Title
An ATM service architecture for the transport of adaptively encoded live video

Authors
Vickers, Brett J.
Suda, Tatsuya

Publication Date
1995

Peer reviewed

An ATM Service Architecture for the Transport of Adaptively Encoded Live Video

ICS Dept. Technical Report #95-45

Brett J. Vickers and Tatsuya Suda
(bvickers@ics.uci.edu) and (suda@ics.uci.edu)

Department of Information and Computer Science,
University of California, Irvine — Irvine, CA 92697-3425

Abstract

The transport of high quality live video requires a video service that does not rely heavily upon accurate *a priori* traffic parameter estimation (as VBR does) and does not unnecessarily sacrifice multiplexing gain and visual quality (as CBR does). A new service capable of supporting the transport of video on a guaranteed as well as a best effort basis is introduced. This service, called quasi-VBR service, is based on the ATM Forum's EPRCA rate control algorithm [4], a modified version of Sriram's T_1/T_2 service discipline [7], and an adaptive video encoding scheme. The results of a simulation study in which quasi-VBR video and ABR data traffic interact are presented. Actual MPEG-1 video sequences are used. Results show that quasi-VBR video service is capable of rapidly adapting to changes in network conditions, providing high quality video service and excellent network resource utilization.

1 Introduction

Applications requiring high quality, live video will undoubtedly become more common as public broadband and private local area ATM networks continue to emerge. Video teleconferencing, live video broadcast, interactive video games, and video surveillance are but a few of the applications that fall into the category of high quality, live video. However, existing ATM network services, namely variable bit rate (VBR) service and constant bit rate (CBR) services, fail to meet many of the needs of live video.

VBR service provides a statistical multiplexing gain and supports guaranteed quality of service through preventive congestion control techniques. Yet preventive congestion control techniques require the video source to accurately predict its traffic characteristics at call admission time. For stored video, *a priori* estimation of traffic characteristics is straightforward; the traffic characteristics are already fully known prior to transmission. For compressed live video, however, prediction of traffic characteristics is often difficult, if not impossible, to perform accurately.

Due to live video's lack of predictability, CBR service is often suggested as the more appropriate service. By adaptively encoding the video signal to achieve a fixed output rate, the CBR source renders its video output process constant and thereby predictable. Unfortunately, in order to obtain traffic predictability, CBR compromises multiplexing gain and causes the visual quality to fluctuate, sometimes severely [1]. Furthermore, it is still difficult to accurately estimate traffic characteristics in advance. A video sequence with a large amount of motion content creates large fluctuations in bit rate, requiring a higher bit rate to maintain visual quality. Yet, the amount of motion in a live video sequence is not always predictable.

High quality live video requires a video service that does not rely heavily upon accurate *a priori* traffic characteristics estimation (as VBR does) and does not unnecessarily sacrifice multiplexing gain and visual quality of video (as CBR does). The service should also provide a minimum guarantee of bandwidth to support at the very least a minimal quality of service. However, the service must also allow the video source to exceed the minimum guarantees when the network has unutilized capacity.

We propose a service that can support video on a guaranteed basis as well as a best effort basis. We further show how it may be supported through a standardized network rate control algorithm, a simple service discipline, and adaptive video encoding. For the remainder of this paper, we shall refer to this class of service as quasi-VBR service.

This paper differs from other studies of rate-based, adaptively encoded video [2, 3] in that it focuses on the required network service architecture. The paper also differs from [2, 3] by exploring the interaction between quasi-VBR video traffic and another unpredictable form of traffic: available bit rate (ABR) data traffic.

In section 2, we present the envisioned quasi-VBR service architecture. The rate control algorithm, the service discipline, and the adaptive encoding scheme are described. Section 3 presents the performance model used to study the quasi-VBR architecture. We discuss experimental results in section 4 and provide some concluding comments in section 5.

2 Service Architecture

VBR and CBR services are guaranteed services; they receive only the amount of network service they request at call admission time. ABR service, on the other hand, relies almost entirely on best effort service. Most ABR traffic is served only when the network has unutilized or unassigned bandwidth to support it. Quasi-VBR service falls between these two service regimes. It requires guaranteed service from the network, but it is also capable of using unutilized bandwidth when it becomes available. The architecture required to provide quasi-VBR service for video consists of three major components: the rate control algorithm, the service discipline, and the adaptive video encoding scheme. These three components are described below.

2.1 Rate Control

For rate control of the quasi-VBR source, we employ the ATM Forum's Enhanced Proportional Rate Control Algorithm (EPRCA), already adopted for rate-based flow control of Available Bit Rate (ABR) data service [4]. EPRCA is a closed loop flow control algorithm that provides rate control on a per-VC basis. Furthermore, it is a "positive feedback" rate control algorithm, where the source does not increase its sending rate unless the network explicitly gives it permission. When the source is not receiving network feedback, it continually decreases its sending rate by an amount proportional to the current sending rate. Contrast this scheme with "negative feedback" algorithms that reduce the sending rate only when feedback indicates congestion and otherwise continually increase the sending rate [6]. EPRCA is more robust, because it continually decreases the source's sending rate, thereby anticipating congestion. It also increases the sending rate only when the network is *un*congested.

