Issues of Quality and Multiplexing when Smoothing Rate Adaptive Video

N.G. Duffield*, K. K. Ramakrishnan†, Amy R. Reibman‡
AT&T Labs–Research

Abstract

We have been examining the transport of compressed video over rate-controlled networks, and have developed a smoothing and rate adaptation algorithm, called SAVE (Smoothed Adaptive Video over Explicit rate networks). SAVE attempts to preserve quality as much as possible, by exercising control over the source rate only when essential, when the delay at the source may exceed a bound. Understanding the impact on the quality of the video using rate adaptation is important and we begin to examine some quantitative measures of video quality in this context. For the form of source rate adaptation we advocate, it appears necessary to evolve the quality metrics typically used to evaluate the efficacy of mechanisms to transport video. We also attempt to understand the dynamic nature of the reduction in the encoded rate, since any prolonged impairment is likely to be noticeable. We study the sensitivity of SAVE to its parameters and network characteristics. Finally, the utility of a proposed scheme is measured by its ability to multiplex a large number of streams effectively. We demonstrate that by smoothing the demand from the source, SAVE helps achieve good multiplexing gains. We examine the effectiveness of SAVE across 20 different traces of entertainment videos using different compression algorithms.

1 Introduction

The desired quality of compressed video to be delivered to the receiver varies widely, depending on the application, the potential cost to the user, and the network infrastructure that is available for transporting the video. We have been studying the transport of adaptive compressed video over ATM (Asynchronous Transfer Mode) networks using the ABR (Available Bit Rate) service [2] and [7]. In [2] we proposed an on-line source smoothing algorithm, called SAVE (Smoothed Adaptive Video over Explicit rate networks), that enables us to obtain better multiplexing gain without degrading the quality of the delivered video stream significantly. SAVE exploits the source buffer to reduce the burstiness of the video over short time scales, and uses the inherent negotiation of the rate based feedback control schemes to manage burstiness over the intermediate time scales. Only when the delay contributed by buffering at the source exceeds a reasonable bound is the quantization value increased.

We believe that using feedback based mechanisms for modifying the quantization parameter infrequently to accommodate both the fluctuations in the load (due to burstiness in the video) as well as the rate allocated by the network (possibly due to congestion) is better than carrying the video using other possible QoS classes that do not use feedback. For example, in unrestricted (or open-loop) VBR (Variable Bit Rate) transport, when buffering in the network is unable to overcome burstiness in the aggregate offered load, frames are lost. For a given transmitted rate, better video quality can be obtained by having the encoder produce coarser quality video by adjusting its quantizer, rather than having the network discard packets arbitrarily. In [6] this was demonstrated in the context of rate-adaptation by the encoder.

In this paper, we build on the work described in [2]. We explore more closely the quality achieved using SAVE. In particular, we study the sensitivity of video quality to the various parameters of SAVE and study the impact of network congestion and feedback delay. Another important issue we address here is the gain from having several sources multiplexed together. The multiplexing gain is achieved by overlapping the "peaks" and the "valleys" of the different sources of video on a link at a given time. The resulting aggregate flow tends to be less bursty than the individual flows. The larger the number of simultaneously active flows, the higher the potential for multiplexing gain. We show that there are significant multiplexing gains to be achieved using SAVE.

Re-negotiated CBR (RCBR) [5] is a mechanism proposed to overcome the medium-term variations in compressed video. It uses re-negotiation of the traffic parameters of a Constant Bit Rate connection using ATM signaling messages. Signaling is often performed by software in the end-systems, and typically involves (although not necessarily) considerable processing including the invocation of

*AT&T Labs–Research, Rm. B139, 180 Park Avenue, Florham Park, New Jersey 07932; duffield@research.att.com
†AT&T Labs–Research, Rm. A155, 180 Park Avenue, Florham Park, New Jersey 07932; kkrama@research.att.com
‡AT&T Labs–Research, Rm. 3-233, 100 Schultz Drive, Red Bank, NJ 07701; amy@research.att.com

Connection Admission Control functions in switches. On the other hand, the explicit rate based feedback mechanisms use RM (Resource Management) cells that are generated in hardware in the ATM adapter (network interface) and almost all the negotiation is performed in hardware both in the end-systems and in the switches, without requiring any CAC (Connection Admission Control) functions. Thus, the negotiation is likely to be more efficient. RCBR could use a source rate adaptation mechanism as we have proposed here with SAVE. However, in the work reported so far with RCBR, the use of such an adaptation has not been studied, and instead the primary emphasis has been to examine the probability of re-negotiation failure. We feel that it is better, even with a scheme like RCBR, that re-negotiation failures are handled gracefully by adapting the quantization parameter of the encoder. Moreover, we are able to address in detail the issue of buffer delay management in the face of delayed network responses to requests from the source for increased bandwidth.

There has been considerable work examining the effectiveness of smoothing of stored video; see e.g. [8, 9, 10, 11, 14]. These require advance knowledge of at least a part of the future sequence of frame sizes. This results in a corresponding playback delay of a few seconds. On the other hand, our approach to smoothing is to make it suitable for a wide range of video applications including interactive ones. Because of this desire to transport interactive, high quality video, we attempt to meet tight delay and quality constraints.

The paper is organized as follows. In Section 2 we briefly describe the SAVE algorithm. In Section 3 we explain the various metrics we use to evaluate the quality of the delivered video. Section 4 summarizes the traces used in the experiments. In Section 5 we investigate the robustness of SAVE with respect to variations in the control timescale of the ABR network. In Section 6 we determine the robustness of quality with respect to network contention, and in so doing establish the degree of multiplexing gain available. In Section 7 we examine the effect of tunable parameters within SAVE on the quality. We then conclude.

2 Description of SAVE Algorithm

The operational setting for SAVE is shown in Figure 1. We briefly describe SAVE here. More detail is provided in [2]. There are four data rates which describe the operation of SAVE. First, uncompressed video is fed into an encoder. The IDEAL RATE is that required by the encoder to encode the frame at ideal quality. The ENCODED RATE will be the rate at which the frame is actually encoded. We assume that the encoder will precisely meet a given target. The adapter is able to make rate requests to the network at least once per frame time. Averaged over one frame time, this is the

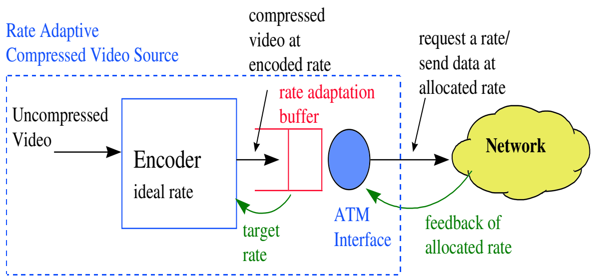

Figure 1. Framework for Rate Adaptive Video in an Explicit Rate Environment

REQUESTED RATE. Finally, the network returns the ALLOCATED RATE to the adapter after some delay primarily involving the propagation delays. Since the network returns a rate for the source to transmit at with every RM cell, the Allocated rate we refer to here is based on the average of the rate returned by the network over a time window of k frames (the results reported here are with $k = 1$). Because the explicit rate-control algorithm used in the network keeps the queueing delays low, the feedback delay is primarily the propagation delay. We consider networks with one-way propagation delays of the order of 50ms to 100ms.

Our aim is to smooth the compressed video using the source buffer while keeping source delays small enough for real-time video: in the region of 100ms or so. Under this constraint, we want the requested rate to be smooth and not unnecessarily large, while at the same time we want the encoded rate to be close to or equal to the ideal rate. This also helps the rate allocation algorithms to stabilize [1]. When a given ideal frame-size cannot meet the delay target given current buffer occupancy and the allocated rate, we encode at a lower rate so as to meet the constraint. We aim to request and be allocated sufficient bandwidth such that this happens rarely. The SAVE algorithm achieves this aim using two disjoint parts. The RATE REQUEST ALGORITHM specifies how the adapter requests bandwidth from the system. The FRAME QUANTIZATION ALGORITHM specifies the rate at which frames are to be encoded.

