Transporting Compressed Video Over ATM Networks with Explicit-Rate Feedback Control

T. V. Lakshman, *Senior Member, IEEE,* Partho P. Mishra,
and K. K. Ramakrishnan, *Associate Member, IEEE*

***Abstract*— We propose a scheme for transmission of variable bit rate (VBR) compressed video for interactive applications using the explicit-rate congestion-control mechanisms proposed for the available bit rate (ABR) service in asynchronous transfer mode networks. Compressed video is inherently bursty, with rate fluctuations over both short and long time scales. This source behavior can be accommodated by the ABR service, since the explicit-rate scheme allows sources to request varying amounts of bandwidth over time. Moreover, when the bandwidth demand cannot be met, the network provides feedback indicating the bandwidth currently available to a connection. In our scheme, the video source rate is matched to the available bandwidth by modifying the quantization level used during compression. We use trace-driven simulations to examine how effective the enhanced explicit-rate scheme is in "rate matching" between the network and the source and the effect on end-to-end delay. We also look at the sensitivity of the proposed scheme to the estimates of the network round-trip times and to inaccuracies in the rate requests made by sources.**

***Index Terms*— ATM, congestion control, packet video.**

I. Introduction

COMPRESSED video traffic is likely to form a significant component of the workload of future networks. Compressed video is inherently bursty with rate fluctuations happening over both short and long time scales. Compressed video is also often rate adaptive, i.e., it is possible to modify the source rate dynamically by adjusting the compression parameters of a video coder. However, unlike rate-adaptive nonreal-time data transfer applications, interactive video requires tight end-to-end packet delay constraints. This delay bound is typically around 200–300 ms, for interactive applications such as video conferencing.

Transporting video over asynchronous transfer mode (ATM) networks has been an active area of research. The methods proposed for transport of compressed video span the spectrum of services offered by ATM networks: constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), and unspecified bit rate (UBR). The focus of the ABR service has been support for bursty data, where there is no clear specification of the source's characteristics [26]. In this paper, we propose a scheme for transmission of VBR compressed video for interactive applications based on the explicit-rate congestion-control mechanisms proposed for ABR.

Ideally, a transport mechanism for compressed video over a packet-switched network should ensure high statistical multiplexing gain, support frequent negotiation for bandwidth between the source and the network to accommodate source burstiness, define a mechanism to allow source rates to be adapted to match the available network bandwidth, achieve a very low frame loss rate, and ensure that end-to-end delays stay bounded. In this paper, we propose an enhanced version of the ATM ABR service, using the explicit-rate option, that allows each of these goals to be met.

In the explicit-rate ABR schemes, *in-band* resource management (RM) cells are periodically transmitted by each source to indicate its desired transmission rate. The network may mark this rate downwards before returning the RM cell back to the source if it is unable to provide the demanded bandwidth. The information returned in the RM cells may be used to adapt the bit rate of the video encoder. This provides a natural way of performing a rate negotiation between the source and the network. This rate renegotiation can be done very frequently, since the RM cell processing is performed in-band, and thus, is not constrained by the limitations of a slow shared signaling channel. Consequently, it is possible to exploit the high short-term correlation of video to accurately predict and renegotiate rates over very short time intervals, leading to higher statistical multiplexing gain.

In our proposed scheme, the video source rate is matched to the available bandwidth returned in the RM cell by modifying the quantization level used during compression. We believe that the overall perceptual quality of the video is likely to be higher with this form of source adaptation to congestion, where the source modifies the quantizer, compared to the situation where the network drops cells or packets under congestion (thus losing frames).

Another advantage of the explicit-rate ABR service is that, unlike traditional best-effort transport in data networks such as the Internet, it can guarantee a minimum bandwidth to individual connections by using admission control. This is particularly useful for video, since it can be used to ensure a minimum level of perceptual quality, even during periods of congestion. We envisage that this minimum bandwidth would have to be determined based on experience gained from

Manuscript received December 4, 1997; revised April 21, 1998; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Keshav. An earlier version of this paper appeared in *Proc. IEEE INFOCOM'96*.

T. V. Lakshman is with Bell Labs, Lucent Technologies, Holmdel, NJ 07733 USA (e-mail: lakshman@research.bell-lab).

P. P. Mishra and K. K. Ramakrishnan are with AT&T Labs-Research, Florham Park, NJ 07932-0971 USA (e-mail: partho@research.att.com; kkrama@research.att.com).

Publisher Item Identifier S 1063-6692(99)08523-4.

"human factors" experiments with a specific class of video and possibly from statistical models of compressed video sources.

The explicit-rate ABR service can also be tuned to keep the queueing delays fairly small by ensuring that the aggregate rate of all sources sharing a link is always less than the link bandwidth and by requiring each ABR source to maintain a smooth flow of cells at the current allowed transmission rate. Thus, the explicit-rate ABR scheme is also designed to achieve a low loss rate. However, for greater flexibility of operation, it may be necessary for switches to segregate ABR connections that are admission controlled and require low delays (e.g., video) from those that are not admission controlled (e.g., bursty data), using a suitable scheduling policy.

Explicit-rate ABR schemes typically attempt to achieve max–min fair fairness when determining how to allocate bandwidth among the set of active connections at each link. Max–min fair allocation results in all connections that are "bottlenecked" at a link being assigned an equal share of the bandwidth. While such a policy may be appropriate for data flows, it is likely to be inappropriate for video flows. For video, there is a relationship between the ultimate "quality of service" achieved by the flow and the degree of rate reduction experienced by the flow relative to its demand. Intuitively, an encoder that is compressing more complex images requires a greater number of bits to obtain the same level of quality and is likely to request more bandwidth than a source transmitting frames for a low-complexity sequence of images. If the network were to simply apply the max–min fairness criterion, then the high-complexity (possibly more activity) video flow would experience a much greater degradation in the video quality compared to the low-complexity flow. To address this issue, we propose the use of a "weighted" max–min fair-share allocation policy. In this policy, a weight is associated with each connection based on the bandwidth demanded for that connection. The link capacity is then divided in proportion to these weights. This weighted max–min fair-share allocation policy mimics the operation of weighted fair queueing [5], [9].

The rest of this paper is organized as follows. In the next section, we describe related work. Sections III and IV provide an overview of the explicit-rate feedback-control mechanism and the proposed enhancements for fair allocation of network bandwidth to video sources. In Section V, we present statistical models for compressed video, which are used for predicting the source's demand. Section VI discusses the source-adaptation mechanisms that we have explored for rate matching the encoder's rate to the network's feedback. Section VII presents simulation results, and concluding remarks are in Section VIII.

II. Related Work

There has been a considerable amount of research over the last few years in investigating various mechanisms for transporting video traffic over communication networks. The proposals using ATM networks may be classified based on the ATM service class that is used.