EPRCA creates a closed feedback loop for each VC by looping congestion-indicating resource management (RM) cells between the source and destination. The source sends an RM cell for every `Nrm` data cells sent. When the destination receives an RM cell, it sends a backward-direction RM cell back to the source. The payload of this backward RM cell contains a binary congestion indication (`CI`) field that is set to `1` if (a) the destination is currently congested, or (b) the last data cell received by the destination had its cell header marked for congestion. On its return path to the source, the backward RM cell initiates a congestion check at each switch hop. If there is congestion on the path from the data source to the destination, then the switch marks the congestion indication bit in the backward RM cell to `1`. Upon receiving a backward RM cell, the source checks the cell's congestion indication field. If `CI=1`, i.e. if congestion is indicated, the backward RM cell is ignored and the sending rate continues to decrease. However, if `CI=0`, i.e. if there is no congestion on the path from the source to the destination, the sending rate at the source is additively increased. Fig. 1 shows the pseudo-code for the EPRCA algorithm.[1]

EPRCA was chosen to support quasi-VBR service for four reasons.

[1]This pseudo-code was taken from ATM Forum Document 94-0438R2 [5]. The pseudo-code for EPRCA's explicit rate mechanism has not been included.

EPRCA Source Pseudo-Code

```
Initialization

  ACR = ICR
  count = 0
  cell_sent = 0
  ADR = ShiftR(ACR, MDF) /* ADR=ACR/2^MDF */

While VC_on_line do

  if (now >= next_cell_time)
    if (ACR > ICR or cell_sent)
      ACR = max(ACR - ADR, MCR)
    endif

    if (cell_to_send)
      send data_cell with EFCI=0
      if (count = 0)
        send RM(DIR=forward, CI=0)
        ADR = ShiftR(ACR, MDF)
      endif
      cell_sent = 1
      count = (count + 1) mod Nrm
    else
      cell_sent = 0
    endif

    next_cell_time = next_cell_time + 1/ACR

    if (receive RM(DIR=backward, CI))
      if (CI = 0)
        ACR = ACR + Nrm*AIR + Nrm*ADR
        ACR = min(ACR, PCR)
        ACR = max(ACR, MCR)
      endif
    endif

  endif

end While
```

EPRCA Parameter Description

ACR	Allowable (Sending) Cell Rate
ICR	Initial Cell Rate
MCR	Minimum Cell Rate
PCR	Peak Cell Rate
ADR	Additive Decrease Rate
AIR	Additive Increase Rate
MDF	Multiplicative Decrease Factor
VC_CI	Congestion state of VC
CI	EPRCA Congestion Indication
EFCI	ATM Header Congestion Bit
DIR	Direction of RM cell
Nrm	One forward RM cell for every Nrm data cells

EPRCA Destination Pseudo-Code

```
Initialization

  VC_CI = 0

While VC_on_line do

  if (receive_data_cell)
    VC_CI = EFCI state of cell
  endif

  if (receive RM(DIR=forward, CI))
    if (VC_CI=1)
      CI = 1
    endif
    send RM(DIR=backward, CI)
  endif

end While
```

Figure 1: EPRCA Algorithm Pseudo-Code

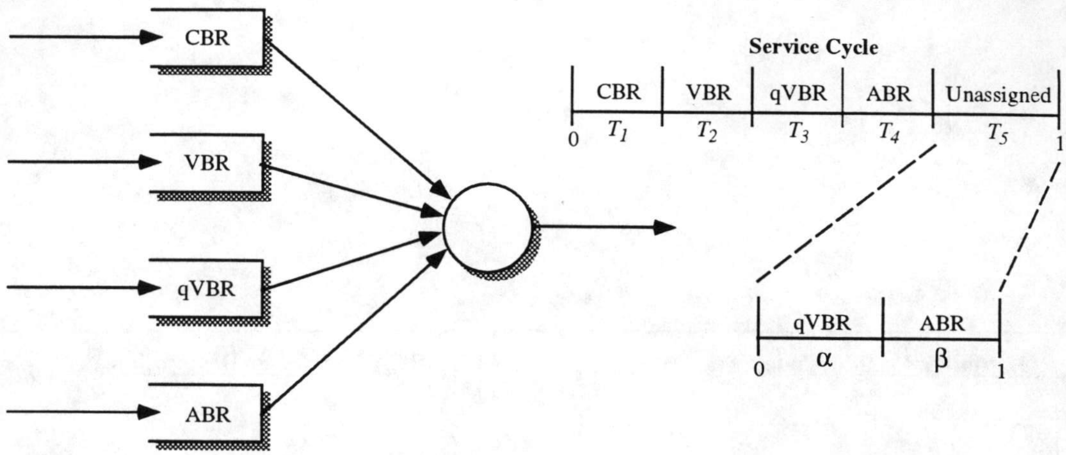

Figure 2: Service Discipline

1. It provides stability and throughput for bursty and unpredictable traffic. Compressed live video, with its relatively bursty behavior, benefits from a rate control algorithm that has already been shown to provide stability for unpredictable, bursty traffic.

2. It guarantees a minimum cell rate. The minimum cell rate guarantee allows live video some semblance of a quality of service, at least in terms of throughput.

3. It scales well. EPRCA has the capability to perform hop-by-hop flow control, rapid feedback, and congestion anticipation. These three mechanisms reduce the impact of large propagation delays.

4. It is a standard. EPRCA is already used for ABR data traffic in ATM networks. Thus, supporting quasi-VBR service requires no new rate control algorithms.

