MediaPlayer™ versus RealPlayer™ - A Comparison of Network Turbulence

Mingzhe Li, Mark Claypool, Robert Kinicki
Computer Science Department
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
{lmz|claypool|rek}@cs.wpi.edu

Abstract — The performance of currently available streaming media products will play an important role in the network impact of streaming media. However, there are few empirical studies that analyze the network traffic characteristics and Internet impact of current streaming media products. This paper presents analysis from an empirical study of the two dominant streaming multimedia products, RealNetworks RealPlayer™ and Microsoft MediaPlayer™. Utilizing two custom media player measurement tools, RealTracker and MediaTracker, we are able to gather application layer and network layer information about RealPlayer and MediaPlayer for the same media under the same network conditions. Our analysis shows that RealPlayer and MediaPlayer have distinctly different behavior characteristics and exposes some of the impact of streaming media on the network and provides valuable information for building more realistic streaming media simulations.

Index Terms — MediaPlayer, RealPlayer, Streaming Multimedia

I. INTRODUCTION

Unlike typical Internet traffic, streaming video is sensitive to delay and jitter, but tolerates some data loss. In addition, streaming video typically prefers a steady data rate rather than the bursty data rate associated with window-based network protocols. Hence, streaming video applications often use UDP rather than TCP, suggesting that video flows may not be TCP-friendly or, even worse, that video flows are unresponsive to network congestion.

Due to commercial streaming products, such as the Windows Media Player™ (MediaPlayer) and RealNetworks RealPlayer™ (RealPlayer), streamed media traffic on the Internet has increased dramatically [JUP01]. Thus it is important to have a better understanding of the network impact of commercial media products to prepare for future Internet growth in streaming media.

Research that attempts to deal with unresponsive traffic [CD01, FKSS01, MFW01, SSZ98] often models unresponsive flows as transmitting data at a constant packet size, constant packet rate, or as "firehose" applications, transmitting at an unyielding, maximum rate. Realistic modeling of streaming media at the network layer will facilitate more effective network techniques that handle unresponsive traffic flows.

This paper investigates the size and shape of streaming flows, which we call *turbulence*[1], for both RealPlayer and MediaPlayer. We develop custom software, which we call *MediaTracker*, to play and record MediaPlayer video streams, and use it with previously developed software [WC02], called *RealTracker*, that plays and records RealPlayer video streams. We design experiments that simultaneously stream both RealPlayer and MediaPlayer videos from the same content and the same Internet servers. We capture application level statistics and network level statistics and analyze the relationship and compare the two types of streams.

The rest of this paper is organized as follows: Section 2 describes our experimental setup; Section 3 analyzes the data obtained from our experiments; Section 4 briefly describes how results from Section 3 could be used to simulate streaming video; Section 5 summarizes our conclusions and presents possible future work.

II. EXPERIMENTS

A. Methodology

To carefully study the behavior of MediaPlayer and RealPlayer streaming video over the Internet, we took the following steps:

- We built a customized version of MediaPlayer, called *MediaTracker*, to playback MediaPlayer clips and record

[1] The term *footprint* is often used in systems work in the context of the basic size a piece of memory of some software. In a network, the size and distribution of packets over time is important, hence our word *turbulence*.

statistics and used a previously customized version of RealPlayer, called *RealTracker* [WC02], to playback RealVideo clips and record statistics (See Section 2.B).

- We accessed Web servers with identical video content for both MediaPlayer and RealPlayer where the video servers themselves were co-located at the same or close to same server node (see Section 2.C).
- For each clip selected, we streamed identical MediaPlayer and RealPlayer clips simultaneously from the servers to one client concurrently receiving the video clips on the customized players. Both application level information and network packets statistics were recorded (see Section 2.D).

B. Tools

MediaTracker records application level information while playing back MediaPlayer clips. MediaTracker was developed using Java Scripts and Windows Media Software Development Kit (SDK)[2] provided by Microsoft for customized MediaPlayer development.

Using the core MediaPlayer engine, MediaTracker plays MediaPlayer clips while recording encoded bit rate, playback bandwidth, application level packets received, lost and recovered packets, frame rate, and quality. MediaTracker supports a customized play list to automate playback of multiple video clips.

