A measurement study of RealMedia streaming traffic

Tianbo Kuang and Carey Williamson

Department of Computer Science, University of Calgary

ABSTRACT

With the growing popularity of real-time audio/video streaming applications on the Internet, it is important to study the traffic characteristics of such applications and to understand their implications on network performance. In this paper, we present a measurement study of RealMedia streaming traffic, where the focus is both on the application-layer view (i.e., the output of the audio/video encoder) and on the network-layer view (i.e., the departure process for network packets emanating from the RealMedia server).

Our main observation is that, although RealVideo can be compressed as Variable-Bit-Rate (VBR) at the application layer, it is often streamed as Constant-Bit-Rate (CBR) at the network layer. The audio and video streams have a hierarchical traffic structure: at large time scales (minutes), the overall bit rate is constant; at medium time scales (seconds), the packets have an on and off pattern due to the interleaving of audio and video; at fine-grain time scales (sub-second), back-to-back packet trains of two or more packets are often seen.

We also note that most CBR-coded RealVideo streams are not long-range dependent (LRD). We attribute this difference to the CBR nature of the coder, which dynamically changes the video frame rate to keep the Internet traffic demands near-CBR over moderate time scales.

Keywords: Multimedia Streaming, Wireless LANs, Network Traffic Measurement, Workload Characterization

1. INTRODUCTION

Continuous-media (CM) streaming has been widely deployed in today's Internet to provide real-time audio and video content to home, school, and office users with wired network connections. According to work done by Chesire *et al.*[1] in 2001, streaming media usage has increased significantly compared to several years ago (e.g., a 1999 study by Wolman *et al.*[2] showed that CM traffic accounted for 18% to 24% of Web-related traffic coming into the University of Washington).

This media streaming trend is expected to grow with the advent of broadband wireless networks, which free end users from the constraints of fixed connections, enabling the retrieval of multimedia content from virtually anywhere at any time. However, wireless media streaming presents challenges, due to the disparities between the characteristics of the wireless propagation environment (e.g., multi-path fading, channel interference, high error rates) and the stringent streaming requirements of continuous media (e.g., delay, delay variation, and error resilience). Although today's compression technology has reduced the size of multimedia files dramatically, CM streaming over wireless networks still remains a challenging problem because of relatively long session durations, during which the wireless channel quality can fluctuate significantly. Moreover, researchers have discovered that Variable-Bit-Rate (VBR) video exhibits long-range dependence (LRD)[3,4] (i.e., burstiness across many time scales). A measurement study of CM traffic characteristics and wireless networks can provide valuable insights into the design of future wireless multimedia networks and applications.

With these goals in mind, we conducted a network traffic measurement and workload characterization study of RealMedia[5] streaming on a wireless LAN. In our study, we focus on the RealMedia traffic streamed from a wired-Internet RealMedia server to a single mobile client on an in-building IEEE 802.11b[6] wireless LAN (WLAN) in the Department of Computer Science at the University of Calgary. The choice of RealMedia is

Author contact information: Department of Computer Science, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4 Phone: (403) 220-6780 Fax: (403) 284-4707 Email: {*kuang, carey*}*@cpsc.ucalgary.ca*

because it is one of the most popular CM streaming applications on the Internet. The choice of IEEE 802.11b WLAN is because it is readily available and it represents a recent trend in broadband wireless network evolution.

This paper studies the RealMedia characteristics at both the application layer and the network layer. The traffic at the application layer, which represents the intrinsic properties of the media source and the codec, is studied by looking at the output of the RealMedia codec. The traffic at the network layer is studied by capturing the traffic from a RealMedia server using the `tcpdump` utility. The latter represents the real workload seen by the network, reflecting all server effects (e.g., smoothing, shaping) on the traffic structure.

The rest of the paper is organized as follows. Section 2 describes related work on Internet audio/video measurement and characterization. Section 3 provides some background on CM streaming, and the RealSystem streaming architecture from RealNetworks. Section 4 describes the experimental setup for our measurement study. The measurement results at the application layer and the network layer are presented in Section 5 and Section 6, respectively. Finally, Section 7 summarizes the results and concludes the paper.

2. RELATED WORK

There are few empirical studies of Internet real-time media streaming at the network layer, though we mention several of them here. Mena *et al.*[7] studied the RealAudio traffic from a popular Internet audio server and found that RealAudio traffic shows behaviour that is not TCP-friendly. They found that RealAudio packet traffic was bursty at small time scales, although the overall bit rate was constant at large time scales. Wang *et al.*[8] conducted a wide-area study of RealVideo traffic from several geographically-different servers to different users. They found that users generally achieve good quality video with an average frame rate of 10 frames per second (fps), though few achieve TV-quality full motion video (24 to 30 fps). Loguinov *et al.*[9] conducted an emulated study of streaming low-bitrate MPEG-4 video to home users in more than 600 major U.S. cities. They reported the results in terms of the packet loss, round-trip delay, one-way delay jitter, packet reordering, and path asymmetry. In another paper, Chesire *et al.*[1] analyzed the stream media workload generated between clients at the University of Washington and servers outside.

The study of video traffic at the application layer, on the other hand, has been carried out extensively. For example, see the works by Beran *et al.*,[3] by Garrett and Willinger,[4] and by Krunz and Tripathi,[10] and the references therein. The general observation is that VBR video exhibits LRD. Recently, Fitzek *et al.*[11] analyzed a large number of MPEG-4 and H.263 compressed full-length video clips. For VBR video, their LRD observations are consistent with the literature. However, for Constant-Bit-Rate (CBR) video, there is no LRD in most clips.

