Multicast Redux: A First Look at Enterprise Multicast Traffic

Elliott Karpilovsky
Princeton University
elliottk@cs.princeton.edu

Lee Breslau
AT&T Labs – Research
breslau@research.att.com

Alexandre Gerber
AT&T Labs – Research
gerber@research.att.com

Subhabrata Sen
AT&T Labs – Research
sen@research.att.com

ABSTRACT

IP multicast, after spending much of the last 20 years as the subject of research papers, protocol design efforts and limited experimental usage, is finally seeing significant deployment in production networks. The efficiency afforded by one-to-many network layer distribution is well-suited to such emerging applications as IPTV, file distribution, conferencing, and the dissemination of financial trading information. However, it is important to understand the behavior of these applications in order to see if network protocols are appropriately supporting them. In this paper we undertake a study of enterprise multicast traffic as observed from the vantage point of a large VPN service provider. We query multicast usage information from provider edge routers for our analysis. To our knowledge, this is the first study of production multicast traffic. Our purpose is both to understand the characteristics of the traffic (in terms of flow duration, throughput, and receiver dynamics) and to gain insight as to whether the current mechanisms support multicast VPNs can be improved. Our analysis reveals several classes of multicast traffic for which changes to the underlying protocols may yield benefits.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network monitoring*

General Terms

Measurement

Keywords

Multicast; VPN; Enterprise Networks

WREN'09, August 21, 2009, Barcelona, Spain.

1. INTRODUCTION

IP multicast [1, 2], which provides an efficient mechanism for the delivery of the same data from a single source to multiple receivers, was first proposed more than two decades ago. It was deployed experimentally on the MBone [3] in the early 1990s and was the subject of a significant amount of research into scalable and robust intra- and inter-domain routing protocols [4, 5, 6]. The MBone, an overlay used primarily to support audio and video conferencing between multicast-capable networks around the Internet, grew rapidly. However, the initial enthusiasm for multicast did not translate into widespread success. The MBone eventually declined, and more importantly, multicast was not adopted in any significant way by service providers.

The failure of multicast to achieve widespread adoption can be attributed to several technical and economic factors. When sources and receivers were in different domains, it was unclear how to appropriately charge for the service. Furthermore, the *de facto* multicast protocol, PIM [4], presented challenges of its own for inter-domain multicast, as it could create inter-provider dependencies. In particular, users in one domain could rely on infrastructure in other domains to carry strictly intra-domain traffic. Moreover, multicast protocols were new, quite complex, and difficult to manage. Finally, the lack of popular multicast applications translated to little interest for deployment.

After languishing for many years on the Internet, multicast is experiencing a resurgence in two main contexts. First, within service provider networks, it is being used to support IPTV applications which benefit from efficient one-to-many distribution. Second, multicast is being deployed in enterprise networks, where the aforementioned issues of management, billing and inter-provider dependencies, are mitigated. In enterprise networks, multicast is used to support a variety of applications, including file distribution, conferencing, and dissemination of financial trading information. Moreover, many enterprises connect their geographically disparate sites via provider-based VPNs. Hence, multicast is becoming increasingly available to VPN customers.

Since the use of multicast in production networks is a relatively recent phenomenon, little is known about its traffic characteristics. However, gaining such knowledge of traffic characteristics is important, as it can help researchers understand whether protocols in the network are adequately supporting the applications. Additionally, such knowledge is needed for proper network planning and provisioning. More-

over, today's multicast VPN services are new and complex (as described in Section 2), with many configuration options based on assumptions regarding usage, engineering guidelines, rules of thumb, and controlled testing. Behavior "in the wild" is poorly understood, yet is critical to uncovering possible issues and improvements regarding design, operation, capacity planning, provisioning, resource usage, and performance. A first step to resolving these issues requires an understanding of traffic in the multicast VPN service.

Studying enterprise traffic from the vantage point of a service provider presents both challenges and benefits. The provider does not have the same visibility into the enterprise traffic as would be possible within the enterprise network itself. This derives from the fact that the enterprise traffic is encapsulated as it crosses the provider network, thereby masking some relevant information. However, we believe that this challenge is far outweighed by the benefits of scale and diversity that a provider domain study affords. Specifically, rather than being limited to studying traffic within a single enterprise, we are able to study traffic in many different enterprise networks. Thus, our results are more general than they would be otherwise.

As such, we undertake a study of enterprise multicast traffic as observed from the vantage point of a tier 1 ISP that provides multicast service to some of its VPN customers. Our purpose is both to understand the characteristics of the traffic and how the VPN service can more efficiently deliver multicast traffic to its customers. By understanding how applications use multicast, we can better understand how to design our networks. To our knowledge, this is the first broad study of production multicast traffic. We collect usage data from provider edge routers that describe the activity of their associated multicast groups. We analyze multiple traffic characteristics, including flow duration, throughput, peak rates, and receiver dynamics. These statistics compactly describe every multicast session and provide information about the benefits (or lack thereof) that the provider network receives from multicast. We also apply clustering techniques to classify flows according to their behavior, allowing us to see if dominant usage patterns emerge.

