An Empirical Study of Real Audio Traffic

Art Mena and John Heidemann

University of Southern California
Information Sciences Institute
Marina Del Rey, CA 90292

artmena@catarina.usc.edu johnh@isi.edu

To appear, IEEE Infocom, Tel Aviv, Israel, March 2000

Abstract

The delivery of multimedia content is a facet of Internet traffic that is rapidly growing in importance. The new generation of World Wide Web sites are relying heavily on extensive multimedia content such as graphics, sound, music and video to attract and retain visitors. While there have been extensive studies on the growth and effects of Hyper Text Transfer Protocol (HTTP) traffic used on the Web, little or no work has been performed in analyzing streaming multimedia traffic. We present the results of a brief study to examine the traffic emanating from a popular Internet audio service using the RealAudio program. We found protocol distributions that show a bias towards non-TCP friendly protocols. In addition, we observed consistencies in audio traffic packet sizes and data rate patterns may be useful as a tool for identifying audio data flows. Our results show that audio flows exhibit significant consistency in data rates and are considerably more persistent than HTTP connections.

***Index terms*—multimedia, streaming audio, internet traffic**

1. Introduction

The rise in popularity of the Internet, and of especially the World Wide Web, has resulted in the web browser becoming the most commonly used Internet application. This popularity has driven an increase in multimedia applications with streaming audio players becoming almost as ubiquitous. Today many sites offer sound and music while numerous sites specializing in providing streaming audio from jukebox libraries as well as feeds from radio stations and live concert events. This audio traffic represents a new class of Internet traffic, which will only increase with the introduction of various new technologies such as MP3, cable modems, and DSL.

While there have been numerous studies of Internet traffic in general [1, 12] and especially web traffic [2] there has been relatively little effort in examining the prominence or effects of streaming media traffic in the Internet. This paper begins to redress this omission by studying RealAudio traffic originating from a major Internet audio source.

Our study makes two contributions to understanding and classifying audio traffic. First, we find that this traffic is much different from current Internet traffic:

- Although audio data can be sent by UDP, TCP, or HTTP, For Real Audio the majority of data (60-80%) is sent by UDP and thus has limited congestion control (Section 4.1).
- Real Audio data is sent at consistent bitrates at medium time-scales (tens of seconds), but at smaller time-scales (single seconds) it is best modeled as a bursty on/off source with off periods in multiples of 1.8 seconds (Section 4.2).
- Real Audio sessions employ one or two flows and utilize multiple protocols. Those that employ two flows (70-80%) use a UDP flow for data and a TCP flow for control. Those using one flow used TCP alone (Section 4.3).
- Like web traffic, user arrivals are strongly correlated to time of day or start of events (Section 4.3). Unlike web traffic, audio flows are very long with a mean duration of 78 minutes (Sections 4.4).

Second, we suggest two ways audio traffic may be easily identified:

- Audio data is highly unidirectional with the bulk of the traffic outbound from the server. We show audio user outbound:inbound byte ratios as high as 50:1 in Section 4.1.
- We found that UDP Real Audio flows may be identified by consistent packet lengths and interdeparture regularity, section 4.2.

Based on these observations, we describe how to simulate current audio users, and we suggest ways to identify long-lived audio traffic.

The results in this paper are preliminary in two senses. First, although we consider traces over medium time scales (5-18 hours), our traces are based primarily on RealAudio servers providing radio-station like streaming data. Data from the playback of individual songs or from conferencing or telephony applications would have much different characteristics. Second, we have only begun to analyze the data, and we are currently able to present only preliminary statistics characterizing traffic behavior. More detailed studies are required.

2. Background

Streaming audio content is delivered from an audio server process to a client application program. The source of the content can be the digital output from an audio codec or a digitized file on the server. These correspond to listening to a radio program or playing a CD with the client program providing much of the expected functionality such as play, stop, change channels, fast forward, rewind, etc.

The client application decodes the digital data and plays it through the clients host's sound system. To compensate for network congestion and jitter, several seconds of audio data is often buffered at the client. Audio encodings are selected to correspond to bit rates available for dial-in users, typically at 16 or 20 kb/s, however higher bit rates are becoming more prevalent.

Streaming audio protocols have been available for many years (early work dates to the mid-1970's [3] and continues until today [4, 5] are examples of recent work), but only recently with widespread commercial use of the Internet and protocols such as RealAudio has this traffic become significant on some links. Music portals such as Broadcast.com provide hundreds of audio channels and have enjoyed phenomenal usage and growth. However there has been little work done to examine audio traffic in detail. As the number of Internet users continues to increase and as high-speed access methods, such as DSL and cable modems, become more ubiquitous, we can expect to see more streaming multimedia traffic on the Internet. Audio traffic has the potential of occupying a sizable fraction of the Internet's bandwidth.

3. Methodology

Several traces of audio data were captured from a popular Internet audio service at Broadcast.com [6]. Trace 1 and 2 were obtained using a Network Associates Sniffer [7] while traces 3 through 5 used tcpdump [8] running on a separate host. The trace host, as well as the Sniffer, was connected to the Switched Port Analyzer (SPAN) port of a Cisco 2924 Fast Ethernet Switch. The SPAN port mirrors the traffic from any port on the switch, which allows us to capture all of the traffic originating from or destined to the audio server. The traces were obtained from five different audio servers at the main Broadcast.com site. They are summarized in Table 1.

Trace	1	2	3	4	5
Date	Mar 99	Mar 99	Jun 99	Jun 99	Jun 99
Start time, GMT	N/A	N/A	16:02	13:32	13:38
Duration	83 sec	141 sec	5.5 hr	10.5 hr	18.2 hr
Packets	134 K	284 K	5.5 M	1.6 M	5.9M
Bytes	38 M	63 M	1.3 G	0.4 G	1.3 G

Table 1. Summary of Traffic Traces

The servers analyzed used the RealServer™ V5.0, from RealNetworks, to provide audio streams. For these servers Broadcast.com typically utilizes Intel Pentium II class hardware running Windows NT or Linux, assigning one system per outgoing audio stream. Traces 1 and 2 were taken from servers that stream live music from radio stations. Traces 3, 4 and 5 were taken from servers that stream events such as radio talk shows and sporting events. We will see that these content differences have an effect on our analysis results.

RealServer delivers digitized audio data with a proprietary protocol utilizing IP Multicast, UDP and TCP, as well as TCP wrapped in HTTP. The client uses the RealPlayer[1] application program to listen to the music. RealServer utilizes a fixed range of port numbers to allow firewalls to pass audio data to its client program. [9]

For each packet captured during a trace 96 bytes were saved. This was done to ensure that we had sufficient information to identify the audio data header. All of the server traffic during the test interval was captured, no packets were dropped by the trace system or the server.