Our work is closest in spirit to those by Mena[7] and by Fitzek,[11] though with three main differences. First, we study both audio and video traffic. Second, we study traffic at both the application layer and the network layer. Third, we study the impact of application-layer traffic on the network layer, and the structural changes in the traffic produced by the transport-layer protocols.

3. BACKGROUND

3.1. Continuous-media streaming

A logical view of a CM streaming system is shown in Figure 1. The encoder compresses the media data and writes it at rate $x(t)$ into a sending buffer (real-time coding) or onto a disk (offline coding). The media server then takes control and transmits the data at rate $y(t)$ to the client through a packet network. Due to network effects, the media data arrives at the receiver with rate $z(t)$. The client plays back the data at rate $x(t)$, with a buffer to compensate for the mismatch between $z(t)$ and $x(t)$. The sending rate $y(t)$ has to meet two constraints: the receiving buffer must not overflow its finite capacity, and the receiving buffer must not be starved. CBR-encoded media data (i.e., constant $x(t)$) can be streamed in either CBR (constant $y(t)$) or VBR (time-varying $y(t)$) form. The same applies for VBR-encoded media, provided that the above constraints on $y(t)$ are met.

The encoder compresses data using *intra-frame* coding and *inter-frame* coding. An intra-frame coded frame is called an I frame. Two types of inter-frames are usually used: P frames and B frames. The former has only one forward reference frame, while the latter has a forward and a backward reference frame. In general, I frames tend to be large, P frames tend to be smaller, and B frames tend to be smaller yet. Some coders produce a PB frame by combining P and B frames in a single frame.

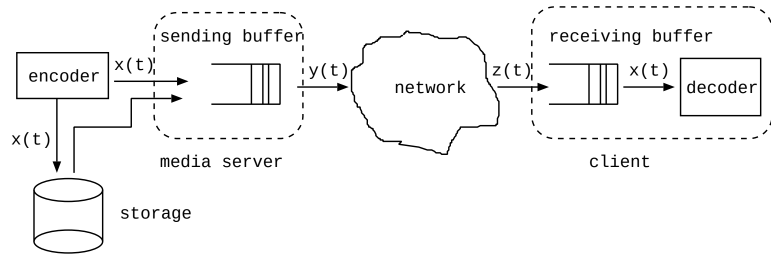


Figure 1. Logical View of Real-time Media Streaming

3.2. RealSystem media streaming architecture

RealSystem[5] is the Internet solution for audio and video streaming proposed by RealNetworks. They provide tools such as RealServer, RealPlayer, and RealProducer, corresponding to the media server, the client, and codec, respectively. The RealSystem architecture supports both real-time and on-demand streaming. We only study on-demand streaming in this paper.

The RealAudio and RealVideo contents are created in advance using RealProducer, and stored in RealMedia File Format (RMFF[12]) files. Before encoding, a target bit rate is chosen, based on the video quality desired for the intended audience. The bandwidth for the audio stream is allocated first, then that for the video stream. One way that RealVideo achieves compression is by skipping frames when needed, so as to achieve a high frame rate for action scenes, and a low frame rate for low-activity scenes.

The RealVideo encoding can be CBR or VBR. While in VBR mode, bits from low motion scenes are "stolen" to make high motion scenes better quality while keeping the average target bit rate unchanged. A lot of video clips on the Internet are coded using *SureStream*[13] technology, wherein multiple versions of the same video with different rates are generated in advance and saved in the same file. In this approach, the client negotiates the suitably-encoded version of the stream to retrieve from the server.

The media data are saved as media packets. Each media packet in the encoded file has a stream number, timestamp, and size, plus a flag indicating whether it belongs to a key frame. The flag helps the server make decisions about which media packets to transmit, retransmit, or skip when problems occur. The RMFF file header contains the target rate, indicating how fast each stream should be delivered.

A streaming session is managed using the Real-Time Streaming Protocol (RTSP[14]). Besides the general start, stop, pause, and fast forward control functions, RTSP is also used to change the delivery parameters of an ongoing streaming session (e.g., using `set_parameter`).

Before sending a media data packet, the RealServer encapsulates it using a *media packet protocol*. These protocols, such as the Real-time Transport Protocol (RTP[15]) or Real Data Transport (RDT[13]) protocol, facilitate the delivery and synchronization of real-time media data. The media packet is then carried by the transport-layer protocol. The default transport-layer protocol for audio/video streaming is UDP (User Datagram Protocol), though RealSystem supports TCP (Transmission Control Protocol) streaming as well (e.g., to traverse network firewalls).

To overcome network delay and delay variation, the RealPlayer buffers incoming data for a few seconds before it starts playing back the streams. If network conditions change during the playback, the RealSystem uses Adaptive Stream Management (ASM[13]) to manage the stream quality (e.g., to change the streaming rate, or to prioritize audio packets).