The Rate Request Algorithm. The requested rate is (essentially) the maximum of two rates: $r_{\rm sm}$ average rate per frame (over some short smoothing window of $w_{\rm sm}$), and $r_{\rm max}$, the maximum rate (over a medium term window $w_{\rm max}$) required to drain a frame from an empty buffer within the delay constraint. Let $f(n)$ be the size of frame n, τ the interframe time and $\tau_{\rm max}$ the buffer delay constraint. Then for the $n^{\rm th}$ frame

$$r_{\rm sm}(n) \quad = \quad (\tau w_{\rm sm})^{-1} \sum_{i=0}^{w_{\rm sm-1}} f(n-i), \qquad (1)$$

$$r_{\max}(n) \quad = \quad (\tau_{\max})^{-1} \max_{i=0,1,w_{\max}-1} f(n-i). \quad (2)$$

The heuristic for this choice is as follows. We clearly want to request a rate that is commensurate with at least the average rate of the source. By choosing $w_{\rm sm}$ quite small we will make this responsive to rate changes within the medium term. (But if there is a Group of Pictures (GOP) structure of period p present in the encoding, we want $w_{\rm sm} \geq p$ to avoid systematic variations in the requested rate). However, if the allocated rate were just $r_{\rm sm}$, when the short term peak to mean ratio exceeds $\tau_{\max}/\tau$ (typically only 2 to 3) large frames will suffer delay beyond τ. So by allocating the maximum of $r_{\rm sm}$ and $r_{\max}$, the large frames typically find the buffer empty, and so drain within time $\tau_{\max}$. By taking $w_{\max}$ sufficiently large we aim to anticipate the large frame typical of a scene. (Actually there is a little more detail from [2]: we take the maximum also with an autoregressive estimate $r_{\rm ar}$ of the maximum frame size divided by $\tau_{\max}$). Finally, we systematically over-request by a factor $\beta > 1$: the requested rate at frame n is then

$$r_{\rm req}(n) = \beta \max\{r_{\rm sm}(n), r_{\max}(n), r_{\rm ar}(n)\}. \quad (3)$$

Frame Quantization Algorithm. Given a buffer occupancy $b(n)$ and allocated rate $r_{\rm all}(n)$ at frame n, the estimated size of a frame which will empty within the delay bound is $f_{\rm avail}(n) = \tau_{\max} r_{\rm all}(n-1) - \max\{0, b(n) - \tau r_{\rm all}(n-1)\}$. We stipulate that no frame can be encoded in a size less than some proportion γ of its ideal size. The encoded size of frame n will be

$$f_{\rm enc}(n) = \min\{f(n), \max\{f_{\rm avail}(n-1), \gamma f(n)\}\}. \quad (4)$$

Choosing $\gamma > 0$ risks that a frame will be delayed more than τ, the inter-frame time. However, since the network component of the delay is a variable quantity, it may be better to risk a frame to be delayed slightly more (but within the maximum buffer delay constraint $\tau_{\max}$) rather than encode only very coarsely.

3 Evaluation Criteria

In this section, we describe our criteria to evaluate the performance of the overall system: delay, quality, and networking performance. While the criteria for video quality are somewhat heuristic, they are based on known characteristics of how humans subjectively rate the quality of video. [4]

Source Delay. The source buffer should not introduce delay so large as to eat into the delay budget of the network; this would make the network less attractive for real-time services. We assume that there is a sufficiently large playout buffer at the receiver to overcome delay jitter. Hence the primary concern for our work is the aggregate delay introduced in the source buffer and the network. We assume that an overall (one-way) delay budget around 200 milliseconds to 300 milliseconds is acceptable, and that, of this, a delay target of about 100 milliseconds for the source buffer is reasonable for interactive applications. We assume the source buffer is large enough to accommodate any backlog arising from a shortfall of the allocated rate from the encoded rate; at the operating point such differences will only last a short time if our delay target is reached.

Quality and Adaptation. We assume that it is desirable to keep the quality of the video transmitted by the source as close to the ideal quality as possible. Our premise is that sources are adaptive enough that even if the encoded rate falls below the ideal rate, the video quality at the receiver will not suffer significant perceptual impairment. We think this is true provided the shortfalls are sufficiently small, rare and short-lived. We shall use the term *cropping* to describe the reduction of the ideal rate to the encoded rate. In particular, cropping entails a reduction in encoding detail rather than a truncation of the size of the image. Cropping can occur either because of network delay in responding to changes in the short-term average rate, or because congestion forces the network to allocate less than the requested rate. In the latter case, the network is unable to know either the actual video quality aimed for, or the effect the reduction of allocated rate will have on quality; rate allocation amongst sources is done entirely on the basis of the requested rates. The rate allocated by the network itself may be based on a weighted max-min fair allocation [7], where the weights are proportional to the requested rate. This enables the network to favor a video source that has a higher requested rate (possibly because of a higher ideal rate) than another, even though they share the same bottleneck.

In evaluating the operation of SAVE with a given video source, we look at the pattern of cropping over the entire sequence of frames. We strive to keep the proportion of cropping below 20%, only exceeding this level for 0.1% of all the frames. At the rates in question, this amount of reduction of the encoded rate will degrade video quality as measured by the Peak Signal to Noise Ratio (PSNR) by about 1-2 dB. We believe that greater than this amount of degradation is generally perceivable by moderately experienced viewers.

We also look at the dynamics of cropping. In addition to meeting the above criteria for cropping amount and frequency over the whole trace, we also want to avoid long sequences of consecutively cropped frames. So we looked at the distribution of the length of bursts of successive frames cropped above (or below) the 20% threshold (other thresholds could also be considered.) We expect that the quality of video is perceived to be equivalent to the quality of the worst segment, provided this segment is long enough

[4]. Thus we will be particularly interested in the maximum burst length of cropping greater than the threshold of 20%. Conversely, during periods when cropping is below the threshold (including when there is no cropping) we expect there to be little impact on perceivable quality, even for experienced viewers.

We aim to keep the maximum burst length of cropping not much greater than any GOP period present in the encoding. Otherwise cropping of the large frame in successive GOPs could lead to noticeable quality reduction over timescales up to 1 second. An example of this would be the cropping of consecutive I-frames in a 12 frame GOP encoded at 24 frames per second. Cropping of an I-frame can impair quality for subsequent frames. In addition, the periodic nature of the impairments will make them more noticeable [4]. For this reason, when there is a GOP structure present, if two frames cropped more than 20% are separated by less than the GOP length, then for statistical purposes we treat all the intermediate frames as though they have been cropped more than 20%.

Robustness to Network Feedback Delay. The rate allocation mechanism of the explicit rate network is not expected to instantaneously allocate a rate in response to requests. To include the time needed for the stabilization of the rate allocation algorithms [1], we need to verify that quality measures are preserved even for a relatively large feedback delay (considered in number of frame times).

Even in the absence of network congestion, frame cropping is likely occur when the short term average demand suddenly changes. Therefore, some frames are likely to suffer cropping until the network allocates an increased rate.

Channel Capacity and Multiplexing Gain. We look at the number of sources that may be multiplexed within a link of a given capacity when our delay constraints and cropping criteria are met. This is eventually the criterion that will guide us to choose one algorithm over another. For statistical multiplexing gain we want to be able to assign capacity to an aggregate of sources at less than the peak of their aggregate requested rate. During transient periods in which the aggregate requested rate exceeds the capacity, the explicit rate mechanism of the network will proportionately reduce the allocation to each source so as to avoid congestion. We use the term *rate-reduction* when the allocated rate is less than the requested rate. We determine the sufficiency of such an allocation by establishing the extent to which such rate-reduction is compatible with the delay and quality targets described above.

Sensitivity to Algorithm Parameters. Finally, SAVE has a number of tunable parameters: w_{sm}, $w_{\max}$, β and γ. We shall investigate the sensitivity of the quality metrics to variations in these parameters.

4 The Experimental Traces

Here we summarize the properties of the 2 sets of traces used in the paper (maintaining conformity with the labeling of [2]).

A. An MPEG-2 encoding of a 40680 frame portion of “The Blues Brothers”, with M=1. There is no periodic structure. The frame rate was 24 frames per second.

E. 19 MPEG-1 traces, each with 40,000 frames, compiled by Rose. They originate from cable transmissions of films and television; see [12] for further details. The GOP is 12 frames with an IBBPBBPBBPBB pattern. For our experiments we assumed a uniform rate of 24 frames per second.