3.1. Important Terms

Internet audio traffic is initiated by a user launching an audio client program on a workstation, contacting an audio server and playing an audio selection. For this study we define a *user* a simply a destination IP address. This IP number may be that of a RealAudio proxy server with the actual user residing behind a firewall. Hence this occasionally makes multiple users actually appear as one (we quantify this effect in Section 4.3). Likewise we define the *server* as the source IP address. Since our traces were obtained directly at the server, the source IP address is known a-priori to be that of a single host.

When a user plays an audio selection a *session* is established which we define as a source/destination IP address pairing. A session typically contains one or more concurrent *flows*, which are two-way exchanges of data on source/destination, IP address/port number pairs.

A session may contain several concurrent flows. The first is a TCP *control flow* which is used to initiate a session, to authenticate the user (as required) and to send control messages to the server (start, stop, pause, etc.). The second is the *audio data flow*, which contains the encoded audio information. The data flow is easily identifiable by the fact that its bandwidth is several orders of magnitude larger than the control flow. We discuss how we distinguish audio flows in Section 4.2.

Finally, since our traces were obtained at the audio server we define *inbound* to be network traffic that is received by the server and *outbound* as traffic that is sent by the server.

4. Trace Results

Our analysis focused on three areas. First we examined the aggregate traffic both inbound and outbound, which includes some web traffic, audio data, and control data. For much of our analysis we excluded extraneous traffic (i.e. ARP, DNS, etc.) which constituted less than 0.1% of the total packets captured. Next we looked at aggregate outbound audio data flows which represent audio content delivered to the user (the bulk of the data). Finally we look at several of the hundreds of individual audio flows extracted from our traces.

4.1. Aggregate Traffic Analysis

This section describes the analysis of the aggregate traffic. The latter includes audio, web, as well as any extraneous traffic and covers inbound and outbound directions. We focus our discussion on trace 3 as it is long enough to capture a range of user behavior. We include traces 4 and 5 in most graphs as supporting evidence. Traces 1 and 2 are too brief to capture long-term trends, but because they represent different content (music rather than talk radio), we present them when they illustrate different behavior. Gross metrics of traces 3, 4 and 5 are shown in the following table.

[1] The program is also known as the RealAudio player but the name was changed when streaming video capability was added. This study only examined audio data, which we refer to as RealAudio traffic to differentiate it from video data.

	Trace 3	Trace 4	Trace 5
Inbound Metrics	1.8M pkts 50 MB	0.9M pkts 216 MB	2.9M pkts 425 MB
Outbound Metrics	3.6M pkts 1,202 MB	0.6M pkts 198 MB	2.9M pkts 866 MB

Table 2 Distribution of Traffic

Table 2 shows the amount of inbound and outbound packets and bytes that were captured during these traces. Of special note is the difference between inbound and outbound traffic, a ratio that ranges from 1:24 to about 1:1. This variation between inbound and outbound traffic is as might be expected of a busy audio server where much of the traffic is outbound. We see this graphically illustrated in Figures 1 and 2, which show the measured bandwidth.

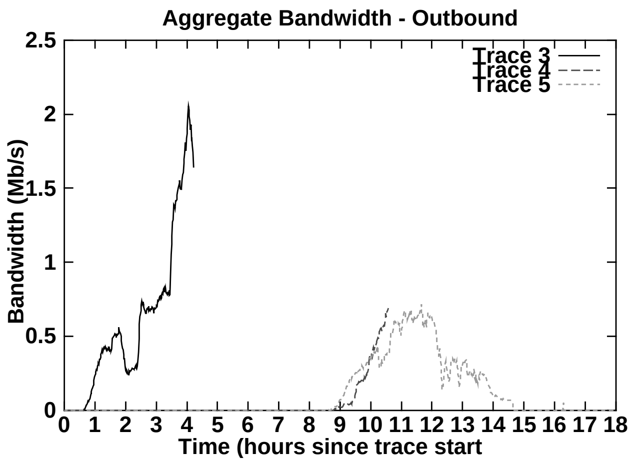

Figure 1. Aggregate Bandwidth, Outbound

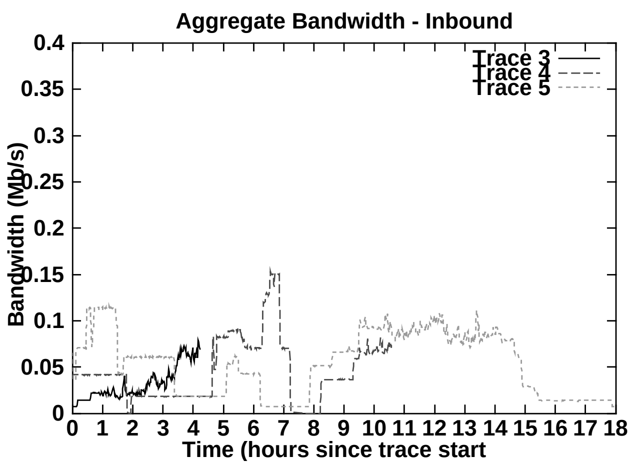

Figure 2. Aggregate Bandwidth, Inbound

The figures show that the traffic is almost entirely outbound with a small fraction inbound. Included in the inbound traffic are flows originating from audio codecs local to the audio server. These flows are rebroadcast to customers. Since these inbound flows are entirely local, and would not traverse the Internet, they are omitted in our subsequent analysis.

Inbound traffic from audio users primarily consists of packet acknowledgements and feedback data from the client program. We found outbound:inbound byte ratios of approximately 28:1, 40:1 and 50:1 for the three traces.

Of interest is that the flows in traces 4 and 5 are inactive across several hours of the measurement period. We believe this effect is caused by time-dependent user requests. Trace 5 illustrates the gradual arrival of additional users and then tapering off of demand; trace 4 captures part of this effect before terminating. All of these traces are from audio servers providing similar content. We have also observed this behavior in 5-minute samples taken over 100 hours from a group of seven similar servers. Discussions with the administrator of these servers indicate that the content being served consists of radio talk shows, which have a definitive start and end time. The pattern we observe is consistent with the event-like nature of such content.

In addition to audio data, the aggregate traffic includes control information, which is generated by a user when they manage the audio flow, as well as some web traffic. The breakdown of this traffic is seen in Table 3. As might be expected, the control flows occupy only a small portion of the aggregate bandwidth. With the bulk of the bandwidth occupied by audio data flows.

	Trace 3	Trace 4	Trace 5
Audio Data	1,160 MB 3.7M packets	403 MB 1.2M packets	1,268 MB 4.6M packets
Control Data	41.3 MB 1.7M packets	10.3 MB 0.3M packets	22.67 MB 1.2M packets
Other Packets	1.0 MB 90K packets	0.8 MB 44K packets	1.0 MB 98K packets

Table 3 Summary of traffic traces

The other data seen in Table 3 includes non-audio related web traffic, as well as some unidentified flows. Since our traces were taken at the audio server, we have eliminated external extraneous traffic. Hence these appear to be remnants of inactive flows, with most having fewer than 1000 bytes, or are local administrative connections to the audio server. We account for over 99% of all bytes and 98% of all packets as RealAudio traffic in each trace; the other data does not represent a significant amount of traffic over this network.

