RealTracer - Tools for Measuring the Performance of RealVideo on the Internet

Yubing Wang
EMC Corporation
Hopkinton, MA 01748, USA
wang_yubing@emc.com

Mark Claypool
Computer Science Department
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609, USA
claypool@cs.wpi.edu

ABSTRACT - The increase in high-bandwidth connections and high-speed computers have spurred the growth of streaming media across the Internet. While there have been a number of studies measuring the performance of traditional Internet traffic, there has not been sufficient wide-scale empirical measurement of video performance across the Internet. The lack of empirical work that measures streaming video traffic may arise from the lack of effective video performance measurement tools. In this paper, we present *RealTracer*, a set of tools for measuring the performance of RealVideo. RealTracer includes *Real-Tracker*, a customized video player that plays RealVideo from a pre-selected playlist and records user-centric video performance information. RealTracer also includes *RealData*, a data analysis tool that helps manage, parse and analyze data captured by RealTracker. We describe the software architecture and usage of RealTracker and the usage of RealData, both publicly available for download. To illustrate the use of RealTracer, we present additional results from a previous study that used RealTracker to measure RealVideo performance across the Internet. Using RealData, that study made several contributions to better understanding the performance of streaming video on the Internet.

1. INTRODUCTION

The increase in high-bandwidth connections and high-speed computers have spurred the growth of streaming media across the Internet. Web sites, traditionally text and graphics only, are increasingly offering streaming videos such as news clips, concerts, taped presentations and sporting events. Applications to access popular streaming media, such as RealNetworks RealOne, Microsoft Windows Media Player or Apple Quicktime, are freely available for most operating systems platforms, providing the potential for nearly every Internet end-host to play streaming video.

The impact of streaming video on the Internet will be largely impacted by the role of commercial streaming media products, whose use has increased dramatically [Jup01]. RealPlayer is installed on over 90% of home PCs, Quicktime claims more than 100 million copies distributed world-wide, and Media Player currently provides 220 million players [CN01]. RealPlayer, provided by RealNetworks[1], is the most popular streaming media player on the US Internet, with over 45% of the commercial market share and an increase of over 50% from the previous year [Jup01].

Over the years, there have been a number of studies measuring the performance of Internet backbones and end-hosts [TMW97, Pax99], as well as detailed studies on the performance of Web clients [KW00, Mah97]. However, there has not been sufficient wide-scale empirical measurement of video performance across the Internet. While the existing studies have been valuable in helping understand Internet performance, they are not sufficient for characterizing streaming video performance since video has application requirements that are different than the majority of Internet traffic.

[1] http://www.real.com

Unlike typical Internet traffic, streaming video is sensitive to delay and jitter, but can tolerate some data loss. In addition, streaming video typically prefers a steady data rate rather than the bursty data rate often associated with window-based network protocols. Recent research has proposed rate-based TCP-Friendly protocols in the hope that streaming media applications will use them [RHE99, FHPW00], but such protocols are not yet widely part of any operating system distribution. For these reasons, streaming video applications often use UDP as a transport protocol rather than TCP. Thus, previous Internet-wide studies that have captured primarily TCP data do not necessarily reflect the traffic characteristics of streaming media.

The lack of empirical work that measures streaming video traffic may largely stem from the lack of effective video performance measurement tools. Capturing streaming media traffic in general is challenging because streaming video applications use a variety of protocols. Moreover, while there have been efforts to develop common standardized protocols, many commercial applications continue to use proprietary protocols.

Internet traffic is commonly measured using the tcpdump[2] utility that can be used to filter packets from a particular application based on protocol and port number. However, the port number for many streaming media servers is not fixed, but is instead negotiated during the initial connection using protocols such as RTSP [RFC 2326]. Extensions to tcpdump [CWVL01, MCCS00] can parse tcpdump data to determine the ephemeral port for a particular session. However, none of these tools have application level information, such as frame rate, data encoding rate, etc. Instead, application level performance must be inferred from network trace information, as in [MH00], adding inaccuracy to application level measures of performance.

Commercial media players, including RealNetworks RealOne and Microsoft Windows Media Player, provide a means of observing performance statistics, such as average bandwidth and encoding rate. However, the same commercial players include no mechanisms for recording data or controlling playout in an automated fashion. In addition, there are often additional performance statistics that can help analyze streaming video performance that are unavailable via the player itself, but can be accessed via an API to the player core or via additional system level measurements.

In this work, we present *RealTracer*, a set of tools for measuring the performance of RealVideo. RealTracer includes *Real-Tracker*, a customized video player that plays RealVideo from a pre-selected playlist. For each video played, RealTracker records user-centric video performance information, including frame rate, jitter and user ratings, and can either send the performance information by email or FTP to a server or save it locally to disk. RealTracer also includes *RealData*, a data analysis tool that helps manage, parse and analyze data captured by RealTracker.

Researchers and practitioners alike can use RealTracer in at least two ways to evaluate RealVideo performance: (1) users can run RealTracker, saving the data locally, and then RealData to analyze RealVideo performance from their own PCs; or (2) users can generate custom playlists and configuration files that have data sent to a central server, and distribute RealTracker to a group of distributed users, allowing analysis RealVideo across a geographically diverse set of PCs.

In this paper, we describe the software architecture and usage of RealTracker and the usage of RealData. To illustrate the use of RealTracer, we present additional results from a previous study that used RealTracker

[2] http://www.tcpdump.org/

to measure RealVideo performance across the Internet from geographically diverse clients to geographically diverse servers. Using RealData, that study made several contributions to better understanding the performance and impact of streaming video on the Internet. In this paper, we illustrate the use of RealData by using data from that study and providing new analysis on the effects of locality of access, such as is provided by Content Distribution Networks, on RealVideo performance.