1) *CBR Transport:* In this mode, the inherently bursty (VBR) output of a video coder is locally buffered at the coder to convert it into a CBR stream. Since the buffering is limited by delay constraints, local feedback is used to adjust the quantizer to prevent buffer overflows and underflows. This results in a variable quality. The advantage of this scheme is that the CBR nature of the stream makes admission control simple. However, the penalty is that there is no attempt to exploit any multiplexing gains possible in the original VBR traffic.
2) *VBR Transport:* In this mode, the traffic generated by the coder is transported in a completely unrestricted (open-loop) manner over the real-time VBR service class [7], [18]. In principle, this results in constant quality. Moreover, since the ratio of the peak rate to the mean rate for compressed video traffic is quite high, there is potential for multiplexing gain and the "effective" bandwidth needed to be less than that for CBR video of the same quality.[1] Admission control for the real-time VBR service requires an accurate source model and an accompanying policing mechanism that ensures that sources indeed conform. Due to this latter requirement, source models in practice are restricted to be simple, characterized by only a peak rate, average rate, and a maximum burst size (this behavior can be easily policed using leaky buckets). Such a simple source model forces admission control to be conservative, since the lack of statistical information regarding source behavior, except the independence of sources, necessitates worst-case assumptions. Hence, even if the coder's output rate is adjusted to conform to these simple traffic descriptors, there is loss of efficiency. Variations of VBR have also been explored that allow a compressed video source to generate data at a VBR while adapting the rate downwards during periods of network congestion using feedback information from network switches [13], [22], [23].
3) *Renegotiated CBR (RCBR):* In this mode, a video coder generates a piecewise linear CBR stream with periodic "renegotiation" of the bit rate between the coder and the network. It is based on the observation that compressed video traffic exhibits rate fluctuations happening over both short and long time scales. In RCBR, short-term fluctuations in the bit rate of a compressed video source are absorbed in the source buffer as in CBR. However, when the source detects an increase or decrease in the bit rate that is likely to persist for a long time, it renegotiates the transmission rate. Thus, RCBR may be viewed as a hybrid of the CBR and VBR approaches that attempts to combine the simplicity of admission control for CBR with the statistical multiplexing advantages of VBR [8]. In the event of a "renegotiation failure,"[2] a source is forced to adapt its coding parameters to match the currently available transmission rate. Since the renegotiation is source initiated, there is no mechanism

[1] This argument is complex to test and quantify.

[2] RCBR appears to make the implicit assumption that such an event is very unlikely. However, guaranteeing a very low renegotiation failure probability requires an admission control check that is of equal complexity to admission control for unrestricted VBR.

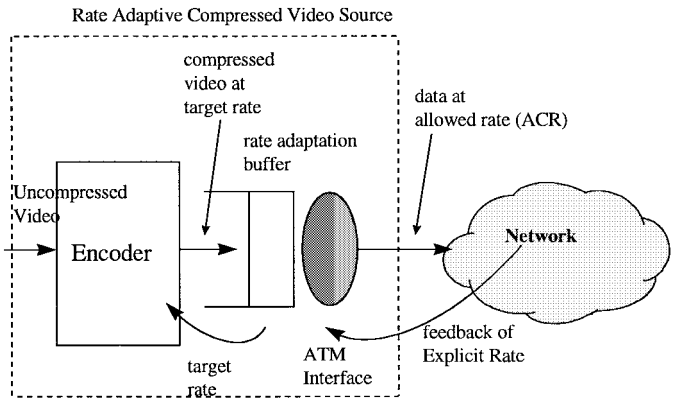

Fig. 1. Framework for rate-adaptive video in an explicit-rate environment.

for the network to inform sources of the abatement of congestion or of newly available bandwidth. Therefore, the source is unable to make use of the newly available bandwidth until the next renegotiation instant.

4) *UBR, Best-Effort Transport:* In this mode, the video is transported as best effort traffic without any rate guarantees from the network. This requires video sources to continuously estimate the available bandwidth and adapt to it. In this case, quality can get unacceptably poor, since there is no minimum rate guaranteed.

The scheme proposed in this paper is an attempt to combine the desirable features of each of the above schemes. We would like to preserve the simple call admission-control capabilities of CBR, the statistical multiplexing advantages provided by unrestricted VBR, the ability to signal rate requirements to the network as allowed by RCBR, as well as the ability to provide explicit feedback from the network. In the rest of this paper, we describe how the explicit-rate ABR service can be used to provide these capabilities.

III. Framework

Fig. 1 shows the framework under which we study the effectiveness of adapting compressed video sources in a rate-controlled network. Uncompressed video from the source is fed to an encoder, which uses a quantization process, followed by Huffman coding to reduce the number of bits required to represent the video signal. We assume that the encoder is capable of coding each video frame to match a target size (in bits). The number of bits used to code a frame affects the quality of the compressed video.

The output bit-stream from the encoder is fed to a rate-adaptation buffer, which is a source buffer that accommodates mismatches in the rate at which the encoder generates data and the rate at which data can be injected into the network. This latter rate is determined by the explicit-rate ABR congestion-control algorithm, in which a source first requests a rate from the network and the network responds with an allowed rate, based on the contention for network bandwidth. The network provides the assurance that the rate allocated to the source will not go below a minimum rate that is negotiated at the time of setting up the connection.

We define the following rates.

- *Nominal Rate:* the rate that is required by the encoder to code the frame at *ideal* quality.
- *Target Rate:* the rate given to the encoder based on the algorithm for smoothing and rate-adaptation. We assume that the encoder will precisely meet the rate it is given as the target for a frame, as long as it is less than the ideal rate for the frame. The difference between the nominal rate and the target rate is a result of our adaptation mechanisms.
- *Demand Rate:* the rate that the source requests from the network based on the prediction.
- *Allowed Rate:* the rate returned from the network, after a feedback delay, in response to the source's requested rate.

IV. Overview of Explicit-Rate Mechanisms for Video

The ABR service has been defined in the ATM forum to support applications that require best-effort service. Although no assurances are made of maintaining low delay or jitter, the feedback-control algorithm attempts to maintain small queues and *feasible* transmission rates for the individual sources (i.e., the aggregate transmission rate of all the currently active sources utilizing a link does not exceed the link capacity). To a great extent, best-effort applications desire a low loss rate. However, no quantitative loss rate requirements have been specified for the ABR service [26]. With the use of explicit-rate mechanisms, and appropriate switch-rate allocation algorithms, we believe the cell loss rates experienced could be small and, therefore, quite acceptable for video transmission. We take advantage of the explicit-rate mechanism's ability to maintain small queues in the network, since that allows us to also depend on a correspondingly small delay in the network, on the average. The ABR service also admits to the notion of a minimum-bandwidth allocation for a source. Although an admission-control mechanism has not been specified, we believe this can be relatively simple and conservative.

The *explicit-rate* mechanism, which we use as the basis for our work here, also attempts to achieve the goal of *max–min fairness* [1] for the source rates, while operating the bottleneck links efficiently. Achieving max–min fairness is important in situations when different links in the network have different demands from sources, and network resources have to be shared equitably. A constructive definition of max–min fairness is provided in [1], [4], and [24]. Intuitively, the criterion ensures that the capacity of a bottleneck resource is equally divided among those flows that are limited by that resource, after allocating the share of the capacity that is requested by flows limited elsewhere in the network.

A. Operation of the Feedback-Control Mechanism

The explicit-rate control scheme depends on a set of co-operating sources periodically probing the network for the appropriate transmission rate.

The two key components of the control algorithm are: 1) the behavior of the source and destination end systems and 2) the behavior of the network elements (switches).

Each source of a virtual circuit (VC) periodically transmits a special *resource management* (RM) cell to probe the state of the network. Each switch identifies and conveys its state of congestion, as well as additional rate information to the source end-system in the RM cell. The source algorithm responds to the feedback information by adjusting the rate of transmission in accordance with a specified policy.