4. EXPERIMENTAL SETUP

4.1. Measurement facility

There are two machines and one wireless access point (AP) in our measurement setup, shown in Figure 2. The RealServer 8.0 software runs on a Linux machine with a 1.8 GHz Pentium 4 CPU. A laptop PC equipped with

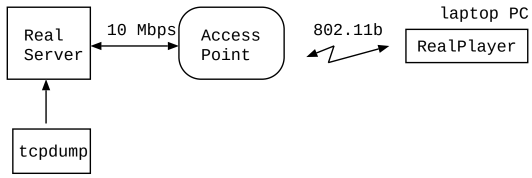


Figure 2. Experimental Setup for Media Streaming Measurements

a Cisco Aironet 350 network adapter was the RealPlayer 8.0 client. The AP is a Lucent RG-1000 residential gateway, connected to the server with a 10 Mbps Ethernet card. The AP uses infrastructure mode, and a maximum retransmission limit of 4 at the MAC layer.

The traffic characteristics at the network layer were studied using the `tcpdump` utility. For each streaming experiment, one trace is captured. Information such as packet size, type, and timestamp can be determined from these traces. For the application-layer traffic, since we study only on-demand RealMedia streaming, we study media packets stored in RMFF files, from which the coding bit rate (also the reading rate at the receiving buffer), the traffic patterns, and frame-level statistics can be determined. Since a UDP packet usually conveys a single media packet, we can easily study the interaction of the two layers.

Note that in the paper, two kinds of timestamps are mentioned: one is the (application-layer) *media timestamp*, the other is the (network-layer) *captured timestamp*. The media timestamp refers to the timeline information in the media packets for synchronization purposes. For example, a video packet with timestamp 100 is coordinated with an audio packet with timestamp 101. These media timestamps are the same every time the stream is downloaded, since they are recorded in the RMFF file. The second (captured) timestamp refers to the time at which a network packet is seen by `tcpdump`. These timestamps are different every time the stream is downloaded. In the rest of the paper, we always refer to the first type of timestamp as the media timestamp, and the second one simply as the timestamp.

4.2. RealMedia workload

Four compressed RMFF files with different levels of scene changes and activity were examined. They are a *Seminar* talk with relatively little motion, a *TV Program* with a rich combination of activity levels and scene changes, a *Movie* clip, and a *Rock Concert* clip. The *Seminar* clip was compressed with multiple target rates using the *SureStream* technology, while the other three clips were all compressed with a single target rate.

Table 1 summarizes the clip information read from the file headers. The clips range in duration from 1 minute to 2 hours. The maximum and minimum rates shown are the target rates for delivering. All four clips are CBR. Only one rate is shown for the *Seminar* clip.

Table 1. Summary Information for the RealMedia Clips Used

	Rock Concert		*TV Program*		*Movie*		*Seminar*	
Item	Audio	Video	Audio	Video	Audio	Video	Audio	Video
Total Packets	432	2,850	14,640	44,577	4,704	101,893	26,724	141,868
Minimum Packet Size (bytes)	320	547	640	842	640	794	304	524
Maximum Packet Size (bytes)	320	685	640	1,009	640	1,007	304	708
Average Rate (kbps)	16.1	184	44.1	180.9	44.1	180.9	8.5	71.5
Maximum Rate (kbps)	16.1	184	44.1	180.9	44.1	180.9	8.5	71.5
Duration (minutes)	1.1		28.3		48		128	
Target Frame Rate (frames/sec)	15		15		25		15	

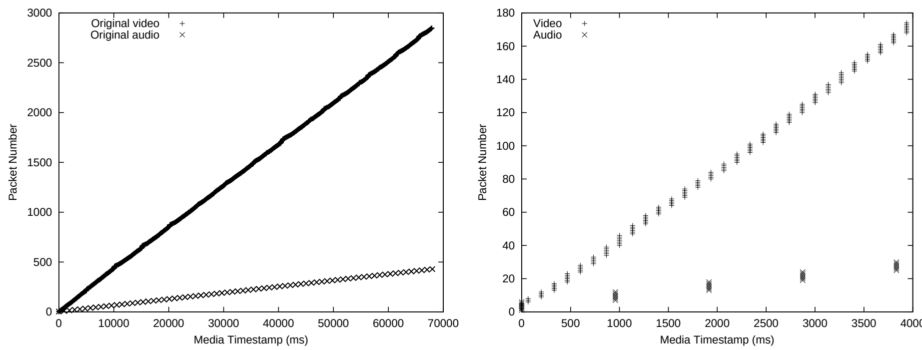
5. APPLICATION-LAYER TRAFFIC CHARACTERISTICS

In this section, we study the application-layer traffic characteristics of the RealMedia streams. We begin with characteristics at the media-packet level, followed by frame-level statistics. Finally, we briefly check the LRD of traffic at this layer.

5.1. Media-packet-level characteristics

5.1.1. RealAudio traffic

Figure 3 plots the media packet number versus media timestamp for the *Rock Concert* clip. In each of the two graphs, the upper line represents video packets while the lower line represents audio packets.



(a) Entire Clip (68 sec) (b) Initial Portion of Clip (4 sec)

Figure 3. Media Packet Number versus Media Timestamp (*Rock Concert*)

Figure 3(a) shows that the number of audio packets per second is roughly constant for the entire trace. However, Figure 3(b), a zoomed-in version of Figure 3(a), shows that audio packets are not evenly distributed in time. Instead, they appear as clusters of 6 packets with the same media timestamp, with time gaps in between clusters. This pattern indicates that the RealAudio codec generates audio traffic in "frames", a basic synchronization unit for audio and video. All packets with the same media timestamp belong to the same frame.