RealTracker, originally developed using RealNetworks' SDK [3] in Microsoft Visual C++ for a Internet-wide RealVideo performance study [WCZ01][4], employs the RealPlayer core video engine that comes with the free basic version of RealPlayer. It records statistics similar to MediaTracker including encoded bit rate, playback bandwidth, and frame rate. RealTracker also supports customized play lists for automatic playback of multiple video clips.

Ethereal[5], a free network protocol analyzer for Unix and Windows, captures data from a network and allows interactive browsing of the captured data. It includes a display filter language and the ability to view a reconstructed stream from a TCP session.

C. Clip Selection

To compare MediaPlayer and RealPlayer under the same network conditions we selected servers that had both MediaPlayer and RealPlayer versions of the same videos. We selected clip sets from the same website with both high (about 300 Kbps) and low (about 56 Kbps) encoded data rates in both MediaPlayer and RealPlayer formats. At one server, we were able to find a pair of very high data rate clips (about 600 Kbps). For all clips, we verified each clip was from the same subnet since media clips that appear on the same Web site may actually be served from different subnets.

Data Set		Encode (Kbps)	Clip Info.
1	R-h/M-h	284.0/323.1	Sports 3:46
	R-l/M-l	36.0/49.8	
2	R-h/M-h	268.0/307.2	Commercial 0:39
	R-l/M-l	84.0/102.3	
3	R-h/M-h	284.0/307.2	Sports 0:60
	R-l/M-l	36.5/37.9	
4	R-h/M-h	180.9/309.1	Music TV 4:05
	R-l/M-l	26.0/49.6	
5	R-h/M-h	217.6/250.4	News 1:47
	R-l/M-l	22.0/39.0	
6	R-v/M-v	636.9/731.3	Movie clip 2:27
	R-h/M-h	271.0/347.2	
	R-l/M-l	38.5/102.3	

Table 1. Experiment Data sets

The above criteria greatly reduced the number of clips available. We collect six sets of clips for our experiments with a total of 26 clips with varied contents, lengths, encoding data rates, all encoded in both MediaPlayer video and RealPlayer video formats. The clip sets chosen are shown in Table 1.

D. Experiment Setup

The experimental setup strives to reduce the effects of the client and concentrate on the effects of the video on the network. The client PC was a Pentium-4 1.8 GHz processor, 512M RAM, AGP 32MB video card, PCI sound card, PCI 10M NIC running Microsoft Windows 2000 professional. The software tools were Microsoft MediaPlayer version 7.1, RealNetworks RealOne Player build 6.0.10.505, and Ethereal version 0.8.20. Since MediaPlayer and RealPlayer can use either TCP or UDP, we forced both players to use UDP as the transport protocol for all experiments since it is more commonly used [WCZ01].

The PC was connected to the WPI campus network[6], which is in turn connected to the Internet. During pilot tests, we verified that at no time during playout of any of the video clips were the CPU or memory overly taxed nor was the maximum last-hop bandwidth the bottleneck.

[2] http://www.microsoft.com/windows/windowsmedia/create/develop.asp
[3] http://www.realnetworks.com/resources/sdk/index.html
[4] RealTracker was formerly known as *RealTracer*.
[5] http://www.ethereal.com/

[6] http://www.wpi.edu/Admin/Netops/MRTG/

All the experiments were run Monday through Friday from 3:00 pm to 6:00 pm, EST, between March 29 and April 11, 2002. Before and after each run, `ping` and `tracert` were run to verify that the network status had not dramatically changed, say from a routing change, during the run.

III. ANALYSIS

A. Network Conditions

The condition of the network during the experimental connections is estimated from the round-trip time and number of hops for each data pair. The experiments ran with a median round-trip time of 40 ms and a maximum round-trip time of 160 ms. Most of the servers were between 15 and 20 hops away, results typical of other streaming experiments [LR01]. The average loss rate reported from `ping` was near 0%, similar to results in [LR01], although we did observe a few packet losses during the experiments. From the information above, we assume that experiments ran under common network conditions without network congestion.