A source specifies a "demand" or desired transmit rate in each transmitted RM cell (in addition to the currently allowed rate), in the *ER field*. Switches compute the rate they may allocate to each VC, and overwrite the ER field with the computed allocated rate if it is lower than what was in the ER field of the received RM cell. As the RM cell progresses from the source to destination, the ER-field value reflects the smallest rate allocated by any of the switches in the path for the VC. On reaching its destination, the RM cell is returned to the source, which now sets its transmit rate based on the allocated in the ER field of the returned RM cell.

The goal of the explicit-rate-based feedback-control algorithm is to respond to incipient congestion, and to allocate rates to the competing sources in a fair manner, while ensuring that the capacity of the network is not exceeded.

There are several switch algorithms proposed for computing the rate to be allocated to a VC [3], [11], [14], [25]. Switches compute an allocated rate for each VC i, based on its requested rate (value in the ER field) $A_i, i = 1, \cdots, n$. VC's are classified as being in a "satisfied" set $\mathcal{S}$ or in a "bottlenecked" set $\mathcal{B}$. The capacity C of the link is allocated to bottlenecked VC's as

$$A_B = \frac{C - \sum_{i \in \mathcal{S}} A_i}{||\mathcal{B}||}. \tag{1}$$

VC's in the satisfied set $\mathcal{S}$ are allocated their requested rate, A_i. Details may be found in [3] and [14]. To keep the dynamics of the switch-rate allocation simple, we implemented a straightforward version of the max–min computation broadly described in [24].

The source policies are a simplified version of [26], where the primary properties of the feedback-control loop have been implemented, without incorporating all the functionality related to the boundary cases. Specifically, the policies relating to the use-it-or-lose-it function and the source decrease function that protects the network against delayed or lost feedback have not been incorporated. The source maintains a currently allowed rate, ACR, which is the rate at which queued cells are transmitted out of the source network interface. Sources maintain a DEMAND (for data sources this may be the outgoing link's rate), used for requesting a rate from the network. When an RM cell returns with an allocated rate ER, the source's allowed rate is changed as follows:

```
if ACR ≥ ER
    ACR = max(min(ER, DEMAND), MCR)
else
    ACR = max(min{ACR + (RIF * PCR), ER}, MCR).
```

Notice that a network indication to decrease the rate takes effect immediately. However, when the allocated rate ER return is higher than the current ACR, ACR increases in additive steps of RIF $*$ PCR. RIF is an increase-factor that is a negotiated parameter, and PCR is the peak cell rate for the connection. ACR always remains above the MCR.

When an RM cell is transmitted, the ER field is set to $\max$ (DEMAND, ACR). RM cells are periodically transmitted, once every Nrm data cells (e.g., $Nrm = 32$), so that the overhead for carrying the probe cells is bounded, while still having a responsive control scheme. A large RIF results in our converging to the returned ER quickly, but with the potential for some transient overload on the network. To keep queues small, RIF may be chosen to be small [in the simulations presented in this paper, RIF was set to a very large value (1)].

For a detailed description of the end-system policies and switch policies that assure max–min fairness while maintaining small queues, we refer the reader to [3], [4], [14], and [26].

B. Enhancements to the Explicit-Rate Scheme for Video

The max–min fairness goal applies a "uniform" criterion for allocation of resources to bottlenecked flows. When there are competing bottlenecked flows at a given resource, the bandwidth allocated to each of these flows is identical. Such a policy is appropriate for data flows but inappropriate for video flows, since for the latter there is a relationship between the ultimate "quality of service" achieved for the flow and the degree of rate reduction experienced by the flow relative to its demand. Intuitively, an encoder that is compressing more complex images requires a greater number of bits to obtain the same level of quality and should, therefore, request more bandwidth than a source transmitting frames for a low-complexity sequence of images. If the network were to apply the max–min fairness criterion, then the high-activity flow would experience a much greater degradation in the video quality compared to the low-activity flow.

We modify the basic max–min fair-allocation scheme to associate a weight with each flow. A weighted fairness criterion is applied at each resource for the different flows placing a demand on it. We use the source's original demand as the basis for assigning a weight to the flow. The weight at a bottleneck for a source i whose demand is D_i is given by

$$W_i = \frac{D_i}{\sum_{j \in \mathcal{F}} D_j} \tag{2}$$

where $\mathcal{F}$ is the set of flows placing a demand on this resource.

The goal of the weighted max–min fair-share allocation algorithm is that VC's receive a weighted fair share of the bottlenecked resource. Thus, the capacity of a bottleneck resource is divided in proportion to the respective weights of the flows that are limited by their bottleneck resource, after allocating the share of the capacity that is requested by other flows limited elsewhere in the network. This weighted max–min fairness mimics the operation of weighted fair queueing [5], [9]. It is more generally applicable than just in this circumstance.

For example, weights may be associated with pricing or other factors.

To achieve a weighted max–min fair allocation, we require the original demand from the source to be available to all of the resources in the network. We introduce an additional field, called "source demand," in the RM cell for this purpose. The source places its demand (based on its bandwidth requirement) in the "source demand" field of the RM cell, and switches only read this field. At the switch, we now have three rates of interest in the RM cell.

- D_i: the original demand of the source, which is written by the source and left untouched as it flows through the network.
- ER_i: the value of explicit-rate field. ER_i is the share of the capacity of this resource that flow i requests after accounting for bottlenecks upstream of this resource. This field is marked down as the RM cell flows through the network.
- CCR_i: the current cell rate of the source.

The goals of our enhanced weighted max–min fair rate allocation mechanism are as follows.

- *Satisfied VC's* (in the terminology of [14]) should receive an allocation equal to their rate requested, ER_i, from this resource. We denote this set $\mathcal{S}$.
- *Bottlenecked VC's* receive an allocation less than their request. The amount of extra capacity left over from the allocation to satisfied VC's is now shared among bottlenecked VC's in proportion to their demands D_i. We denote this set $\mathcal{B}$.

The steps in the weighted max–min fair rate allocation algorithm of the switch are similar to those in [24]. We perform the computation to determine the allocation upon arrival of an RM cell, let us say from VC_j. Let A_i, $i = 1, \cdots, n$ be the requests for each of the n VC's. The ER value in the RM cell for VC_j is considered to be the request A_j, while the current state of the allocations at the switch for the other VC's are considered as their requests A_i. Let the fair share computed by our algorithm for any VC_i be F_i. The exception from [24] is a check to see if a VC is bottlenecked or not, based on whether the weighted share of the leftover capacity is less than the rate request for that VC. Initially, the set $\mathcal{S}$ is empty. The share for all bottlenecked VC's, C^b, is computed as

$$C^b = C - \sum_{i \in \mathcal{S}} A_i. \tag{3}$$

Here, A_i is the request for a flow from the set $\mathcal{S}$ of currently satisfied flows. C is the capacity of the channel. The weights for the remaining flows, in the set $\mathcal{B}$ are recalculated again as

$$W_i = \frac{D_i}{\sum_{i \in \mathcal{B}} D_i}. \tag{4}$$

If the rate request A_i for the remaining bottlenecked flows in $\mathcal{B}$ is such that the following equation is satisfied:

$$A_i < C^b * W_i \tag{5}$$

then flow i is removed from the set $\mathcal{B}$ and put into the set of satisfied VC's $\mathcal{S}$. Also, the fair share for these VC's in $\mathcal{S}$ is $F_i = A_i$. The available capacity for bottlenecked VC's, C^b, is correspondingly reduced following (3). The weights for the remaining flows in $\mathcal{B}$ are recalculated using (4). We repeat this operation for each of the flows in $\mathcal{B}$, and remove those flows whose rate requests satisfy (5).