RealNetwork's RealServer supports a number of transport protocols including Multicast, UDP, TCP and HTTP. Protocols seen in both inbound and outbound traffic are shown in Table 4. Approximately one third of the traffic utilized TCP or HTTP as the transport protocol with HTTP traffic identified as TCP traffic originating from the audio server on port 80. A minor portion of the latter includes server administrative traffic as the RealServer includes a web-based administrative system. The remaining traffic, occupying approximately two thirds of the aggregate, utilizes UDP.

UDP traffic does not include transport-level congestion control, which implies that much audio traffic depends on application-level congestion control. Development and deployment of TCP-friendly congestion control for audio data is important for network stability [10]. Finally, while RealServer supports multicast, no multicast packets were observed in any of the traces.

	Trace 3	Trace 4	Trace 5
Bytes			
UDP	723 M (60 %)	415 M (79%)	955 M (74 %)
TCP/Non-HTTP	432 M (36 %)	68 M (17%)	304 M (24%)
TCP/ HTTP	47 M (3.9 %)	18 M (4%)	36 M (2%)
Multicast	0	0	0
Packets			
UDP	3.68 M (67 %)	1.26 M (80%)	4.52 M (77%)
TCP/Non-HTTP	1.66 M (30 %)	0.26 M (17%)	1.21 M (21%)
HTTP/TCP	0.14 M (3 %)	0.05 M (3%)	0.12 M (2%)
Multicast	0	0	0

Table 4 Aggregate Traffic

4.2. Analysis of the Aggregate Flows

After considering aggregate data we examined individual flows (as determined by source and destination ip-address/port-number pairs). We examined all flows, control as well as audio data flows. Using both the port numbers and the observed quantity of outbound data we determined that most of the flows (86-91%) were audio related. We identified outbound audio flows as those comprised of more than 100K bytes sent by the audio server. This simple heuristic is possible due to our particular test scenario but could be refined by a more detailed examination of the data packet format. Although RealServer uses specific port numbers, they alone were not always a reliable determination of the flow type, most likely due to the use of audio proxy servers.

Inbound audio flows were identified in a similar manner. The remaining flows observed in the traces were unidentifiable using port numbers and our simple audio flow heuristic. However the traffic associated with these is minimal (see Table 3). Of the flows directly attributable to RealServer most do not contain audio data. Many were control flows or were unidentifiable. The breakdown of flows identified as containing audio data is shown in Table 5.

	Trace 3	Trace 4	Trace 5
Audio Data Flows	1460	324	837
Inbound	14	20	42
Outbound	1446	304	795

Table 5 Summary of Audio flows

The audio flows in the table include several that are inbound to the server originating from one or more audio codecs. These were ignored in subsequent analysis, as this effort focused entirely on the outbound audio data flows.

The flow arrivals and duration for Traces 3 through 5 are seen in Figures 3 and 4. Here we show the data as cumulative distribution functions (CDFs) for the parameter. Figure 3 shows the distribution of the time between arriving flow connection requests, while Figure 4 shows the duration of the flows. The flow duration plot shows that half of the flows lasted longer than 45 minutes, which is significantly longer than HTTP connections. This finding does have significant implications to the overall Internet traffic as it indicates that audio streaming traffic is considerably more persistent than other traffic types such as HTTP [11] as well as telnet and FTP [12]. Unfortunately we were unable to correlate the flow duration with the length of the event broadcast by the server.

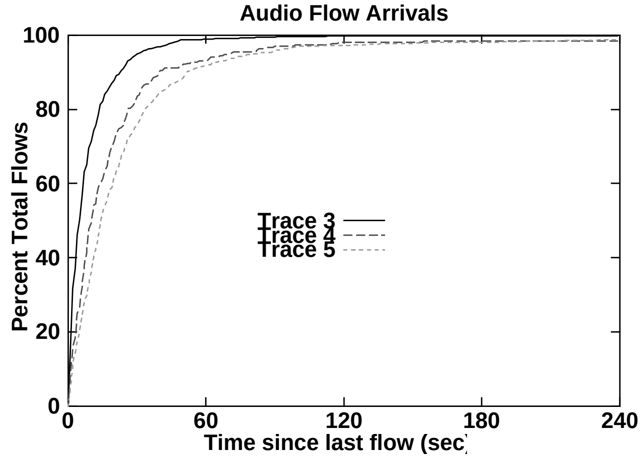

Figure 3. Audio Flow Interarrivals at the Server

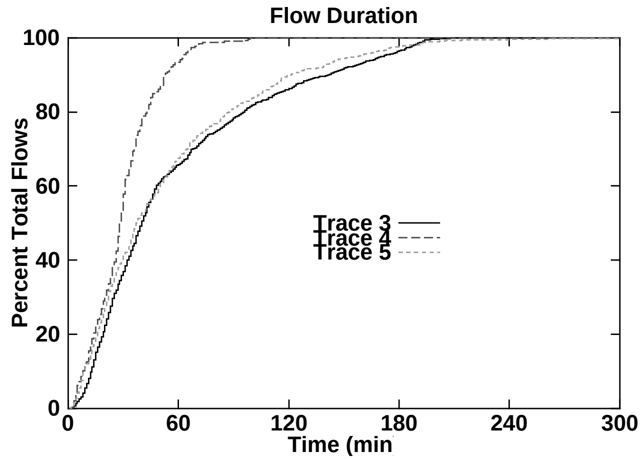

Figure 4. Flow Duration

Hence we do not know what percentage of the event the flow durations represent.

We hypothesize that the flow durations are heavy-tailed and are currently investigating that aspect. This characteristic has implications with network-level traffic policing which can affect audio flows even if the flow identification is done infrequently.

The transport protocol used by outbound audio data flows is shown in Table 6. As might be expected, the transport protocol makeup of the audio flows shows a similar distribution pattern as seen with the aggregate traffic (Table 4). The vast majority of the flows use UDP as their transport protocol. The reason for the differences seen in the table may be related to the content being delivered or the server configuration. This reliance on UDP raises congestion control concerns with audio traffic.

	Trace 3	Trace 4	Trace 5
Total audio flows	1460	324	837
UDP flows	1165 (81 %)	217 (71%)	611 (77%)
TCP flows	281 (19 %)	87 (29%)	184 (23%)

Table 6 Outbound audio data flows

Round Trip Time (RTT) can be used to help determine the distance from the server to a client. It was computed by measuring the time difference between an outbound packet sequence number and the reception of its associated acknowledgement. To avoid problems with retransmission ambiguities we evoked Karn's algorithm [13]. Because we don't yet interpret UDP feedback, we only compute RTT for the TCP audio flows. Figure 5 shows the result of this RTT and shows that the data appears heavy tailed.