The rest of this paper is arranged as follows: Section 2 provides background needed to help understand the RealTracer tools and performance results presented here; Section 3 describes the RealTracer tools in detail; Section 4 presents some results we obtained in a wide-area study using the RealTracer tools; Section 5 introduces some related work; Section 6 summarizes our conclusions and presents possible future work.

2. RealVideo BACKGROUND

RealNetworks provides the most popular streaming media player, called *RealPlayer* (the latest version is called RealOne[3]), on the U.S. Internet. In January 2001, 25.9 million U.S. Internet users at home used a RealNetworks player, up 47.6 percent from January 2000; 21.5 million used Microsoft Windows Media Player, up 31.2 percent; and 7.3 million used Apple QuickTime, down 8.4 percent [Jup01]. RealNetworks also has the largest share of Internet users at work: in January 2001, 10.5 million U.S. Internet users at work used a RealNetworks player, up 52.1 percent from 2000; 9.0 million used Media Player, up 39.9 percent from 2000; and 1.9 million used QuickTime, up 8.5 percent from 2000.

2.1 Connections and Protocols

RealServer primarily uses Real Time Streaming Protocol (RTSP) [RFC 2326] for session control and supports the RTP standard [RFC 1889] for framing and transporting data packets. Earlier versions of RealServer used the Progressive Networks Audio (PNA) protocol and, for backward compatibility, newer real servers and players still support this protocol.

RealServer uses two network connections to communicate with RealPlayer clients: one for communicating control information with the client, and one for communicating the actual data. RealServer uses the control connection to request client configuration parameters and to send information such as clip titles, and clients use the control connection to send instructions such as fast-forward, pause, and stop. The video clips themselves, on the other hand, are actually streamed over the data connection.

At the transport layer, RealServer uses both TCP and UDP for sending data. The initial connection is often in UDP, with control information then being sent along a two-way TCP connection. The video data itself is sent using either TCP or UDP. The actual choice of transport protocol used is automatically determined by the RealPlayer and Real-Server. This auto-configuration of protocols can be overridden by the user, but is the default and recommended setting for RealPlayer [Rea00b].

2.2 Buffering

For each video clip, RealPlayer keeps a buffer to smooth out the video stream in case of changes in bandwidth, lost packets or variance in packet arrival rates (jitter). Data enters the buffer as it streams to RealPlayer, and leaves the buffer as RealPlayer plays the video clip. If network congestion reduces bandwidth for a few seconds, for example, RealPlayer can keep the clip playing with the buffered data. If the buffer empties completely, RealPlayer halts the clip playback for up to 15 seconds while the buffer is filled again.

2.3 Bandwidth Characteristics

RealSystem uses a technology called *SureStream* in which one RealVideo clip is

[3] http://home.real.com/

encoded for multiple bandwidths [Rea00a]. A RealPlayer connects to a single video URL and the RealServer determines which stream to use based on the RealPlayer's specified minimum and average bandwidths. The initial playback data rate of the video stream is based on the maximum client bit rate (a RealPlayer configuration parameter) and other video settings. The actual video stream served can be varied in mid-playout, with the server switching to a lower bandwidth stream during network congestion and then back to a higher bandwidth stream when congestion clears. Unlike previous versions of RealPlayer, RealOne does not need to rebuffer data during this switching. If packets are lost during video delivery, special packets that correct errors are sent to reconstruct the lost data.

A portion of a RealVideo clip's bandwidth first goes toward the audio, leaving the remainder of the bandwidth for the video. For example, a 20 Kbps RealVideo clip (typical for a 28.8 modem) with a 5 Kbps RealAudio voice codec will leave 15 Kbps for the video, while an 11 Kbps music codec will leave only 9 Kbps for the video. The RealOne player also integrates streaming media with HTML pages to support Web browsing, which may consume a portion of a RealVideo clip's bandwidth with HTML page downloads.

Most RealVideo clips are created with a *Scalable Video Technology* option that allows RealServer to automatically adjust the video stream according to the client's connection and computer processing speed [Rea00a]. If the client machine is unable to play the clip at the encoded frame rate, the frame rate will gradually reduce in a controlled fashion to maintain smooth video.

2.4 Adaptive Stream Management

In addition to the *SureStream* technology described above, RealSystem 8 uses a protocol called Adaptive Stream Management (ASM) [CGLLR01] in order to adapt to the actual bandwidth and loss on the channel. ASM uses a set of rules to describe various means of channel adaptations, which are stored in the compressed media as the ASM rulebook. During the initial phase of communication, the ASM rulebook is transferred to the client. In turn, the client collects the information about the channel, parses the ASM rulebook, and sends the server a request to subscribe to a rule or combination of rules that match the current statistics of the channel. When the server receives the request, it passes it to the file format plug-in, which in turn begins to stream data according to the rule request.

3. RealTracer Tools

RealTracer includes *RealTracker*, a customized player that can play streaming RealVideo clips and record system performance statistics as well as user ratings, along with *RealData*, a tool that helps analyze the statistical data collected by RealTracker. RealTracker and RealData are standalone programs that can run independently on Windows 98/95, NT and 2000.

With RealTracker, users can create their own *playlist* and specify the means of gathering statistics. RealTracker plays videos on the playlist, gathering system and RealVideo performance statistics.

With RealData, users can easily analyze the RealVideo statistics collected by RealTracker. RealData can sort the Real-Tracker statistics by various performance metrics, such as bandwidth, frame rate, jitter etc. RealData provides a tool that can generate cumulative density data for measured bandwidth, frame rate and jitter. RealData can also export the statistical data into a comma-delimited file suitable for use in most spreadsheet programs, such as Microsoft Excel, for further analysis.