Many of the experimental results were carried out using 1- to 38-fold aggregations of the 19 traces from set E. Each trace was used twice, all traces having randomized phase and ordering within the aggregation. Experiments were repeated over different randomizations.

5 Sensitivity to Network Feedback Delay

We examine the sensitivity of SAVE to the feedback delay from the network in the context of Trace A. In the experiments we assumed the feedback delay for the network to allocate requests was fixed:

$$r_{\mathrm{all}}(n) = \begin{cases} r_{\mathrm{req}}(n-\delta) & n > \delta \\ r_0 & n \leq \delta \end{cases} \tag{5}$$

Here r_0 is an initial rate given to the source until the network responds to the first rate request after the feedback delay of δ. We used the mean ideal rate as r_0.

We display the extent of degradation measured by our quality metrics in Table 1. The target for the proportion of frames not suffering more than 20% cropping of the frame size is clearly met. In fact, only 0.3% to 0.4% of the frames suffer any cropping at all. As seen in the table, the reduction in quality shows a slight sensitivity to the feedback delay. We feel that this reduction is well within reasonable levels for our target for quality degradation.

Even with a feedback delay of 4 frame times (about 166 milliseconds) the maximum delay contributed by the source buffer was 105ms, the next largest delay being 98ms, again deemed acceptable based on our target of 100 ms for the source buffer delay.

Also shown in Table 1 are the dynamic measures of quality. We look at the statistics of the length of bursts of consecutive frames that have more than 20% cropping, and those of the complementary bursts of frames for which

F.back Delay	Proportion of Cropping			Mean Burst Crop	
	$> 0\%$	$> 20\%$	$> 50\%$	$\leq 20\%$	$> 20\%$
1 fr.	0.0032	0.0011	0	2257	3.28
2 fr.	0.0036	0.0012	0	2031	3.25
3 fr.	0.0037	0.0014	0	2030	3.85
4 fr.	0.0042	0.0016	0	1933	4.00

Table 1. Proportion of Frames experiencing a degradation in quality for trace A. Also shown are mean number of successive frames having less than 20% cropping (if any), and mean burst length of successive frames failing the 20% cropping criterion.

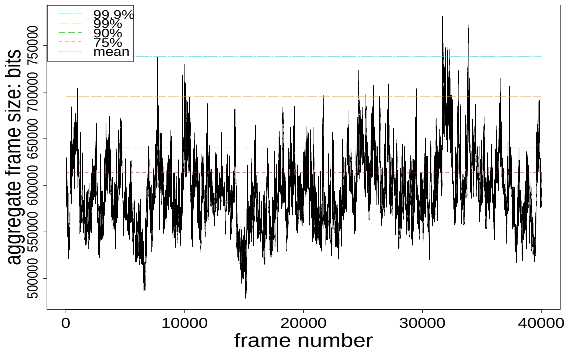

Figure 2. SMOOTHING BY AGGREGATION. Aggregated requested rate of 19 40000-frame MPEG-1 video traces in set E. The vertical range is about 1/2 of the lowest value

cropping, if present, is no more than 20%. In the table we see that the mean burst length of cropping $> 20\%$ ranges from 3.25 to 4, for varying feedback delays. In contrast, the mean length of the complementary bursts is around 2000. When the feedback delay is 4 frames, the longest burst of cropping $> 20\%$ was 8 frames, or about 1/3 of a second. Furthermore, there were only 26 such bursts, involving only 67 frames in total. This was for the trace of 40680 frames. This will produce a very slight visual impairment, probably visible only to the most experienced and critical viewers.

6 Multiplexing Gain and Sensitivity to Network Congestion

In this section we investigate the bandwidth requirements of the aggregate request rate from a number of sources. We want to determine the extent to which statistical multiplexing across sources is feasible: we aim to be *able* to allocate less than the aggregated peak requested rate of sources. The extent to which this is possible depends

Capacity (% of peak)	delay		fraction cropped	
	mean	max	$> 0\%$	$> 20\%$
100%	20.4	99.0	0.003	0.0006
90%	20.5	99.9	0.004	0.0006
80%	20.8	102.2	0.006	0.0008
70%	21.1	103.8	0.008	0.0010
60%	21.5	104.9	0.011	0.0017
50%	21.8	106.2	0.013	0.0022

Capacity (% of peak)	burst crop $> 20\%$		burst crop $\leq 20\%$	
	mean	max	mean	min
100%	2.2	7.3	2900.	108.
90%	2.2	8.9	2600.	106.
80%	2.7	13.8	2297.	79.
70%	3.3	20.2	1869.	28.
60%	7.2	75.6	1405.	25.
50%	9.0	96.9	1235.	20.

Table 2. IMPACT OF STATISTICAL REDUCTIONS IN ALLOCATED RATE. Averaged impact on single traces within 5-fold aggregation in set E. Capacity expressed as quantile of peak aggregated requested rate. As capacity decreases, observe insensitivity of delay and mean burst length of cropping over 20

on the impact to quality experienced by individual sources caused by transient contention. We are concerned with establishing the minimum capacity required; we intend to deal elsewhere with the problem of characterizing the statistical behavior of requests for the purposes of admission control.

Rate-Reduction, Cropping, and Statistical Multiplexing. A time-series of the aggregate *requested rate* from 19 traces of set E is shown in Figure 2. Consider a channel carrying the aggregated traffic from these sources. If the capacity C of a channel is less than the maximum aggregate requested rate, then periods of contention will occur from frames n (of the aggregate stream) for which the aggregate requested rate $R(n)$ exceeds the capacity C. The ABR rate-allocation algorithm responds to the demand exceeding the available bandwidth by allocating bandwidth to individual sources in proportion to their requests, the proportion being such that the total allocation equals the available bandwidth. We achieve this by using a weighted max-min fair allocation algorithm in the network [7]. Thus when $R(n) > C$ the rate allocated to each source will be a proportion $C/R(n)$ of its request.

Simulating Rate-Reductions in Aggregates. We can gauge the effect of attempting to use less capacity than the peak aggregate rate as follows. For a given set of traces, we run the SAVE algorithm to construct the requested rate for each trace as before, then sum to yield the aggregate requested rate $R(n)$. For a capacity C, we construct the proportional rate-reduction process

$$p(n) = \max\{1, C/R(n)\}. \tag{6}$$

To asses the impact of contention on an individual source we rerun the SAVE algorithm, but now proportionately reducing the allocated rate, subject to the network roundtrip delay δ. Hence (5) is replaced by

$$r_{\mathrm{all}}(n) = \begin{cases} p(n-\delta) r_{\mathrm{req}}(n-\delta) & n > \delta \\ r_0 & n \leq \delta \end{cases} \tag{7}$$

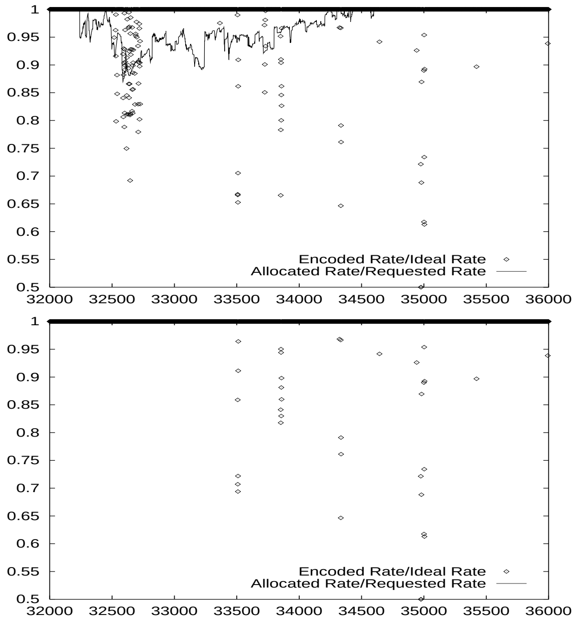

Figure 3. PATHWISE COMPARISON OF CROPPING AND RATE-REDUCTION. TOP: Segment of trace of Set A, impacted by contention with 10 traces from Set E, the total allocated at the 90^{th} %-ile of the peak requested rate. Encoded/Ideal rate ($=$ $1-$ cropping proportion) drops below 1, partially in response to contention. BOTTOM: Encoded/Ideal rate for same trace without contention. The horizontal axis is the frame number.