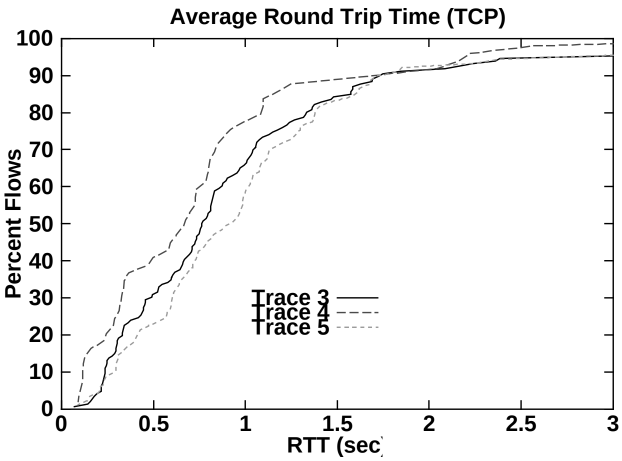

Figure 5. Audio Flow Round Trip Time (TCP)

Figure 6 shows a Cumulative Distribution Function for measured outbound audio data flow rates. Of interest are the concentrations at approximately 6.5, 16 and 20 Kbps. These correspond to "natural" RealAudio encoding rates and are shown as vertical lines on the plot.

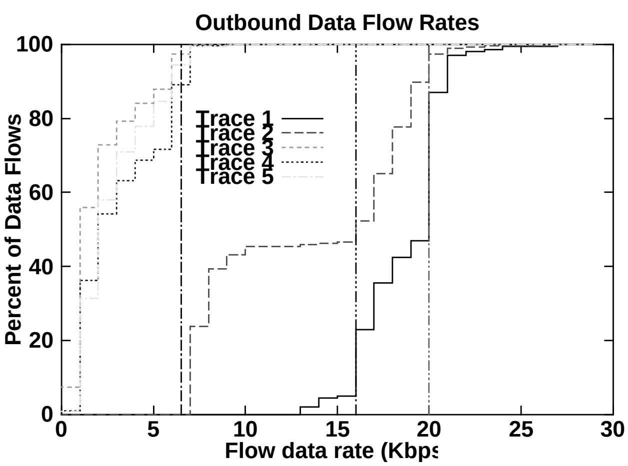

Figure 6. Audio flow data rates

The highest rate, 20 Kbps, corresponds to minimal stereo quality optimized for a 28.8 dialup modem. We show data from all 5 traces here as there are significant difference between them. For the most part, traces 1 and 2 show data rates that dominate at 16 and 20Kbps, while traces 4, 5 and 6 show rates at 1 to 5 Kbps. Since we observe very different data rates across different traces, we conclude that data rates are strongly dependent on the content being served. Traces 1 and 2 are dominated by music radio station streams, which need high bandwidth (i.e. 16 and 20 Kbps) to provide acceptable quality sound. On the other hand, traces 3, 4 and 5 are from servers that stream radio talk shows and sporting events. We hypothesize that their bandwidth requirements are less stringent than music due to their lower dynamic range and more efficient compression, hence the lower observed data rates. However we were surprised that data rates for traces 3-5 do not correspond more closely to a standard RA data rate.

In addition to sending data at specific rates, RealAudio traffic is dominated by specific packet sizes. Figures 7 and 8 show those sizes for TCP and UDP, respectively.

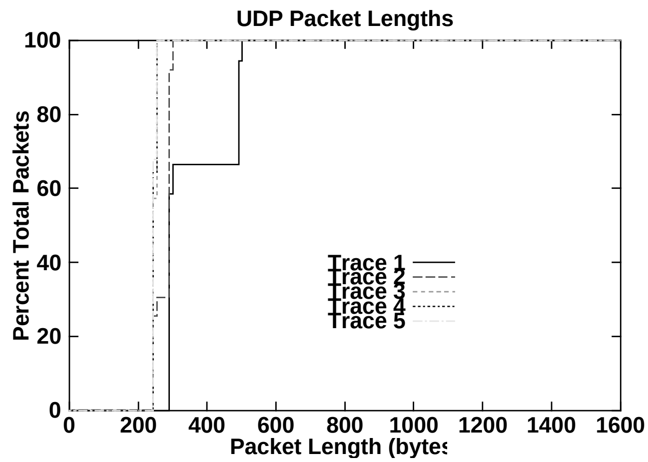

Figure 7. Data flow packet length distribution (UDP)

Again all five traces are shown in the diagrams. UDP packet lengths, seen in Figure 7, show significant regularity with concentrations at 244/254, 290/300 and 490/502 bytes, corresponding to 1 and 6.5, 16 and 20 Kbps audio flows respectively. This regularity in packet length could constitute a simple heuristic for identifying RealAudio UDP traffic in a trace file. There is additional regularity in the flow with the larger packet (254, 300 and 502 bytes) sent after every 5 of the smaller packets.

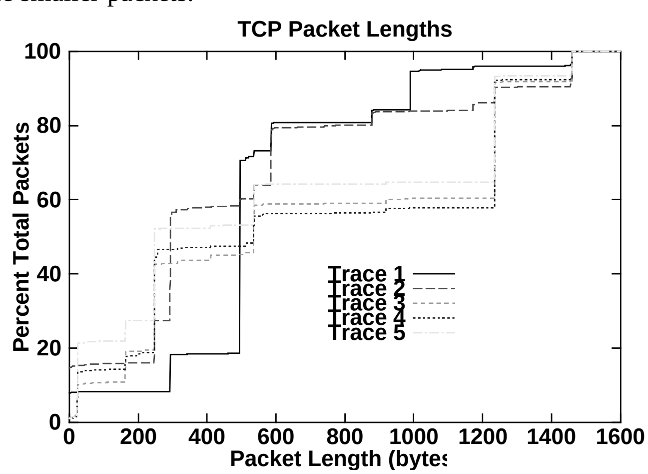

Figure 8. Data flow packet size distribution (TCP)

As can be seen in Figure 8 the TCP packet lengths exhibited less regularity than UDP with a concentration of packet lengths seen at 293 and 495 bytes. These correspond roughly to the 16 Kbps and 20 Kbps audio flows respectively. The packet size is often sent in multiples of fixed increments due to the audio codecs. Hence we see concentrations at 293, 586, 879 and 1172 for 16 Kbps flows as well as 495 and 990 for 20 Kbps flows.