Section 3.1 describes the design, implementation and operation of

RealTracker and Section 3.2 describes the operation of RealData.

3.1 Design, Implementation and Operation of RealTracker

RealTracker is a customized RealVideo player with a customized front-end interface to gather user information and a customized back-end to record performance statistics. Section 3.1.1 describes the structure of RealTracker and section 3.1.2 shows the operation of RealTracker for conducting RealVideo performance studies.

3.1.1 Design and Implementation of RealTracker

Before describing the design of RealTracker, we briefly introduce the structure of a RealSystem client. A RealSystem client such as RealPlayer consists of two major parts, the *top-level client* and the *client core*. The core handles data transport with RealServer and provides RealSystem features such as delay buffering and repairing. The top-level supplies the user interface and communicates with the core and other components, such as rendering plug-ins, through RealSystem interfaces. The RealSystem Software Development Kit[4] provides the interfaces required to link to the client core.

RealTracker has three layers in terms of software structure, as illustrated by Figure 3.1. The first layer includes four modules that provides the RealTracker user interface, the `user information collection` module records the user information entered by the user and the system information detected by RealTracker; the `RealVideo statistics collection` module collects and displays RealVideo statistics through the interfaces provided by layer two; the `data delivery` module delivers the statistics to the locations specified by the user via either email or ftp; the `RealVideo display` module plays out the streamed video by calling the layer two interface.

The second and third layers together form a RealSystem client where the second layer represents the top-level client and the third layer corresponds to the client core. The second layer acts as a bridge between the first layer and the third layer. It includes two RealSystem interfaces: `IRMAPlayer` and `IRMAClientAdviseSink`. The former lets the top-level client control the presentation playback by beginning, pausing, stopping, or seeking in a presentation timeline. Through this interface, the RealSystem client can also gain access to the client engine, stream objects, and stream source objects. The latter lets the top-level client receive notifications from the client core about changes in a presentation's playback status and the corresponding RealVideo statistics. The second layer of RealTracker also includes a module called *Jitter Analysis*, which measures the frame-level jitter, a statistic that is not provided by `IRMAClient-AdviseSink`. The third layer is the RealVideo core that provides the foundation of RealTracker. Since the second layer is the most important part of RealTracker, in the following paragraphs, we describe the three modules within this layer in detail.

To create a new `IRMAPlayer` interface, a client engine object is first created. On start-up, the top-level client loads the client core (`rmacore.dll`) and calls that library's `CreateEngine` method to return a pointer to the client engine object. The top-level client can then use `IRMAClientEngine` to utilize the client core and create a player object by calling the `CreatePlayer` method. On shutdown, the top-level client calls the C-style function `CloseEngine` to close the client engine.

[4] http://www.realnetworks.com/resources/sdk/index.html

To add a new `IRMAClientAdviseSink` interface, the top-level client registers for a specific `IRMAPlayer` object through that object's `AddAdviseSink()` method. Through the method `OnStatisticsChanged()` provided by this interface, the top-level client can receive notifications from the client core about changes in the presentation statistics. In this method, the top-level client can further retrieve the statistics of both client and server by accessing resources in the RealServer Property Registry, which is a dynamic repository for a variety of server and client properties maintained by RealServer. Among the available properties, RealTracker focuses on a comprehensive set of statistics for the clients such as the average bandwidth, average latency, current frame rate, etc.

There are no entries in the RealServer Property Registry for jitter and frame inter-arrival time. Therefore there are no means for RealTracker to retrieve these statistics through `IRMAClientAdviseSink` interface or other RealSystem interfaces. Instead, RealTracker measures jitter as the standard deviation of inter-frame playback times. Previous study [WCZ01] shows that the result of frame-level jitter is consistent with our previous measurements of packet-level jitter for different end-host network bandwidths [CR99]. And furthermore, to the user, the frame-level jitter is more perceptible than the packet-level jitter, and hence more closely related to the quality of RealVideo as perceived by the user. To measure the frame-level jitter, RealTracker catches the frame playback time through the `ForceRedraw()` method provided by the `IRMASite` interface. This method provides a callback mechanism that allows RealTracker to measure the inter-frame playback time that, in turn, is used to calculate the frame-level jitter.

