A Smoothing Proxy Service for Variable-Bit-Rate Video

Jennifer Rexford
Networking & Distributed Systems
AT&T Labs – Research
Florham Park, NJ 07932
jrex@research.att.com

Subhabrata Sen*
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
sen@cs.umass.edu

Andrea Basso
Speech & Image Processing
AT&T Labs – Research
Red Bank, NJ 07701
basso@research.att.com

Distributed audio and video applications, such as distance learning and entertainment services, rely on the efficient transmission of multimedia streams. However, these applications typically generate bursty, high-bandwidth traffic that is ill-suited to the poor throughput, delay, and loss characteristics of the Internet.

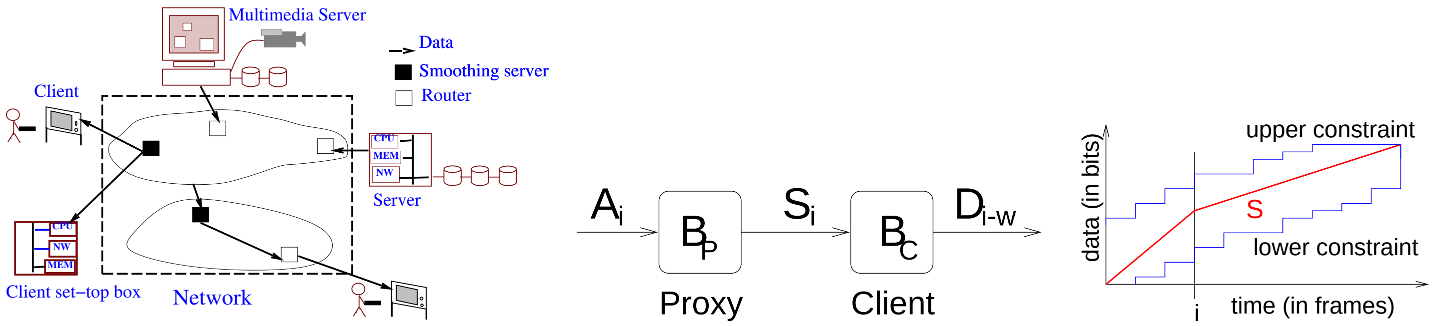
Compressed multimedia streams exhibit significant burstiness on multiple time scales, due to variations in detail and motion within and between scenes, as well as the frame structure of the encoding scheme. In addition, burstiness is inherent in emerging orchestrated media applications that combine video, audio, and text sources in a single stream. Variable-bit-rate traffic complicates the design of real-time retrieval, transport, and provisioning techniques that can achieve high resource utilization. In addition, a network service provider typically does not control the entire path from the multimedia source(s) to the end users, making it difficult to coordinate an efficient transmission. We contend that service providers can address these problems by deploying *proxy services* that perform *online smoothing* of variable-bit-rate streams inside the network. Figure 1(a) shows an example where smoothing can occur at network access or peering points. In this context, we focus on providing high-quality, non-interactive video to the growing number of users with broadband Internet access.

Smoothing Proxy for Broadband Users

For a wide range of videocast applications, such as course lectures or prerecorded movies, many users may be willing to tolerate a playback delay or several seconds or even minutes in exchange for higher quality and/or a lower price. We argue that a proxy service can perform online smoothing via *workahead transmission* into the client playback buffer by delaying playout of the multimedia stream. Our approach is a hybrid of techniques for smoothing a live, interactive stream at a video encoder and smoothing a prerecorded stream at a video-on-demand server. Due to tight delay constraints, smoothing of interactive video targets *small* timescale burstiness (say, within an MPEG group-of-pictures) and relies on reducing the encoding quality during scenes with high bandwidth requirements. Reductions in quality typically occur at scene changes, or in intervals with detail or motion, where the effects are most noticeable to the user. In contrast, workahead smoothing with a 1–60 second delay can remove the *medium* timescale burstiness within and across scenes *without degradation in the video quality*.

In addition, by reducing variability within a stream, workahead smoothing reduces network resource requirements *without depending on statistical multiplexing* (though it is still possible to achieve statistical multiplexing gains after smoothing). Smoothing within a stream is particularly important for supporting high-bandwidth multimedia applications on broadband access media, which provide limited opportunities for multiplexing gain. For example, a high quality MPEG-2 stream consumes 2-5 megabits/second, and emerging hybrid fiber-coax networks typically provide a 30 megabits/second downstream channel that is shared amongst multiple users. In addition to removing the inherent burstiness in multimedia streams, online smoothing can also compensate for the jitter introduced by upstream nodes. Reducing jitter at the proxy is especially useful if the path from the proxy to the client lies within a single service provider's domain and has predictable throughput and delay.