Note that this ignores some second order effects: reduction of the allocated rate for a given frame may increase buffer occupancy at the end of that frame time, and hence the rate request on the next frame may be increased to meet the delay target. We overcome this approximation when we report results using a network cell-level simulation, where we obtain an accurate characterization of the effects of a reduced rate being returned by the network. We show a sample of the effects of contention in Figure 3. The points show the ratio of encoded to ideal rate (i.e., $1-$ cropping proportion). The base behavior without contention is shown at the bottom. At the top is shown the modified ratio in the presence of contention, as reflected by the ratio of allocated to requested rate. Note additional cropping that now occurs between frames 32,500 and 33,000. Most of the cropping is less than 20% (i.e., encoded/ideal rate greater than 0.8) even though allocated rate is about 90% of the requested rate.

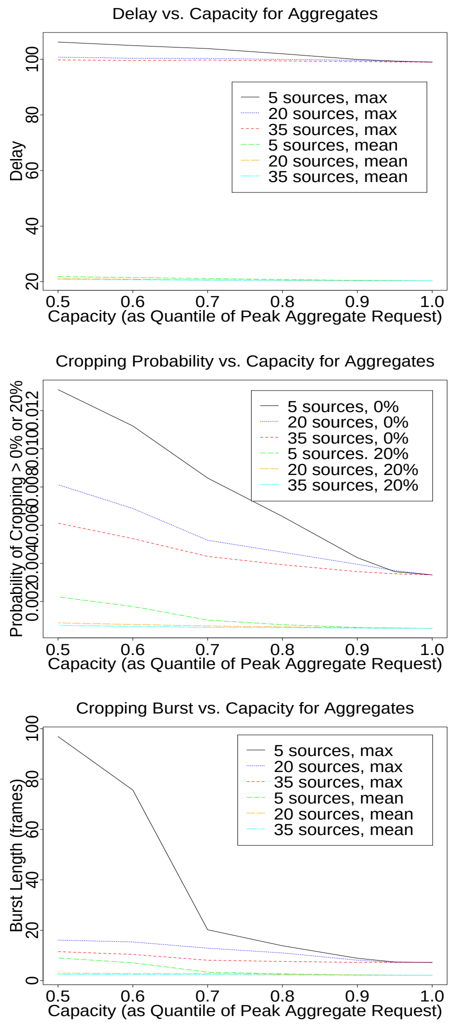

Figure 4. VARIATION WITH AGGREGATION SIZE OF IMPACT OF RATE-REDUCTION THROUGH REDUCED CAPACITY. For aggregations of 5, 20, 35 sources, averaged impact on single traces within aggregate; capacity expressed as quantile of peak aggregate requested rate. TOP: mean and max. delay. MIDDLE: proportion of cropping $>$ 0% and $>$ 20%. BOTTOM: mean and max. burst length of cropping $>$ 20%.

Impact on Quality of Statistical Multiplexing. We assess the impact of statistical multiplexing on quality of single sources by subjecting SAVE to the simulated rate-reduction process $p(n)$. We explore the variation of quality metrics with aggregation size and allocation. Table 2 summarizes experiments with 5 fold aggregates of traces from set E.

The statistics are averaged over individual traces. Figure 4 shows the variation in delay, cropping and burst length of cropping as a function of capacity, over a range of aggregations from 5 to 20 to 35 sources. Both show that mean and maximum of the delay are quite insensitive to both aggregation size and the degree to which capacity is set below the peak aggregated requested rate. This shows that one of the design criteria of SAVE (cropping to avoid delay) operates well over a wide range of conditions. The price to be paid for this is the frequency and duration of cropping. Both increase as the allocated rate is reduced. Our criterion that the *maximum* burst of cropping $> 20\%$ should not exceed a GOP (12 for the traces of Set E) is met when capacity is (not much) less than the 90^{th} percentile of the peak aggregate requested rate. The overall cropping criteria (no more than 1 in 1000 frames to be cropped more than 20%) is already satisfied at this capacity.

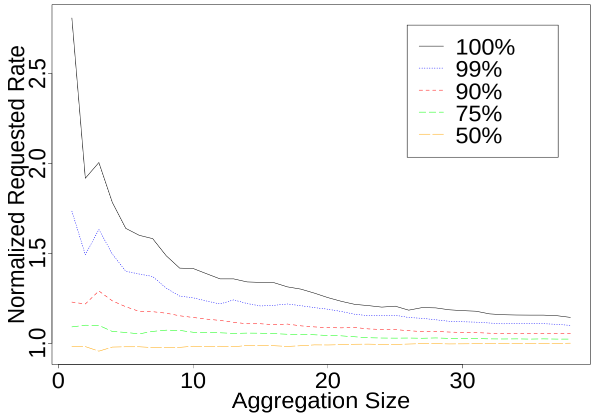

Figure 5. SMOOTHING THE REQUESTED RATE BY AGGREGATION. Quantiles of the aggregated rate for 1- to 38-fold random phase aggregations of traces from Set E. Quantiles normalized as a multiple of mean aggregate requested rate.

As the size of the aggregate increases, performance measures improve for capacity at a given quantile of the peak aggregate rate. Thus, averaging across sources means that it will be possible to obtain the desired performance with capacity equal to smaller quantiles of the peak rate for large aggregations. However, smoothing means that the quantiles of the aggregate requested rate (expressed as a proportion of the mean) become closer as the number of sources in the aggregate increases; see Figure 5. Thus the quality obtained becomes relatively insensitive to which quantile is chosen. In the example in Figure 5 the peak of the aggregate requested rate is only about 20% higher than the mean for aggregations of 20 or more sources.

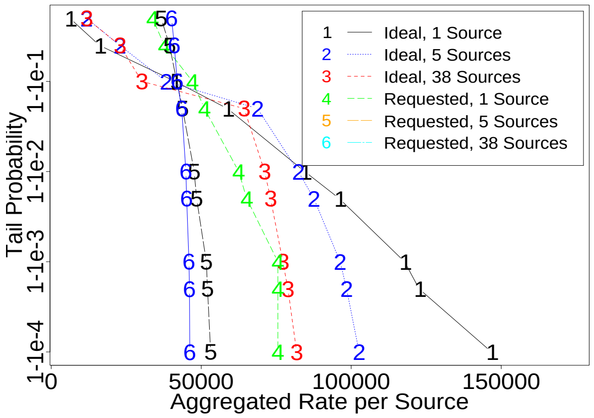

Figure 6. RATE SMOOTHING AND AGGREGATION Tail distribution, for ideal and requested rates per source, for aggregations of 1,5 and 38 sources of Set E. High quantiles of requested rate are lower for requested rate.

The smoothing by SAVE of individual sources reduced the variability of the requested rate compared with the ideal rate. This smoothing within sources in turn makes smoothing across sources in an aggregate more effective. We demonstrate this by displaying in Figure 6 the tail distribution of the aggregate rate per source for ideal and requested rate, for aggregations of 1,5 and 28 sources from Set E. The high quantiles of the requested rate (above the 90%-ile) take much lower values. The variability of the ideal rate does not approach that of the requested rate, even for very large aggregations. The price of this reduction in variability that quantile below about the 90%-ile, and indeed the mean, are higher for the requested rate than the ideal rate. An important potential benefit of the reduction of the high quantiles is for measurement based admission control. We think of an effective bandwidth for a source as a rate required to accommodate its extremes of variability. The narrowness of the distribution of the requested rate means that measurements of the effective bandwidth will be less influenced by statistical errors than for the ideal rate.