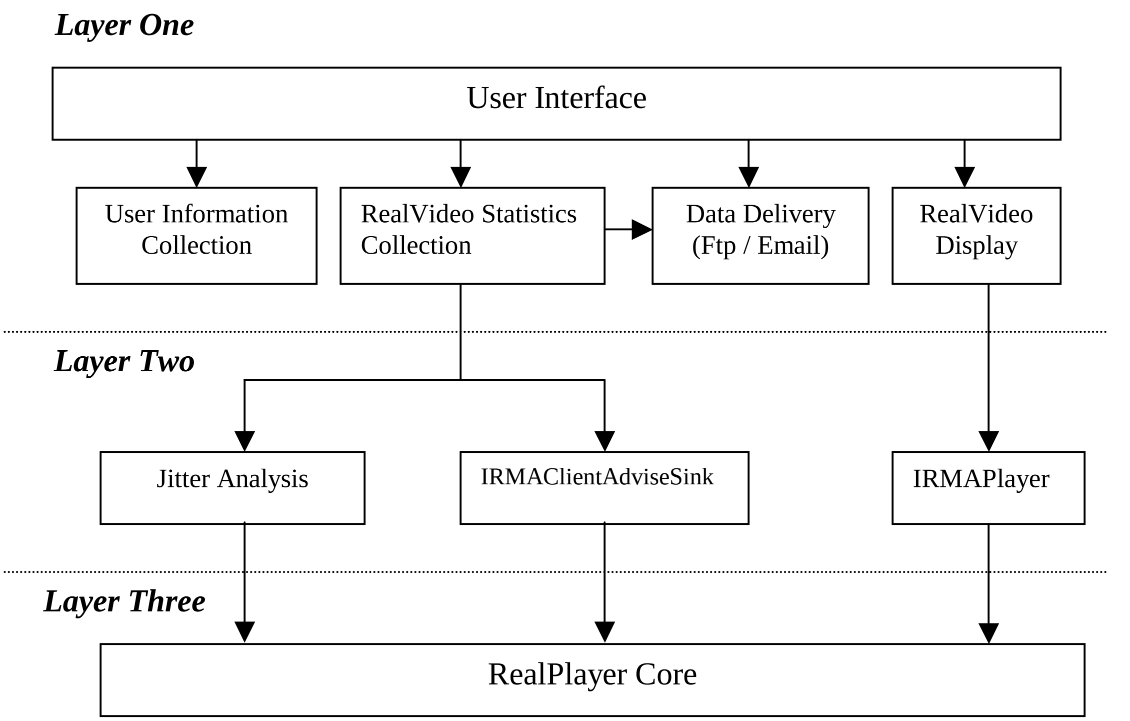


Figure 3.1 RealTracker Structure

3.1.2 Operation of RealTracker

There are two targeted users of RealTracker. The first are users who want to conduct RealVideo performance studies by distributing RealTracker with specific configuration files and playlists as in [WCZ01], where RealTracker automatically delivers the RealVideo performance data back to the distributor (see Section 3.1.2.1). The second are users who collect RealVideo statistics by using RealTracker to display the RealVideo clips on the playlist created by the distributor or by themselves (see Section 3.1.2.2).

3.1.2.1 For the Distributor

To plan a RealVideo performance study using RealTracker, the distributor needs to create a playlist and a configuration file. The playlist is a text file that contains a list of RealVideo clips to be played by each user. RealTracker randomizes the order of clips in the playlist. An example of the playlist file is shown as follows:

```
US - Space 01
rtsp://srnyc006.media.globix.net/space/b010511_sp_scram_2.rm
US - Space 02
rtsp://srnyc006.media.globix.net/space/b010411_sp_sts100lonchakov_2.rm
US - Space 03
rtsp://srnyc006.media.globix.net/space/b000803_en_spacecowboys_2.rm
US - Space 04
rtsp://srnyc006.media.globix.net/space/s001102_expd1_dock_1.rm
```

The configuration file lets the distributor specify two email addresses and two FTP sites to which the RealVideo statistics will be delivered. RealTracker will automatically save the configuration into a file called `RT.cfg`. Users running RealTracker must put `RT.cfg` into the same directory as the RealTracker executable: `RealTracker.exe`.

3.1.2.2 For the User

Before users can run RealTracker, they must have RealPlayer or RealOne (either the free version or a subscribed version) installed and the RealTracker support files[5]: `Aux98.dll` or `AuxNT.dll`, as appropriate.

Upon startup, RealTracker requests country, state, and network configuration information from the user as depicted in Figure 3.2a. In addition, RealTracker automatically detects the Operating System type, CPU type, available RAM and IP address.

Upon clicking “OK” the main window pops up. The main window, depicted in Figure 3.2b, provides a playlist for video clip selection and allows users to start and stop playing the clip. Clicking “Play” begins playout of the first video clip in the playlist. RealTracker then proceeds sequentially through the playlist to the end. Clicking “Stop” terminates the playout of the current video.

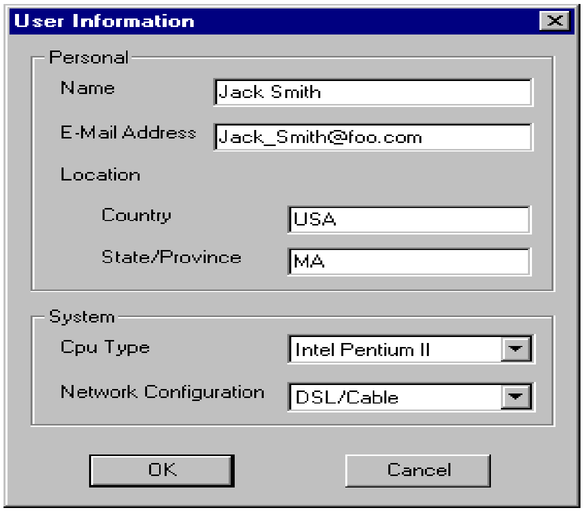
When each clip finishes playout, the user is solicited to assess the video quality by providing a numeric rating from 0-10 as depicted in Figure 3.2c.

While the video is playing, RealTracer gathers system statistics: encoded bandwidth, measured bandwidth, transport protocol, encoded frame rate, measured frame rate, playout jitter, frames dropped and CPU utilization.

The user data and specific clip statistics are then sent via both email and FTP to a server specified in the `RT.cfg` file. Users can also choose to save statistics locally to disk.