The rest of the paper proceeds as follows. Section 2 discusses protocol specifics. Section 3 discusses how the data is collected and challenges of inference. Section 4 shows our results. We compare our work with related research in Section 5, and conclude in Section 6.

2. MULTICAST VPN OVERVIEW

In this section we provide an overview of how multicast and Multicast VPNs (MVPNs) are supported by the ISP. A description of the specification upon which the MVPN service is based can be found in [7]. While MVPN service may evolve in the future, the description provided here represents current industry practice.

2.1 IP Multicast

With traditional unicast, every connection has exactly two endpoints; multicast, on the other hand, replaces point-to-point delivery with the idea of *groups*. Multicast groups use specially designated IP addresses, known as multicast addresses, taken from the 224/4 address block [8]. Hosts can join or leave groups (on behalf of applications) at will. Packets sent to the group address are forwarded to all members.

The requirement to deliver the exact same data to multiple end hosts provides the opportunity to reduce network resource consumption. Multicast protocols create a distribution tree on the network topology for every group [1]. This tree reaches all group members. When packets are transmitted to the group, a single copy is sent along the initial branch from the source. When the tree branches, the router at the branch point replicates packets and sends separate copies along each branch. Figure 1 demonstrates this capability. As a result, the amount of bandwidth needed on shared path segments can be greatly reduced.

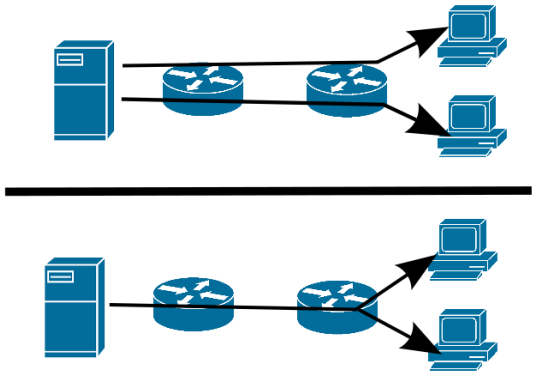

Figure 1: On top: a server sending data to clients using unicast. On bottom: efficient distribution with multicast.

In general, every multicast tree requires a routing table entry on every router along the tree. Thus, the total amount of routing state can potentially be proportional to the number of multicast groups.

2.2 Multicast VPNs

In a VPN, the customer attaches to the provider network at one or more locations. At each such provider location, one or more Customer Edge (CE) routers attaches to a Provider Edge (PE) router, as depicted in Figure 2. The PE receives packets from attached CEs and transports them across the backbone to other PEs, which then deliver them to attached CEs. In the case of Multicast VPN, a customer multicast packet entering at an ingress PE, may be destined to receivers at multiple customer locations. Thus, the provider will be required to deliver the packet to one *or more* egress PEs. The customer multicast packet is encapsulated using GRE [9] and transported across the backbone to the PE routers using IP multicast. That is, the customer multicast packet is encapsulated in a provider multicast packet between the ingress and egress PEs.

Within the provider backbone, every VPN is assigned a unique multicast group, called a *Default MDT (Multicast Distribution Tree)*. When a customer attaches itself to a set of PEs, those PEs join the associated multicast group. The Default MDT acts as a broadcast channel among the PEs and serves two purposes. First, customer domain multicast control messages are transmitted over the Default MDT. Second, customer multicast packets (*i.e.*, application traffic) are initially transmitted over the Default MDT. These

[1]For simplicity we omit many details of multicast routing; some trees may be specific to an individual sender while others may be used by all senders to a group.

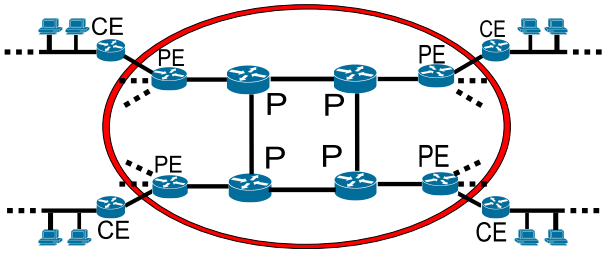

Figure 2: Setup of the MVPN network. Circle encloses the provider network. Customer edge (CE) routers attach to provider edge (routers). The core provider (P) routers do not interact with customer routers directly, but transit traffic between PEs.

multicast packets are delivered to all PEs to which the customers attach. However, some of these PEs may not have group members downstream of their attached CEs. Packets that reach such PEs are dropped.

Since packets may be dropped at some PEs, this Default MDT mechanism has the potential to waste bandwidth in the provider network. To mitigate this problem, a second kind of multicast group, referred to as a *Data MDT*, is used in the backbone. When an ingress PE detects that the sending rate of a sender to a group exceeds a configured threshold (based on duration and throughput), that sender's traffic to the group is moved from the customer's Default MDT to a customer-specific Data MDT in the provider backbone. A PE will only join this announced Data MDT if there are receivers for the corresponding customer group downstream of its attached CEs. The Data MDT mechanism does not waste bandwidth; packets are only delivered to PEs that have downstream receivers. However, Data MDTs consume additional resources (in the form of routing table entries) in the provider network.

3. METHODOLOGY

In this study, we collect and analyze data about enterprise VPN multicast traffic seen on a large, tier-1 ISP. Along the way, we ran across several analysis issues. We disclose our challenges and solutions here.