*The work of this author was performed while visiting AT&T Labs – Research



(a) Internetwork model (b) Smoothing model (c) Smoothing constraints

Figure 1: Computing a schedule for online smoothing

Proxies have been proposed for a number of other multimedia services, such as transcoding, selective discard, and retransmission. However, online smoothing introduces new challenges in controlling the low-level *timing* of frame and packet transmissions. We are investigating these issues by building a prototype of the proxy smoothing service in a video streaming testbed. For ease of deployment, our prototype uses a commercial off-the-shelf operating system and network adapter. Conceptually, smoothing operates at the frame level and assumes that transmission proceeds continuously at the scheduled rate. In reality, though, each video frame consists of multiple packets, requiring the proxy's transmission schedule to obey packet boundaries. Online smoothing requires changes in the packet transmission rate on a relatively small timescale, though not necessarily at the packet or frame level, to avoid underflowing or overflowing the buffer space at the proxy or the client. Although the network adapter could conceivably perform traffic shaping, these devices are expensive and not yet widely available, and typically do not allow the host to dynamically change the shaping parameters. Finally, online smoothing requires information about the sizes of incoming frames while transferring data at high rates, requiring an efficient design that minimizes data copying. In the remainder of the discussion, we describe our smoothing model and the platform for our prototype implementation.

Online Smoothing Model

The basic architecture of the proxy smoothing service extends our earlier work on online smoothing of variable-bit-rate video [1, 2]. The proxy has a B_P-bit buffer and receives A_i bits of the stream by time i, as shown in Figure 1(b). The buffer allows the proxy to delay transmission of incoming traffic, within a window of w frame times. The client has a B_C-bit buffer and plays D_{i-w} bits of the stream by time i, where time is measured in frame slots and D_j is the total size of the first j frames. In the absence of jitter or smoothing at upstream nodes, the proxy arrival vector is simply a shifted version of the client's playout vector (i.e., $A_i = D_{i+w}$). The proxy computes a schedule for the number of bits S_i to transmit by time i, based on constraints that avoid underflow and overflow of the two buffers. In particular, the schedule must satisfy

$$\max\{D_{i-w}, A_i - B_S\} \leq S_i \leq \min\{D_{i-w} + B_C, A_i\}$$

for all $i = 0, 1, \ldots, N+w$ for a video with N frames and a smoothing window of w frames, as shown in Figure 1(c). Since, the frame sizes are not known in advance, the proxy *incrementally* computes the constraint curves and the resulting schedule S. The proxy computes the schedule on a sliding window of w frames, using algorithms initially developed for smoothing prerecorded video. For example, we employ the shortest-path algorithm [3], which runs in $O(N)$ time on a sequence of N frames and produces a schedule S that minimizes the peak and standard deviation of the transmission rates. Executing this algorithm in every time slot allows the proxy to incorporate the most recent frame arrivals, at the expense of increased processing requirements. In general, the proxy could compute a schedule every α time units, where $1 \leq \alpha \leq w$. This is particularly important in scaling the proxy to smooth multiple simultaneous video streams.

