2009 11th IEEE International Symposium on Multimedia

Demonstration of vDelay: A Tool to Measure Capture-to-Display Latency and Frame rate

Omer Boyaci, Andrea Forte, Salman Abdul Baset, and Henning Schulzrinne
Department of Computer Science, Columbia University
{boyaci,andreaf,salman,hgs}@cs.columbia.edu

1. Introduction and related work

We present vDelay [4], a tool for measuring the capture-to-display latency (CDL) and frame-rate of real-time video applications such as video chat and conferencing.

Real-time video chat applications have three key software components: a video encoder that compresses the video captured from the camera, a video decoder that decompresses the video received over the network, and a playout buffer that smooths the playout of received video due to network delay variations. These software components impact capture-to-display latency (CDL) and frame-rate of the real-time video played at a receiver application. Capture-to-display latency is the total time to encode and decode a video frame, playout buffer time, and latency of the network path. Along with bit-rate, these two metrics provide quick insights into the performance of a real-time video application.

vDelay has three important properties. First, it does not require any change in the source code or executable of a real-time video application. Thus, it can be used to measure the CDL and frame-rate of closed source video applications. Second, vDelay does not require any specialized hardware. Third, it is written in Java so it can be used to measure CDL and frame-rate of a real-time interactive video application on any operating system.

Existing video latency measurement tools involve the use of a specialized hardware. For example, OmniView [3] is a tool that uses a specialized PCI card. Our goal is to measure video latency without the use of any specialized hardware.

Yoshimura *et al.*designed a module for a video streaming application that measures for each frame the deviation from the playout time. Their approach does not calculate CDL or frame-rate, and requires changing the video application. Further, their approach is targeted towards video streaming applications. The adelay [1] tool can be used to measure mouth-to-ear latency.

2. Measuring CDL and frame-rate

The key to measuring CDL and frame-rate lies in embedding a timestamp in the caller's video, and retrieving that timestamp at the callee. The timestamp is the current system time at the machine running the caller user agent. Assuming that the machines running the caller and callee user agents are time synchronized within an acceptable error, the capture-to-display latency is the difference between the timestamp retrieved from the caller's video and current system time at the machine running the callee user agent. This difference can also be used to calculate the inter-frame display time at the callee user agent. Further, since every new frame must have an increasing value of a timestamp, the number of frames within a time period can be used to calculate the frame-rate of the received video.

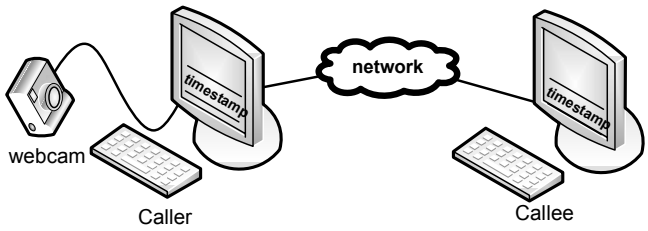

Figure 1: vDelay setup.

We use a trick to embed the timestamp in the caller's video that does not require any change to the video chat application. The timestamp, i.e., the current system time at the machine running the caller user agent, is displayed at the monitor of the machine running the caller user agent every t time units. A webcam is attached to the machine running the caller user agent and faces the LCD monitor. Thus, it captures the current image on the LCD monitor which includes the timestamp. The caller user agent then encodes this captured frame including the timestamp, and sends it over the network to the callee user agent which decodes the frame and displays it on its attached LCD monitor. An application running on the same machine as the callee user agent grabs the timestamp from the received frame, and calculates the time difference between the timestamp grabbed from the received frame and local system time. The timestamps are processed to calculate CDL and frame-rate. Figure 1 shows the setup for measuring CDL and frame-rate. The novelty of this approach lies in the fact that no additional hardware is needed and no modification to the software of any real-time video application is required.

We considered three approaches for displaying the timestamp at the caller user agent: as (1) an EAN-8 barcode [2], (2) numeric characters, and (3) a progress bar. From experimentation, we found that displaying timestamp as a barcode was the most attractive option. Since barcodes such as EAN-8 have a built in checksum mechanism and because barcode reading is very fast. Figure 2 shows a screen shot of the receiver side vDelay application.

978-0-7695-3890-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ISM.2009.129

Figure 2: Screen shot of the receiver side vDelay application. FPS, CDL, and FRR statistics are shown at the top of the image. The barcode received from the caller user agent is also visible.