A histogram analysis (not shown here) of the inter-packet times of the *Rock Concert* clip shows that two peaks are present: one at 0 ms, and the other at 958 ms. The peak at 0 ms represents packets that are part of the same frame, as stated previously. The peak at 958 ms provides additional information about the traffic structure. Since there are 6 packets in an audio frame, each 320 bytes in size, the average data rate is

$$\frac{\text{6 pkts/frame x 320 bytes/pkt x 8 bits/byte}}{\text{0.958 sec/frame}} = \text{16.03 kbps (kilobits/sec)}$$

for each frame interval, yielding a constant bit rate overall. This calculation result is consistent with the audio data rate shown in Table 1.

Similar observations apply for the other clips, except that the audio inter-frame time period is 1,856 ms for the *TV Program* and *Movie* clips. The *Seminar* clip has multiple inter-frame times since it was compressed with multiple target rates. However, a dominant inter-frame time is seen if only a specific streaming rate is studied.

5.1.2. RealVideo traffic

The RealVideo packets in Figure 3 exhibit the same clustered structure as the audio packets, though with three noticeable differences. First, the encoded media packet rate is different, as reflected by the differing slopes for audio and video streams in the plots. Second, the video stream uses variable-size packets and a variable number of packets per frame. Third, the inter-frame timing structure in the video stream is more subtle. Although the histogram of inter-packet times for each clip shows two dominant peaks, there are several other interval occurrences. The peak at 0 ms again reflects packets belonging to the same frame. The other values, however, indicate that a RealVideo codec generates a variable-frame-rate video stream.

A more careful examination of the video traces reveals that the time gaps between video frames are either a multiple of the reference frame interval (i.e., the inverse of the target frame rate), or only a few milliseconds apart. Figure 4 illustrates this by plotting media packet size versus media timestamp for a 400 ms portion of the *Movie* stream. As shown in Figure 4, there are 4 media packets at time 640 ms, 1 packet at time 641 ms, 2 packets at time 642 ms, and 1 packet at time 643 ms, for a total of 8 media packets, constituting 4 frames, each with an inter-frame time of 1 ms. Other video packets are all about 160 ms away from these packets. The 160 ms gap is related to the frame rate mechanism of the codec. The 1 ms gap, however, cannot be attributed to the frame skipping algorithm. While we do not know whether RealVideo uses PB frames or not, our guess is that this behaviour is related to the B frames: because a B frame needs two reference frames, it has to be encoded/decoded after the backward reference frame. The RealVideo codec seems to generate B frames and its backward reference P frame together for convenience of decoding. The codec decides whether or not to use a B frame and how many B frames, depending on the content of the video. This hypothesis explains why sometimes we see a cluster of frames, and sometimes we do not.

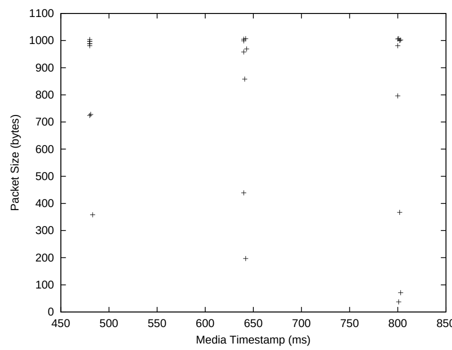


Figure 4. Illustration of Video Inter-Frame Time Structure (*Movie*)

Looking at the example in Figure 4 again, we know that there are 4 frames near 640 ms. By summing the media packet sizes observed at each media timestamp value, we can calculate the frame sizes, which are 3401 bytes, 858 bytes, 1204 bytes, and 969 bytes, respectively. The pattern of sizes suggests that these are P, B, B, and B frames. The next cluster starts 160 ms later with frame sizes 2783, 1044, 1367, and 1073 bytes, suggesting the same frame structure. The first cluster has a span of four reference frame intervals (the target frame rate is 25 fps for this clip, yielding a reference frame interval of 40 ms.). Other clustered frames have similar patterns. These findings lend credence to our previous hypothesis about the RealVideo structure.

5.1.3. Implications of audio and video multiplexing

The multiplexing of RealVideo and RealAudio with the above patterns provides an explanation for the network-layer observations by Mena *et al.*,[7] who found that RealAudio traffic has a consistent bit rate at medium time scales and bursty on-off patterns at small time scales. This is because after sending a constant number of audio packets back-to-back, the encoder stops sending audio packets and starts sending video packets. When this process repeats periodically, the audio traffic exhibits an on-off pattern.

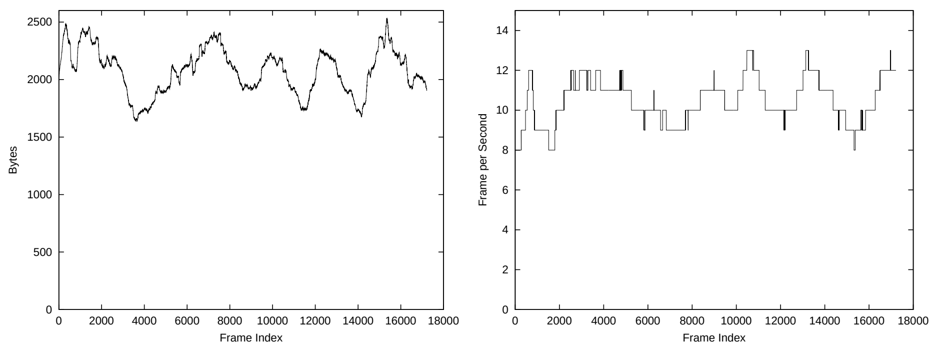
The foregoing explanation also suggests that the server may send video packets in a similar fashion at the network layer: trains of video packets, separated by audio packets. We will investigate this further in Section 6. Another implication is that RealVideo traffic at the application layer actually has two levels of burstiness: the frames are clustered in small groups, and the packets are clustered within each frame. If a server sends video packets according to the media timestamps, the video traffic could be very bursty.