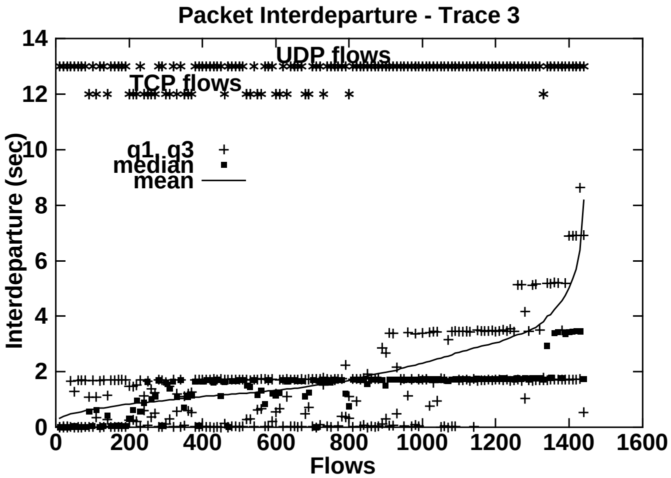

Figure 9. Packet Interdeparture - Trace 3

Streaming audio data typically flows at a uniform rate. We analyzed the packet interdeparture time, which is the time difference between subsequent packets as they depart from the audio server. Figure 9 summarizes several kinds of information about packet interdeparture time for each audio flow. First, we computed the mean interdeparture time for all packets of each flow (shown as the line) and ordered the flows by this measure. In an idealized rate-based flow, this mean could represent packet data transfer rate with a normal distribution around the mean.

To evaluate how well Real Audio traffic is similar to this ideal, we also present median and first and third quartiles. (To improve the clarity of these individual data points we show these statistics only for every tenth flow.) Two aspects of these statistics suggest that audio traffic is not an ideal rate-based flow. First, we observe that median values (shown as dark squares) are consistently much lower than the mean (the line). This indicates that packet interdepartures do follow a normal distribution but instead have a long tail. Second, we observe clustering of the third quartile and median around multiples of about 1.8 seconds. This trend indicates that there is some more complex pattern in the underlying transport protocol. Rather than sending data smoothly, Real Audio sends short bursts of packets separated by gaps. In Section 4.4 we examine two individual flows and characterize this behavior more completely.

4.3. Analysis of Aggregate Users

After considering flows we examined per-user behavior to see if users impart a larger structure on activities. Our definition of a user is a host IP number, which communicates with an audio server. The total number of active users, those that sent or received more than 100 Kbytes in one or more flows, is shown in Table 7.

	Trace 3	Trace 4	Trace 5
Active Users	1397	298	753
Inbound Users	13	10	25
Outbound Users	1384	288	728

Table 7 Summary of audio users.

As with flows, the inbound users are hosts local to the audio server and represent flows from one or more audio codecs. Only the outbound users cause traffic on the Internet. The following table shows more details on the outbound audio flows.

	Trace 3	Trace 4	Trace 5
Active outbound audio flows	1460	324	837
Active outbound audio users	1384	288	728
Mean Number of Flows per user	1.06	1.13	1.15

Table 8 Summary of audio flows.

For the most part the RealPlayer client application limits a listener to one audio source at a time. Hence we would expect to see only one flow per user. However the data indicates that a user can have multiple flows. The "user" in this case is an audio proxy server, which serves the same purpose as an HTTP proxy effectively hiding the actual user from the source and our trace.

The total flow distribution per user is shown in Figure 10. Users employing TCP establish a single flow for both control and audio data, while those employing UDP typically establish one TCP flow for control and one UDP flow for audio data. Figure 10 shows that most of the users have two flows, which means that most of the audio data flows utilize UDP. The percentages seen in Figure 10 roughly correspond to those in the aggregate traffic shown in Table 4.

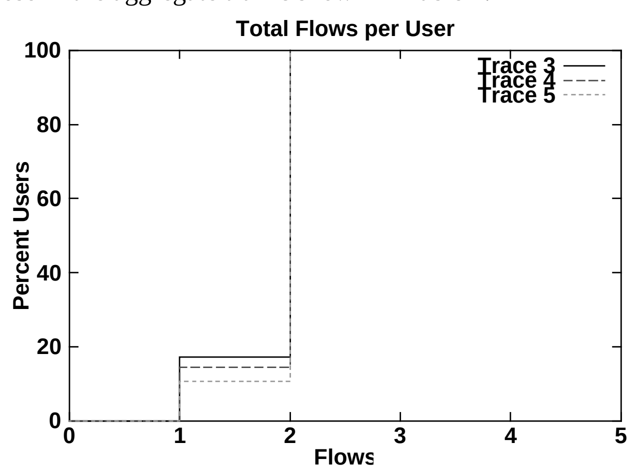

Figure 10. Total number of flows per user

For outbound audio flows the vast majority of the users have only a single audio flow. For those with multiple flows, when we look at the number of concurrent flows per user we see that nearly all of the users (e.g. except for 1 out of 1397 in trace 3) have at most one concurrent audio flow. This strongly suggests that there is no significant use of audio proxy servers in our traces. Nearly all cases of a single user accessing multiple audio flows is due to a single user sequentially selecting the audio selection. Even this behavior is fairly rare, occurring on only 5-10% of the users (Figure 11).

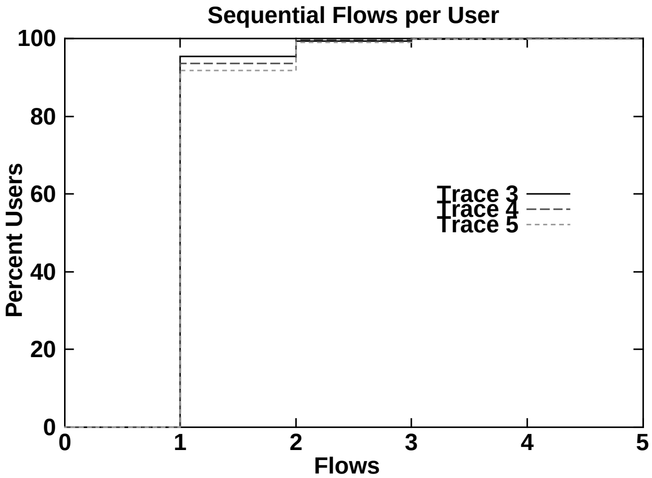

Figure 11. Sequential audio flows per user

To determine the user arrival rate we located the arrival time of the first audio packet for each active audio data flow. Figure 12 indicates that long-term user interarrival times are approximately exponential. This observation is consistent with web user interarrival [2]. This long-term summary does not accurately represent short-term user behavior, since Figure 1 indicates data rates are time-dependent.