B. Bandwidth and Encoding Data Rate

The encoded data rate in Table 1 was not from the link description provided by the Web page, but instead was captured by our customized video players. For the same advertised data rate, the RealPlayer clips always had a lower encoding rate than the corresponding MediaPlayer clip. For example, two clips advertised as needing a 300 Kbps connection yielded a 284 Kbps encoded rate for the RealPlayer clip and a 323 Kbps encoded rate for MediaPlayer clip. RealPlayer's higher bandwidth consumption may be because of its buffering and playback mechanism, as described in Section 3.E.

C. IP Packet Fragmentation

Large application frames sent over UDP can result in IP fragmentation. Figure 1 shows the network layer packet arrival pattern for one high encoding rate pair (a 250 Kbps MediaPlayer clip and a 217 Kbps RealPlayer clip). The MediaPlayer packets have a very regular pattern, with groups of packets and a constant number of packets in each group. Further investigation of the packet types using Ethereal reveals that each packet group is composed of one UDP packet and the remaining packets are IP fragments. All the packets in one group except the last IP fragment are 1514 bytes. The last fragment size is different for each clip but is the same within each clip. The default Maximum Transfer Unit (MTU) for Windows of 1500 bytes[7] suggests that MediaPlayer servers send large application layer frames that are then fragmented by the operating system to the size of the MTU.

The fact that no IP fragments were observed in any of the RealPlayer traces suggests that RealServer breaks application layer frames into packets smaller than the MTU to avoid IP fragmentation.

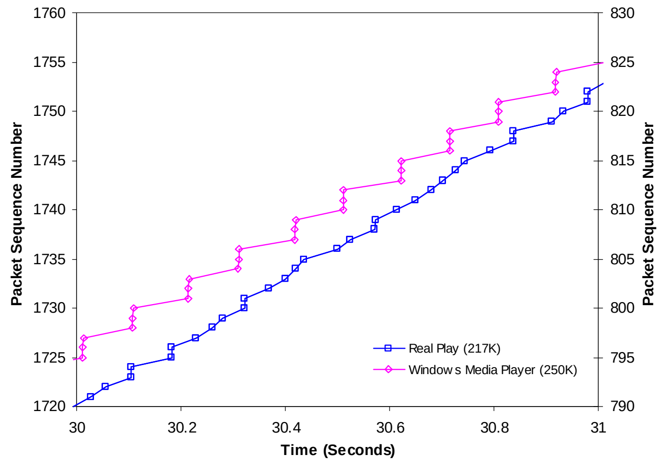

Figure 1. Packet Arrivals vs. Time (Data Set 5, Single Clips)

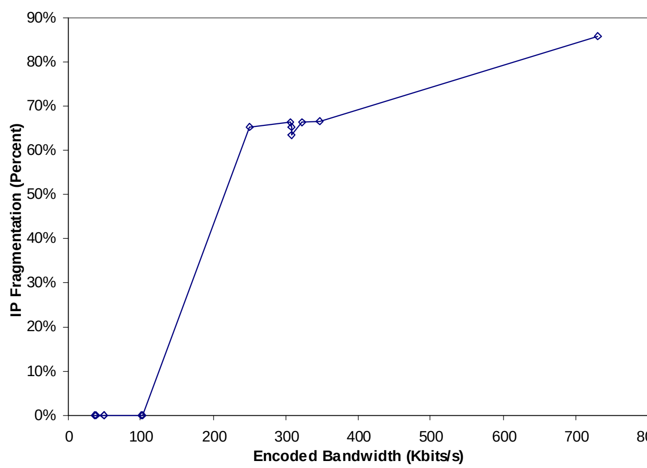

Figure 2. IP Fragmentation vs. Encoded Data Rate (All MediaPlayer Clips)

Figure 2 depicts MediaPlayer IP fragmentation for different encoding rates. The fragmentation percentage increases with encoded rate. For example, for clips encoded at 300 Kbps 66% of the packets are IP fragments, while below 100 Kbps there is no fragmentation. IP fragmentation can seriously degrade network goodput during congestion, since a loss of a single fragment results in the larger application layer frame being discarded. At its worst, fragmentation leads to congestion collapse in the network [FF99]. Fragmentation based congestion collapse can occur when some of the cells or fragments of a network-layer packet are discarded (e.g. at the link layer),

[7] http://support.microsoft.com/default.aspx?scid=kb;EN-US;q140375

while the rest are delivered to the receiver, thus wasting bandwidth on a congested path.