Finally, we have a set of flows in $\mathcal{B}$ which have their rate requests A_i such that

$$A_i \geq C^b * W_i. \tag{6}$$

The fair share F_k, ($k \in \mathcal{B}$) for the remaining flows in $\mathcal{B}$ is then given by

$$F_k = C^b * W_k. \tag{7}$$

This allocation is then indicated in the ER field of the RM cell corresponding to VC_j, and the RM cell is now forwarded downstream toward the destination.

The enhanced weighted max–min fairness algorithm thus allocates a share, as in (7), to the flows whose rate requests are greater than the weighted share of the capacity available to the individual flow. VC's that are not bottlenecked at this resource will receive their rate request, as observed in the ER field of an RM cell from that VC. VC's bottlenecked at this resource and, hence, limited by this resource receive a weighted share of the resource's capacity that may be allocated to bottlenecked VC's. The guarantee is also that the allocation to bottlenecked VC's is higher than the allocation to a satisfied VC at this resource.

Finally, the local allocation for VC_j at a switch is computed as $A_j = \min(F_j, \mathrm{CCR}_j)$, where CCR_j is the current rate that the VC is transmitting, as indicated in the RM cell. If a source is transmitting at $\mathrm{CCR}_j < F_j$, this is the bandwidth allocated to the source. This allocation by a switch allows a downstream bottleneck to convey its allocation to an upstream switch one round-trip time later [14].

C. Convergence Delays of the Allocation Algorithm

The distributed rate allocation algorithm achieves max–min fairness by an iterative process. There is a "global iteration" achieved by RM cells flowing from the source to destination and back, collecting the rate allocation information for the flow. Further, there is a "local iteration" at each switch link to determine the allocation to each of the flows sharing that link.

At the first step of the global iteration, the allocation of all the flows sharing the first-level (tightest) bottleneck is determined. Subsequently, in each of the next steps of the global iteration, the allocation of flows sharing the next-level (next-tightest) bottlenecks is determined, progressing from one bottleneck level to the next, until we finally make the allocation of the rates to the flows sharing the Kth-level (loosest) bottleneck in the network. It is shown in [4] that an upper bound on convergence time of such distributed algorithms determining a max–min fair allocation is approximately $K *$ RTT, where RTT is the maximum round-trip delay for control information to propagate from the source through the network to the destination, and back to the source; and K is the number of different bottleneck rates. There may be significant queueing delays for RM cells, as well as propagation delays (e.g., in a

wide-area network (WAN), which contribute to RTT. As a result of a richly connected network, each link having diverse numbers of flows sharing them or with sources having different demands, the number of distinct rates K may be large as well. Thus, the time to converge to a final rate allocation for all the flows, once the demands have stabilized, may be larger than a frame time, based on the results in [4]. The source rate-adaptation policy needs to be cognizant of this, as we discuss in Section VI.

V. Source Models for Video

The enhanced explicit-rate mechanism, proposed in this paper, assumes that each source can periodically indicate its bandwidth requirements to the network via RM cells. This requires the source to forecast the encoder bit rate requirements over small time intervals, e.g., on the order of a few network round-trip times. This demand forecasting can be done using models that characterize the statistical behavior of video sources. Several such models have been proposed in the literature [7], [16]–[18].[3] Any of these models could be used in conjunction with the explicit-rate control mechanism proposed in this paper, as long as it is able to accurately predict the short-term rate requirements of a video source. In this paper, we have chosen to use the gamma-beta autoregressive (GBAR) source model [15]. This model has been shown to accurately model video teleconferencing sources when using H.261-like coding schemes.

A. Models for Video Teleconferences

In [17] and [18], traffic models for video teleconferences using H.261 and H.261-like coding were formulated by examining data recorded during several 30-min video teleconferences. A key observation is that traffic models look similar, despite the sequences differing in the details of the coding scheme. The important features of the video teleconference models can be summarized as follows. The number of cells per frame can be modeled by a stationary process. The marginal distribution of the number of cells per frame follows a gamma distribution (negative binomial if a discrete distribution is used), and so the number of cells per frame is given by

$$X(t) = \frac{\lambda(\lambda t)^{s-1}}{\Gamma(s)} e^{-\lambda t} \tag{8}$$

where $\Gamma(s)$ is the gamma function defined as

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt. \tag{9}$$

The parameters s and λ are called the *shape* and *scale* parameters, respectively, and these can be obtained from the mean and variance of the source. Let ρ be the lag-1 correlation. These correlations are typically very high for teleconference sources with $\rho = 0.98$ for the source studied in [18]. This high correlation makes fairly accurate short term forecasting feasible. In [19], a very simple forecasting rule is used successfully. The rule is $X_{n+k} = \mu + \rho^k(X_n - \mu)$, where ρ is the correlation coefficient. μ, the mean number of cells per frame, is computed on-line. An accurate model is the DAR(1) model [18], which is a Markov chain determined by three parameters: the mean, variance, and ρ. The transition matrix is computed as

$$P = \rho I + (1-\rho)Q \tag{10}$$

where ρ is the autocorrelation coefficient, I is the identity matrix, and each row of Q consists of the negative binomial (or gamma) probabilities $(f_0, \cdots, f_K, F_K)$, where $F_K = \sum_{k>K} f_k$ and K is the peak rate. The DAR(1) model matches the autocorrelation of the data over approximately 100 frame lags. This match is more than sufficient for our purposes, since our forecasting horizon is a few round-trip times, which correspond to only three or four frame lags at most. Knowing the mean, variance, and lag-1 correlation of the source, forecasts can be made using the DAR(1) model, given only the number of bits in the current frame. The DAR(1) model can be used with any marginal distribution, and was used in [16] to model entertainment and MPEG-2 coded video sequences with marginal distributions which are not gamma distributed. For teleconference sequences, since the marginal distributions are gamma distributed, this generality is not necessary. Moreover, the DAR(1) model has "flat spots" which make its sample paths "look" different from those of the data when comparisons are made for a single source (for multiplexed data sources they are indistinguishable [7]). Though these flat spots may not affect our results, for the teleconferences we use a statistical model more specialized for modeling accurately the short-term fluctuations of single teleconference sources.

This model called gamma-beta autoregressive [GBAR(1)] model, was proposed by Heyman [15]. Like the DAR(1) model, the GBAR(1) model is also a three parameter model requiring only knowledge of the mean, variance and lag-1 correlation of the source. It relies on the observation that video teleconferences have gamma-marginal distributions and exponentially decaying autocorrelations up to lags of about 100 frames. The main features of the model (described in detail in [15]) are summarized below, since we use it as our forecasting model.

