the relationship between applications and packet inter arrival times. A second interesting area is using the monitor's capabilities to measure unicast packets containing multicast control traffic to provide insight into the relationship between control and multicast traffic. Finally, a viable method for supporting multiple source groups in an SSM-only network is an outstanding hurdle to widespread SSM deployment.

Acknowledgments

The authors would like to acknowledge the many suggestions of Jambi Ganbar, Aaron Striegel, Greg Miller, Colleen Shannon, Xiyan Shi and Kevin Thompson. In addition, we thank Howie Xu for his work on the Apptel Linux driver and David Moore of CAIDA for the IP hash code.

References

[1] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, pages 78–88, Jan./Feb. 2000.

[2] J. Jamison, R. Nicklas, G. Miller, K. Thompson, R. Wilder, L. Cunningham, and C. Song. vBNS: not your father's internet. *IEEE Spectrum*, 35(7), July 1998.

[3] R. Beverly, G. Miller, and K. Thompson. Multicast performance measurement on a high-performance IP backbone. *Computer Communcations*, 24(5-6):461–472, 2001.

[4] P. Rajvaidya, K. Almeroth, and K. Claffy. A scalable architecture for monitoring and visualizing multicast statistics. In *DSOM*, pages 1–12, 2000.

[5] B. Mah. Measurements and observations of IP multicast traffic. Technical Report UCB/CSD-93735, University of California at Berkeley, March 1993.

[6] M. Handley, C. Perkins, and E. Whelan. Session announcement protocol. RFC 2974, IETF, October 2000.

[7] K. Almeroth. A long-term analysis of growth and usage patterns in the multicast backbone. In *Proceedings of IEEE INFOCOM*, March 2000.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. RFC 1889, IETF, January 1996.

[9] K. Sarac and K. Almeroth. Supporting multicast deployment efforts: A survey of tools for multicast monitoring. *Journal of High Speed Networking*, March 2001.

[10] K. Thompson, G. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Transactions on Networking*, pages 10–23, November 1997.

[11] S. McCreary and K. Claffy. Trends in wide area IP traffic patterns: A view from Ames Internet Exchange. In *ITC Specialist Seminar*, September 2000.

[12] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder. OC3MON: Flexible, affordable, high-performance statistics collection. In *Proceedings of INET*, June 1997.

[13] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and K. Claffy. The architecture of CoralReef: an Internet traffic monitoring software suite. In *Proceedings of PAM*. CAIDA, April 2001.

[14] Applied telecom. http://www.apptel.com/point.htm.

[15] S. McCanne and V. Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In *Proceedings of USENIX Technical Conference*, 1993.

[16] Spirent communications. http://www.spirentcom.com.

[17] J. Mogul. Observing TCP dynamics in real networks. In *ACM SIGCOMM Symposium on Communications Architectures and Protocols*, pages 305–317, 1992.

[18] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast-sparse mode. RFC 2362, IETF, June 1998.

[19] K. Claffy, H. W. Braun, and G. Polyzos. A parameterizable methodology for Internet traffic flow profiling. *IEEE Journal of Selected Areas in Communications*, 13(8):1481–1494, 1995.

[20] Amnis systems. http://www.optivision.com.

[21] C. Shannon, D. Moore, and K. Claffy. Beyond folklore: Observations on fragmented traffic. *To appear in IEEE/ACM Transactions on Networking*, 2003.

[22] R. Stevens. Plans for creating access grid technology for the national machine room, 1999. http://venues.accessgrid.org/AG/venues.php.

[23] Multicast beacon server. http://dast.nlanr.net/Projects/Beacon.

[24] D. Meyer and B. Fenner. Multicast source discovery protocol (MSDP). Work in progress internet-draft, IETF, September 2001.

[25] B. Fenner. Duplicate multicast packets. http://www.aciri.org/fenner/mcast/dups.html.

[26] Z. Albanna, K. Almeroth, D. Meyer, and M. Schipper. IANA guidelines for IPv4 multicast address assignments. RFC 3171, IETF, August 2001.

[27] D. Meyer. Administratively scoped IP multicast. RFC 2365, IETF, July 1998.

[28] D. Meyer and P. Lothberg. GLOP addressing in 233/8. RFC 2770, IETF, February 2000.

[29] D. Meyer. Extended assignments in 233/8. RFC 3138, IETF, June 2001.

[30] B. Levine, J. Crowcroft, C. Diot, J. J. Garcia-Luna-Aceves, and J. Kurose. Consideration of receiver interest for IP multicast delivery. In *Proceedings of IEEE INFOCOM*, pages 470–479, 2000.

[31] M. Handley. Sdr: Session directory tool, 1995.

[32] H. W. Braun. BGP-system usage of 32 bit Internet address, November 1997. http://moat.nlanr.net/IPaddrocc/.

[33] S. McCreary. IPv4 address space utilization, August 1998. http://www.caida.org/outreach/resources/learn/ipv4space/.

[34] SSM IETF Working Group. An overview of source-specific multicast (SSM). Work in progress internet-draft, IETF, November 2002.

TABLE IV
Summary of Packet-based Results

1. Multicast traffic is time-of-day and day-of-week dependent with intra-day traffic increasing by as much as 300%.
2. The packet size distributions were highly variable, often with many large packets and strong modes.
3. Only 0.5% of the multicast traffic was fragmented while 3.2% of the traffic was marked ‘don’t fragment‘.

TABLE V
Summary of Flow-based Results

1. A flow timeout value between 90 and 256 seconds offers a balance between minimizing stale state and new state creation.
2. The majority, 76%, of multicast flows are very short-lived, and do not contribute significantly to multicast byte and packet volumes.
3. Existing protocols that initiate control events based on multicast application traffic, MSDP and PIM, need to employ rate or packet based thresholds.
4. The quantity of duplicate packets attributable to building multicast trees over ATM does not justify the complexity of employing multicast ATM virtual circuits.
5. IANA reserved blocks of multicast address space are indiscriminately used.
6. Analysis of Source-Specific Multicast reveals only a small number of groups with different simultaneous sources, but often with many different sources participating