D. Packet Sizes

The MediaPlayer packet sizes also show more regularity than RealPlayer packet sizes. MediaPlayer packets have a high density at one packet size while RealPlayer packet sizes are distributed over a larger range and do not have a single peak density point. In a typical low date rate clip, over 80% of MediaPlayer packets may have a size between 800 Bytes and 1000 bytes. For high data rate clips, MediaPlayer has two high density distribution packet sizes, one at 1500 bytes contributed by the UDP and IP fragments, and another at the size of the last IP fragment, the remaining part of the large application layer packets.

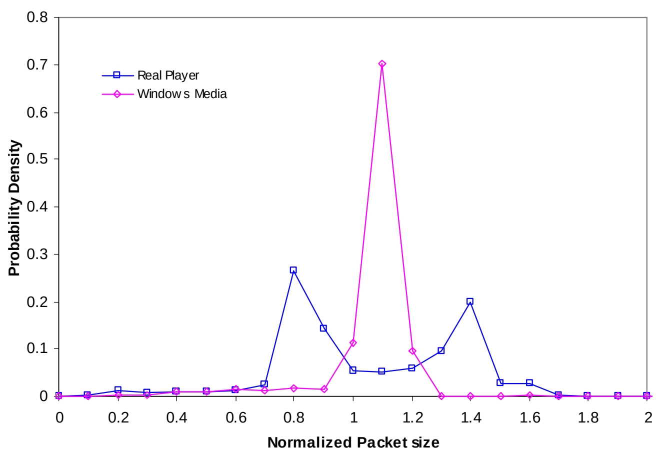

Figure 3. PDF of Normalized Packet Size (All Data Sets)

We summarize the packet size distributions for all experiments by normalizing the packets by the average packet size seen over the entire clip. Figure 3 shows a PDF of the normalized packets. The sizes of MediaPlayer packets are concentrated around the mean packet size, normalized to 1. The sizes of RealPlayer packets are spread more widely over a range from 0.6 to 1.8 of the mean normalized packet size.

E. Packet Interarrival Times

CBR traffic has fixed-size packets and a constant packet arrival rate. The difference in packet interarrival times, also known as jitter, can cause degradations to video perceptual quality that are as serious as packets loss [CT99].

For high data rate MediaPlayer clips, we consider only the first UDP packet in each packet group to remove the noise caused by the IP fragments. Figure 4 shows Cumulative Density Functions (CDFs) of the normalized packet interarrival times. The CDF of packet interarrival times for RealPlayer has a gradual slope as packets arrive over all ranges of the normalized interarrival times. In contrast, the CDF of packet interarrival times for MediaPlayer is quite steep around a normalized interarrival time of 1, indicating that most packets arrive at constant time intervals. This packet interarrival analysis combined with the packet size analysis from Section 3.D suggests that MediaPlayer traffic has a more constant bit rate than RealPlayer traffic.

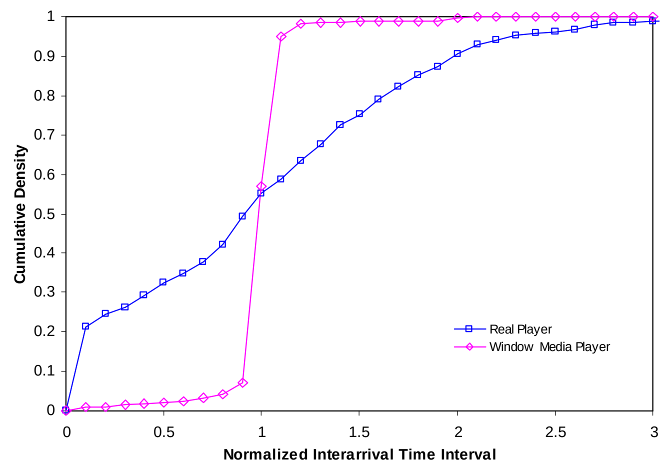

Figure 4. CDF of Normalized Packet Interarrival Times (All Data Sets)