Let $Ga(s, \lambda)$ denote a gamma distributed random variable with shape parameter s and scale parameter λ. Let $Be(t, r)$ denote a beta-distributed random variable. The density function of the beta distribution is given by

$$f_\beta(x) = \frac{\Gamma(t+r)}{\Gamma(t+1)\Gamma(r+1)} x^{t-1}(1-x)^{r-1}, \qquad t, r > -1. \tag{11}$$

The GBAR(1) model uses the following facts.

1) The sum of independent $Ga(s, \lambda)$ and $Ga(q, \lambda)$ random variables is a $Ga(s+q, \lambda)$ random variable.
2) The product of independent $Be(t, s-t)$ and $Ga(s, \lambda)$ random variables is a $Ga(t, \lambda)$ random variable. The forecasting rule for the GBAR(1) model is given by:

$$X_n = A_n X_{n-1} + B_n. \tag{12}$$

Since, for video teleconferences, we want the distribution of X_n (and naturally X_{n-1}) to be $Ga(s, \lambda)$, (the shape and scale parameters being obtained from the empirical mean and

[3] These models were primarily designed to solve the problem of admission control for open-loop VBR transport of video.

TABLE I
PARAMETERS FOR VIDEO-TELECONFERENCE SEQUENCES

Seq	Bytes/ cell	Mean cells/frame	Variance	Lag-1 correlation
A	14	1506	262861	.981
B	48	104	882	.984
C	64	130	5535	.985
D	64	170	11577	.970

variances as was done for the DAR model), we pick A_n to be a $Be(t,\ s-t)$ random variable and B_n to be a $Ga(s-t,\ \lambda)$ random variable. It may be easily verified from (1) and (2) that when X_{n-1}, A_n and B_n are mutually independent, X_n is $Ga(s,\ \lambda)$ distributed as desired. Also, the lag-k autocorrelation function is given by $\rho(k) = (t/s)^k$. Using this t is determined since we know $\rho = \rho(1)$ and s (from the mean and variance of the data). The forecasting computation is simple: given X_{n-1} multiply it by B_n, a sample from an independent beta-distributed random variable, and then add A_n, drawn from a gamma distribution. Both distributions have parameters which need to be computed only once from the mean, variance, and lag-1 correlation of the teleconference sequence of interest.

For four video teleconferences, we used the GBAR(1) process for short-term prediction of the number of cells per frame, given the number of cells per frame for the current frame. The mean, variance, and 1-lag correlation needed for the predictions is given for each of the sequences in Table I.

VI. SOURCE-ADAPTATION MECHANISM

We use a source buffer between the encoder and the ATM layer to provide isolation between two control loops. The first control loop is at the ATM layer, where the source adapts its transmission rate, ACR, based on the feedback from the network. The second control loop, local at the source, uses the source buffer occupancy and a smoothed value of the ACR.

A. Demand Prediction

A rate-adaptive ABR video source needs to estimate its future bandwidth DEMAND D_i and send out an RM cell requesting this bandwidth at least a feedback-delay amount of time earlier than when the rate is needed. Let us assume, for simplicity, that this is done on a per-frame basis. If one were to look at a timeline for the operation, a rate request would be made at time t, based on the predicted size for the frame to be transmitted at time $t+T$. Here, $T = \text{RTT} + \epsilon + F + \delta$, where RTT is the round-trip delay; ϵ is the time for the encoder to adapt to a new rate; F is the frame time (assuming that the encoding of the frame also takes a frame time, e.g., 33 or 40 ms); and δ is the delay in the source end-system needed to packetize the data and hand it down to the ATM adaptation layer.

There are several issues with just using the straightforward prediction of a single frame size $F+$ RTT later. Since there may be considerable variation in the frame sizes, the time we look ahead in the prediction has to be precise. For example, if the response from the network comes too late for the coder, we could be encoding according to a rate required for an earlier frame. If the response comes too early, this rate may be superseded by a subsequent rate feedback. Thus, we may be coding at a rate suitable for a frame to be transmitted at a later time. Furthermore, the rate received from the network for this frame (in time for it to be encoded at RTT $+ \epsilon$) is implicitly assumed to be available later, at $t + \text{RTT} + \epsilon + 2F + \delta$ when the frame transmission is completed. If the rates received in subsequent RM cells are different, this may lead to the frame being delayed. However, this delay may be acceptable if MCR is large enough.

It is not always desirable to take the straightforward approach of requesting bandwidth on a frame by frame basis, since it requires a very accurate estimate of the look ahead time T. In case there are errors in estimating this value, the bandwidth allocated by the network may lag or lead the bandwidth required by the source. Since accurate knowledge of the various delays, especially the RTT, is not possible, it is instead preferable to use a simple smoothing technique to limit the sensitivity to errors in this estimate. This involves predicting the bit rate of several frames—from the next frame to be transmitted to the frame that will be transmitted one RTT later—and using the average rate over all these frames as the demand D_i to the network. Specifically, it may be desirable to predict the requirements of N frames in advance (as a moving window), and compute a D_i (placed at time t) based on the average rate for these N frames. The choice of the size of the moving window N for averaging the demand also depends on the coding scheme used (e.g., H.261 for video teleconferencing; MPEG for entertainment video). Using an averaging interval that is larger than a frame time is desirable. For example, using an average over several frames, such as a GOP for MPEG may be appropriate. Note, however that the simulation results reported in Section VII do not use this smoothing.

B. Encoder Rate Adaptation

At the ATM layer, the source policy adapts the transmission rate ACR, every time an RM cell is received back at the source. There is potential for considerable variation in the rate returned to the source, based on changing conditions in the network. In addition, when the network is unable to grant the demand D_i indicated by a source, the encoder has to adapt it's bit rate to match the bandwidth granted by the network, to prevent overflow of the source buffer between the encoder and the ATM layer. The source buffer serves to isolate the encoder from the rapid changes in the rate provided by the network, and also acts as an integrator of the difference between the encoder's desired rate and the allowed rate, ACR, over time.

There are several options available for adapting the encoder's quantization level to the allowed transmission rate ACR at the ATM layer.

1) Directly use the instantaneous ACR as the coder's rate to encode the next frame.
2) Use information on the occupancy of the source buffer, between the coder and the ATM layer, to modify the encoding rate.
3) Use a combination of the source buffer and the recent history of ACR returned to adapt the coder's rate. The simulation results reported below use this approach.

Using the first option would imply directly using the feedback information from the network to adjust the coding rate for the next frame. There is an immediate connection between the feedback from the network and the coder. This works well if the estimate of the feedback delay is perfect and also if the network returns an ER value that is very close to the DEMANDed rate. Neither of these are likely. We want the source to adapt its rate to changing conditions in the network. Moreover, it is difficult to estimate the feedback delay. Another important problem is that during the transient convergence period when the network is attempting to converge to the final weighted max–min fair rate, the returned ER and hence ACR for the source is continually changing. Using an RIF value that limits the step-size with which the source may build up its ACR toward the returned ER also makes this matching quite difficult. We also believe that the potentially rapid fluctuations of the coding rate adversely impacts the quality of the video.