Single source behavior and comparative multiplexing gain. When SAVE is employed for transport of a single source's video stream, then we shift the focus away from the statistical properties of aggregations and instead ask what is the *constant rate* required for a given source so that SAVE achieves the quality targets. In our experiments on this topic, it was necessary to set the parameter γ (the minimum proportion of the frame to be encoded) to zero. Since the rate given to the source is constant, when $\gamma > 0$ we are unable to adaptively increase the allocated rate. When this

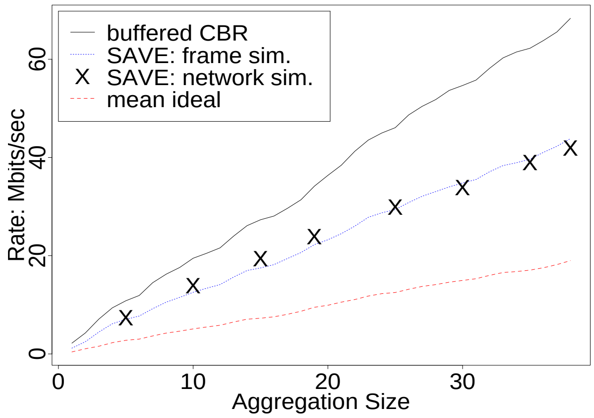

Figure 7. UTILITY OF EXPLICIT RATE FOR MULTIPLEXING GAIN. Bandwidth requirements, normalized by mean ideal rate, for 1- to 38-fold aggregates of Set E. LOWER CURVE: SAVE, 90^{th} percentile of aggregate requested rate. UPPER CURVE, CBR rate for 100ms source buffer. Also shown: mean ideal rate, and capacity estimate based on full network simulation described below.

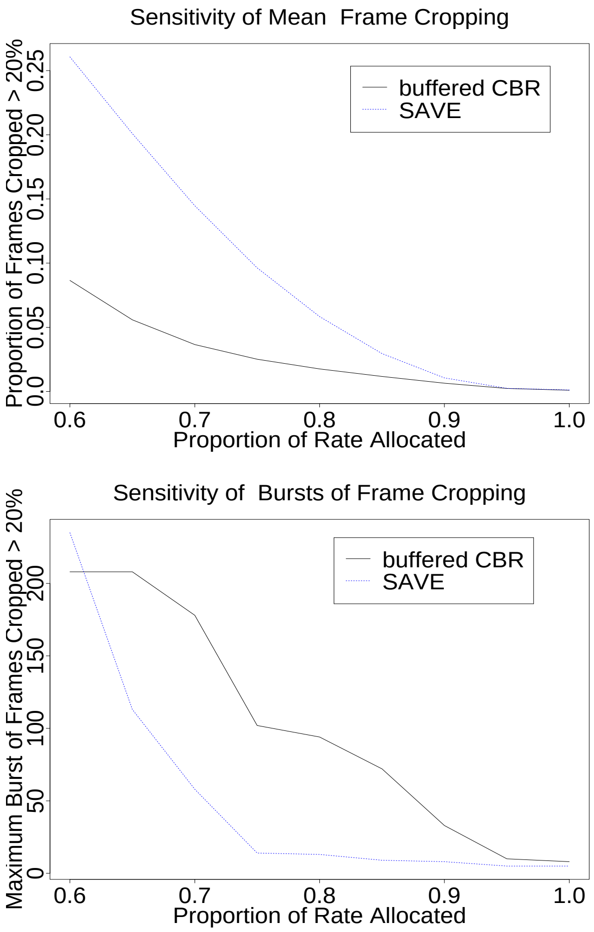

Figure 8. COMPARATIVE SENSITIVITY OF FRAME CROPPING IN SAVE AND BUFFERED CBR TO SYSTEMATIC RATE-REDUCTION OR UNDERALLOCATION OF BANDWIDTH. TOP: average cropping more sensitive for SAVE. BOTTOM: burst cropping more sensitive for CBR.

occurs over lengthy periods of high activity it leads to either excessive delays or buffer overflow. We refer to this mode, in which SAVE fits frames to a buffer that is drained at constant rate (CBR), as "buffered CBR".

Over the 19 traces of Set E, the required CBR rate was found to exceed the mean ideal rate by a factor of between about 2.4 and 6.0, with a mean of about 3.6. For the trace of set A it is about 2.2 times the mean ideal rate. To examine the multiplexing performance of the SAVE algorithm working in tandem with an explicit rate network, we compare these CBR bandwidth requirements to the bandwidth required when using SAVE in the network it was designed for, a network with explicit rate feedback. For CBR, we use the sum of the CBR rates of the constituents of the aggregate. For SAVE, we use the 90^{th} percentile of the peak aggregated requested rate. These are displayed for 1- to 38-fold aggregations from set E in Figure 7. The CBR, SAVE and mean ideal rates (for data alone, neglecting any protocol overhead) are in approximate proportions $3.6 : 2.3 : 1$.

The higher rate allocation for the buffered CBR reflects the necessity to allocate a sufficient rate to accommodate long-lived trends found in the video sources. If bandwidth allocated by the network is slightly lower than needed, degradation of quality will occur, often over a long burst. Since this is an inherent property of the video source, we expect that similar allocations will be required for other algorithms which adapt and transport real-time video at constant rate and similarly high quality. We compared the buffered CBR rates with the peak smoothed rate of the algorithm in [10] under the same delay constraint; the rates were similar.

We quantified the relative sensitivities of SAVE and buffered CBR to systematic rate-reduction (for SAVE) or underallocation (for buffered CBR). One motivation for this is trying to understand the effect of errors in traffic characterization at admission control time. We compared the sensitivity of the quality metrics (frequency of cropping $>$ 20%, and maximum burst length of such cropping). For SAVE we reduced the time-varying allocated rate by a fixed proportion; for buffered CBR we reduced the constant rate by the same proportion. The results for trace A are summarized in Figure 8 for proportions from 0.6 to 1. Burstiness of cropping is not very sensitive for SAVE for rate-reductions down to at least 0.75, but then increases rapidly. Buffered

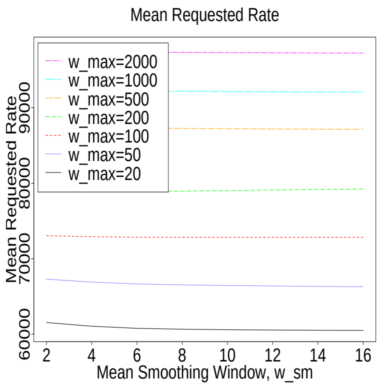

Figure 9. MEAN REQUESTED RATE VS. w_{sm}, trace set A, for varying $w_{\max}$

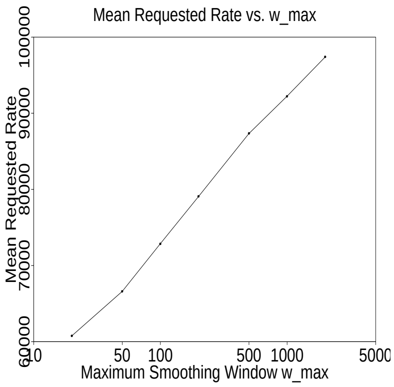

Figure 10. MEAN REQUESTED RATE VS. $w_{\max}$, averaged over $w_{\rm sm}$

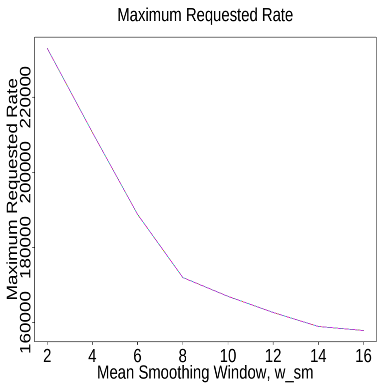

Figure 11. sc Maximum Requested Rate vs. $w_{\rm sm}$, trace set A. Curves coincide for $w_{\max}$ from 20 to 2000.

CBR on the other hand is quite sensitive to underallocation, for reasons outlined above. Cropping frequency for SAVE is more sensitive, in this case degrading to 1 in 1000 for a systematic rate-reduction of about 0.9.

The ordering of the sensitivities to rate-reduction by the network has the consequence that if we relax the quality target on maximum burst length of cropping more than 20%, and replace it by one on, say, the mean burst length, the performance gap between SAVE and buffered CBR narrows somewhat. In this case the ratios of required capacities for aggregates of buffer CBR, SAVE, and the ideal rate were found to be in ratios 3.2 : 2.1 : 1.

Validation through a Network Level Simulation. The foregoing analysis was based on the requested rate generated by SAVE operating on individual traces. The aggregated requested rate was in turn used to estimate the impact of contention on individual sources. While having the simplicity that only individual traces need be analyzed, this approach has the potential drawback that the impact of rate-reduction on buffer occupancy is ignored. Rate-reduction can cause increased buffer occupancy, which can in turn cause cropping to be increased. In addition, the feedback delay has the potential to vary because of queueing at the switches, although we try to keep this small. This motivates the evaluation of SAVE in a full network simulation at the cell level. The rate allocated by the network, including the dynamics of contention are accurately reflected in the network level simulation.