F. Buffering Mechanism

Delay buffering is a well-known technique [RKTS94, SJ95] used to remove jitter. Data enters the buffer as it streams to the player and leaves the buffer as the player displays the video. If network congestion causes a large interarrival time between packets, the player keeps the video smooth by playing buffered data. Both RealPlayer and MediaPlayer use delay buffering to remove the effects of jitter. Figure 5 depicts the bandwidth used over time for one data set. When streaming begins, RealPlayer transmits higher than the playout rate until the delay buffer is filled, at which time it transmits at the playback rate. The streaming duration is shorter for RealPlayer than for MediaPlayer since RealPlayer transmits more of the encoded clip during the buffering phase than does MediaPlayer. MediaPlayer always buffers at the playback rate resulting in a less bursty data rate.

In Figure 5, the buffering rate of RealPlayer in proportion to the playout rate is higher for the low data rate clip than it is for the high data rate clip. Figure 6 depicts the ratio of buffering rate to playout rate for all RealPlayer clips. This ratio decreases as the encoding rate increases. For example, for the low data rate clips (less than 56 Kbps), the buffering rate to playout rate ratio is as high as 3, while for the very high data rate clip (637 Kbps), the buffering rate to playout rate ratio is close to 1, possibly because the bottleneck bandwidth is insufficiently small for a higher buffering rate.

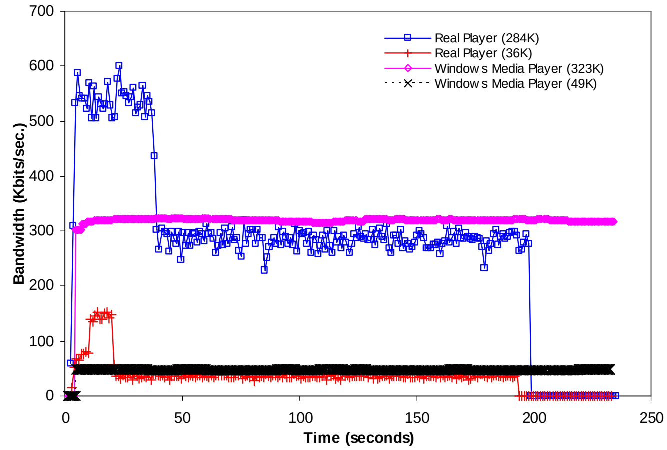

Figure 5. Bandwidth vs. Time (Data Set 1, Single Clips)

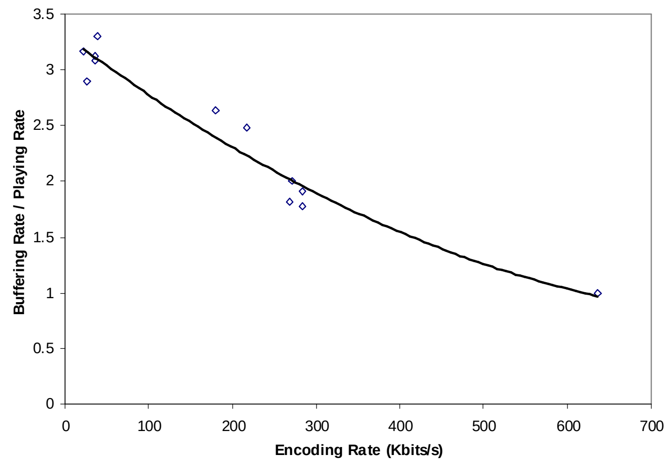

Figure 6. Buffering Rate/Playback Rate vs. Encoding Rate for RealPlayer Clips (All Data Sets)

At the same size buffer, RealPlayer begins playback of the clip before MediaPlayer. If RealPlayer and MediaPlayer begin clip playback at the same time, MediaPlayer has a smaller buffer and may suffer from more quality degradations due to jitter. From the user perspective, RealPlayer either begins clip playback sooner or has a smoother playout than MediaPlayer. From the network perspective, RealPlayer generates burstier traffic that is harder for the network to manage.