2.2 Service Discipline

In addition to providing feedback and rate control, quasi-VBR service must be a part of the service discipline utilized in the output ports of ATM switches. We study a simple FCFS service discipline adapted from the T_1/T_2 scheme [7]. In our modified service discipline, each service class — CBR, VBR, quasi-VBR and ABR — is allocated a FIFO queue in each output port of each ATM switch. Service between the four queues is cycled, with the service cycle divided to give connections within each service class their guaranteed quality of service. Any fraction of the service cycle unused by a service class is offered to the quasi-VBR and ABR queues on a best effort basis.

Fig. 2 illustrates this service discipline in more detail. Let T_1, T_2, T_3 and T_4 denote the fraction of link capacity assigned to the guaranteed service of the CBR, VBR, quasi-VBR and ABR queues, respectively. Let T_5 denote the fraction of link capacity that is unassigned. This unassigned bandwidth is divided between quasi-VBR and ABR services since they both require best effort service. Let α denote the fraction of unassigned link capacity allocated to quasi-VBR service, and let β denote the fraction of unassigned link capacity allocated to ABR service ($\alpha + \beta = 1$).

During call admission, the value of T_5 is observed to determine whether there is enough link capacity to support the new call. If not, the call is rejected. Otherwise, link capacity is transfered by granting a portion of the unassigned link capacity to the service required by the new call. For example, acceptance of a CBR call entails increasing T_1 at the expense of T_5. Conversely, when a call terminates, link capacity is returned to the unassigned bandwidth pool.

The service discipline entails the use of a fixed-length service cycle. During this service cycle, a total of up to S cells is served from the four service queues. The four queues are served in a round-robin fashion until they each receive their guaranteed fraction (T_i) of the service cycle and, thereby, the link capacity. The CBR and VBR queues, which receive only guaranteed service, are served up to $\lceil T_1 \times S \rceil$ and $\lceil T_2 \times S \rceil$ times per service cycle, respectively. The quasi-VBR and ABR queues, which benefit from the unassigned bandwidth on the link, receive up to $\lceil (T_3 + \alpha T_5) \times S \rceil$ and $\lceil (T_4 + \beta T_5) \times S \rceil$ guaranteed cell services per cycle, respectively.

Counters are used to keep track of the total number of cells sent from each queue during the service cycle. In the case of CBR or VBR, once the queue's counter reaches its guaranteed number of services for the cycle, the queue may receive no more services during that cycle. However, in the case of quasi-VBR or ABR, the queue may receive service even after the counter reaches its guaranteed number of services. This is because quasi-VBR and ABR services receive best effort service in addition to guaranteed service, and their queues may be served if the VBR and CBR queues exhaust themselves before receiving their full guaranteed service. If both the quasi-VBR and ABR queues have received all their allocated services by the time a best effort service opportunity arrives, then the two queues receive alternating best effort cell services.

While this modified T_1/T_2 service discipline is studied primarily due to its simplicity, other rate based service disciplines, such as Earliest-Due-Date [8] or Fair Queueing [9], may also be used with a quasi-VBR service.

2.3 Adaptive Encoding

In order to respond gracefully to congestion feedback, the video encoder must alter its behavior dynamically. Simply discarding or buffering video data is not an ideal way to achieve high quality, realtime video service. More adaptive solutions for responding to rate-based network feedback include alteration of the video frame rate and alteration of the lossy encoder's quantization parameters.

Altering the frame generation rate. To meet the target sending rate established by the rate control algorithm, a source may alter its frame rate. A video source transmitting at 30 frames per second, for instance, may respond to a congestion indication by encoding every other frame. When congestion subsides, the source may restore the original frame rate. While this technique is adaptive, it only adjusts the sending rate by an amount equal to the size of a video frame. Thus, its adaptability is too coarse-grained. Furthermore, even slight reductions in frame rate can become obvious to the user, especially when the video sequence contains a significant amount of motion.

Altering the quantization of the encoding process. The quantization stage of digital video encoding introduces an element of information loss. This loss may be controlled to dynamically adjust the video source's bit rate to meet the target sending rate established by the rate control algorithm. Loss adjustment is controlled by the encoder's quantization step (Q step) parameter. The Q step is an integer that varies between 1 and 31, with higher values resulting in more information loss and thereby a lower bit rate. By adjusting the Q step, the encoder can increase or decrease the number of bits generated in such a way that the resulting video quality is altered gracefully. One simple way to perform this Q step adjustment is to monitor the occupancy of the output buffer at the user-network interface. If the encoder varies the Q step according to buffer occupancy, an effective feedback loop between the network and the encoder is generated. More complicated Q step adjustment algorithms use fluctuation damping techniques [2] and multiple buffer monitors [10].

The chief advantage of adaptive Q step adjustment is that it provides a relatively fine-grained tuning of the video bit generation rate. Furthermore, small changes in Q step are barely visible to the human eye.

Because adaptive frame rate adjustment is too coarse-grained, we have opted for adaptive quantization as a mechanism for controlling the sending rate.

3 Performance Model

To study the service architecture described in section 2, the network model shown in Fig. 3 has been used to perform a series of simulations. Two types of service have been included in the model: ABR data and quasi-VBR video. CBR and VBR services have not been included due to their more predictable nature.