Chat application	Version	Video codec	Resolution	Bitrate (kb/s)	Fps	CDL (ms)	Std. dev (ms)
Live Messenger	14.0.8064	H.264	640x480	600	23	69	16
Gtalk	v1.0.8.0	H.264	512x300	1000	27	99	16
X-Lite	3.0.47546	H.263+	320x240	400	27	102	15
Yahoo	9.0.0.2152	N/A	320x240	72	3	113	23
eyeBeam	1.5.19.5	H.264	640x480	400	27	129	16
AIM	6.8.14.6	N/A	240x180	120	9	147	57
Tokbox (LL)	2.01 2351	N/A	270x200	320	24	148	72
Skype (HQ)	4.0.0.215	VP7	640x480	560	20	238	22
Tokbox (HL)	2.01 2351	N/A	270x200	320	23	342	69

Table 1: Comparison of video chat applications. The results are sorted by capture-to-display latency (CDL). LL, HL and HQ are abbreviations for low latency, high latency, and high quality.

3. Results

For all the video chat applications (listed in Table 1), we ran the experiment for ten minutes and repeated it twice. The Tokbox application completely runs in browser and only depends on the availability of a Adobe Flash player. In Tokbox, the caller user agent sends packet over TCP to a Flash server maintained by Tokbox which forwards these packets to the callee user agent over TCP and vice versa. With the exception of Tokbox, the caller user agent sends packets directly to the callee user agent. Other than Tokbox and Yahoo Messenger, all the video applications send packets over UDP. For Skype, the video session was of high quality (HQ) as indicated by an icon in the received video.

The sender and receiver are synchronized through NTP. We also measured the synchronization manually before each experiment by sending a ping packet which is captured via wireshark on both sides. We observed a difference of 6ms at most. The monitor refresh time may also effect our measurements because the frame buffer is synched to monitor in every 16ms, which may explain the standard deviation of 16ms.

Table 1 shows the delay performance of these video applications. The results are sorted by capture-to-display latency. As mentioned before, Tokbox forwards packets from a caller user agent to a callee user agent through servers which are based in different geographical locations. The use of a server in different location impacts the CDL. Therefore, we report the minimum and maximum observed CDL for Tokbox which are abbreviated as LL (low latency) and HL (high latency) in Table 1.

Our results indicate that amongst all video chat applications, Windows Live Messenger has the best CDL value. For Tokbox (LL), Tokbox (HL), and AIM, the standard deviation of CDL is more than 50 ms. We conjecture that Tokbox has a high standard deviation for CDL due to the packet scheduling at the server relaying media packets. For AIM, we attribute the high standard deviation to the video encoding function.

X-Lite and eyeBeam achieved the highest frame-rate per second (fps). Except for Yahoo Messenger and AIM, the frame-rate of all video chat applications is above 20 frames per second. As for the CPU utilization of the machine running the caller user agent, we measured that Skype uses 44% of the CPU, the maximum amongst all applications. Gtalk tops the bit-rate comparison at 1,000 kb/s.

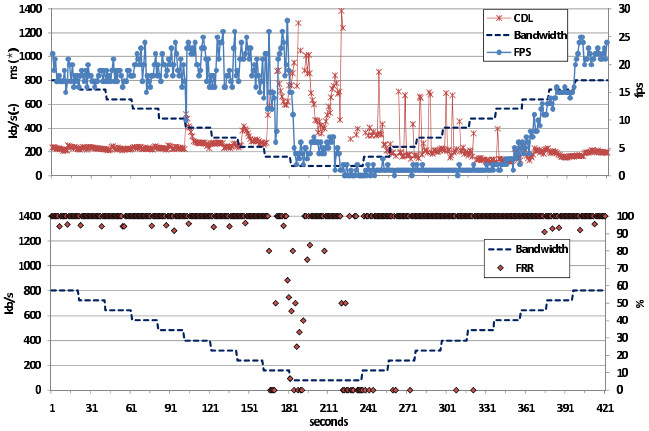

Figure 3: CDL, fps, and FRR for Skype as a function of time when the available bandwidth is adjusted as a step function.

vDelay can be used to measure CDL and frame-rate of a video chat application under controlled network conditions. Such use provides a powerful testing mechanism for application developers. One instance is shown in Figure 3, which shows the performance of Skype when the available bandwidth of a video session is adjusted as a step function. The figure shows that Skype suffers from a high jitter in frame-rate as the available bandwidth is gradually decreased. With the decrease in available bandwidth, CDL starts to increase indicating the impact of network queuing and playout buffer adjustments. The CDL graph shows large spikes when available bandwidth is below 400 kb/s.

References

[1] adelay. A tool to measure mouth-to-ear latency. http://www.cs.columbia.edu/irt/software/adelay/.

[2] EAN barcode. http://en.wikipedia.org/wiki/European_Article_Number.

[3] OmniView. http://www.omnitek.tv/admin/old_support/AVdelay1[1].pdf.

[4] O. Boyaci, A. Forte, S. Baset, and H. Schulzrinne. vdelay: A tool to measure capture-to-display latency and frame rate. In *Multimedia, 2009. ISM 2009. Eleventh IEEE International Symposium on*.