3.1 Data Collection

Between January 4, 2009 and February 8, 2009, we monitored multicast traffic using a specially designed, lightweight poller. The poller contacts the PE routers that support enterprise VPN traffic using SNMP [10]. We mine customer provisioning information to configure the poller to understand the mapping between multicast groups and associated PEs [2].

We refer to PEs that inject traffic into the backbone as senders, and PEs that receive traffic from the backbone as receivers. *Note that in our analysis, "receivers" always refers to egress PEs.* These definitions arise from the fact that we study multicast traffic over a backbone network, with PEs on the boundary between the backbone and the customers.

[2]One multicast group is assigned per Default MDT per VPN, and one multicast group is assigned per Data MDT per VPN. However, a VPN may have multiple Data MDTs.

```
Timestamp:    1237114394
Group:        239.255.255.128
Source:       58.122.125.122
Receiver:     58.122.16.89
Counter:      16938168
Status:       Okay
```

Figure 3: Format for the multicast entries received from the poller. The entries are synthesized for illustration.

Each PE tracks the amount of data sent to and received from each Default and Data MDT using a byte counter. Note that multicast groups are uniquely identified by an (S, G) tuple, where S is the IP address of the sender and G is the multicast group address. Due to the way in which multicast is configured in the provider backbone, data received on Default MDTs at egress PEs is reported on a per-group basis, rather than for each (S, G) pair. Data sent by ingress PEs is reported on a per-source basis. Given this setup, the data we collect is a series of records in the form of Figure 3.

We poll all the PEs at five minute intervals. While we would ideally like to poll at a smaller interval, we were limited by the operational constraints of a large backbone network. Routers were already supporting SNMP polling to support a wide variety of management functions, and we needed to limit the additional load placed on them. As a side note, not all successive polls were spaced exactly five minutes apart, due to variability in the polling process. For the time frame of our study, actual polling intervals ranged from 2.4 to 13.2 minutes. However, 99.7% of the poll intervals were between 4.8 minutes and 5.2 minutes. Thus, except for a few outliers, the polling is very consistent. Overall, we collected and analyzed approximately 88 million records.

The post-analysis of the polled results needs to handle two issues. As with any polling-based approach, a poll may not return the requested data object. This can occur for a variety of reasons. For example, SNMP requests or responses (and subsequent retransmissions) could be lost or the router could take too long to respond. In such cases, the request is considered to have "timed out" and values are not reported. Also, if configuration information is outdated, the poller may request information for a multicast group that is no longer a part of a particular PE router. However, in practice these problems occur infrequently. Approximately 99.3% of all polls were successful; 0.4% failed due to time outs and 0.3% failed due to outdated configuration information.

Second, in some cases we observed discrepancies (beyond what one would expect due to packet loss) between the amount of data reported by senders and receivers for a (S, G) pair. As a sanity check, we examined such discrepancies for a small sample of the Default and Data MDT data. For the Default MDT, senders and receivers usually agreed on the amount sent, although there was noticeable dissent as well. We analyzed these dissenting cases and found several possible explanations. In some of the cases, we identified the data as obviously incorrect (*e.g.*, sending rates faster than the line speed) which we believe were due to bugs in the way the data was reported. We also found that our polling method did not always accurately report data when a multicast group spans domains (as is the case for VPNs in the provider network that have presence on multiple continents). There were also

cases in which we could not successfully identify the problem. When we could identify the cause of problems they often were associated with the amount of data reported received at egress PEs for Default MDTs. Thus, the analysis we can perform on the Default MDT is limited to aggregate analysis based on sending data at ingress PEs. However, the Data MDT had a higher rate of agreement. As such, we have general confidence in the accuracy of the data.

3.2 Default MDT Analysis

Our analysis for the Default MDT is further limited by several factors:

- Every PE router in a VPN is always attached to the Default MDT. Thus, it is impossible for us to study the dynamics of receivers that leave or join.
- Every PE router sends constant keep-alive messages (*i.e.*, PIM Hello messages) to the Default MDT. These messages are recorded by the PE routers as incoming data. Thus, every PE is always sending data, and it is difficult to tell when a true multicast flow starts or stops.
- Moderate to high bandwidth flows (which are of greater interest) are usually transmitted on the Data MDT.

The first problem is unavoidable, given our setup. The second problem can be solved using a threshold value, stating that flows that generate less than a certain amount of data per interval have ended. Since the keep-alive messages are fairly consistent in size and frequency, it is possible to model the background load generated by them. However, determining the threshold accurately is difficult, as variability in the messages raise the problem of accidentally filtering low bandwidth flows. Finally, since the Default MDT typically only carries low bandwidth flows, the total impact of it on the multicast network is limited. As such, we perform limited analysis on the Default MDT (avoiding any per-flow results) and focus most of our attention on the Data MDT.