The network consists of three switches (SW1, SW2, SW3) with three bidirectional pairs of quasi-VBR video streams communicating between video sources V1 and V2. These video streams consist of actual video data with unique characteristics. A number of ABR data sources along the path between V1 and V2 offer an interfering load, with a fraction of the load destined for V2. With the exception of the 4 Mbps bottleneck link between SW3 and V2, all links have a carrying capacity of 50 Mbps. Each link's propagation delay is 5 ms, corresponding to a round trip delay of 40 ms for V1 and V2 — a value typical for a cross-continental connection.

The following describes the parameters and other details of the performance model, including the buffer sizes used in sources and switches, the characteristics of the quasi-VBR video traffic, and the characteristics of the interfering ABR data traffic.

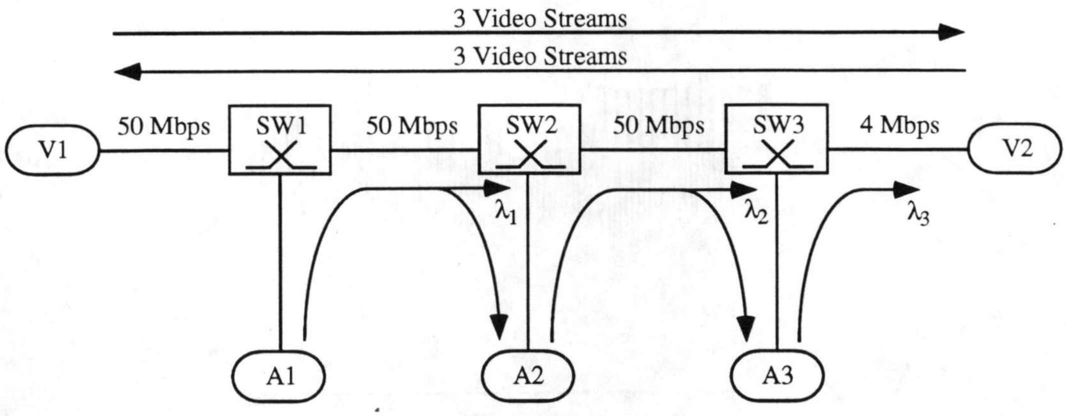

Figure 3: Performance Model

3.1 Source and Switch Buffers

The service discipline described in section 2.2 is implemented in the output ports of switches SW1, SW2 and SW3. For the purposes of this paper's performance study, the buffer space in each of the output ports is not shared and is divided evenly between the quasi-VBR and ABR services. In each output port of each switch, a total of 100 cell buffers are reserved for quasi-VBR service, while another 100 cell buffers are reserved for ABR service. As previously mentioned, CBR and VBR services are not considered. At V1 and V2, output video buffers of 100 cells are utilized. The buffer sizes used in this study result in a worst case quasi-VBR end-to-end delay of 87 ms for $\alpha = 0$ and 153 ms for $\alpha = 1$.

3.2 Quasi-VBR Video Traffic

The simulation utilizes a publicly available MPEG-1 encoder [11] capable of adaptively adjusting the Q step within the range $[1, 31]$. A small value for Q results in low information loss and thereby a higher signal-to-noise (SNR) ratio. Conversely, a large value for Q produces a significantly degraded video quality but a lower bit generation rate.

The three video sequences sent from V1 to V2 and from V2 to V1 are labeled A, B, and C, and have unique traffic characteristics as shown in Table 1. Sequence A has the highest information content of the three, although at a Q step of 8 each of the sequences produces nearly the same average bit rate. Figs. 4, 5 and 6 show the number of bits per frame for selected portions of the three video sequences at Q=1. The large fluctuations in frame size create bursty video behavior and reflect the sequence of the three different MPEG frame types: I, B and P. I frames are intracoded only. That is, when encoding an I frame, the encoder uses information only from the present frame. When encoding a P frame, or predicted frame, only the differences between the present frame and the last reference (I or P) frame are encoded. Finally, when encoding a B frame, the encoder performs bidirectional interpolation. It encodes differences between the present frame and the previous *as well as* the following reference (I or P) frames. I frames tend