With the second option, we take advantage of the local source buffer between the encoder and the ATM layer to "integrate" the effects of both the differences between the coder's desired rate and the feedback rate. The buffer also smooths out some of the errors in our estimation of the feedback delay. By considering the source buffer as the point of isolation between the encoder's rate and the rate sustainable by the network, we use the source buffer occupancy B_{occ} to determine the encoder's rate. We try to maintain the source buffer levels between a low threshold Q_{low} and a high threshold Q_{high}. We use a rate reduction function, below the nominal rate the coder needs for the best quality (R_T), that is a linear function of the buffer occupancy in the range (Q_{low}, Q_{high}). The average encoder rate λ_{avg} is determined from the following:

$$\lambda_{\text{avg}} = R_T * \frac{B_{\text{occ}} - Q_{\text{low}}}{(Q_{\text{high}} - Q_{\text{low}})}. \tag{13}$$

While in principle this serves the function of smoothing the encoding rate used, it completely isolates any drastic deviation of the network's feedback. As a result, large differences between λ_{avg} and ER may lead to unacceptable queue build-up locally at the source, resulting in either exceeding our delay targets or loss locally from the source buffer.

We chose to use the third option, which uses a combination of the source buffer and the recent history of ACR returned to adapt the coder's rate. The source buffer also allows us to smooth out errors in our estimation of the feedback delay. Thus, we minimize rapid fluctuations in the coding rate, avoiding any adverse impact on the quality of the video.

The encoder rate-adaptation function accounts for both the ACR and the state of the source buffer. The following function is used:

$$\lambda_{\text{avg}} = \text{ACR}_{\text{avg}} - \left\| \alpha * \frac{(B_{T-F} - \text{SETPOINT})}{\text{time\_horizon}} \right\|. \tag{14}$$

Here, B_{T-F} is the predicted buffer size at the time the encoder is given the rate to code the frame, and SETPOINT is the desired buffer setpoint at the local source buffer. α is a small gain factor. The time_horizon is the interval over which we try to bring the predicted buffer B_{T-F} down to the level of the SETPOINT. The time horizon is typically of the order of a few frame times (chosen to be five, in our simulations), so that the delay for a frame is not adversely affected. The constraint for choosing the buffer SETPOINT is that the contribution to the delay by the source buffer is not excessive. Similarly, ACR_{avg} is also computed over an interval of a few frame times ($\approx$5 frame times). Similar control mechanisms have been proposed earlier for control of bursty data traffic sources [20], [21].

It is desirable to use a longer averaging interval and reduce the frequency with which the source demand and the encoder's rate are modified. Altering the source rate frequently may result in impairment of the user perceived quality of the resulting video. Also, the amount of time taken by the allocation mechanism to converge to the weighted max–min fair rate may be significant. It has been shown in [4] that it takes a period of RTT per distinct rate in the eventual rate vector for all of the source rates to converge to their final rates. Of course the amount of "damping" is also dependent on how much source delay we can tolerate. A constraint we use as a rule of thumb is that the end-to-end delay should not be greater than about 200–300 ms/frame.

C. Minimum Cell-Rate Selection and Admission-Control Issues

For compressed video flows, it is necessary to limit the number of admitted connections to limit the degradation in user perceived quality due to contention among the various compressed video flows. Thus, a user may wish to set limits on the average and worst-case degradation that can be tolerated. From a network's perspective, the user's quality requirements need to be mapped into a bandwidth requirement, possibly in the form of an equivalent bandwidth specification. Moreover, since for the ABR service, admission control is typically performed based on the minimum cell rate (MCR) specified during connection setup, it may be necessary to derive an appropriate value of the MCR from the equivalent bandwidth specification. Another alternative is to perform admission control based on the equivalent bandwidth and separately specify the MCR. In Section VII, we examine these issues in greater detail.

In the work presented in this paper, we assume that the network uses a separate queue to isolate the compressed video flows from other classes of service that may or may not be admission controlled. We also assume that a scheduling policy is used that causes these queues to be served in proportion to the rates allocated to each of these classes.

VII. Simulation Results

We use trace driven simulations to study the performance of the enhanced explicit-rate control scheme, and quantitatively justify some of the observations made in the previous sections. The simulations are used to address the following questions.

- How effective is the control scheme in dynamically adapting the encoder source rate to match the network bandwidth?

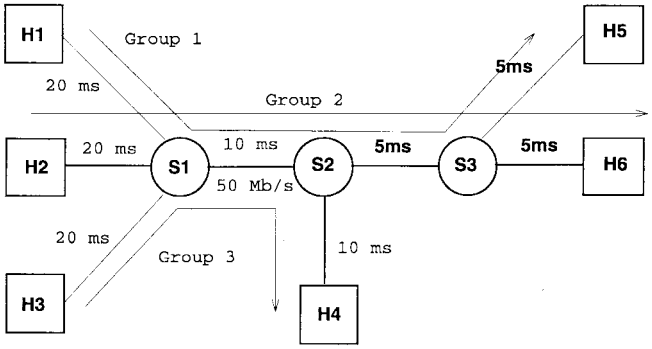

Fig. 2. Simulation configuration.

- How is the overall performance impacted by the accuracy in predicting the source bit rates?
- How effective is the control scheme in ensuring fairness?
- What is the impact of multiple bottleneck links?

In all of the simulations, a single long trace of measured video-teleconferencing data (38 137 frames) is used to derive the frame sizes for each video traffic source. Each simulation is run for 100 s. The video sources generate 25 frames/s. The mean bit rate of the entire trace is about 4.4 Mb/s. Sources start staggered apart at intervals of 40 ms (one frame time apart). The initial rate, ICR [26] is chosen to be 500 cells/s, and the source buffer SETPOINT is 200 cells. We use the "fan-in" configuration shown in Fig. 2, with 6–24 active video sources, spread evenly across Groups 1–3, feeding into a common bottleneck link (link S1 $\rightarrow$ S2). The round-trip propagation delays are typically 80 ms. Each receiver is assumed to implement a play-out buffer; the target end-to-end (one-way) delay is 300 ms.

The primary metrics we use to evaluate the performance of our adaptation scheme are:

- the average and minimum rate allocated by the network for each connection;
- the average and minimum *cropping* ratio for each connection—where the cropping ratio for a single video frame is defined as actual encoded frame size/nominal frame size;
- the end-to-end delay to transmit an entire video frame for each connection.

We evaluate the fairness properties by comparing the average values of the rate, actual encoded frame size and the cropping ratio for each connection. In selected cases, we also compare the time-varying behavior of the allocated rate and cropping ratio for each of the connections.