3.3 Data MDT Analysis

For the Data MDT, we track flows (*a.k.a.* sessions) by observing the amount of data transmitted by the source for a given multicast group. A flow is defined by the Data MDT records associated with it, indicating its start and end times, as well as the amount of data sent and receiver dynamics. We calculate the amount of data sent by taking the difference between byte counters in successive sender records. We define *throughput* as the total amount of bytes sent averaged over the entire flow duration. We define *peak rate* as the maximum amount of bytes/second seen between any two consecutive polling intervals. In addition, we keep track of the number of receiver PEs joined to a particular group and record when a PE router joins or leaves the group. We also cluster flows to determine if dominant behavior patterns emerge. We employ a variant of the k-means algorithm [11] that applies a "simulated annealing"-style approach to the problem, along with a local search heuristic. We take into account the following characteristics: duration, throughput, peak rate, maximum number of receivers, and average number of receivers. Because such clustering algorithms usually assume equal variance for each characteristic, we normalize all characteristics to z-scores (*i.e.*, mean of 0 and standard deviation of 1) before clustering.

Although this analysis may appear straightforward, there are many corner cases. In particular, there are issues determining when a particular flow starts or stops, and determining the amount of data sent during a session. Some of these issues revolve around a condition we refer to as a *counter reset*, when a subsequent polling record either has a byte value of 0 or has a lower value than the previous record. These occur often in the data and may signify that a new flow has begun. However, interpreting what they actually mean in a particular situation is quite difficult. This problem is further complicated by the fact that some of the routers use a 32-bit byte counter which may overflow. Thus, when we see a counter value that is less than the previous one, did the session actually reset? Did the session end in-between a polling interval, with a new session starting up and taking its place? Did the counter overflow? Because of the difficulty in analyzing these cases, we adopted a simple rule to classify when the flows start and stop in this situation: *if the counter value is zero, a new flow has begun; otherwise, it is the same flow.* Note that in the case of a counter reset, instead of subtracting two intervals (to get the amount of data sent), we simply use the byte counter value of the last interval.

Another problem that arises is error messages (*e.g.*, SNMP time outs). Although they are rare, they can cause problems when analyzing when flows start and stop. If one sees a valid flow record, a sequence of error messages for the following intervals, and then another valid flow record, do we have two separate flows, or one longer flow? For the Data MDT analysis, we consider flows separated by error messages to be separate flows. Our analysis thus has a bias that reports more flows, each of shorter duration.

Another related issue is receiving a partial result from a query, *i.e.*, a router returns part of an answer but "times out" before returning all records. For simplicity, flow entries missing from the partial result are considered to be terminated. Although we did not quantify the number of such cases, we believe them to be sufficiently rare to not warrant serious investigation.

Up until this point, we have assumed that we have consecutive records to calculate quantities like duration, throughput, *etc.* However, for the Data MDT, it is possible that very short lived flows (less than approximately 10 minutes) may only have a single record associated with them. This behavior happens because of a temporary sending spike, which will boost a Default MDT multicast session into the Data MDT for a short period of time. Because we can only infer information from consecutive pairs of records, these single record sessions are difficult to analyze. As such, we label these flows as having 0 second duration with 0 kilobits/sec throughput and 0 kilobits/sec peak rate. This provides them with a unique identifier in our analysis.

4. RESULTS

4.1 Default MDT Analysis

Figure 4 displays a range of the sending rates seen in the Default MDTs for each (S, G) pair per polling interval. Not all sending rates are shown. For example, there is a large step in the CDF at approximately 0.02 kilobits/sec, equivalent to approximately 800 bytes per 5 minute interval. We highly suspect that this value is related to the minimum amount of background PIM messages sent by every router

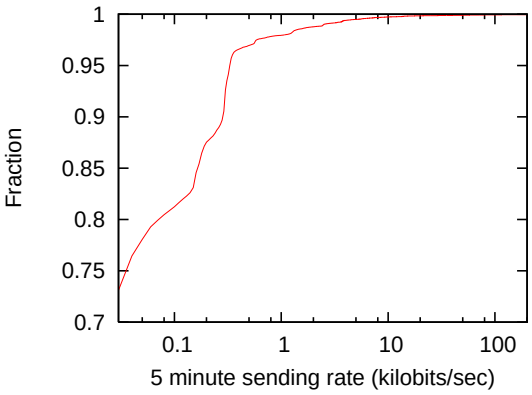

Figure 4: CDF of sending rates seen in the Default MDT between intervals.

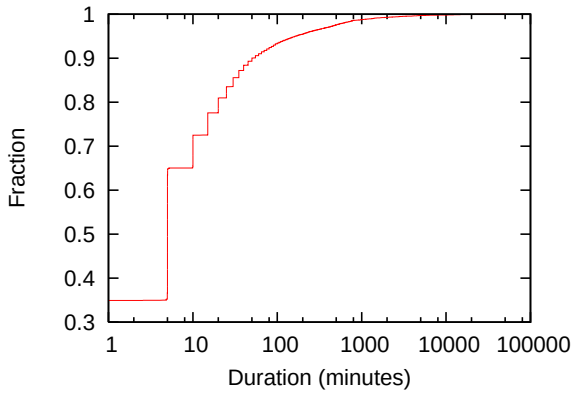

Figure 5: CDF of Data MDT session durations.