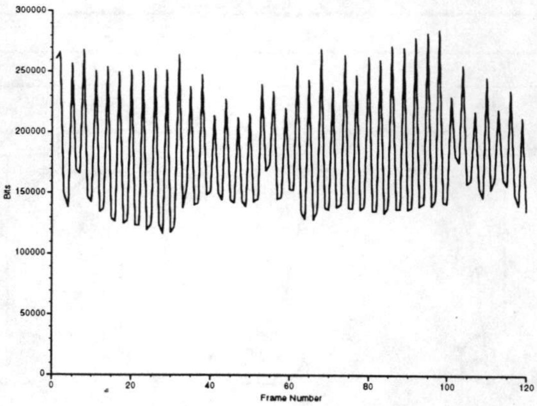

Figure 4: Video Sequence A

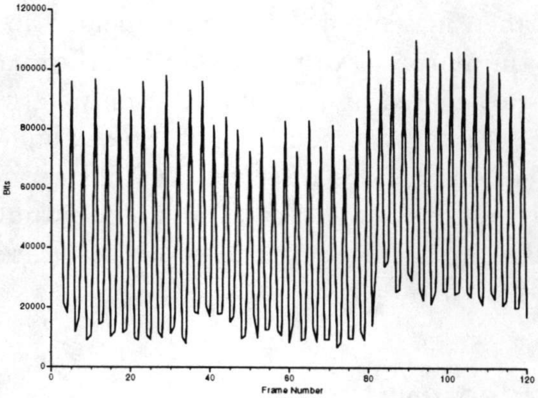

Figure 5: Video Sequence B

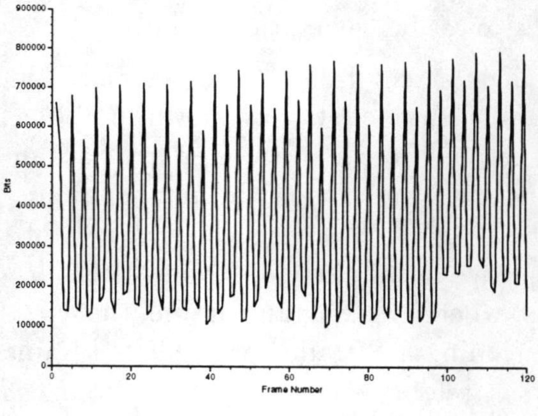

Figure 6: Video Sequence C

Video Sequence	Frame Rate	Mean bit rate (Q=1)	Mean bit rate (Q=8)
A	30 Hz	4.9 Mbps	1.1 Mbps
B	25 Hz	1.3 Mbps	0.6 Mbps
C	7 Hz	2.4 Mbps	1.1 Mbps

Table 1: Video Sequence Characteristics

to be largest, B frames the smallest, and P frames in between. Each of the video sequences used in the simulation are encoded with a repeating frame pattern of IBBPBB.

Although the three video sequences have very different traffic characteristics, they are given the same call admission parameters to simulate the difficulty of establishing accurate traffic parameters for live video in advance. They are each given a guaranteed minimum cell rate (MCR) of 1500 cells per second (approximately 0.6 Mbps) and a peak cell rate (PCR) of 118000 cells per second (approximately 50 Mbps). Furthermore, the video sequences begin with an EPRCA initial cell rate (ICR) equal to the minimum cell rate to prevent network overload immediately after call establishment. A forward RM cell is sent once for every 16 data cells sent ($N_{RM} = 16$), the Additive Increase Rate (AIR) is initialized to 0.5, and the Multiplicative Decrease Factor (MDF) is initialized to 10.

3.3 Interfering ABR Data Traffic

To study the interaction between quasi-VBR video service and ABR data service, interfering best effort data traffic has been introduced throughout the network. As with the quasi-VBR video traffic, the ABR data traffic is controlled by the EPRCA algorithm, though with a much smaller guaranteed minimum cell rate (MCR = 10 cells per second). Furthermore, ABR traffic does not share the same queues as quasi-VBR traffic in the output ports of ATM switches.

ABR data sources A1, A2 and A3 generate packets whose lengths are exponentially distributed with a mean size of 384 bytes, and whose packet interarrival times are exponentially distributed with a mean of 1/11800 seconds. After segmentation into ATM cells, the resulting Poisson ABR data load offered by A1, A2 and A3 is approximately 40 Mbps. A large portion of this ABR data is destined to travel only a single switch hop as shown in Fig. 3. The remaining ABR traffic is sent across the bottleneck link to V2. We denote the mean cell arrival rate of the interfering traffic destined for V2 as λ_i, where i indicates the ABR source.

In order to study the interaction between ABR data traffic and quasi-VBR video traffic on the bottleneck link, the nature of the ABR data traffic must first be characterized. Let us first define the *best effort capacity* of the bottleneck link as the amount of link capacity not assigned to supporting CBR, VBR, or the minimum guaranteed cell rates of quasi-VBR (i.e., $T_5 \times C$, using the terminology of section 2.2). The *ABR load* ρ is then defined as the fraction of best effort capacity actually used to deliver ABR data traffic. For our performance model in which there is no CBR or VBR traffic, the ABR load on the bottleneck link is given by the following equation.

$$\rho = \frac{(1 + 1/N_{RM})\sum_{i=1}^{m_1} \lambda_i}{C - (1 + 1/N_{RM})\sum_{i=1}^{m_2} MCR_i}.$$

The numerator describes the intensity of the ABR data offered to the bottleneck link, while the denominator gives the best effort capacity of the link. m_1 is the number of ABR data sources sharing the bottleneck link, and m_2 is the number of quasi-VBR video sources sharing the link. In our simulations, $m_1 = m_2 = 3$. C is the capacity of the bottleneck link (9000 cells per second). MCR_i is the guaranteed minimum cell rate for quasi-VBR video source i ($i = 1, \ldots, m_2$). N_{RM} is the EPRCA parameter indicating the number of data cells sent for each resource management cell. We assume that $N_{RM} = 16$ for both quasi-VBR and ABR services. All remaining ABR EPRCA parameters are identical to the quasi-VBR EPRCA parameters.

By varying the ABR load ρ, we are able to study the impact of ABR data traffic on the quality of the resulting quasi-VBR video.

4 Experimental Results

Both transient and regular behavior are investigated for the performance model presented in the last section.