5.2. Frame-level statistics

Frame sizes were calculated by summing all packets with the same media timestamp. A statistical summary of frame size characteristics appears in Table 2. All clips have a mean frame size that is larger than the median

Table 2. Frame-Level Statistics for Media Clips

Item	*Rock Concert*	*TV Program*	*Movie*	*Seminar*
Total Frames	511	18,230	9,661	77,442
Minimum Frame Size (bytes)	895	54	117	56
Median Frame Size (bytes)	2,930	1,684	1,041	864
Maximum Frame Size (bytes)	13,573	21,388	15,218	13,875
Mean Frame Size (bytes)	3,053	2,060	1,269	875
Standard Deviation (bytes)	1,449	1,915	849	753



(a) Frame Size versus Time (b) Frame Rate versus Time

Figure 5. Illustration of Inverse Relationship Between Frame Size and Frame Rate (*TV Program*)

frame size, suggesting an upper tail to the distribution. This "heavy tail" property is most pronounced for the *TV Program* clip.

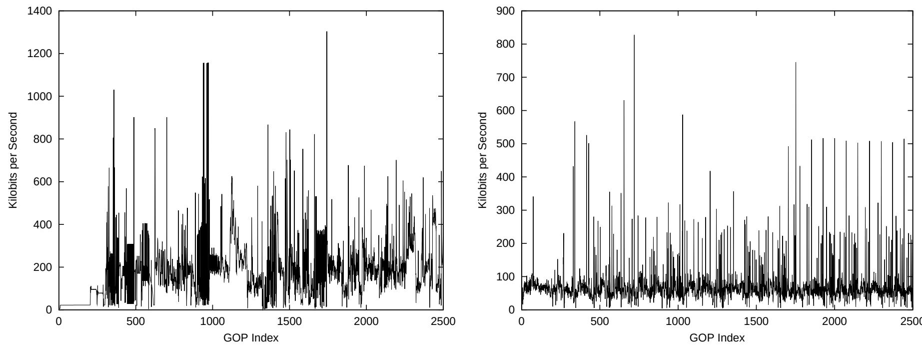
The marginal distribution of the frame size is of limited interest here, however, since the inter-frame time varies. Recall that RealVideo occasionally skips frames to constrain the total bit rate to the target bit rate when the frame sizes are large. Figure 5 illustrates this effect by plotting the smoothed (window size 1000) frame size and frame rate for the *TV Program* clip as a function of time. A simple study like this of frame size versus frame index may exaggerate the burstiness of the traffic, if the inter-frame time is not considered. A rate trace avoids this problem.

The rate trace is calculated by dividing the frame size by its inter-frame time span. Moreover, the cluster of frames reported earlier likely represents the unit that the codec handles, so we add together the sizes of the frames within a cluster, and divide this sum by the time span between clusters. With a slight abuse of MPEG video compression terms, we call this a *Group of Pictures (GOP)* interval. Figure 6 shows the traffic profile calculated in this way for the *TV Program* and the *Seminar*. They both have some bursty peaks. However, the *TV Program* seems to have some slowly varying components (i.e., scenes) as well.

To see the slowly varying components of different clips clearer, a moving average is applied to each rate trace. The smoothing window sizes for all clips are 500, except for the *Rock Concert*, which uses a window size of 100 (since it has fewer data points). The results are shown in Figure 7. One can easily tell that the *Movie* and *Seminar* clips are roughly CBR, while the *TV Program* still shows considerable variation, suggesting a VBR-coded stream. (The smoothed plot of the *Rock Concert* clip is not shown here because it is too short. When plotted on its own, its bit rate is CBR too.)

5.3. Long-range dependence

The VBR traffic raises the question of the existence of LRD, which is briefly discussed in this section. Figure 8 shows the autocorrelation functions of all rate traces, each with a maximum lag that is meaningful to its trace length. The autocorrelation function of the *TV Program* does not decrease to zero even after a long lag. For



(a) *TV Program* (b) *Seminar*

Figure 6. Application-Layer (Codec) Traffic Profile for Two Media Clips

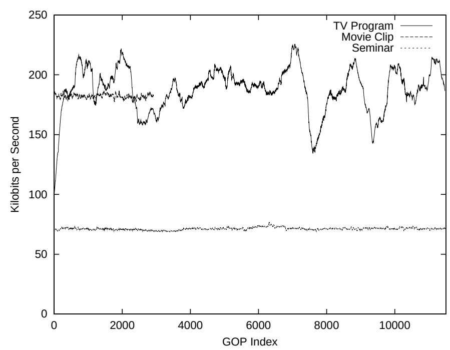
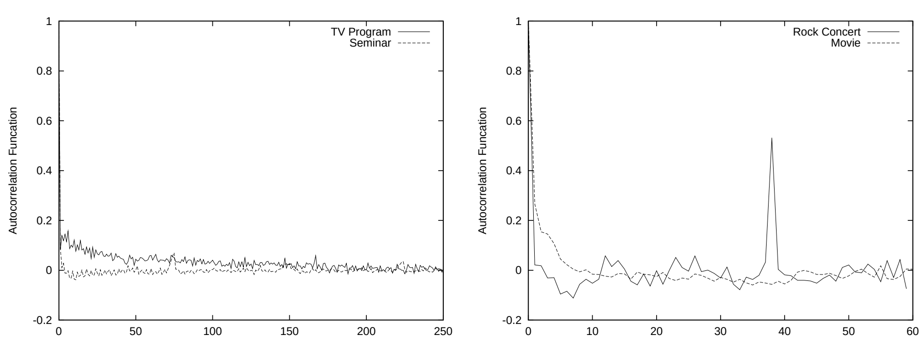