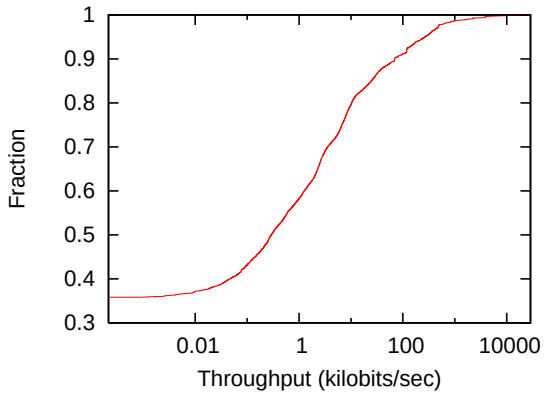

Figure 6: CDF of average throughput per Data MDT flow.

in the network. Likewise, we observed a very small number of rates (less than 0.05%) in excess of 200 kilobits/sec. After manually inspecting several cases and finding them to not be in agreement with the receiving records, we removed them for this analysis.

There are several important aspects. First, the vast majority of the Default MDT rates are quite low, with approximately 99.5% of all rates less than 5 kilobits/sec. Second, we notice a very small number of flows that consume moderate bandwidth, surpassing 100 kilobits/sec. While it may seem odd for these "large" flows to be transmitted on the Default MDT (as they should be switched to the Data MDT), there are several reasons why we may see them here. A flow changes to the Data MDT only if it maintains a certain throughput for a given duration. Thus, it is possible that a very bursty flow may never meet the duration threshold, while still sending large amounts of data. Another possibility is that these rates represent flows that were eventually switched to the Data MDT; thus, these rates may reflect the short period of time before these flows were changed (and thus had their traffic registered in the Default MDT). It is also plausible that a router may be configured to never promote flows from the Default MDT to the Data MDT. In the first two cases, by reducing the duration threshold, these flows can properly be handled by the Data MDT, at the potential cost of increased routing state churn (since more entries will transition between the Default and Data MDTs and vice-versa).

4.2 Data MDT Analysis

Figure 5 shows a CDF of flow durations. The stepwise nature arises from the fact that almost all poll intervals are five minutes. A significant portion of flows (approximately 70%) last 10 minutes or less. Ideally, such short lived flows, if sufficiently low bandwidth, would be kept entirely in the Default MDT to prevent an increase in routing state. Whether there exists a way to identify these flows and prevent them from moving to the Data MDT remains an open problem. While most flows are short-lived, a very small number lasts more than a week.

Figure 6 displays the average throughput seen in the Data MDT flows (as compared with Figure 4 for the Default MDT). Quite surprisingly, we see many flows (more than 70%) that send at less than 5 kilobits/sec. One reason for this phenomenon is that a large percentage of flows (36%) are so short-lived that they only have a single Data MDT record, and thus no accurate estimate for their throughput (and are defaulted to a value of 0 kilobits/sec). Since the Data MDT should primarily have high throughput flows, future research on multicast might investigate different mechanisms for switching flows to the Data MDT, as well as identifying situations when such transitions are appropriate.

Figure 7 shows the distribution of peak rates for the Data MDT. Peak rates are calculated by taking the maximum throughput seen across polling intervals. Peak rates are, on average, approximately 1.6 times greater than the average throughput. It is also interesting to note that for both peak rate and throughput, there are a small percentage of high bandwidth flows, indicating that multicast Data MDT trees may significantly reduce the amount of traffic sent over a network (relative to the Default MDT).

Figure 8 tracks the dynamics of receivers per session, measuring the maximum number of receivers seen, where a receiver is an egress PE in the backbone network. Surprisingly, a very large fraction of the flows (almost 50%) only have a single egress PE. In other words, using unicast to transport these flows across the backbone would use bandwidth just as efficiently (and incur less routing state overhead). These re-

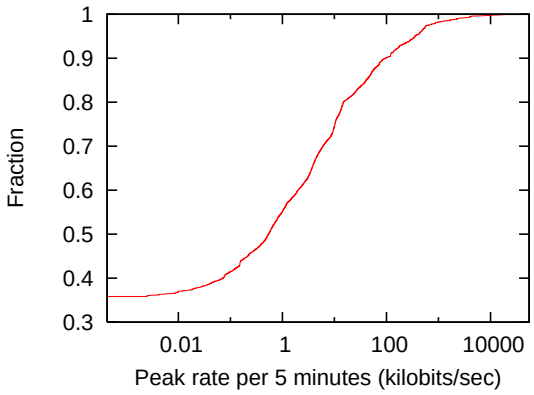

Figure 7: CDF of peak rates of Data MDT flows.

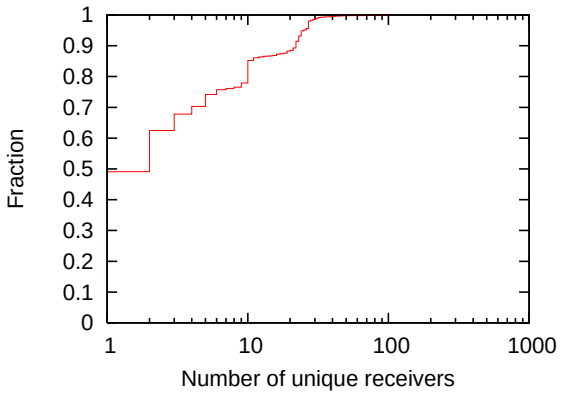

Figure 8: CDF of maximum number of receivers per Data MDT session.

sults imply that there may be an opportunity to reduce multicast state overhead; if an appropriate mechanism can be used to identify these flows in advance, a significant amount of multicast overhead can be removed by using unicast encapsulation across the backbone. As a separate note, there is a significant fraction (approximately 20%) of flows that reach at least 10 different egress PEs during their lifetime.