We simulated the operation of SAVE in an ATM network using the ABR service [13]. The explicit rate option is used, with the switches implementing a weighted max-min fair rate allocation algorithm [7]. The simulation was performed at the granularity of ATM cells, rather than video frames. We used a relatively simple configuration of all the sources "fanning in" to a network of 3 switches in tandem. The bottleneck is the link between the second switch and the destination host, the sink for all the video sources. The capacity of the bottleneck link was tuned in order to find the minimum capacity required for acceptable quality in the individual sources. The initial one way delay is 50 milliseconds. The simulation was run for approximately 20,000 frames from each source.

The results from the network level simulation are displayed as points in Figure 7. The other traces in the figure have been adjusted for protocol overhead. All include a factor for the ATM cell header (about 10%); the SAVE frame simulation also contains a factor for Resource Management cells (about another 3%). We see that the full network simulation and the frame level simulation of individual traces are in close agreement. The difference between them was at most 10%.

7 Parameter Sensitivity of the SAVE Algorithm

Sensitivity to Smoothing Parameters. The SAVE smoothing algorithm has two components of smoothing, as described in Section 2: the short-term smoothing window $w_{\rm sm}$ and the medium term maximum window $w_{\max}$. We nominally choose $w_{\rm sm}$ to be of the order of 12 frames, the GOP length of the traces in set E. The maximum window $w_{\max}$, which aims to capture the characteristic of the large frames of a scene, was nominally set to 1000 frames. In this section, we study the sensitivity of SAVE, specifically its quality and bandwidth usage, to these two smoothing parameters.

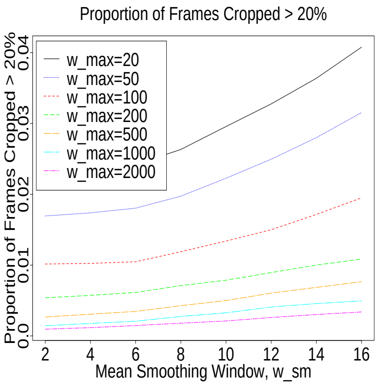

Figure 12. PROPORTION OF FRAMES CROPPED $>$ 20% VS. w_{sm}, trace set A for varying $w_{\max}$

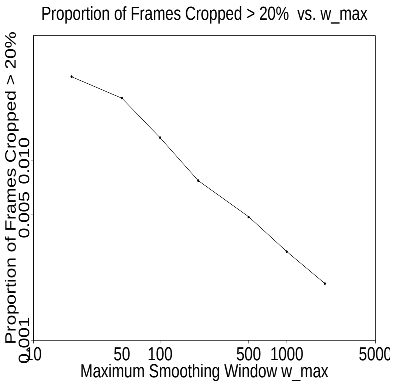

Figure 13. PROPORTION OF FRAMES CROPPED $>$ 20% VS. $w_{\max}$, trace set A, averaged over w_{sm}.

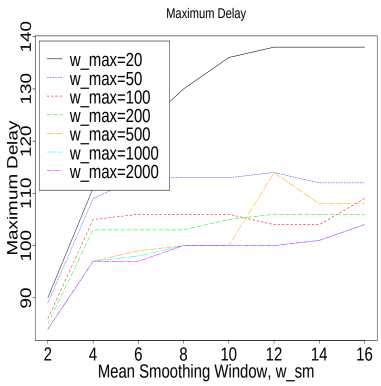

Figure 14. MAXIMUM DELAY VS. w_{sm}, trace set A for varying $w_{\max}$.

Figure 9 shows the variation of the mean requested rate as both $w_{\max}$ and w_{sm} are varied. As we observe from the figure, there is very little sensitivity to the value of w_{sm}. There is a small reduction in the mean requested rate as w_{sm} is increased from 2 to 6 or 8 frames. Beyond that, there is almost no change in the mean requested rate. However, it is clear that there is considerable sensitivity to the value of $w_{\max}$. We isolate this in Figure 10 by plotting the average over w_{sm} of the mean requested rate as a function of $w_{\max}$. Note the logarithmic horizontal scale. As $w_{\max}$ is increased, the mean requested rate increases, reflecting the fact that one large frame tends to have an effect over a longer period of time. The mean requested rate increases slowly with $w_{\max}$–approximately logarithmically–going from 60000 bits/frame when $w_{\max}$ is 20, to nearly 90000 bits/frame when $w_{\max}$ is 1000.

The maximum requested rate is insensitive to $w_{\max}$, as shown in Figure 11: the curves of the maximum request as a function of w_{sm} coincide for $w_{\max}$ in a range from 20 to 2000. However, the maximum requested rate is a decreasing function of the smoothing window w_{sm}. It reduces rapidly as w_{sm} increases up to 8. Beyond that, we begin to reach a point of diminishing returns, and the maximum requested rate reduces slowly as w_{sm} goes beyond the GOP value of 12. The form of dependence on $w_{\max}$ and w_{sm} indicates that the largest requested rate is governed by the short-term average over a few large frames and the mean requested rate is governed by the large frames observed over a longer, scene-level, timescale.

The impact on quality is likely to be of more interest as the two parameters are varied. Figure 12 shows how the proportion of frames cropped by more than 20% varies. The primary sensitivity is once again to $w_{\max}$. For values of $w_{\max}$ greater than 100, the proportion of frames cropped over 20% remains below 0.5% for the entire range of w_{sm} we examined. For smaller values of $w_{\max}$, less than 100, the proportion of cropped frames shows a small amount of sensitivity to w_{sm}. However, the primary sensitivity of the quality is to $w_{\max}$; see Figure 13. Note the log-log scale: the proportion of frames cropped more than 20% decays slowly, as a power law, in $w_{\max}$ (in fact $\approx$ const. $\cdot\, w_{\max}^{-0.85}$). For the acceptability criterion we selected (no more than 0.1% of frames cropped $>$ 20%), $w_{\max}$ needs to be reasonably large, of the order of 500 frames or more. This supports our initial intuition that $w_{\max}$ needs to capture the scene-level behavior, which is likely to be of the order of a few seconds. We also observed the statistics of the burst length of cropping with varying w_{sm} and $w_{\max}$; Generally, the mean burst length ranged from 1.5 to 3.5, increasing with w_{sm}. The maximum burst length was a decreasing function of $w_{\max}$, in a range from 25 to 2, and generally increasing with w_{sm}. Finally, Figure 14 show that maximum source buffer delay is reasonably insensitive to w_{sm} once $w_{\max}$ is greater than about 50 frames; it is within the 100ms target for $w_{\max}$ $\geq$1000. In summary, $w_{\max}$ is the main tunable parameter determining quality.

Sensitivity to Target Source Buffer Delay. Up to now, we have examined the performance of the algorithm with a source buffer delay target of 90 milliseconds. We now examine the sensitivity of quality to this delay target, as we vary it from 50 milliseconds to 120 milliseconds, for Trace A. Figure 15 shows the variation of the mean and peak for the requested rate from the source as we vary the delay target. The mean ideal rate (in terms of bits/frame) is 54823

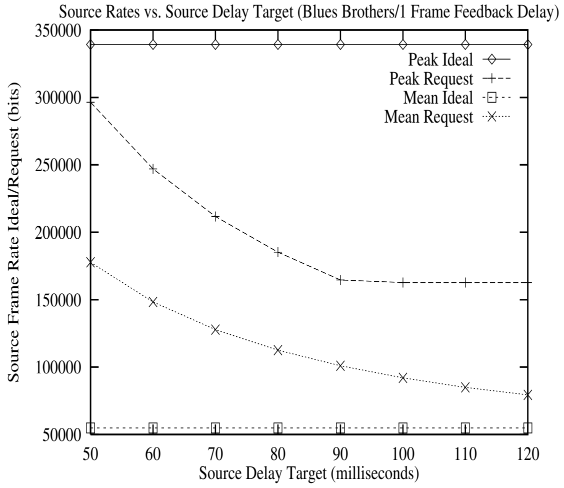

Figure 15. Behavior of the Requested Rate for Trace A for varying Source Buffer Delay Target

bits. The mean request rate (to achieve the desired quality target) obviously is higher, but decreases as the delay target at the source becomes larger. When the delay target is 50 milliseconds (the frame time is 41.6 milliseconds), then the mean request rate is almost 3.24 times the mean ideal rate. We need to request a large rate from the network to ensure that the buffer occupancy is kept low to accommodate the 50 ms delay target. However, when we go to a target delay of 100 milliseconds, the mean request rate is 1.68 times that of the ideal. The larger the source buffer, the smaller the overall rate that needs to be requested from the network while still meeting our quality target. Also shown in Figure 15 is the behavior of the peak for the ideal and requested rates. The peak requested rate drops rapidly from 296400 bits for a source target delay of 50 milliseconds, to 164653 bits for a delay of 90 milliseconds. However, the peak rate does not drop further, and in fact remains flat for this trace, when the delay target increases further from 100 to 120 milliseconds. It is important to note that the ratio of the peak ideal rate to the peak requested rate (when the target delay is 90 milliseconds) is about 2.06, which is a substantial benefit derived from using SAVE.