A. Rate Matching

The primary goal of rate based feedback control is to allow traffic sources to match the available network bandwidth. Ideally, a video source should get its demanded bandwidth as long as the network is lightly loaded. On the other hand, when the network is unable to grant the source demand D_i, the source should be able to adapt its bit rate downwards in a timely fashion to ensure that the per-packet delays and loss rates stay bounded. In the first set of experiments, we study the performance of the enhanced explicit-rate scheme

TABLE II
AVERAGE VALUE OF CROPPING RATIO FOR CONNECTIONS 1–6 WITH VARYING NUMBER OF CONNECTIONS

	Cn 1	Cn 2	Cn 3	Cn 4	Cn 5	Cn 6	Avg. Bottleneck Link Utilization
# of Conns							
6	0.97	0.97	0.97	0.97	0.96	0.96	0.563
12	0.82	0.83	0.82	0.82	0.81	0.81	0.949
18	0.57	0.57	0.57	0.57	0.56	0.56	0.960
24	0.43	0.43	0.43	0.43	0.43	0.43	0.963

TABLE III
MINIMUM VALUE OF CROPPING RATIO FOR CONNECTIOINS 1–6 WITH VARYING NUMBER OF CONNECTIONS

	Cn 1	Cn 2	Cn 3	Cn 4	Cn 5	Cn 6
# of Conns						
6	0.51	0.51	0.49	0.50	0.52	0.52
12	0.37	0.32	0.39	0.38	0.38	0.33
18	0.26	0.26	0.26	0.24	0.25	0.24
24	0.19	0.19	0.19	0.18	0.20	0.19

TABLE IV
AVERAGE RATE (MEGABITS/S) FOR CONNECTIONS 1–6 WITH VARYING NUMBER OF CONNECTIONS

	Cn 1	Cn 2	Cn 3	Cn 4	Cn 5	Cn 6
# of Conns						
6	5.88	5.71	6.11	6.31	6.82	7.32
12	4.94	4.80	5.12	5.28	5.72	6.14
18	3.41	3.32	3.54	3.66	3.97	4.26
24	2.60	2.53	2.69	2.78	3.02	3.25

as the number of active connections is varied between 6 and 24 connections, while keeping the bottleneck link capacity unchanged. This causes the average bottleneck link utilization to vary between 56% (6 sources) and 99% (24 sources).

Table II enumerates the values of the cropping ratio for connections 1–6, with a varying number of connections. With six sources active, at 56% bottleneck utilization, the rate requested by a video source is always granted by all switches on the path. Even so, the actual size of the transmitted frames is sometimes smaller compared to the nominal frame size. This occurs because of local queue build up resulting from errors in predicting the future demand and/or small timing mismatches between when the requested bandwidth is available to the source to when it is actually needed. Due to the use of the setpoint algorithm (14), the encoder reduces its bit rate in response to such a queue buildup.

As the number of connections is increased, the aggregate bandwidth demands of the sources exceeds the network capacity more frequently. This causes a decrease in the average actual bit rate of each source (Table IV). As a result, the ratio of the (actual/nominal) frame sizes also reduces as shown in Tables II and III. These ratios capture the average and worst-case degradation in the "quality" for each active stream over the entire duration of the simulation.

Fig. 3 shows the end-to-end frame delay histogram. As the number of active connections is increased, the network utilization increases and, consequently, the mean and variance of the end-to-end delay increases because of greater queueing in

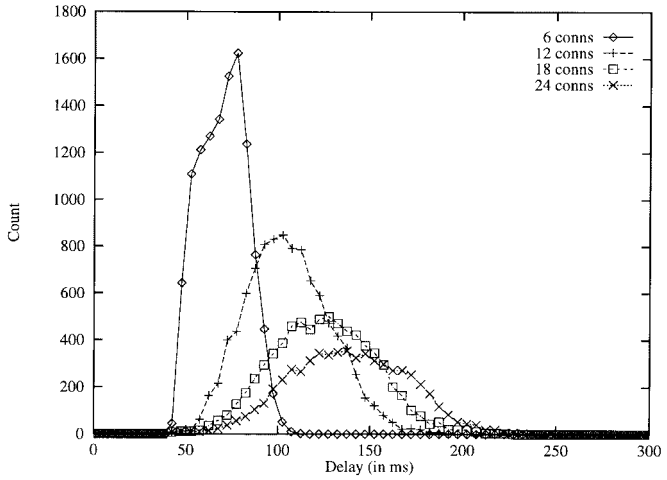

Fig. 3. Histogram of per frame delays with varying number of connections. Propagation delay = 80 ms.

source and switch buffers. However, the end-to-end delay stays below the 300-ms target, even with 24 active connections, when the bottleneck link utilization was 99%.

B. Observations on Admission Control

In our simulations, the MCR value was set to zero and no explicit admission-control checks were enforced. However, our simulation results provide insights into understanding how the ABR admission-control framework may need to be extended for compressed video sources.

Say that the video quality is acceptable to a user as long as the average value of the cropping ratio is greater than 0.50, and the minimum value of the cropping ratio is greater than 0.20. Table III, shows that with 18 connections, the average and maximum cropping ratios are approximately 56% and 24%, respectively. When the number of connections is increased to 24, the maximum cropping ratio drops to 18%, while the average cropping ratio drops to 43%. Therefore, for this particular definition of user satisfaction, quality is close to being acceptable with 18 connections. This suggests that we need to ensure that no more than 18 connections are admitted, with the limit being the bottleneck link S1 $\rightarrow$ S2.

For the ABR service, admission control is typically performed based on the MCR, specified during connection setup. Therefore, a naive approach to ensure that no more than 18 connections are admitted on the link S1 $\rightarrow$ S2 would be to pick a MCR of 2.77 Mb/s for each connection (this value is derived by dividing this link's bandwidth equally among all 18 connections). However, the value of the MCR also determines the minimal amount of bandwidth that is always "reserved" for a connection, and thus limits the degree of bandwidth sharing across connections. As shown in Table V, with the MCR value set to zero, the actual minimum rate seen by each of the 18 connections is much lower than 2.77 Mb/s (it is typically between 0.8–1.1 Mb/s across all 18 connections in this experiment), while achieving the quality targets as specified by the average and the minimum cropping ratios of 0.5 and 0.2. This suggests that setting the MCR value to 2.77 Mb/s would likely cause a reduction in both the average

TABLE V
MINIMUM RATE ALLOCATED (MEGABITS/S) FOR CONNECTIONS 1–6 WITH VARYING NUMBER OF CONNECTIONS

	Cn 1	Cn 2	Cn 3	Cn 4	Cn 5	Cn 6
# of Conns						
6	1.74	1.32	1.32	1.77	1.78	1.91
12	1.39	1.19	1.19	1.46	1.59	1.67
18	0.90	0.84	0.83	0.93	1.13	1.11
24	0.68	0.60	0.60	0.71	0.76	0.83

and the maximum cropping ratios below the acceptable levels. Alternatively, if one were to pick an MCR of 1.1 Mb/s, we would admit as many as 45 connections on the S1 $\rightarrow$ S2 link. This is clearly unacceptable as well.

Thus, we observe that it is desirable to have two rates specified for a connection: a MCR that is used to determine the minimum rate that is always assured to a connection, and a rate similar to the equivalent bandwidth [6] that is used for admission control. For our simulation configuration and traffic sources, these values would be 1.1 and 2.77 Mb/s, respectively.

We note that these conclusions are preliminary in nature. Further work is needed to understand the relationship between the two rates, as well as the consequences of modifying the manner in which admission control is performed for the ABR service.

C. Effectiveness of Source Rate-Adaptation Policy

The goal of the source rate-adaptation policy is to shield the encoder from frequent fluctuations in the ACR value granted by the network while retaining the ability to react when the network reduces the rate granted to a source. The overall effectiveness of the source rate-adaptation policy is dependent on the choice of the source buffer setpoint and the time interval used for estimation and control in (14), as well as the accuracy of the predictions.