4.1 Transient Behavior

To study of the impact of rapidly changing bandwidth availability on quasi-VBR service, we simultaneously initiate transmission of quasi-VBR video streams A, B and C but terminate their transmission at different times. Stream A transmits for a real time of 30 seconds, stream B terminates 20 real seconds into the simulation, and stream C terminates after only 10 real seconds. A small ABR load ($\rho = 0.18$) is applied, allowing the quasi-VBR connections to use most of the bottleneck link capacity.

Each time a quasi-VBR video stream terminates transmission, the remaining quasi-VBR streams take up the newly available bandwidth through the service architecture's combination of rate control and adaptive video encoding. Fig. 7 illustrates this effect by plotting the utilization of the bottleneck link as a function of time. Note the sudden drops in link utilization at $t = 10$s due to the termination of video sequence C. Shortly after this drop in utilization, streams A and B exploit the newly available link capacity as expected. The same effect occurs at $t = 20$s when stream B terminates and stream A exploits the unused link capacity.

The oscillations in utilization are due to the non-negligible propagation delay of the links and the binary feedback characteristic of EPRCA. When a source receives an RM cell indicating a lack of congestion, it increases its sending rate, often exceeding the ideal sending rate. Thus, short-term network congestion results. Until the congestion clears and an RM cell propagates back to the source, EPRCA forces the source to continually decrease its sending rate. This process repeats itself, resulting in the oscillation seen in Fig. 7.

Nevertheless, even with these oscillations, the average utilization of the bottleneck link is approximately 90%, an acceptable result for reactive congestion control in a wide area environment. The oscillations can be dampened by using the explicit rate control mechanism of EPRCA. However, we leave investigation of quasi-VBR service's use of the explicit rate control mechanism for future work.

Fig. 8 plots the peak signal-to-noise ratio (i.e., a measure of the difference between the original video sequence and the received video sequence) of stream A as a function of time. The solid line indicates the peak SNR averaged over each 10 second interval. Note that each time another video stream sharing the bottleneck link drops out, the peak SNR of stream A adaptively increases as expected. This figure illustrates the speed with which quasi-VBR video service can adapt to changing network conditions, improving visual quality whenever possible.

4.2 Regular Behavior

In addition to studying the transient behavior of the quasi-VBR service architecture, its time-averaged regular behavior is also investigated. In this set of simulations video streams A, B and C initiate bidirectional transmission simultaneously and continuously transmit for 30 real seconds. The ABR load on the bottleneck link is varied to study the interaction between quasi-VBR video and ABR data.

Fig. 9 plots the averaged peak signal-to-noise ratio of reconstructed video stream A as a function of the ABR load ρ. Three curves, corresponding to different values of α, are shown. Recall that α indicates the fraction of unassigned link capacity given to quasi-VBR service. A value of $\alpha = 0$ corresponds to the case where the bottleneck link's quasi-VBR queue receives best effort service only when there are no ABR cells to serve. Thus, the quasi-VBR queue receives only its guaranteed link capacity ($T_3 \times C$) plus any link capacity the ABR service fails to utilize. This setting of α results in a linearly decreasing quality of video as the ABR load approaches 1. Curves for two other settings of α are also shown, one for the case where quasi-VBR service is allocated 40% of the unassigned link capacity, and another for the case when it is allocated 80% of the unassigned link capacity. As Fig. 9 shows, the signal-to-noise ratio of stream A evens once the ABR load reaches the point where ABR traffic exceeds its allocation of unassigned link capacity. These results suggest that by shifting best effort priority from ABR data service to quasi-VBR video service, the network can stem the decline in video quality as ABR load increases. Streams B and C showed similar results.

Fig. 10 shows the effect of varying α on the utilization of the bottleneck link. Note that the closer α is to 1, i.e., the more unassigned bandwidth we grant to quasi-VBR service, the faster the utilization converges toward 100% as the ABR load increases. This result suggests that quasi-VBR video service is more stable than ABR data service. The reason for quasi-VBR's stability lies mostly in the continuous nature of video traffic. Because adaptively encoded video is continuous in nature, feedback cells are constantly generated, resulting in a steady and timely