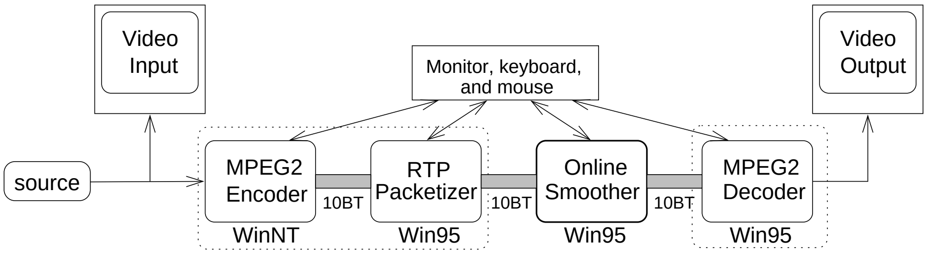


Figure 2: MPEG-2 video streaming platform

Applying the online smoothing model to MPEG-1 and motion-JPEG video traces illustrates that a smoothing window of 1–60 seconds offers substantial reductions in the peak rate and rate variability of the transmission rates. Smoothing at this timescale enables the proxy to remove burstiness beyond the inherent variability of the MPEG group-of-pictures structure. By smoothing over scenes with different average frame sizes, our online algorithm reduces the peak bandwidth by an additional factor of two, depending on the video. These performance gains are achievable for MPEG-1 and motion-JPEG streams with 1–8 megabytes of storage at the proxy and the clients. With the decreasing cost of high-speed memory, video set-top boxes and personal computers can easily devote several megabytes of buffer space to video smoothing, in exchange for these substantial reductions in network resource requirements. Generating the transmission schedule for a 1-second (30-frame) smoothing window requires approximately 12 milliseconds (on a R4400 workstation), making it feasible for a single proxy to smooth multiple video streams at a network access point.

Prototype of Smoothing Proxy

We are designing and evaluating the proxy smoothing service in the context an MPEG-2 multimedia platform [4, 5], as shown in Figure 2. For ease of debugging and performance tuning, the video source is initially implemented on two PCs that perform MPEG encoding and RTP packetization. The digitized output of a video source is fed to a real-time hardware/software MPEG-2 audio/video encoder. The encoder generates constant-bit-rate video, with burstiness on a 1-second timescale. A new version of the device driver supports constant-quality MPEG-2 video, which will allow us to evaluate the online smoothing service on variable-bit-rate streams with medium-time-scale burstiness. The second PC executes a real-time software RTP packetizer (compliant with the IETF RFC 1889, RFC 1890, and RFC 2055 [6, 7]) and a packet streamer. The third PC will execute our smoothing algorithm and shape the output stream. The fourth PC is the client, which includes an RTP depacketizer, a network adaptation layer, and a hardware video/audio MPEG-2 decoder.

We are in the process of building the smoothing proxy on a 300 MHz Windows 95 PC using UDP sockets in WinSock2, focusing on the efficient transmission of a single variable-bit-rate MPEG-2 video stream. The software design consists of three components: packet capture and accumulation of frame-size information, computation of the transmission rates, and packet delivery. The proxy paces the data transfer at the socket level by periodically initiating transmission of a group of packets. To perform within the operating system overheads, we estimate that the proxy will schedule transmission at the time scale of two frame slots (66.6 milliseconds). The proxy spreads the transmission of the group of packets across the two-frame interval, with the kernel and Ethernet adapter controlling the actual emission of packets. This periodic scheduling model requires minor adjustments to the smoothing constraints in Figure 1(c). As part of our ongoing prototyping effort, we will conduct experiments to determine the minimum granularities that the system can support for shaping the outgoing packets and computing the online schedule, particularly when the proxy handles multiple video streams. These results should lend additional insight into the challenges of developing multimedia smoothing proxies on a conventional PC platform. These services have the potential to bring high-quality video services to consumer and business clients with broadband Internet access.

References

[1] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online smoothing of live, variable-bit-rate video," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video*, pp. 249–257, May 1997.

[2] J. Rexford and D. Towsley, "Smoothing variable-bit-rate video in an internetwork," in *SPIE Symposium on Voice Video and Data Communications: Multimedia Networks: Security, Displays, Terminals, and Gateways*, November 1997.

[3] J. D. Salehi, Z.-L. Zhang, J. F. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," in *Proc. ACM SIGMETRICS*, pp. 222–231, May 1996.

[4] A. Basso, G. L. Cash, and M. R. Civanlar, "Transmission of MPEG-2 streams over non-guaranteed quality-of-service networks," in *Proc. Picture Coding Symposium*, 1997.

[5] A. Basso, M. R. Civanlar, and G. L. Cash, "Lessons learned on real-time MPEG-2 transmission over ATM networks," in *Proc. SPIE Digital Compression Technologies and Systems for Video Communications*, vol. 2952, pp. 182–193, 1997.

[6] D. Hoffman, G. Fernando, V. Goyal, and M. R. Civanlar, "RTP payload format for MPEG1/MPEG2 video." IETF RFC 2250, January 1998.

[7] M. R. Civanlar, G. L. Cash, and B. G. Haskell, "RTP payload format for bundled MPEG (draft-civanlar-bmpeg-01)." Internet Draft, February 1997.