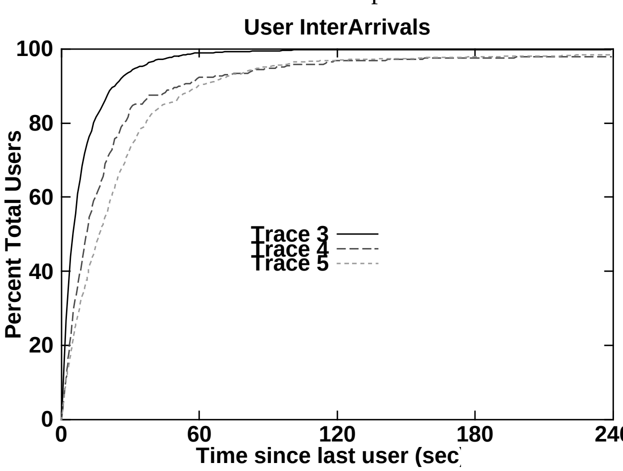

Figure 12. User Arrivals

We measured user duration as the difference between the first and last audio data packet ignoring the control flow. Figure 13 summarizes user durations. Of these we consider trace 5 the most accurate because that trace captured the entire cycle of user arrival and departure, while traces 3 and 4 were terminated with many active users. (Figure 1). Half of the users stayed more than 30 minutes with about 75% staying over an hour. These account for the lengthy flow durations seen in Figure 4. Since we were unable to determine the duration of the actual audio program to which the users were listening, we do not know what percentage of the program Figure 13 represents. We hypothesize that one may be able to derive a user duration distribution function for broadcast events, which takes the event duration as one of the input variables.

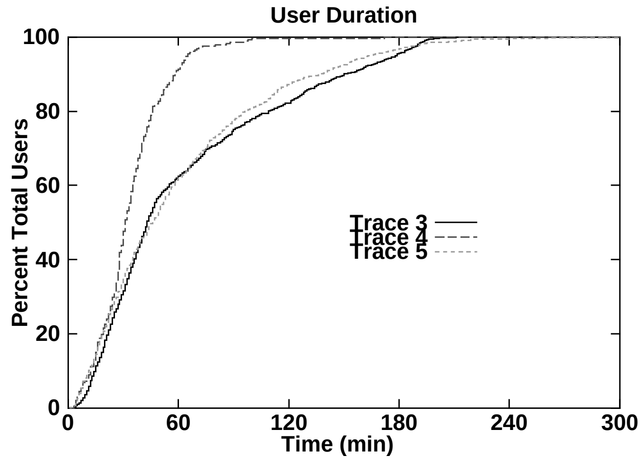

Figure 13. User Duration

4.4. Analysis of Individual Flows

Our analysis of packet interdeparture times indicated surprising regularity in the quartiles of packet interdeparture times across all flows (Figure 9). To interpret this behavior we examined several flows in detail. We summarize the behavior of the two flows below:

Example flow	A	B
Trace	3	3
Transport	Udp	udp
Mean bandwidth	6.7Kbp/s	2.2Kb/s
Mean interdeparture	0.324 sec	4.124 sec
Median interdeparture	0.002 sec	1.730 sec

Table 9 Example Flows

We examine these flows two ways. First, we consider the packet interdeparture times of two adjacent packets. Given packets sent at times t_0, t_1, and t_2, we compute $\delta_1 = t_1 - t_0$ and $\delta_2 = t_2 - t_1$, then graph the point (δ_1, δ_2).

Figure 14 and 17 show the results for the two example flows. In this type of graph, a 45 degree line indicates evenly paced data, while points clustered along the axes indicates that two closely spaced packets are usually followed by a packet spaced further apart. Second, we graph the CDF of packet interdeparture times. Figures 14, 15 and 16 present

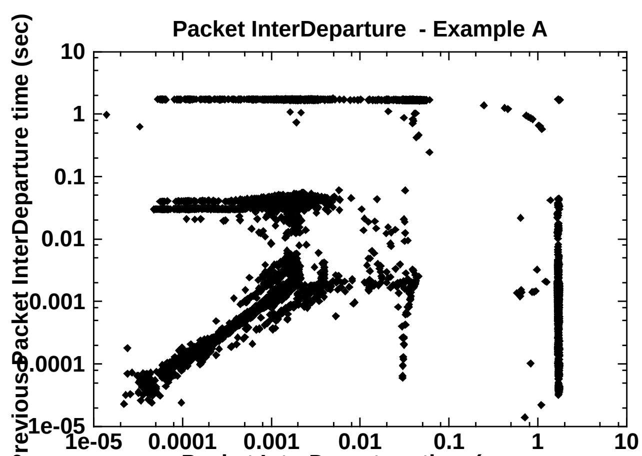

Figure 14. Packet Interdeparture comparison, example flow A

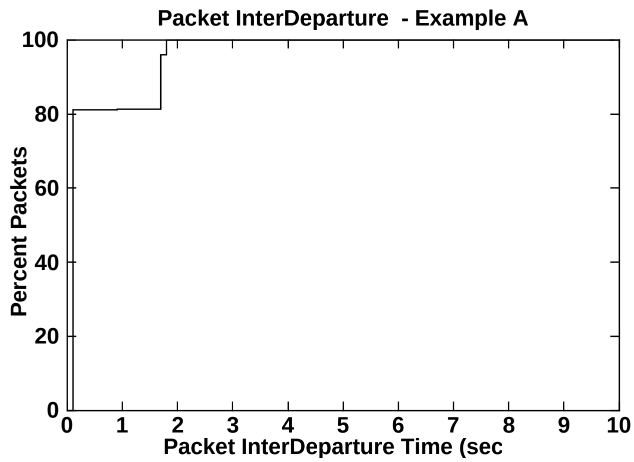

Figure 15. Packet Interdeparture CDF, example flow A

these analyses of example flow A. Flow A represents a "typical" flow like many of the 1397 flows in trace 3. Figure 14 is plotted on a log-log scale to show the detail present at small time-scales. First, note the preponderance of data points concentrated with deltas less than 0.03s. These indicate packets paced at very short intervals (the diagonal line) or sent back-to-back (the heavy weighting at about y=0.03s). Second, we observe a heavy concentration of interdepartures at about 1.8s. These indicate that most of the packets depart in short intervals, less than 0.3s, but a sizable fraction is sent at 1.8s intervals. The interdeparture CDF, Figure 15, corroborates these observations, showing that 80% of the packets are sent with very small intervals and the remaining 20% at about 1.8s.

In Figure 16 we show the data packets plotted against time for a small portion of the flow for example A, illustrating these statistics. The bursty nature of the flow is clearly evident indicating that RealAudio data is sent with a rate-based on-off process. For this flow a burst of a few (typically 6) packets are sent out, followed by an off period of about 1.8s.