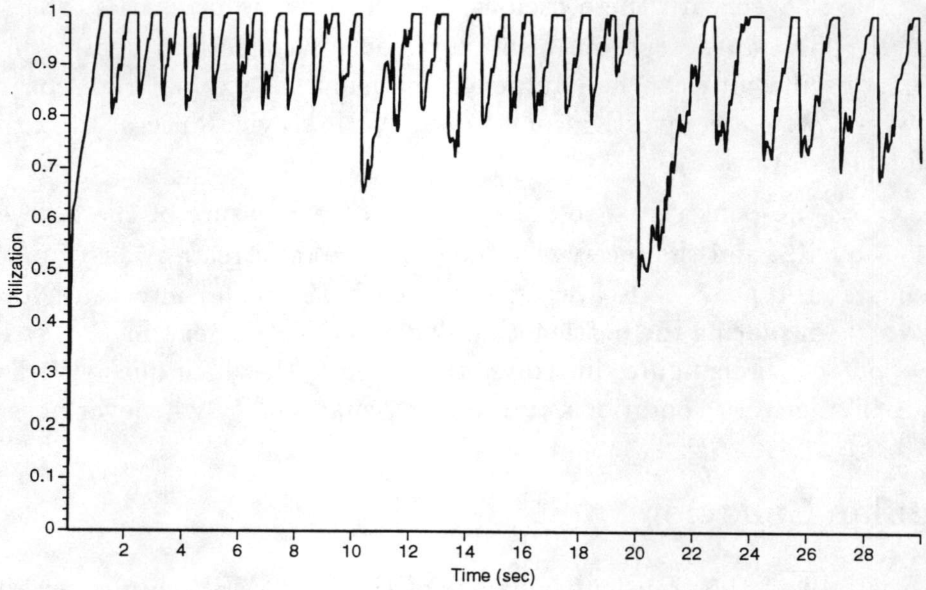

Figure 7: Bottleneck link utilization v. time

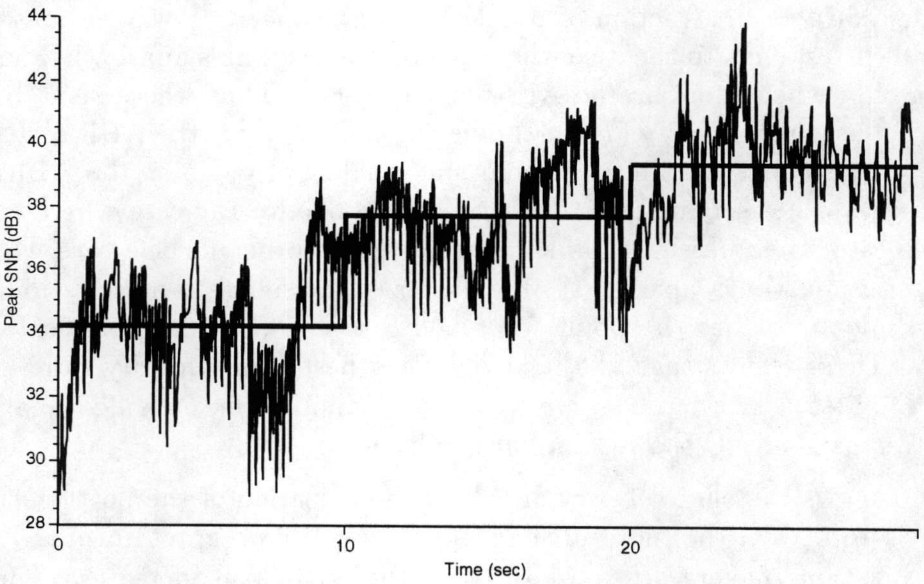

Figure 8: Peak SNR of sequence A v. time

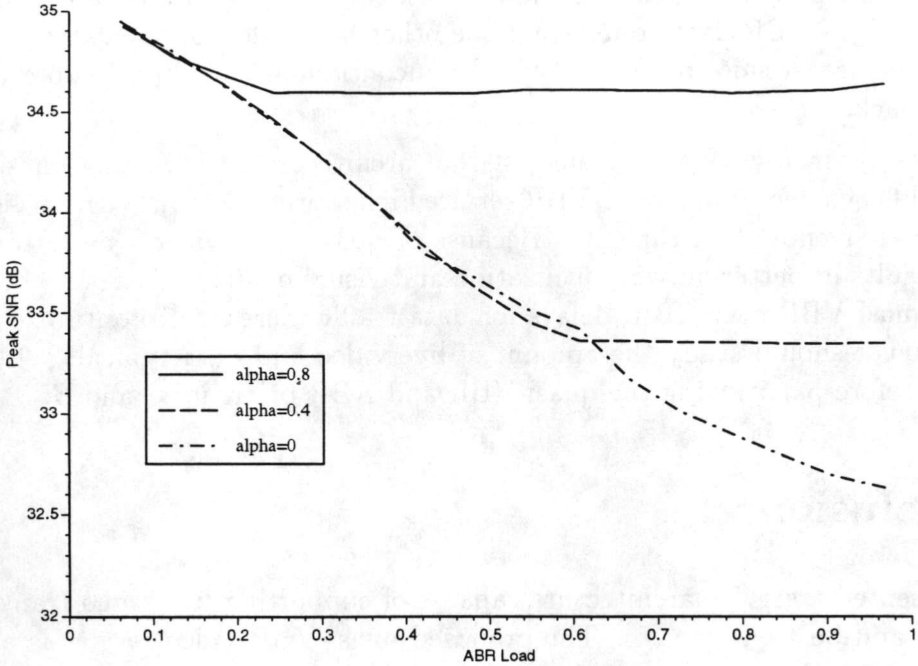

Figure 9: Peak SNR of sequence A v. ABR load ρ

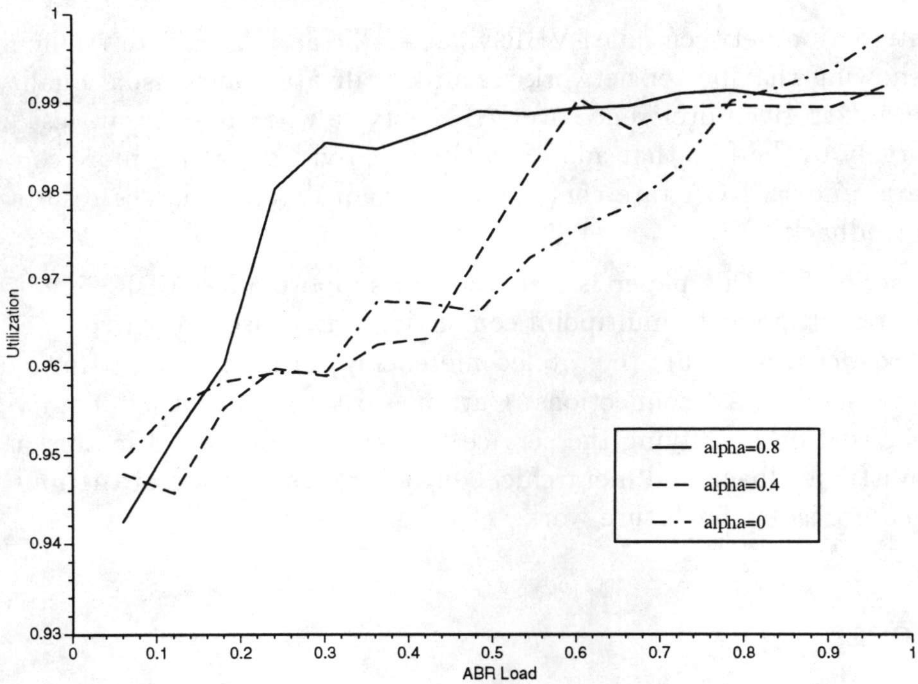

Figure 10: Bottleneck link utilization v. ABR load ρ

stream of congestion feedback to the video source. This timely feedback allows the quasi-VBR video source to adapt itself quickly to the network congestion state, producing traffic to exploit unused bandwidth. ABR data sources, on the other hand, do not typically provide continuous output. Packet data tends to be bursty and noncontinuous, yielding a more erratic stream of network feedback.

The results in Figs. 9 and 10 suggest that architectures supporting quasi-VBR and ABR services should consider giving quasi-VBR service higher priority than ABR service when it comes to doling out best effort link capacity. Because of quasi-VBR video's more continuous nature, favoring it results in better network utilization and visual quality. There is, however, a tradeoff; by favoring quasi-VBR over ABR, delays for data traffic increase. To optimize the tradeoff, the network manager should study the amount of live video and data typically offered by users on the network before partitioning the quasi-VBR and ABR priorities α and β.

5 Conclusion

We have presented a service architecture capable of supporting live video transmission on both a guaranteed and best effort basis. The proposed quasi-VBR video service is provided through a combination of EPRCA rate control, a modified T_1/T_2 service discipline, and adaptive quantization step encoding. Parameter estimation for live video is simplified by requiring the user to submit only a simple minimum cell rate at call admission time. By allowing the user to exceed the minimum cell rate through best effort service, the quasi-VBR service architecture improves network resource utilization and visual quality of the transmitted video.

The interaction between quasi-VBR video traffic and ABR data traffic has been studied, with results showing that better network resource utilization and visual quality are obtained if quasi-VBR service is given preference over ABR service when doling out best effort bandwidth. This is due largely to the fact that adaptively encoded video is much more continuous in nature than ABR data, and EPRCA rate control of continuous traffic sources results in a steady flow of congestion feedback.