Table 3 shows more details on the degradation in quality relative to the source target delay, varying from 50 to 120 milliseconds. Also shown is the mean delay at the source. The 99.99th percentile and higher (also the maximum delay) are consistently below the target delay we chose for the particular experiment.

We observe that the proportion of frames that suffer a degradation in quality increases slightly with increasing delays – a somewhat counter-intuitive result. This is because the uncertainty increases as the size of the source buffer increases. For larger target delays, we should see the source buffer is full enough to cause a subsequent frame to be cropped slightly more frequently. However, no frame suffers a degradation of 50% or more in any of these experiments. We are also within the quality target of no more than 0.1% of frames suffering more than 20% degradation.

Target Delay (ms)	Cropping $\geq 0\%$	Cropping $\geq 20\%$	Mean Delay (ms)
50	0.0012	0.00017	13.9
60	0.0014	0.00022	16.7
70	0.0017	0.00034	19.3
80	0.0023	0.00071	21.9
90	0.0032	0.00106	24.5
100	0.0041	0.00120	27.1
110	0.0048	0.00145	29.8
120	0.0054	0.00182	32.4

Table 3. Proportion of frames suffering a degradation in Quality as a Function of Source Target Delay

Over-request β	Cropping $> 0\%$	Cropping $> 20\%$	Mean Delay (ms)
0.9	0.0886	0.0212	34.2
1.0	0.0082	0.0018	27.8
1.05	0.0032	0.0011	24.5
1.10	0.0022	0.0007	23.0
1.20	0.0007	0.0002	20.8
1.30	0.0003	0.0001	19.1

Table 4. Proportion of frames suffering a degradation in Quality as a Function of Source Rate Over-request

It is clear from the above that there is a tradeoff between the source target delay and the rate requested of the network. The requested rate goes up as the target delay becomes smaller. Note that SAVE is still able to maintain quality within acceptable levels even while meeting the much smaller delay target.

Sensitivity to Over-request of Network Rate. As described in Section 2, the SAVE algorithm systematically over-requests a rate from the network by a factor $\beta > 1$ relative to its calculated needs; see equation(3). This allows a built-up source buffer to drain while awaiting feedback from the network, especially on arrival of a large frame. Subject to achieving our quality targets, the smaller we can make β the better, since this will use less network resources.

In the results reported so far, we have used a β of 1.05. In this subsection, we examine the sensitivity to β, using

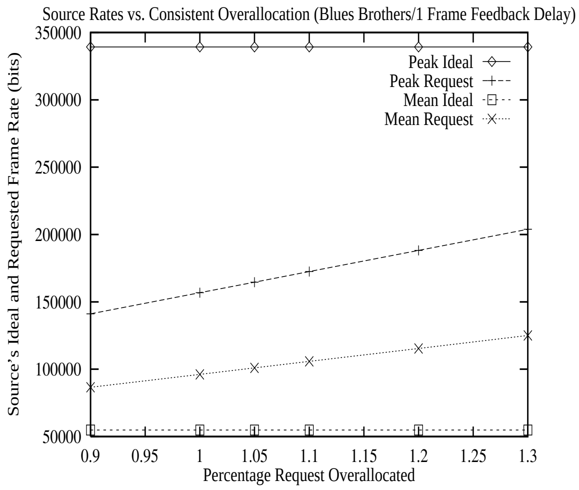

Figure 16. Behavior of the Requested Rate for Trace A for varying extent of over-request

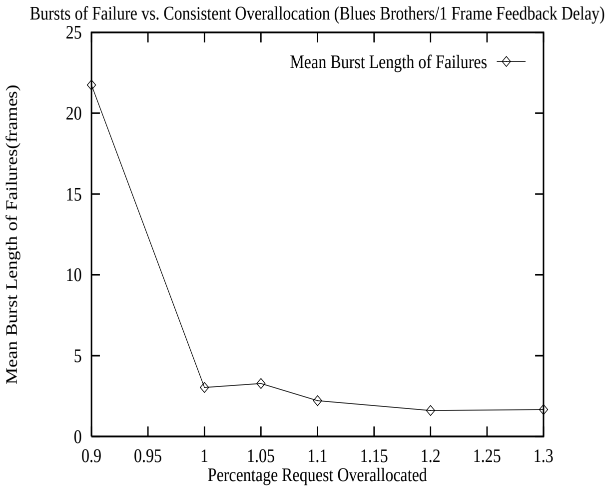

Figure 17. MEAN BURST LENGTH OF CROPPING > 20%. Trace set A, for varying extent of over-request.

trace A, varying β from 1.3 down to 0.9, especially with respect to the extent to which frames are cropped. When β is 1.0, this implies that there is no over-request. For β of 0.9, we are persistently asking for only 90% of the required rate, causing a buildup at the source buffer. This has the potential of degrading the quality, since it would trigger the reduction in the target rate to the encoder and result in cropping the frame.

Figure 16 shows the mean and peak for the requested rate, with varying β. As expected, the peak request rate increases linearly with β. The mean requested rate has a slightly lower slope reflecting a small amount of short-term smoothing that occurs with the source buffer. Also shown in the figure are the mean and peak values for the ideal rate for comparison. As observed before, the requested rate is higher than the long-term mean ideal rate, but substantially lower than the peak ideal rate. However, we believe that the increase in the requested rate is small, even with β of 1.3, but the improvement in the quality we observe appears to be worthwhile.

Examining quality, when β = 1.0, Table 4 shows that the proportion of frames suffering cropping greater than 20% is 0.18%, slightly above our target of 0.1%. However, even when we consistently only ask for 90% of the rate required, the percentage of frames that suffer a cropping greater than 20% is still only 2%. In fact less than 9% of the frames suffer *any* cropping at all. SAVE also doesn't allow the mean delay in the source buffer to grow excessively. With β of 0.9, the mean source delay is 34.2 milliseconds, still quite reasonable.

We also examined the impact of β on the dynamics of cropping, shown in Figures 17 and 18. The mean burst length of cropping $>$ 20% is relatively small, ranging from less than 1 to about 3 frames, as long as β is 1.0 or greater. The lengths of these bursts are relatively insensitive for β greater than 1.0. However, when β =0.9, we see that the mean burst length of cropping $>$ 20% grows quickly to over 20, indicating that the quality may not be acceptable. Thus, it is useful to look more carefully into the dynamic behavior of SAVE, to show that in fact we are impairing quality by bringing β down to 0.9.

It is revealing to examine the effect of β on the length of the complementary bursts of consecutive frames cropped less than 20%. Increasing β from 1.0 to 1.3 dramatically increases the mean complementary burst length from near 2000 frames (a duration of over 1 minute) to nearly 14,000 frames (over 7 minutes). Getting β down to 0.9 results in a mean length of complementary bursts of about 250 frames. This is comparable to the mean burst length of frames cropped $>$ 20%, of nearly 25 frames. We believe that there is a benefit in having a $\beta \geq 1.0$, so that we can have a long enough period of almost unimpaired video.

We saw in Section 6 that SAVE is reasonably robust with respect to statistical reduction in the allocated rate that occurs as a result of network congestion. The results of this section show that SAVE is also robust with respect systematic rate-reductions, when the allocated rate is lower than the requested rate.