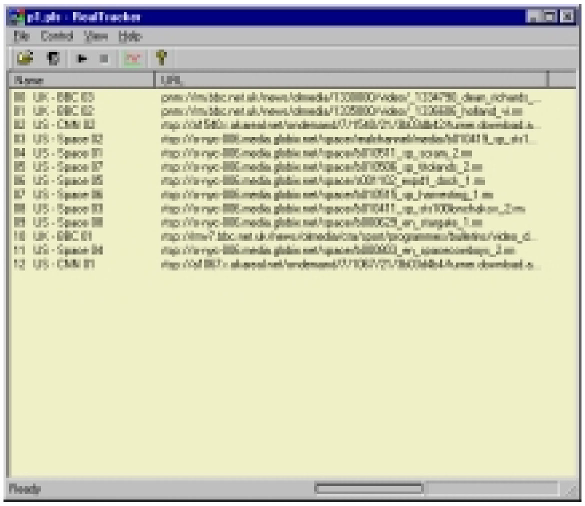
If so desired, the user can control the length of the clip playout and the requests for quality ratings using the “Options” button. The defaults are to play the clip for 1 minute and request a rating for each clip, proceeding to the next clip after 10 seconds if no rating is given.

[5] See http://perform.wpi.edu/real-tracer/



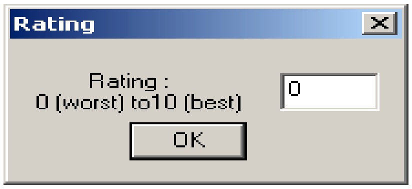
Name: Jack Smith
E-Mail Address: Jack_Smith@foo.com
Country: USA
State/Province: MA
Cpu Type: Intel Pentium II
Network Configuration: DSL/Cable

Figure 3.2a RealTracker User Information Window



Name	URL
00 UK - BBC 03	[illegible]
01 UK - BBC 02	[illegible]
02 US - CNN 02	[illegible]
03 US - Space 02	[illegible]
04 US - Space 01	[illegible]
05 US - Space 07	[illegible]
06 US - Space 05	[illegible]
07 US - Space 06	[illegible]
08 US - Space 03	[illegible]
09 US - Space 08	[illegible]
10 UK - BBC 01	[illegible]
11 US - Space 04	[illegible]
12 US - CNN 01	[illegible]

Figure 3.2b Figure 3.2b RealTracker Main Window



Rating :
0 (worst) to10 (best) 0

Figure 3.2c RealTracker Clip Rating Window

3.2 Operation of RealData

3.2.1 Real Folder

RealData provides a file management structure called a *RealFolder* that is used to manage statistics gathered by RealTracker. To create a RealFolder, users click on "RealFolder" in the main menu and choose "New Folder". A pop-up window lets users enter a new folder's name. After clicking on the "OK" button, the new folder shows up in the left window under "RealTracer". To add files into this folder, users select the folder in the left window and click on "RealFolder" in the main menu and then choose "Add Files". RealData can only process the files collected by RealTracker, so only files collected by RealTracker should be added into a RealFolder. RealData allows users to save the Real-Folders created. To save the created RealFolders, users click on "File" in the main menu and choose "Save" or "Save As" and then enter the name for the RealFolders. To open a previously saved RealFolder, users click on "File" and choose "Open".

3.2.2 RealData Files

Each RealData file is associated with a RealVideo clip that was played by RealTracker. After adding a file into a RealFolder, the statistics data associated with a RealVideo clip show up in the right window. If a RealFolder is selected, all the RealData files within that folder show up in the right window. To view more details about a particular RealData file, a user can double click on the file selected in the right window. Another window then pops up, in which the upper window shows the basic information regarding the user and the corresponding RealVideo clip, such as user name, location, IP address, RealVideo clip's URL etc, and the lower window shows the statistics data, recorded per second, for that particular RealVideo clip. All the RealData files and corresponding statistics data within one RealFolder can be exported to a comma delimited file, suitable for import into most spreadsheet programs, such as Microsoft Excel, for further analysis. To do so, a user clicks on "Tool" in the main window and chooses "ExportToExcel".

3.2.3 Cumulative Density Data

RealData provides a tool that can generate cumulative density data (used for drawing CDF graphs) for measured bandwidth, frame rate and jitter. To generate a CDF for the recorded bandwidth, for example, a user clicks on “Tool” in the main window and chooses “CDF” and then chooses “BANDWIDTH_ALL” or “BANDWIDTH_SELECTED”. If a user wants to generate cumulative density data for all the files within a RealFolder, s/he chooses “BANDWIDTH_ALL”, while choosing “BANDWIDTH_SELECTED” generates cumulative density data for the selected RealVideo clips. The cumulative density data is then saved as a comma-separated text file for further analysis or graphing.

4. Results

In this section we present results obtained from our use of the RealTracer tools. The results presented are meant to show some of the possible usages of the RealTracer tools rather than the general results about the use of streaming media on the Internet. More details on the results from our previous use of the RealTracer tools can be found in [WCZ01].

Section 4.1 describes our approach of using RealTracker to conduct a wide-scale empirical study of RealVideo traffic from several Internet servers to many geographically diverse users; Section 4.2 uses RealData to analyze results from the study and to present new analysis not presented in [WCZ01].

4.1 Approach

In order to empirically evaluate the performance of RealVideo across the Internet using RealTracker, we employed the following methodology:

- Set up a Web site to allow users to download RealTracker.
- Select RealVideo servers from geographically diverse Web sites and choose diverse video clips from thoses sites.
- Solicit users to run RealTracker and gather data.
- Analyze the results using RealData.

In order to let users easily download RealTracker, we set up a Web site for the RealTracker software and corresponding support files, along with the detailed usage instructions. We also configured RealTracker so that the data could be sent via both email and FTP to a server at Worcester Polytechnic Institute.

We chose RealServers accessible through Web pages from 6 geographic regions: Asia, Australia, Europe, Japan, North America, and South America. Within each region, popular RealNetworks sites were chosen from: Australia, Brazil, Canada, China, Italy, Japan, the United Kingdom, and the United States. Figure 4.1 depicts a geographic representation of the RealServer sites chosen. From each site, we selected a variety of video content among all the videos that were offered.