Finally, we perform clustering analysis on these characteristics (as well as on the average number of receivers seen per flow). In order to determine the number of clusters, we plot the unexplained variance for various numbers of clusters. Figure 9 shows the amount of variance that can be explained with different clusters, where variance is defined as the sum of the L_2 norms from each flow to its closest cluster. We choose to label our flows with 4 clusters, as $k = 4$ is at the knee of the curve; it explains a significant fraction of variance while allowing us to label flows in a manageable manner. The cluster points are described in Table 1. A short name is given to each cluster point, highlighting its dominant characteristic. It is important to keep in mind that each cluster represents an average of many flows, and there is variance within a given cluster. From these points, we see many interesting aspects.

In the first cluster, called *unicast*, we see flows that are long-lived, very high throughput, with few egress PEs (usually one). As such, these backbone flows are not truly benefiting from multicast, but are adding routing state to the network. Although there are only a few flows that fall in this cluster, there may be value in investigating mechanisms to more efficiently support these flows, particularly if the applications they represent become more common.

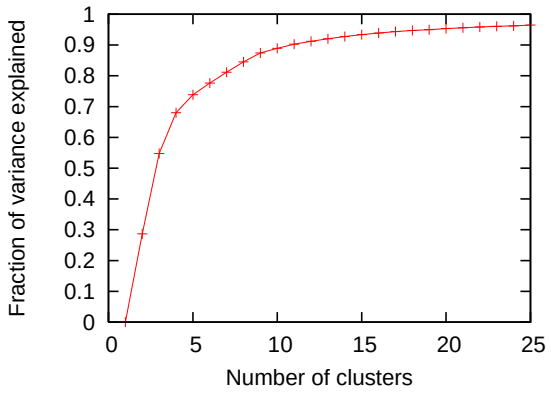

Figure 9: Total variance explained for a given number of clusters for Data MDT flows. L_2 norm is used.

The second cluster, *limited receivers*, contains most flows. It represents flows that are approximately 1 hour long, moderate bandwidth, with very few receivers. In fact, this cluster typically has a maximum of 3 receivers and an average of 2. Any improvements or optimizations that apply to these flows could be particularly beneficial to a network, given the size of the cluster.

The third cluster, *long lived*, represents flows that lasted approximately 1 month and consumed moderately high bandwidth. Moreover, the number of receivers, although small, is large enough to imply that multicast technologies are a good mechanism for handling these flows. However, the total number of flows in this category is very small. As such, the impact of these flows over the entire network is minimal.

Finally, we have a cluster, *well-fitted*, that represents moderate length, moderate bandwidth flows with a large number of receivers. The flows in this cluster benefit greatly from using multicast. They are called *well-fitted* because these types of flows are the ones that derive much benefit from multicast (given the number of receivers). Reducing the router state imposed by these flows is tricky, as they represent "typical" multicast traffic that the protocol was designed for. Although they are the second largest cluster group, they are small in number, and thus are probably not a large contribution to total resource consumption. As such, optimizing them may not be of primary importance.

4.3 VPN Analysis

Finally, we analyze how individual customers (more precisely, the VPNs assigned to them) use multicast.

Figure 10 depicts the amount of time each VPN spends with at least one flow in the Data MDT, for a time period of one week. While there are a few VPNs that hardly use the Data MDT (less than 30% of their total time), the majority spend at least half of their time in the Data MDT.

Figure 11 plots the number of customers with active Data

Short Name	Duration	Throughput	Peak Rate	Max Rcv.	Avg. Rcv.	Flows in cluster
"Unicast"	29 hours, 6 min	11.8 mbits/sec	22.5 mbits/sec	1.4	1.2	0.1%
"Limited rcv."	1 hour, 12 min	39.7 kbits/sec	56.3 kbits/sec	2.7	2.1	86.5%
"Long lived"	28 days, 2 hours, 10 min	604.9 kbits/sec	983.9 kbits/sec	9.5	3.1	0.05%
"Well-fitted"	59.3 min	20.5 kbits/sec	30.7 kbits/sec	25.4	19.6	13.3%

Table 1: The four cluster centers.

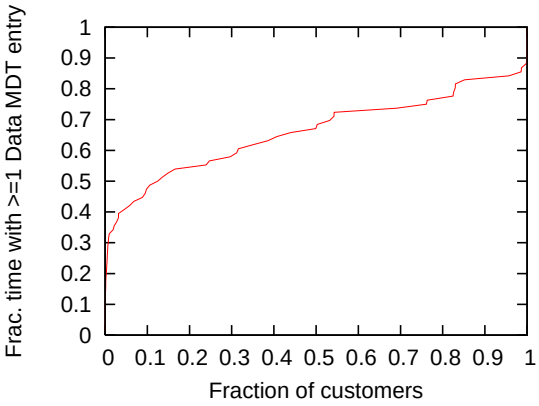

Figure 10: CDF of time spent with activity in the Data MDT, on a customer basis.