G. Packets Received by Network Layers

Packets received by the operating system will be received later by the application. MediaTracker allows us to record the time application layer packets are received. Figure 7 compares the time the network layer receives the packets to the time the application layer receives the packets. Although the figure only shows 4 seconds of data, the same pattern occurs for all MediaPlayer clips over the entire clip duration. The operating system receives packets in regular intervals of 100 ms, while the MediaPlayer application receives packets in groups of 8, once per second. The difference between the time the application receives a packet and the time that the operating system receives the packet may be due to packet interleaving [PHH98]. We are not able to gather application packets in RealTracker.

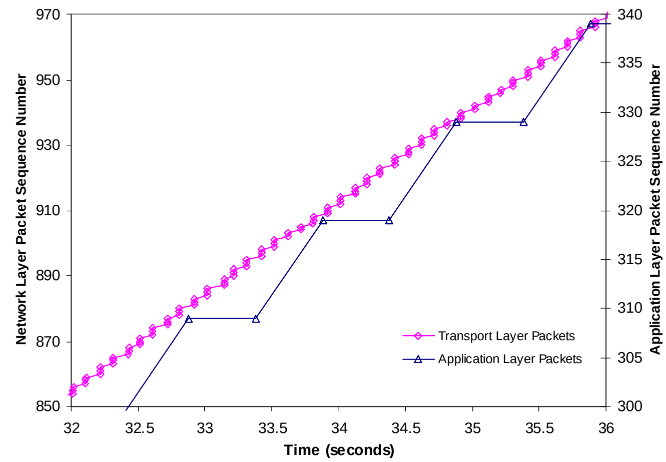

Figure 7. Packets Received by Network Layers for MediaPlayer (Data Set 3, Single Clips)

H. Frame Rate

Video quality is often measured by frame rate where a higher frame rate yields smoother motion in a video. In each clip set, the frame sizes for the MediaPlayer and RealPlayer were the same. In our high data rate experiments, MediaPlayer and RealPlayer both reached the full-motion rate of 25 frames per seconds. The lower data rate MediaPlayer clips played out at less than 15 frames per second. The equivalent RealPlayer clips played out at a significantly higher frame rate than the corresponding MediaPlayer clip. Figure 8 graphs frame rate versus playout bandwidth for all clip data sets. For the low, high and very high clips, the average frame rate is plotted versus average playout rate, along with standard error bars, and connected by lines. Similar to the results for frame rate versus encoded rate, RealPlayer has a higher frame rate than MediaPlayer for the same bandwidth.

IV. SIMULATION OF VIDEO FLOWS

Empirical experiments with live video streams are often difficult because of variable network conditions and the costs involved with deploying large numbers of video clients. However, by using simulations the previous section results may be useful for streaming video protocol designs, new network router queue management disciplines that react to streaming video flows, and understanding interactions between streaming audio and traditional traffic. We briefly sketch out the design of such simulations.

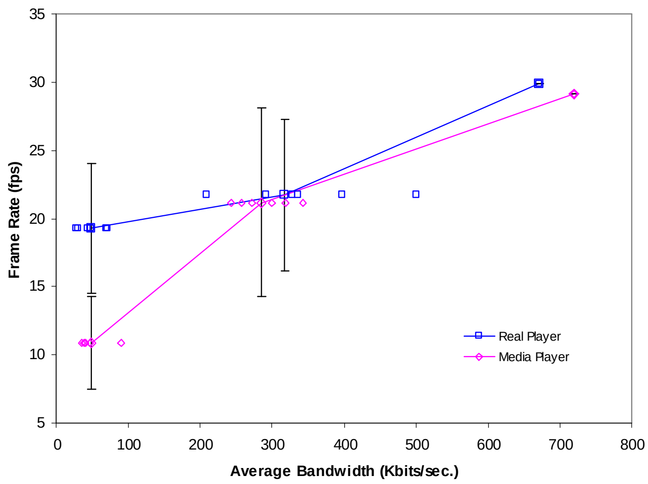

Figure 8. Frame Rate vs. Average Bandwidth (All Data Sets)

One can model a video player in the simulated network by selecting an RTT based on average conditions in our experiments. Encoding rate and clip length from one of the data sets in Table 1 can be used and packet sizes can be modeled from distributions based on Figure 3 while packet interarrival intervals could be based on Figure 4 distributions. MediaPlayer packets should include IP fragmentation rates based on Figure 2. RealPlayer data rates for the first 20 seconds (for low data rate clips) to 40 seconds (for high data rate clips) should be higher than the encoded rate based on Figure 6.