Figure 7. Smoothed Codec Traffic Rate for Selected Media Clips

Rock Concert, while the autocorrelation is close to zero, it has a periodic nonzero component. This corresponds to the key frames every 5000 ms (about 40 frames). Whether or not there is LRD in these traces needs further study. The autocorrelation function for the *Seminar* clip seems to drop to zero rapidly. The autocorrelation function for *Movie* decays faster than the one for *TV Program*, but slower than those for the other two clips.

The LRD properties of the traces were further studied using a wavelet-based method described by Abry and Veitch.[16] The energy versus transform level plot is shown in Figure 9. The *TV Program* has a linear part



(a) *TV Program* (b) *Rock Concert*

Figure 8. Frame-Level Autocorrelation Structure of Media Clips

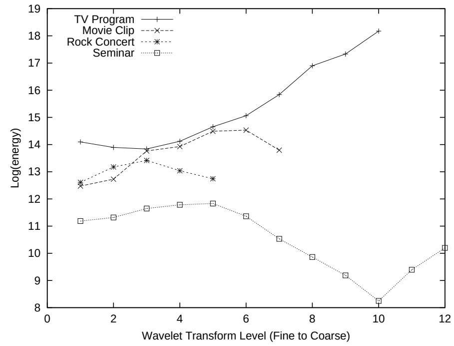


Figure 9. Testing for LRD in Frame-Level Structure of Media Clips Using Wavelet Transform

including the largest transform level, clearly indicating an area of LRD. The *Rock Concert* and *Seminar* do not have a linear area that includes the largest transform level, indicating no LRD. The *Movie* has a rough linear area, but it does not include the largest transform level. It is unclear whether LRD exists here or not.

A minimum squared error estimate of the slope gives an estimated Hurst parameter of 0.75 for the *TV Program* clip (calculated from level 3 to 10). It thus implies that this VBR-coded stream exhibits LRD, while some CBR-encoded streams (e.g., *Rock Concert*, *Seminar*) do not have LRD, and some CBR-encoded streams (e.g., *Movie*) may have weak LRD components. These results are consistent with the findings of Fitzek,[11] who reports that most CBR-coded variable frame rate H.263 streams do not exhibit LRD. However, they also found a few CBR traces that have weak LRD.

One possible explanation is that the LRD primarily represents the intrinsic nature of video. If an encoder is running without constraint (i.e., to generate a pure VBR trace), its output will reflect the underlying characteristics of the video. However, for CBR video, the encoder is forced to discard some bits, even frames, thus breaking the underlying video features. At the same time, some small variation LRD components may survive after this process. That is why we can see some weak LRD in the trace. However, more solid evidence is needed to support this argument. We are currently investigating this point.

6. NETWORK-LAYER TRAFFIC CHARACTERISTICS

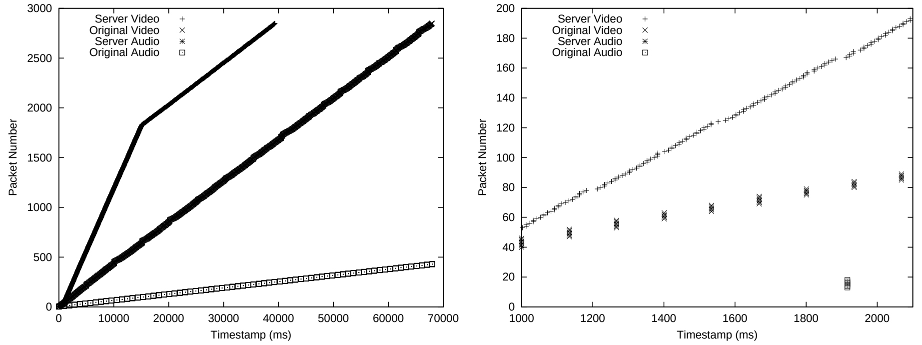
In this section, we study the measured traffic at the network layer (i.e., $y(t)$ in Figure 1), when UDP is used as the transport-layer protocol. We focus on the relationship between application-layer and network-layer traffic.

6.1. Network-layer traffic patterns

Figure 10 shows the packet number versus the captured (network-layer) timestamp for the *Rock Concert* clip. Figure 10(a) shows the entire trace, while Figure 10(b) shows a 1 second portion of the trace. The most obvious observation is that network packets are sent at a consistent rate. The higher rate at the beginning is due to the client's request to fill up the receiving buffer quickly. After the buffer is filled, the streaming rate is reduced to the target rate. Other clips have similar streaming patterns. In particular, the *TV Program* was streamed as CBR, although it was compressed as a VBR video.