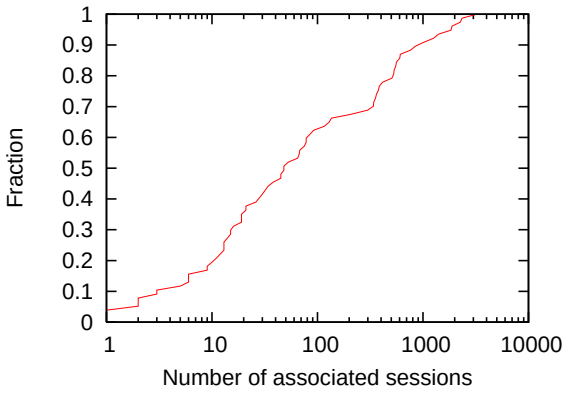

Figure 12: CDF of Data MDT sessions per customer.

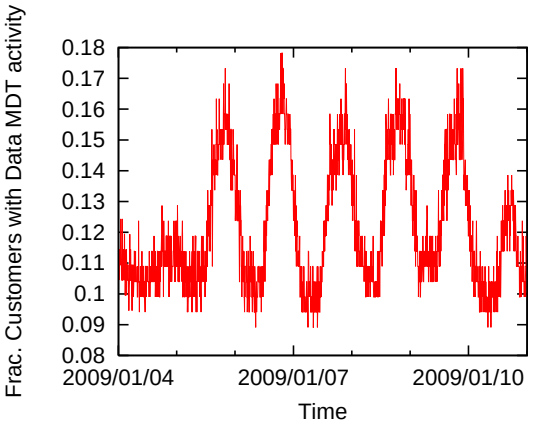

Figure 11: Time versus customers with Data MDT activity.

MDT entries over time. Very strong diurnal and weekday/weekend patterns can be seen. This corresponds roughly with what we would expect from enterprises, as they are more likely to be active during business hours. There is significant traffic at night and on weekends as well.

Figure 12 plots the CDF of the number of sessions per VPN. In this, we see that most VPNs (50%) only engage in a small number of flows over the course of one month. However, there are several heavy hitters that engage in hundreds (and up to a thousand or more) flows during this time period. Thus, although many VPNs use the Data MDT, they often use it in varying amounts.

Lastly, we investigated whether types of companies (*e.g.*, retailers, financial, *etc.*) use the Data MDT in different ways. We label the VPNs with categories and look at the clusters that they fall into. We summarize them in Table 2. Categories consist of health related industries, manufacturers, retailers, finance, tech, information services (including consulting and analysis firms), natural resources (either extraction or conservation), and other.

Some interesting trends can be seen. First, the *unicast* style flows are almost entirely confined to the manufacturers and financial companies. The *limited receivers* category was prominent across all categories. For the *long lived* flows, we see a clear dominance in the health related industries. Finally, *well-fitted* flows were varying across industries.

These results shed some light on our understanding of multicast. The fact that many industries had many flows in the *limited receivers* category suggests that many different kinds of applications have this behavior. In contrast, *unicast* and *long lived* had clear dominators, possibly indicating this behavior is specialized to certain applications. Because of the variance in the *well-fitted* category, it is difficult to draw hard conclusions about the types of traffic that fit this classification. Overall, these results suggest that optimizations to general multicast networks should focus on the *limited receivers* case, and that there is potential for optimizations to the *unicast* and *long lived* flows for specialized cases.

5. RELATED WORK

Our study is unique because, to our knowledge, there have been no other published results analyzing real VPN multicast traffic. However, there have been several areas of research related to multicast in general.

As previously mentioned, the MBone was a multicast backbone network that was free to use. Because not all routers that interconnect networks were multicast enabled, the MBone used tunnels between multicast islands. Unfortunately, this limitation presented scaling challenges as the bandwidth ef-

Category	% of *unicast*	% of *limited receivers*	% of *long lived*	% of *well-fitted*
Health Services	0	94.3	2.9	2.9
Manufacturers	0.3	78.5	0.03	21.1
Retailers	0	94	0	6
Finance	0.4	86.1	0.02	13.5
Tech	0	99.6	0	0.4
Information Services	0	75.1	0.04	24.8
Natural Resources	0	100	0	0
Other	0.3	98.5	0	1.2
Average	0.1	86.5	0.05	13.3

Table 2: A breakdown of how flows from each cluster center fall into different enterprise categories. Average for all flows across all groups is given at bottom.

ficiency of multicast can be reduced when tunneling, rather than native multicast is used. While there have been studies done on the MBone [12, 13], as well as general IP multicast [14], our work differs from these since we evaluate VPN customers within a single ISP, as opposed to inter-ISP multicast traffic.

To better aid operators, a tool known as VMScope [15] was created to help with network management. It can remotely monitor multicast VPNs and determine characteristics such as packet loss and latency. Deployed at a single location in a network, it provides a congenial interface for operators seeking high-level information about multicast sessions. We consider this work tangential to ours, as we are primarily concerned with longer-term characteristics such as flow duration and throughput.