V. SUMMARY AND FUTURE WORK

This work presents an empirical study comparing the impact on the network for RealPlayer and MediaPlayer, Our analysis shows that high bandwidth MediaPlayer traffic can have up to 80% IP fragmentation rates, while RealPlayer has none. MediaPlayer packet sizes and inter-packet times are typical of CBR flows, while RealPlayer packet sizes and inter-packet times vary considerably more. For all encoding data rates, RealPlayer buffers at a higher rate than does MediaPlayer, making RealPlayer burstier. For low encoding data rates and the same average playout bandwidth, RealPlayer has a higher average frame rate than MediaPlayer.

Requiring equivalent content in both RealPlayer format and MediaPlayer format on a single site limited the range of our study. Despite this, the results presented here should be useful to network practitioners seeking insight into the practices and differences in commercial streaming video players. Network researchers should be able to use the results to produce more realistic video traffic for popular simulators, such as NS.

This study examined video clip traces obtained directly at a single player. It would be interesting to examine traces at an Internet boundary, such as the egress to our University, or at least at several players and more clips. Such analysis might reveal interactions between the media flows that our single client studies did not illustrate.

We wish to thank Yubing Wang and Zheng Zuo for their development of RealTracker and Yin Zhang for his preliminary version of MediaTracker.

VI. REFERENCES

[CD01] A. Clerget and W. Dabbous. "Tag-based Unified Fairness", In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, USA, Apr 2001.

[CT99] M. Claypool and J. Tanner. "The Effects of Jitter on the Perceptual Quality of Video", In *Proceedings of the ACM Multimedia Conference*, Vol. 2, Orlando, Florida, USA, Nov 1999.

[FF99] Floyd, S., and Fall, K. "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, Aug 1999.

[FKSS01] W. Feng, D. Kandlur, D. Saha, and K. Shin. "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness", In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, USA, Apr 2001.

[JRCC00] J. van der Merwe and R. Cceres and Y. Chu and C. Sreenan. "mmdump: A Tool for Monitoring Internet Multimedia Traffic", *ACM Computer Communication Review (CCR)*, Vol. 30, No. 5, Oct 2000.

[JUP01] Jupiter Media Metrix. "Users of Media Player Applications Increased 33 Percent Since Last Year", *Press Release*, Apr 2001. http://www.jup.com/company/pressrelease-.jsp?doc=pr010403

[LR01] D. Loguinov and H. Radha. Measurement Study of Low-bitrate Internet Video Streaming, In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, USA, Nov 2001.

[MFW01] R. Mahajan, S. Floyd, and D. Wetherall. "Controlling High-Bandwidth Flows at the Congested Router", In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, Nov 2001.

[PHH98] C. Perkins, O. Hodson and V. Hardman. "A Survey of Packet-Loss Recovery Techniques for Streaming Audio", *IEEE Network Magazine*, Sep/Oct, 1998.

[RKTS94] R. Ramjee, J. Kurose, D. Towsley and H. Schulzrinne. "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks", In *Proceedings of IEEE Computer and Communications*, Toronto, Ontario, Canada, 1994.

[SJ95] D. Stone and K. Jeffay. "An Empirical Study of Delay Jitter Management Policies", *ACM Multimedia Systems*, Vol. 2, No. 6, Jan 1995.

[SSZ98] I. Stoica, S. Shenker, and H. Zhang. "Core-Stateless Fair Queuing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks", In *Proceedings of ACM SIGCOMM*, Vancouver, Canada, 1998.

[WC02] Y. Wang and M. Claypool. "RealTracer - Tools for Measuring the Performance of RealVideo on the Internet", *Technical Report WPI-CS-TR-02-12*, CS Dept, Worcester Polytechnic Institute, May 2002.

[WCZ01] Y. Wang, M. Claypool and Z. Zuo. "An Empirical Study of RealVideo Performance Across the Internet", In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, USA, Nov 2001.