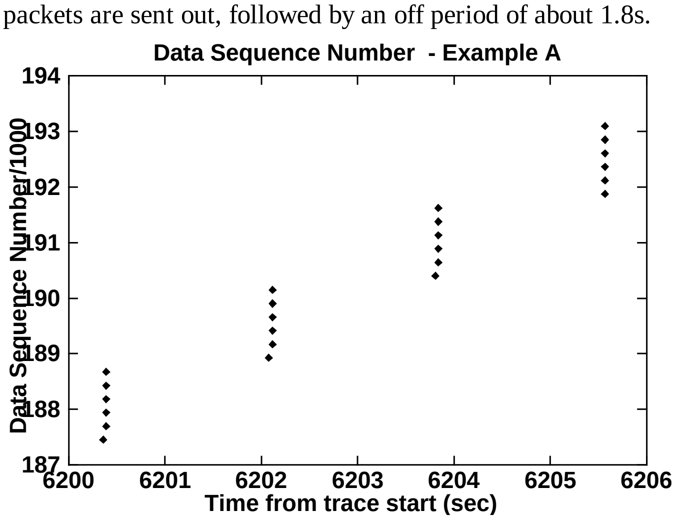

Figure 16. Data Sequence Number, example flow A

Figures 17 and 18 present the two views of interdeparture times for example flow B. Flow B represents flows from trace 3 with unusually high mean interdeparture times. Figure 17 is

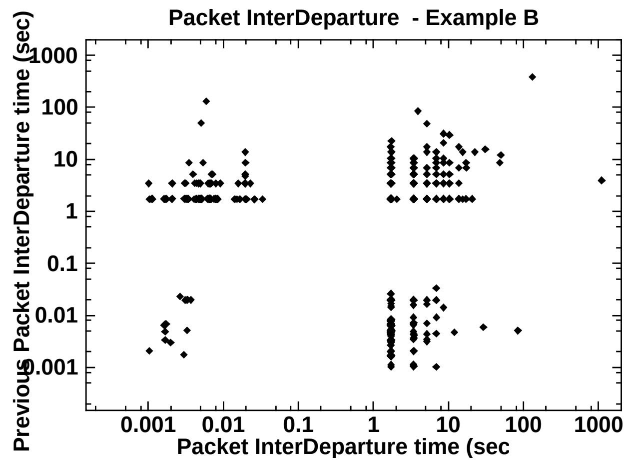

Figure 17. Packet Interdeparture comparison, example flow B

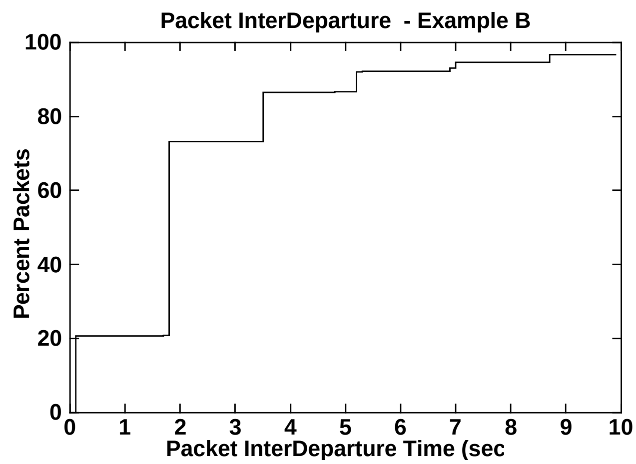

Figure 18. Packet Interdeparture CDF, example flow B

shown on a log-log scale, but note the change in scale compared to Figure 14. In this flow there is a lengthy pause as shown by the data point at about (200, 300). This may represent a user explicitly requesting a pause in transmission, or it may represent some other disruption. Note the differences in the cumulative distribution between this example (Figure 18 and the previous example (Figure 15). Only 20% of the packets departed in less than 0.1s compared to 80% before. This results in a lower bit rate. In Figure 17 we can also see several departure plateaus at multiples of 1.8s. This indicates that the on-off sending process is checked at intervals of approximately 1.8s.

5. Simulation of Audio Flows

The data in section 4 can provide insight into the behavior of streaming audio traffic. This information can offer insight into streaming audio protocol design, new network routing disciplines that might identifying and react to non-TCP-friendly flows, or investigations into interactions between streaming audio and traditional traffic. Laboratory experiments with live audio streams will be difficult, though, because of variable network conditions and the difficulty of purchasing the server and deploying hundreds of clients. We therefore suggest an algorithm that allows simulation of an audio server. The algorithm we describe here is simplified and assumes no proxy servers and that each user listens to exactly one audio flow. Based on our analysis in Section 4.3, this covers 99% of the users.

First, one would place a simulated server in a network. Since we observed that server usage is very time-dependent (Figure 1), one could either chose to model a server at a given time of day (busy or less busy) or one serving an event. Assuming that the simulation is to understand network behavior in extreme situations one would probably select a busy period, so user interarrival times could be based on the distribution in Figure 12. We can place the user in the network by selecting an RTT based on Figure 5. We cannot describe user location in more detail based on the information obtained from our traces. Since we assume no users are proxy servers and each user retrieves a single audio flow (a simplification), we would next select the audio flow duration (Figure 4), and audio data rate (Figure 6). Both of these distributions are content-dependent, so a complete simulation would vary these distributions or only present conclusions for a given class of content.

Finally, one would select a packet length corresponding to retrieval rate (6.5 Kbps) and send data at regular intervals based on Figure 14. One could introduce noise based on Figure 14 as well. If one was interested in modeling bulk audio data, a flow with these parameters might be constructed. A more detailed model would include the control channel feedback of 11B packets. Once a bitrate has been selected, traffic must be generated at that rate. Based on Figures 14, 15 and 16, RealAudio does not emit data at a steady rate, but rather as series of closely-spaced packets followed by a long gap. This can be modeled as an on-off process as we described in section 4.5.

6. Prior Work

There has been little work in the analysis of streaming multimedia traffic although the examination of Internet traffic in general often includes multimedia. Paxton [1] conducted a landmark study of Internet traffic in general, while Danzig et al [12] focused on an analysis of TCP and FTP traffic. Thompson et al [14], examined Internet traffic on an OC-3 trunk for two time periods of 24 hours and 7 days during May and August 1997. The study identified RealAudio traffic and found that it exhibited flow and byte percentages ranging from 0.5% to 2.5% of total traffic. However they only looked at UDP Real Audio flows and their audio flows only lasted for 10-30 seconds, transferring 20 kilobytes on average. Our traces show that audio flows last much longer and transfer considerably more data. This discrepancy may be due to more recent increased usage of the RealPlayer client application, the proliferation of audio services such as Broadcast.com, or differences in audio content. The latter may range from individual songs, talk radio programs or commercial music radio shows.

Traces in other domains have been better studied. Our work was inspired by Mah's examination of web traffic traces and his resulting empirical model [2]. Although he did not consider streaming multimedia, we followed his example of simulating traffic based on CDFs of real data rather than mathematical models.

7. Future Work

This present work is only a brief beginning to the analysis of streaming multimedia traffic on the Internet. Much work remains including obtaining additional packet traces, deriving flow identification methods as well as examining congestion and self-similarity.