Some research has continued on multicast protocol improvement. For example, Chainsaw [16] is a peer-to-peer overlay multicast system that does not rely on trees for message propagation. There has been information theoretic work on multicast in coded packet networks, where outgoing packets are generated from incoming packets [17]. Additional theoretical work has been done to shown that reencoding packets in the middle of a network can result in a large increase to the maximum sending rate [18]. However, this research focuses on theoretical improvements to multicast. Our work supplements this research by providing real usage information to guide future work.

Finally, there has been a large amount of research concerning multicast support for IPTV. For example, the channel surfing problem has been given considerable study, where user behavior is expected to change during commercial breaks [19, 20]. Moreover, measurement studies have been done for IPTV multicast [21]. Because we study a different domain and are not focused on a single application, we consider this work to be complementary to ours.

6. CONCLUSION

Due to the growth of VPN multicast traffic, it is extremely important to understand how organizations are using multicast. Without this information, it is impossible to know how this service will continue to grow and change over the years. Additionally, such information can help us optimize multicast protocols to consume fewer resources.

Our results from a tier 1 ISP show several interesting aspects of multicast traffic. We see a wide distribution of flow duration, although most flows tend to be short-lived. Likewise, many flows use low or moderate amounts of bandwidth, with a small number of flows with very high throughput. Moreover, we see potential opportunities to make the Multicast VPN service more efficient. The large number of single egress PE flows in the provider domain indicates that unicast could be used as a replacement, resulting in no impact to efficiency but considerably less routing state. Likewise, a significant number of flows only communicate with a handful of receivers; converting these to unicast streams would decrease bandwidth efficiency, but greatly cutback on memory requirements.

There is future work to consider. First, it would be interesting to do a longer, longitudinal study on multicast traffic to understand the evolution of enterprise customer behavior. Second, this research does not extensively evaluate memory and bandwidth trade-offs present in today's networks. Further analysis to identify mechanisms for optimizing multicast memory usage multicast should be considered. Finally, a lower level application layer diagnosis of multicast traffic could provide insights into how particular applications are leveraging multicast technologies.

7. REFERENCES

[1] S. E. Deering and D. R. Cheriton, "Host Groups: A Multicast Extension to the Internet Protocol." `http://tools.ietf.org/html/rfc966`, December 1985. Request for Comments: 966.

[2] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Transactions on Computer Systems*, vol. 8, no. 2, 1990.

[3] B. O'Sullivan, "The Internet Multicast Backbone." `http://ntrg.cs.tcd.ie/undergrad/4ba2/multicast/bryan/index.html`. Accessed March 2009.

[4] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "An architecture for wide-area multicast routing," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, 1994.

[5] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT)," *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, 1993.

[6] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocols." `http://tools.ietf.org/html/rfc1075`, November 1988. Request for Comments: 1075.

[7] E. Rosen and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs." `http://www.ietf.org/internet-drafts/draft-ietf-l3vpn-2547bis-mcast-08.txt`, March 2009. Network Working Group Internet Draft.

[8] Z. Albanna, K. Almeroth, D. Meyer, and M. Schipper, "IANA Guidelines for IPv4 Multicast Address Assignments." `http://tools.ietf.org/html/rfc3171`, August 2001. Request for Comments: 3171.

[9] D. Farinacci, T. Li, S. Hanks, D. Meyere, and P. Traina, "Generic Routing Encapsulation (GRE)." `http://tools.ietf.org/html/rfc2784`, March 2000. Request for Comments: 2784.

[10] D. Harrington, R. Preshun, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks." `http://tools.ietf.org/html/rfc3411`, December 2002. Request for Comments: 3411.

[11] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 2002.

[12] K. Almeroth and M. Ammar, "Multicast group behavior in the Internet's multicast backbone (MBone)," *Communications Magazine, IEEE*, 1997.

[13] K. Almeroth, "A long-term analysis of growth and usage patterns in the Multicast Backbone (MBone)," *IEEE Infocom*, 2000.

[14] B. Mah, "Measurements and Observations from IP Multicast Traffic," tech. rep., UC Berkeley, 1994. CSD-94-858.

[15] L. Breslau, C. Chase, N. Duffield, B. Fenner, Y. Mao, and S. Sen, "VMScope: a virtual multicast VPN performance monitor," in *INM '06: Proceedings of the 2006 SIGCOMM workshop on Internet network management*, 2006.

[16] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and E. E. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *in Peer-to-Peer Systems IV, Volume 3640*, 2005.

[17] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Transactions on Information Theory*, vol. 52, June 2006.

[18] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, June 2005.

[19] C. Cho, I. Han, Y. Jun, and H. Lee, "Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method," *The 6th International Conference on Advanced Communication Technology*, 2004.

[20] D. Smith, "IP TV Bandwidth Demand: Multicast and Channel Surfing," in *IEEE International Conference on Computer Comunications*, 2007.

[21] K. Imran and M. Mellia and M. Meo, "Measurements of Multicast Television over IP," in *IEEE Workshop on Local and Metropolitan Area Networks*, 2007.