Once the servers and videos were selected, we solicited friends, family and colleagues from various parts of the world to help in the study. Since it was fairly easy for us to obtain data points from Massachusetts, we asked friends and colleagues on campus and at work to solicit help from people they knew outside of Massachusetts. We also posted messages asking for help to the `rec.video` newsgroup and `end2end-interest` mailing list.

We then gathered data from users running RealTracker for an 11 day period from June 4, 2001 to June 15, 2001. Figure 4.1 depicts a geographic representation of the locations of users that ran RealTracker.

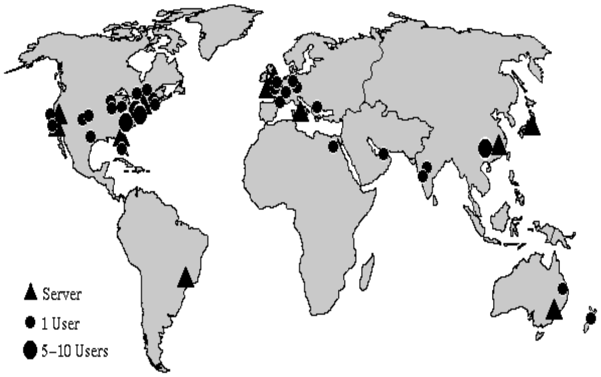


Figure 4.1Geographic Depiction of RealServers and Users

4.2 Results and Analysis

As described in Section 3, the system statistics gathered by RealTracker while a clip is being played includes: encoded bandwidth, measured bandwidth, transport protocol, encoded frame rate, measured frame rate, playout jitter, frames dropped, CPU utilization and perceptual quality ratings. Here, our analysis of these statistics using RealData focuses on frame rate and bandwidth.

A total of 63 users from 12 different countries participated in the study, playing a total of 2855 clips, among which 2100 are played by the users from the US and 352 are played by users in Europe.

It may be expected that servers in "wired" geographic areas, say North America, will provide better streaming video performance than other locations, say Brazil. Figure 4.2 depicts a Cumulative Density Function (CDF) of the frame rate for the servers used in our study, separated into 5 different geographic regions. The 5 regions all provide very similar frame rate distribution shapes, although the median of the best frame rate distribution is about 13 fps and the median of the worst frame rate distribution is about 8 fps. Australia and Europe have the best frame rate distributions, with Europe providing a larger percentage of frame rates above 20%. Asia provides the worst frame rates, but the differences at very low frame rates is small, and Asia servers actually have a larger percentage of frame rates above 15 fps than do North America servers.

While the peering richness of a client's ISP may largely determine video performance, this is difficult to measure. Thus, we consider geographic region, which is easy to measure, in place of client ISP information. Similarly to servers, it may also be expected that users in well "wired" geographic areas will observe better frame rates than users in more technologically remote areas. Figure 4.3 depicts a CDF of frame rate for the users in our study, separated into 4 geographic regions. In this case, geographic region appears to more clearly differentiate streaming video performance than it did in the case of the servers. Australia/New Zealand provides the worst frame rates for all ranges, with 75% of videos having fewer than 3 fps and less than 10% of videos having more than 15 fps. Clips played in Europe have the best frame rates up to 15 fps, with only 15% of videos having less than 3 fps and 25% of videos getting more than 15 fps. North America is slightly better than Asia up to the 15 fps rate. Europe, North America and Asia all provide about the same percentage of videos with frame rates above 20 fps.

Users often connect to local servers either explicitly or implicitly to get better performance. Content Delivery Networks (CDNs), in particular, attempt to make content more local for each user. While local access has been shown to be effective for improving the performance of Web content [KCZ01], the effects of local content on the performance of streaming media are not yet studied. We use RealData to analyze whether local access can achieve better video performance, too.

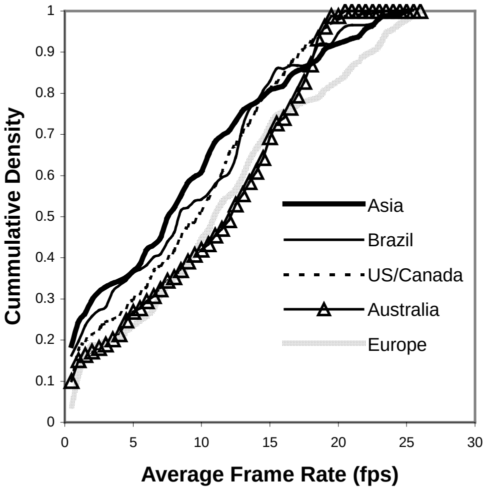


Figure 4.2 CDF of Frame Rate for Real-Servers in Different Geographic Regions

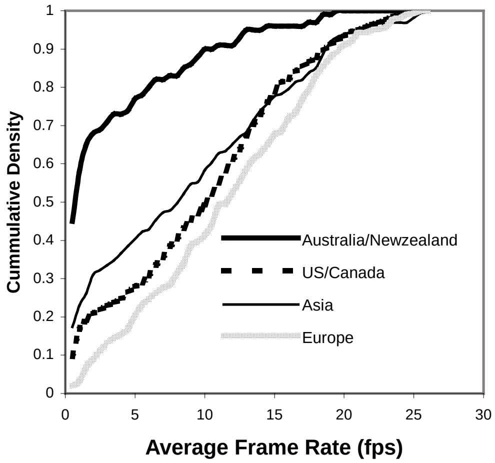


Figure 4.3 CDF of Frame Rate for Users in Different Geographic Regions

Figure 4.4 depicts a CDF of frame rate for the combinations of RealServers and users in different geographic regions. The access of US users to US servers provides a similar frame rate distribution shape as that of US users to non-US servers. The access of European users to European servers provides slightly better frames rate than that of European users to non-European servers. Overall, the 4 combinations all provide very similar frame rate distributions. This is consistent with our observation in Figure 4.2, which suggests there is very little difference in streaming video that is served from different geographic regions. In contrast, the bandwidth distribution shows strong locality as shown by Figure 4.5. The US users to US servers achieve better bandwidth than that of US users to non-US servers while the European users to non-European servers have better bandwidth than that of European users to European servers. This suggests media streaming may benefit from the effects of localized proxy caches and CDNs.