8 Conclusions

In this paper we have examined the quality that can be achieved with an online source smoothing algorithm for compressed video, called SAVE. The adaptation of com-

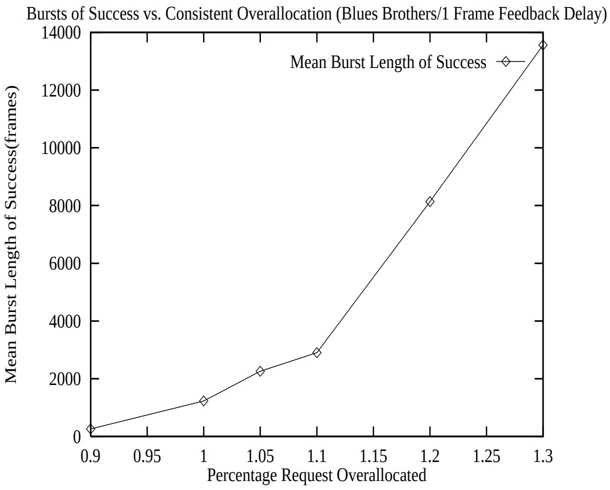

Figure 18. MEAN BURST LENGTH OF CROPPING $<$ 20%.. Trace set A, for varying extent of over-request

pressed video is based on negotiation of the bandwidth with the network using feedback-based congestion control and adapting the quantization parameter when necessary. For this form adaptation, it appears necessary to evolve the quality metrics typically used to evaluate the efficacy of mechanisms to transport video. The metrics relate to how frequently we have to adapt the video and by how much. We are quite stringent in setting quality targets for SAVE, suggesting that no more 1 in 1000 frames suffer more than a 20% cropping. Further, we examined the burstiness of the impairment, because we believe that having a long string of consecutive frames that are cropped results in noticeable degradation in quality. Our paper examined how effective SAVE is, in meeting our quality targets and builds on the results reported in [2]. We also looked at the "traditional" measures of effectiveness, in the context of our algorithm: the multiplexing gain, the source buffer delay and the behavior of the rate requested from the network.

We believe our algorithms need to be robust to variations in the feedback delay and to network congestion, failing which we would see buffer occupancy at the source increase. We find that SAVE's adaptation based on smoothing the short-term variations in ideal frame rate (of the order of a GOP) and tracking the maximum ideal frame size over a scene time-scale allows us to be relatively insensitive to feedback delay.

Smoothing and aggregation of flows help reduce variability in the rate requested of the network. When we go to a 38 fold aggregation of the traces in set E, the various quantiles of the requested rate (ranging from 100% to 50%) become quite close to the mean aggregate requested rate. (Although we don't detail it here, the same convergence of quantiles holds for the ideal rate; see [2]. But we find that convergence to the mean is faster for the requested rate.) In addition, SAVE tolerates a reasonable amount of reduction in the rate allocated to the flows. With a 5 fold aggregation of set E, we see that we can tolerate the situation where the capacity of the network is as low as 70% of the peak requested rate (in this case), and still maintain acceptable quality.

We examined the sensitivity to network congestion by considering the effect on one trace (trace A) when there are 10 other traces (from set E) that are multiplexed together. Even when the allocated rate from the network is 90% of the requested rate, we are able to maintain quality in all the different dimensions: delay, the overall proportion of frames that are cropped more than 20% and the size of the burst of frames that are consecutively cropped.

Taking this 90% of the requested rate being allocated by the network as a reasonable rule of thumb for SAVE to compare multiplexing gain, we compared the benefit of the rate-adaptive video (SAVE) with a source-buffered CBR model. We see that for equivalent quality, the bandwidth required by SAVE (90%ile) is 60% of the peak bandwidth required by a buffered CBR service. And this benefit of SAVE only seems to grow with aggregation size, a highly desirable characteristic. Moreover, SAVE has lower sensitivity to the bursty cropping of frames when the rate allocated by the network is less than the requested rate. Buffered CBR has a significantly larger burst of frames that would be cropped if the allocated rate from the network reduces below 90% of what is desired.

We demonstrated SAVE's sensitivity to its parameters - the target of the source buffer delay, the amount by which the source "over-requests" a rate from the network, and its smoothing windows. There is a tradeoff between the source target delay and the rate requested (especially the mean rate) of the network. The requested rate goes up as the target delay becomes smaller. But, SAVE is still able to maintain the quality targets with a smaller delay target - all the way down to 50 milliseconds. We showed that it is important to understand the dynamic nature of the quality reduction, where long bursts of frames are impaired. When the source systematically under-requests the rate needed from the network (asking for 90% of the rate needed by SAVE), it appears that the average "quality" measured over the length of the trace (trace A) doesn't seem to degrade much. But we suffer in having longer bursts of frames that are impaired in quality.

Finally, we showed that the most significant parameter in SAVE is the maximum smoothing window $w_{\max}$. The proportion of frames cropped decreases as a power law with $w_{\max}$, and the mean requested rate goes up logarithmically with $w_{\max}$. $w_{\max}$ serves as a dial that we may use to tradeoff between quality and the average of the rate we request from the network. The short-term smoothing window, $w_{\rm sm}$ helps us in reducing the peak requested rate from the network, without much impact on the quality. The maximum requested rate goes down as $w_{\rm sm}$ goes up to about a GOP, beyond which we see diminishing benefits. Thus, the combination of $w_{\max}$ and $w_{\rm sm}$ can be used manage the resources that we use from the network, while managing quality to be within acceptable limits.

We believe SAVE demonstrates considerable promise as a method of online rate-adaptation for compressed video.

Acknowledgments

We gratefully acknowledge the assistance of Partho Mishra in letting us use the network level simulator he has developed. We thank Frank Kelly, Peter Key, and the anonymous reviewers for their comments.

References

[1] A. Charny, K.K. Ramakrishnan, A. Lauck, “Time Scale Analysis and Scalability Issues for Explicit Rate Allocation in ATM Networks”, *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, August 1996.

[2] N.G. Duffield, K.K. Ramakrishnan & A.R. Reibman, “SAVE: An algorithm for smoothed adaptive video over explicit rate networks”, Proceedings of IEEE Infocom' 98, San Francisco, April 1998.

[3] M.W. Garrett, W. Willinger, “Analysis, modeling and generation of self-similar VBR traffic”, In *Proceedings ACM Sigcomm 94*,London, UK, August 1994, pp.269-280.

[4] B. Girod, ”Psychovisual aspects of image communications”, *Signal Processing*, vol. 28, pp. 239-251, 1992.

[5] M. Grossglauser, S. Keshav, & D. Tse, “RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic”, Proceedings of the ACM SIGCOMM' 95 Conference, Sept. 1995.

[6] H. Kanakia, P.P. Mishra, A.R. Reibman, “An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport”, Proceedings of the ACM SIGCOMM' 93 Conference, Sept. 1993.

[7] T.V. Lakshman, P.P. Mishra, K. K. Ramakrishnan, “Transporting Compressed Video over ATM Networks with Explicit Rate Feedback Control”, Proceedings of the IEEE Infocom 1997 Conference, Kobe, Japan, April 1997.

[8] T. J. Ott, Lakshman T. V., Tabatabai A., “A Scheme for Smoothing Delay Sensitive Traffic Offered to ATM Networks”, Proceedings of IEEE Infocom 1992, pp. 776-785, May 1992.

[9] A.R. Reibman and A.W. Berger, “On VBR video teleconferencing over ATM networks,” in IEEE Global Telecommunications Conference (GLOBECOM), December 1992.

[10] A. R. Reibman and A. W. Berger, ”Traffic descriptors for VBR video teleconferencing over ATM networks”, *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 329-339, June 1995.

[11] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, D. Towsley, “Online smoothing of live, variable bit-rate video”, Proceedings NOSSDAV, 1997.

[12] O. Rose. ”Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems”. University of Wuerzburg. Institute of Computer Science Research Report Series. Report No. 101. February 1995.

[13] “ATM Forum Traffic Management Specification Version 4.0,” af-tm-0056.00, ATM Forum, April 1996, available as: ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.pdf

[14] James D. Salehi, Zhi-Li-Zhang, James F. Kurose, Don Towsley, “Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing”, Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, pp. 222-231, May 1996.