Not described in this paper is a method to support quasi-VBR video service for point-to-multipoint or multipoint-to-multipoint connections. Because many live video applications are multicast or broadcast in nature (e.g., teleconferencing), it is important that quasi-VBR service be able to support multicast connections. Current work on supporting adaptive multicast video transport is focused on modifying the service architecture described in this paper through the use of probabilistic feedback and hierarchical encoding. Description of the multicast quasi-VBR service architecture is left for future work.

References

[1] A.R. Reibman and A.W. Berger. "Traffic Descriptors for VBR Video Teleconferencing Over ATM Networks," in *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 329–339, June

1995.

[2] H. Kanakia, P.P. Mishra and A. Reibman. "An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport," in *Proc. ACM SIGCOMM*, pp. 20–31, Sept. 1993.

[3] Y. Omori, T. Suda and G. Lin. "Feedback-Based Congestion Control for VBR Video in ATM Networks," in *Proc. of the 6th IEEE Workshop on Packet Video*, Sept. 1994.

[4] "ATM Forum Traffic Management Specification, Version 4.0," ATM Forum Document 95-0013R2, pp. 44-53, April, 1995.

[5] "Closed-Loop Rate-Based Traffic Management," ATM Forum Document 94-0438R2, Sept, 1994.

[6] P. Newman. "Backward Explicit Congestion Notification for ATM Local Area Networks," in *Proc. of IEEE Globecom*, pp. 719–723, Houston, Texas, Nov. 1993.

[7] K. Sriram. "Dynamic Bandwidth Allocation and Congestion Control Schemes for Voice and Data Multiplexing in Wideband Packet Telephony," in *Proc. of IEEE Globecom*, pp. 1003–1009, 1990.

[8] D. Ferrari and D. Verma. "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," in *IEEE J. Selected Areas of Commun.*, Vol. 8, No. 3, pp. 368–379, April 1990.

[9] A. Demers, S. Keshav and S. Shenker. "Analysis and Simulation of a Fair Queueing Algorithm," in *Proc. of ACM SIGCOMM*, pp. 3–12, 1989.

[10] Simulation Model Editorial Group. "MPEG Video Simulation Model Three (SM3)," ISO/IEC JTC1/SC2/WG11 Document MPEG90/041, pp. 54–55, July 1990.

[11] A.C. Hung. "PVRG-MPEG Codec 1.1 Documentation", Portable Video Research Group, Stanford University, June, 1993. Contact achung@cs.stanford.edu.