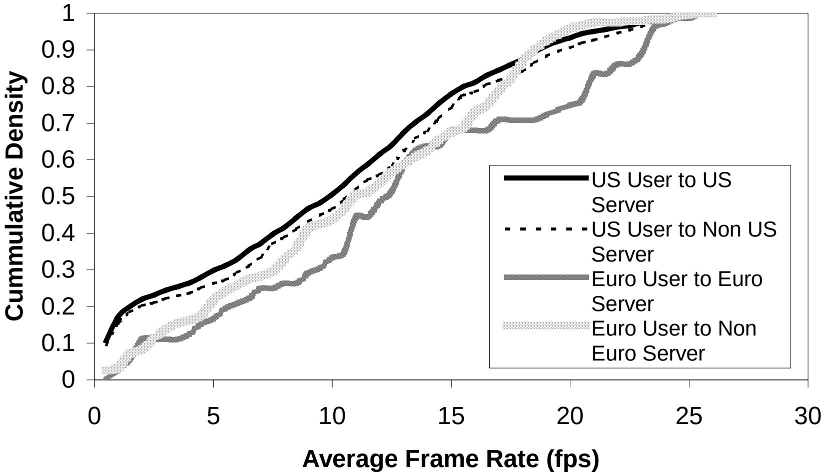


Figure 4.4 CDFs of Frame Rate for Users/RealServers in Different Geographic Regions

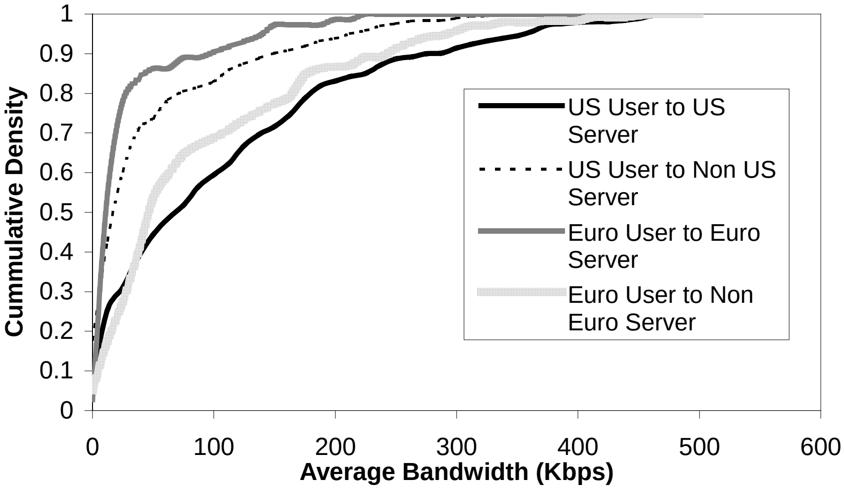


Figure 4.5 CDFs of Bandwidth for Users/RealServers in Different Geographic Regions

5. RELATED WORK

5.1 Multimedia Tools

[MCCS00] described the mmdump tool for parsing typical multimedia control protocols in order to dynamically setup filters to capture multimedia data traces. They presented the design and implementation of mmdump and demonstrated the use of mmdump through results from monitoring live traffic on a major ISP network. Although mmdump obtains both packet level data and embedded control information, it does not provide application level information such as frame rate, frame jitter, or video encoding rate. Nor does mmdump facilitate distribution to other users for large, wide-area data collection.

Most commercial media players all provide ways that users can view video performance statistics in real-time as videos play. Microsoft Windows MediaPlayer[6], Real-Networks RealOne[7] player, and Apple Quicktime[8] all enable users to monitor frame

[6] http://windowsmedia.com/

[7] http://www.real.com/

[8] http://www.apple.com/quicktime/

rates, loss rates and more, through pulldown menus in the Graphical User Interfaces. However, the same players provide no mechanisms to save performance data and do not facilitate automate playout in a controlled fashion. Microsoft, Real-Networks and Apple all include Software Development Kits (SDKs) that enable users to embed the proprietary video players into custom products, but do not provide general tools to measure performance.

5.2 Internet Measurement of Multimedia

[LR01] streamed low-bit-rate video over dialup connections across all 50 states in the U.S., analyzing network level statistics such as packet loss and round-trip time. While their work studied the impact of these parameters on streaming media traffic, they used a proprietary streaming media protocol that may not be representative of commercial video products.

[MH00] presented the results of a brief study examining the traffic emanating from one popular Internet audio service using RealAudio. While their results were useful in identifying data protocols used and flow lengths, they did not provide information on RealVideo, which potentially uses a much higher fraction of bandwidth that do RealAudio streams.

[CWVL01] collected traces from RTSP streaming-media sessions, which included both audio and video, from clients from a large university to servers in the Internet. They compared streaming-media workloads to traditional Web-object workloads, and explored the effectiveness of caching on streaming-media workloads. While their analysis provided additional information on session length and bandwidth usage, they did not provide additional video performance metrics such as frame rate or image size.