Data Sources

This study examined packet traces obtained directly at the audio server. We have considered moderate length traces (5-18 hours) of broadcast radio. Studies of longer durations and other kinds of content (for example, individual songs, conferencing, or Internet telephony) would provide insight into other traffic patterns. It would also be interesting to examine packet traces at a major Internet boundary such as the entrance to a university or an Internet Service Provider in order to determine the percentage of audio traffic.

Audio Congestion Control

Support for TCP-Friendly congestion control is important to the health of the Internet [10]. Two approaches to TCP-friendly audio are new audio protocols [15] and a general congestion manager [16].

Multimedia flow identification

Better means for identifying multimedia flows are needed, particularly if non-TCP friendly flows are to be detected. In addition to port number, We have suggested that packet size and interdeparture times indicate Real Audio traffic. Regularity in packet size and rate can sometimes work well, but not always. Furthermore the data packet may contain both audio and video information. Hence it may be necessary to examine the data payload in order to positively identify the packet. Further complicating this aspect are numerous other streaming multimedia technologies such as Real Audio G2, Liquid Audio, MP3 and the Microsoft media player, although the trend is to support standard protocols, such as RTSP. Efficient router-level means of identifying these flows is an open issue.

Self Similarity

Another important question for multimedia traffic is to understand its aggregate behavior across a range of time-scales. Ethernet [17] and web traffic [18] have been found to be self-similar at large time-scales and to have a much richer behavior at small time scales [19]. Our analysis suggests that multimedia streams exhibit regular behavior at small time scales (see section 4.5). It remains to be seen whether longer-term behavior (Figure 4) prompts self-similarity.

8. Conclusions

This paper has analyzed traces of RealAudio traffic to better understand its characteristics. From our results we find that audio flows differ from typical telnet, FTP and HTTP flows in several important areas. First we found that audio flow durations are significantly longer than typical Internet web flows, so identifying and policing audio traffic may be feasible. Second, 60-70% of audio flows use UDP, and thus require application-specific congestion control. Third, audio flows exhibit a significant amount of regularity in packet lengths, bit rates and interpacket arrivals. These characteristics may be useful in identifying audio flows. Finally audio traffic does not always exhibit steady state characteristics. We have seen long term trends, on the order of hours, in traffic that may be related to the geographical location of listeners or the time of day. However this may be dependent on the audio content. After describing background and our methodology we examine traces in the aggregate, per-user, and per-flow. We also consider how to reproduce this traffic in a simulation to support controlled experiments.

The traditional access method for most Internet users has been the dial-in modem. As high-speed Internet access methods, using cable-modems for example, become cheaper and more ubiquitous, streaming audio and video become increasingly possible. Early examples of the popularity of this traffic include widespread commercial use of RealAudio today and increasing interest in MP3 and Internet telephony. Multimedia traffic may soon make up a substantial fraction of Internet traffic. Our paper suggests that such traffic will have many different characteristics than today's traffic.

Acknowledgements

The authors would also like to thank Henry Heflich, Gary Nelson and Ed Luczycki of Broadcast.com. This study would not have been possible without their support, assistance and hard work in obtaining the necessary packet traces. We would also like to thank Ted Faber for his views and comments on the paper.

References

[1] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics", PhD. Dissertation, University of California, Berkeley, April 1997, ftp://ftp.ee.lbl.gov/papers/vp-thesis/dis.ps.gz.

[2] B. Mah "An Empirical Model of HTTP Network Traffic", In Proceedings of the IEEE Infocom, pp. 592-600, Kobe, Japan, IEEE. April 1997, http://www.ca.sandia.gov/~bmah/Papers/Http-Infocom.ps

[3] Danny Cohen, "Specifications for the Network Voice Protocol", Technical Report ISI/RR-75-39, USC/Information Sciences Institute, March, 1976.

[4] V.J. Hardman, M.A. Sasse, M. Handley, and A. Watson. "Reliable Audio for Use over the Internet" In Proceedings of the INET, Hawaii, The Internet Society. July, 1995. http://info.isoc.org/HMP/PAPER/070/

[5] Van Jacobson, "Multimedia Conferencing on the Internet", Tutorial at SIGCOMM '94; published as a special issue of the SIGCOMM Newsletter, August, 1994.

[6] Broadcast.com, http://www.broadcast.com

[7] Network Associates, Inc., http://networkassociates.com

[8] http://ee.lbl.gov

[9] Real Networks Inc., Real Audio Firewall Support, http://service.real.com/firewall/

[10] Sally Floyd, Kevin Fall, "Promoting the Use of End-to-End Congestion Control in the Internet" IEEE/ACM, February 1998, ftp://ftp.ee.lbl.gov/papers/collapse.feb98.ps .

[11] Jeffrey Mogul, "Network Behavior of a Busy Web Server and its Clients", Technical Report N. 95/5 , DEC Western Research Laboratory, October 1995, http://www.research.digital.com/wrl/techreports/abstracts/95.5.html.

[12] P. Danzig, S. Jamin, R. Caceres, D. Mitzel and D. Estrin, "An Empirical Workload Model for Driving Wide-area TCPIP Network Simulations", Journal of Internetworking: Research and Experience, Vol. 3 No. 1), pp. 1-26, March, 1992, http://netweb.usc.edu/jamin/traffic/traffic.ps.Z

[13] W. Richard Stevens, "TCP/IP Illustrated Volume 1". Addison-Wesley Publishing Company, 1994, pp 299-301.

[14] Kevin Thompson, Gregory Miller, Rick Wilder "Wide Area Internet Traffic Patterns and Characteristics (extended version)", IEEE Network Magazine, Vol. 11, No. 6, pp.10-23, Nov/Dec 1997, http://www.vbns.net/presentations/papers/ MCItraffic.ps.

[15] Reza Rejaie, Mark Handley, and Deborah Estrin. "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet", In Proceedings of the IEEE Infocom, IEEE, March 1999. http://catarina.usc.edu/reza/papers/infocom99.ps.

[16] H. Balakrishnan, H. S. Rahul, and S. Seshan. "An Integrated Congestion Management Architecture for Internet Hosts", In Proceedings of the ACM SIGCOMM , pp. 175-188, Cambridge, MA, USA, ACM, September 1999. http://inat.lcs.mit.edu/publications/BRS99.ps.

[17] Will Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", IEEE/ACM Transactions on Networking, Vol.2 No.1, February 1994, pp. 1-15. http://www.acm.org/pubs/citations/journals/ton/1994-2-1/p1-leland/

[18] Mark Crovella, Azer Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Transactions on Networking, Vol 5, No. 6, pages 835-846, December 1997. http://www.acm.org/pubs/citations/journals/ton/1997-5-6/p835-crovella/.

[19] Anja Feldmann, Anna C. Gilbert, Polly Huang, and Walter Willinger. "Dynamics of IP traffic: A study of the role of variability and the impact of control", In Proceedings of the ACM SIGCOMM , Cambridge, MA, USA, ACM. August 1999.