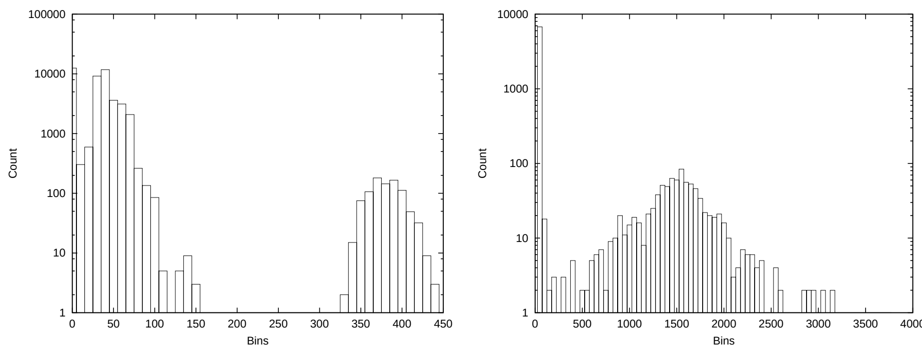
Figure 10(b) shows a zoomed-in version of the *Rock Concert* traffic captured at the server from 1000 ms to 2100 ms. Audio packets on the network exhibit a bursty pattern at small time scales. The graph also confirms our prediction that video packets also have an on-off bursty pattern at small time scales, reflecting the application-layer traffic pattern at the network layer.

Figure 11 shows the histograms of the inter-packet times of video (Figure 11(a)) and audio (Figure 11(b)) streams. (Note that the y-axis is log scale). We use the *TV Program* clip in this case because its length reduces the effect of the start-up streaming rate on the results. Examining the graph, three clusters of inter-packet times are evident: one at 0 ms, one near 30 ms, and one around 380 ms. We seek to explain each of these peaks.



(a) Entire Clip (b) Initial Portion of Clip (1 sec)

Figure 10. Network-Layer Structural Characteristics of RealServer Streaming Data (*Rock Concert*)



(a) Video Stream (b) Audio Stream

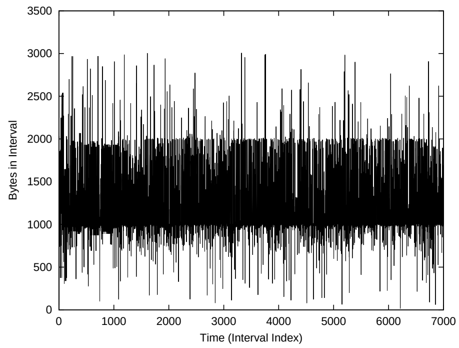
Figure 11. Histogram of Inter-Packet Time (*TV Program*)

The peak at 0 ms is an artifact of our measurement methodology. Since `tcpdump` is run on the same machine hosting RealServer, the captured network-layer packets are actually in the network device driver queue, *before* being sent to the network interface. This is why we see inter-departure times of 0 ms.

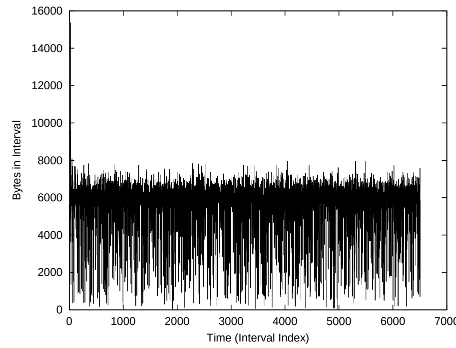
Next, we consider the peak at 30 ms. It cannot be caused by system call overhead since the time scale is too large. Rather, given that the total streaming bit rate is 225 kbps and the mean video packet size is 842 bytes, the time needed to send an average-sized packet is about 30 ms (842*8/225,000). So this peak simply indicates that RealServer sends packets into the socket buffer back-to-back. There are also many inter-packet times in the 40-100 ms range. In fact, the counts here are roughly equal to those in the 0 ms bin. Since the maximum packet size of 1008 bytes takes at most 1008*8/225,000=36 ms to send, the 40-100 ms inter-packet times indicate that RealServer must send multiple packets (two or more) to the socket at a time (probably to reduce socket system call overhead). The 40-100 ms time represents the time RealServer waits for the packets to drain, before doing the next send.

We suspect that the peak around 380 ms represents the time interval that RealServer stops sending video packets while it is processing audio data. The variations around the peaks are caused by three factors: RealVideo packets are variable-size; RealServer sometimes sends more than one packet to the socket at a time; and the processing time within the operating system kernel varies.

Similarly, Figure 11(b) shows that the audio stream has three cluster peaks at 0 ms, 40 ms, and 1550 ms. Recall that all audio packets are 640 bytes in size, leading to a per-packet time of 640*8/225,000 = 23 ms. The peak near 40 ms occurs because RealServer writes two audio packets into the socket and then waits for twice the regular inter-packet time. In every frame there are 16 audio packets, so the total time is 640*16*8/225,000=364



(a) Aggregated within 20 ms



(b) Aggregated within 200 ms

Figure 12. Network-Layer Traffic Profile Emanating from RealServer (*Rock Concert*)

ms. This value is consistent with the time that the RealServer stops sending video data (see the discussion of the 380 ms peak in the previous paragraph).