6. CONCLUSIONS and FUTURE WORK

In recent years, the use of commercial streaming products has increased dramatically due to the growth of high-bandwidth connections and high-speed computers. The future impact of streaming video on the Internet will be largely impacted by the role of commercial streaming media products. However, there has not been sufficient wide-scale empirical measurement of video performance across the Internet, which may largely stem from the lack of effective video performance measurement tools.

In this paper, we have presented the design, implementation and operation of a new tool suite, called RealTracer, for measuring and analyzing RealVideo performance. RealTracer includes RealTracker**,** a customized player that can play streaming RealVideo clips, record system performance statistics as well as user ratings, along with RealData, a tool that helps analyze the statistical data collected by RealTracker. RealTracker is implemented using the RealSystem RealOne G2 Software Development Kit, which provides two major interfaces, `IRMAPlayer` and `IRMAClient-AdviseSink`, to control RealVideo playback and gather the corresponding statistics. With RealTracker, users can create their own playlist and specify the means of gathering statistics. With RealData, users can sort the RealVideo statistics, generate cumulative density data, and export the statistical data in a comma-delimited file.

We have successfully used RealTracer to conduct a wide-scale empirical study of RealVideo traffic from several Internet servers to many geographically diverse users [WCZ01]. The RealTracer tools helped us find:

- The average RealVideo clip streamed over the Internet has good quality, playing out at 10 fps and, aided by a large, initial delay buffer, with nearly imperceptible amounts of inter-frame jitter. Users connecting to the Internet with modems and/or slow computers still have their PC or their network connection as the video performance bottleneck, while typical new computers connecting to the Internet via DSL or Cable modem achieve even slightly better performance than corporate network connections to the Internet. This suggests that increasing broadband connections for home users are pusing the bottlenecks for video performance closer to the server.

- There is very little difference in streaming video that is served from different countries, but there are distinct performance differences from video that is received in different countries.
- While local access has been shown to be effective for Web content, media streaming may also benefit from the effects of localized proxy caches and CDNs.

RealTracker only records user-centric video performance information. Our future work could seek to broaden the data set of both users and servers. In doing so, RealTracker could also gather statistics from Real-Servers. RealServer plug-ins can monitor resources in the RealServer Property Registry, which is a dynamic repository for a variety of server and client properties. These properties include such values as the number of clients currently connected, the total bandwidth being utilized, and a comprehensive set of statistics for each of these clients. A monitor plug-in can monitor any registry property, receiving notification when RealServer updates the property. Monitor plug-ins can also add their own properties to the registry and receive notifications when those properties change.

The major commercial competitor to RealNetwork's RealPlayer is Microsoft's MediaPlayer[9]. Developing similar tools to RealTracer for Media Player, perhaps a *MediaTracer*, would enable an empirical study of more general video performance on the Internet.

7. ACKNOWLEDGMENTS

We would especially like to thank Zheng Zuo for his help in implementing the RealTracer tools.

8. RealTracer AVAILABILITY

The RealTracer programs and source code are publicly available at:
http://perform.wpi.edu/real-tracer/

9. REFERENCES:

[CGLLR01] G.J. Conklin, G.S. Greenbaum, K.O. Lillevold, A. F. Lippman, Y.A. Reznik, Video Coding for Streaming Media Delivery on the Internet, *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 11, No. 3, March 2001.

[CN01] D. Cunningham and N. Francis. An Introduction to Streaming Video, Cultivate Interactive, *European Commission's Digital heritage and Cultural Content (DIGICULT)*, May 2001.

[CWVL01] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and Analysis of a Streaming-Media Workload, In *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, March 2001.

[FHPW00] S. Floyd, M. Handley, J. Padhye and J. Widmer. Equation-Based Congestion Control for Unicast Applications, In *Proceedings of ACM SIGCOMM Conference*, pages 45-58, 2000.

[Jup01] Jupiter Media Metrix. Users of Media Player Applications Increased 33 Percent Since Last Year, Press Release, April 2001.
http://www.jup.com/company/pressrelease-.jsp?doc=pr01040

[9] http://www.microsoft.com/windows/windowsmedia/en/default.asp

[KWZ01] B. Krishnamurthy, C. Wills, and Y. Zhang, On the Use and Performance of Content Distribution Networks, In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, California, USA, November 2001.

[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, California, USA, November 2001.

[MCCS00] J.E. van der Merwe, R. Cáceres, Y-H. Chu, C.J. Sreenan. mmdump - A Tool for Monitoring Internet Multimedia Traffic. *ACM Computer Communication Review*, 30(4), October 2000.

[MH00] A. Mena and J. Heidemann. An Empirical Study of Real Audio Traffic. In *Proceedings of the IEEE Infocom*, pp. 101-110. Tel-Aviv, Israel, IEEE. March, 2000.

[Rea00b] Real Networks Incorporated, *RealPlayer 8 User Manual*, copyright 2000.

[RFC 2326] H. Schulzrinne, A. Rao, R. Lanphier, Real Time Streaming Protocol (RTSP), *RFC 2326*, Apr. 1998.
Http://www.rfc-editor.org/rfc/rfc2326.txt

[RFC 1889] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 1889, Jan. 1996.
Http://www.rfc-editor.org/rfc/rfc1889.txt

[RHE99] Rezza Rejaie and Mark Handley and D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet, In *Proceedings of IEEE INFOCOM Conference*, 1999.

[WCZ01] Yubing Wang, Mark Claypool and Zheng Zuo. An Empirical Study of RealVideo Performance Across the Internet, In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, California, USA, November 2001.