From the foregoing discussion, a hierarchical model could be derived for streamed RealAudio and RealVideo traffic: at large time scales (minutes), the overall bit rate is constant; at intermediate time scales (seconds), on-off patterns represent the interleaving of audio and video data; and at fine-grain time scales (sub-second), back-to-back packet trains separated by time gaps represent the "packet batching" behaviour of the server to reduce system call overhead.

6.2. Long-range dependence

We have shown that RealServer changes the traffic characteristics dramatically if the video is VBR-coded. The stream is sent as CBR overall no matter how it is coded initially. So even if traffic from an encoder has LRD, it may be changed by the server streaming policy. We study the LRD at network layer in this section. Since at this level video frames are not of importance, we count the number of bytes in a specified interval of time. Since we know that the VBR-compressed *TV Program* clip exhibits LRD, we focus solely on this clip.

Figure 12 shows the resulting network traffic profile for the *TV Program* clip. Figure 12(a) shows the number of bytes per 20 ms interval, while Figure 12(b) shows the number of bytes per 200 ms interval. The thick dark horizontal bands in each of these plots show that (as expected) they resemble a CBR stream. Using the wavelet-based method, no linear area including the highest transform level is identified in the transformed domain. So even though there are some local irregularities, they are smoothed out in the aggregation process.

7. CONCLUSIONS

In this paper, we conducted a measurement study of RealMedia streaming traffic. Four different RealMedia clips were used in the study, with durations ranging from 1 minute to 2 hours. Traffic data were collected using `tcpdump` near the RealMedia server. Our workload characterization study focuses on both the application-layer view (i.e., the output of the audio/video encoder) and the network-layer view (i.e., the departure process for network packets emanating from the RealMedia server). In addition, the relationship between the two layers of traffic is investigated.

There are three main results from our study. First, the RealAudio codec generates pseudo-CBR traffic, with burstiness at small time scales. The application-layer traffic characteristics are preserved at the network layer. Second, RealVideo comes in two forms: VBR and CBR. In both cases, the burstiness at fine-grain time scales is similar to that in RealAudio traffic. In fact, the video traffic has two distinct types of burstiness: groups of frames are generated together, and packets from the same frame are grouped together. Consistent with previous results in the literature, there are strong LRD components found in the VBR-compressed video clips. However, there is little or no LRD in CBR video. Third, when compressed media is streamed by RealServer, it is typically streamed as Constant-Bit-Rate (CBR). Both the streamed audio and video have a three-tiered structure. At large time scales (minutes), the overall bit rate is constant. At medium time scales (seconds), the packet pattern is bursty due to the interleaving of audio and video. At fine-grain time scales (sub-second), back-to-back packet trains of two or more packets are often seen due to "packet batching" by the server. The RealServer changes the VBR traffic dynamics of the application dramatically. VBR-compressed video loses its LRD because of an overall constant bit rate, despite some local irregularities.

Our next step is to model this popular multimedia workload for network simulation purposes.

ACKNOWLEDGMENTS

Financial support for this research was provided by iCORE (Informatics Circle of Research Excellence) in the Province of Alberta, and by the Natural Sciences and Engineering Research Council of Canada, through NSERC Research Grant OGP0121969.

REFERENCES

1. M. Chesire, A. Wolman, G. Voelker, and H. Levy, "Measurement and analysis of a streaming-media workload," *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, March 2001.
2. A. Wolman, G. Voelker, N. Sharma, N. Cardwell, M. Brown, T. Landray, D. Pinnel, A. Karlin, and H. Levy, "Organization-based analysis of web-object sharing and caching," *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, October 1999.
3. J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Trans. Commun.* **43**, pp. 1566–1579, February/March/April 1995.
4. M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," *Proceedings of ACM SIGCOMM'94*, September 1994.
5. RealNetworks, "RealSystem production guide: RealSystem release 8 with with RealProducer 8.5," **http://service.real.com/help/library/index.html**.
6. ANSI/IEEE, "Standard 802.11b - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specificiations: Higher-speed physical layer extension in the 2.4 GHz band," September 1999.
7. A. Mena and H. Heidemann, "An empirical study of RealAudio traffic," *Proceedings of IEEE INFOCOM*, pp. 101–110, March 2000.
8. Y. Wang, M. Claypool, and Z. Zuo, "An empirical study of RealVideo performance across the Internet," *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, pp. 295–309, November 2001.
9. D. Loguinov and H. Radha, "Measurement study of low-bit-rate Internet video streaming," *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, pp. 281–293, November 2001.
10. M. Krunz and S. Tripathi, "On characterization of VBR MPEG streams," *Proceedings of ACM SIGMETRICS*, pp. 192–202, May 1997.
11. F. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network* **15**, pp. 40–54, November/December 2001.
12. R. Agarwal, J. Ayars, B. Hefta-Gaub, and D. Stammen, "Internet draft draft-heftagaub-rmff-00.txt: Realmedia file format," March 1998.
13. RealNetworks, "Working with RealProducer 8 codecs," **http://service.real.com/help/library/index.html**, May 2000.
14. RFC2326, "RFC2326: Real time streaming protocol (RTSP)," **http://www.ietf.org/rfc/rfc2326.txt**, April 1998.
15. RFC1889, "RFC1889: A transport protocol for real-time applications," **http://www.ietf.org/rfc/rfc1889.txt**, January 1996.
16. P. Abry and D. Veitch, "Wavelet analysis of long range dependent traffic," *IEEE Transactions on Information Theory* **4**, pp. 2–15, January 1998.