We first examine the sensitivity to the smoothing function applied in the source-adaptation policy (14). The demand is predicted $N = 2$ frames in advance. We examine the behavior over a short time interval of 1 s to illustrate the variation of the frame sizes (hence, bit rate) over time in the trace data. We use six active sources to examine the behavior under light load so that the allowed transmission rate, ACR, is a function of the source-adaptation policy rather than being limited by the network. Figs. 4 and 5 show the nominal frame size, the predicted frame size, and the target rate λ_{avg}, as computed in (14), for three values of the time_horizon: 1, 5, and 15 frame durations.

We note that with a larger time window for averaging, the target rate provided to the encoder is relatively smooth, and the amount of deviation from the "nominal" frame size is also somewhat less. For example, with a 15 frame smoothing interval, λ_{avg} in Fig. 5 varies only between $\approx$180–160 kb/s. Using a larger interval for smoothing may result in the differences between the prediction (hence, DEMAND) and λ_{avg} being significant (reflecting the fact that there is smoothing in the encoder's target rate, but not in requesting an "averaged" demand from the network). This could result in more queueing

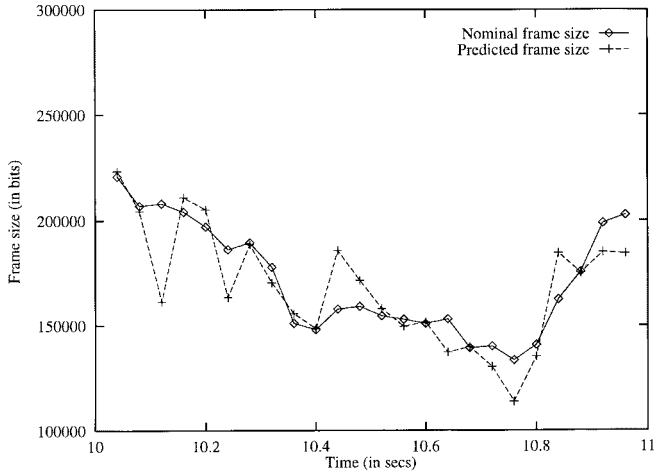

Fig. 4. Effect of prediction accuracy on source rate adaptation.

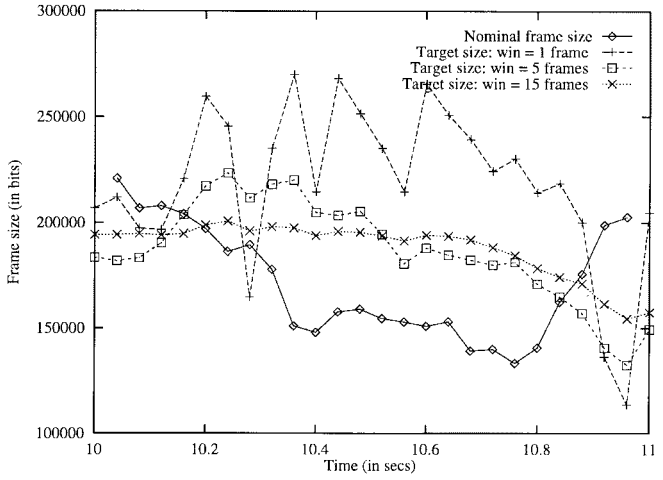

Fig. 5. Effect of window size on source rate adaptation.

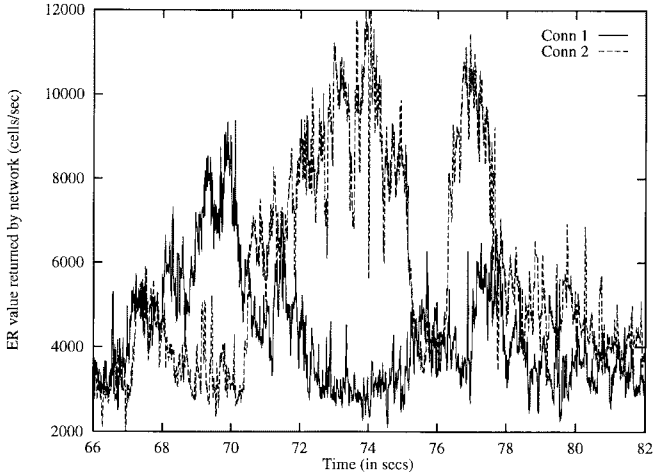

Fig. 6. Time-varying rate allocation (ER) with weighted max–min fair-switch allocation policy.

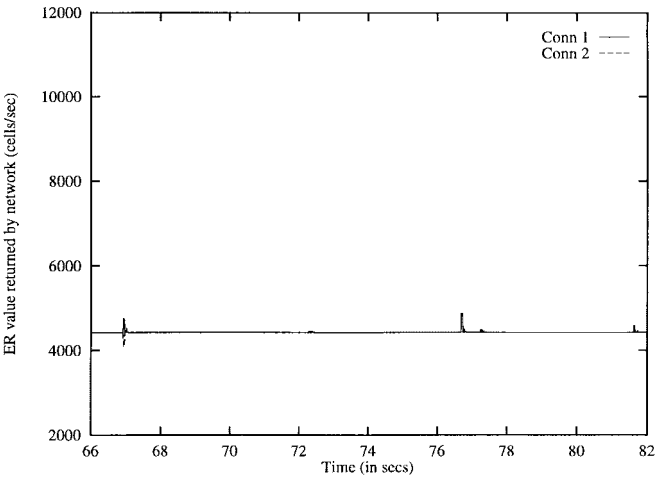

Fig. 7. Time-varying rate allocation (ER) with unweighted max–min fair switch allocation policy.

at the source (when the prediction is lower but the target rate to the source is higher), which is undesirable.

In contrast, with a averaging interval of one frame, λ_{avg} varies considerably, from 270 to 160 kb/s. The deviation from the "nominal" frame size is as much as the predictions are from the actual trace's frame sizes. But, using a small averaging "window" results in λ_{avg} being more responsive to the predicted frame size. Thus, if the rate returned from the network is equal to the DEMAND, then the match between ACR and λ_{avg} is close. This results in very little buffer build-up at the source. Errors in the predictions, however, directly impact the target rate provided to the encoder. Thus, the quality of the final video is dependent on the accuracy of the prediction.

D. Fairness

Tables II and IV illustrate the primary effect of the weighted max–min fair-share allocation algorithm—the cropping ratio, averaged over the length of the trace, is almost identical for all of the connections. This implies that when each of the connections have a slightly different demand, the weighted max–min fair-share algorithm impacts them proportionately, as we originally intended. Figs. 6 and 8, which show the rates returned in the ER field and the cropping ratios for two selected connections, illustrate this point. Observing the time-varying behavior is important, as the average-case behavior of the algorithms may be indistinguishable (unless we use a metric that integrates the effect of the rate-reductions over time). The figure emphasizes the point that the ER value computed using the weighted max–min fair-share allocation algorithm for the two connections varies over time, so as to match each connection's time-varying demand.

In contrast, a network that is enforcing an unweighted max–min fairness attempts to allocate equal shares of the bottleneck bandwidth to connections, as long as they can use it. For example when there are 24 active connections, and since every connection always demands more than 2 Mb/s, while sharing the common bottleneck link (S1 $\rightarrow$ S2), they are allocated an equal share of that bottleneck bandwidth. This allocation stays unchanged for as long as the demands from all the connections exceeds the capacity (likely for the entire lifetime of the connections). Fig. 7 illustrates that the ER values returned to connections 1 and 2 over a 16-s interval are

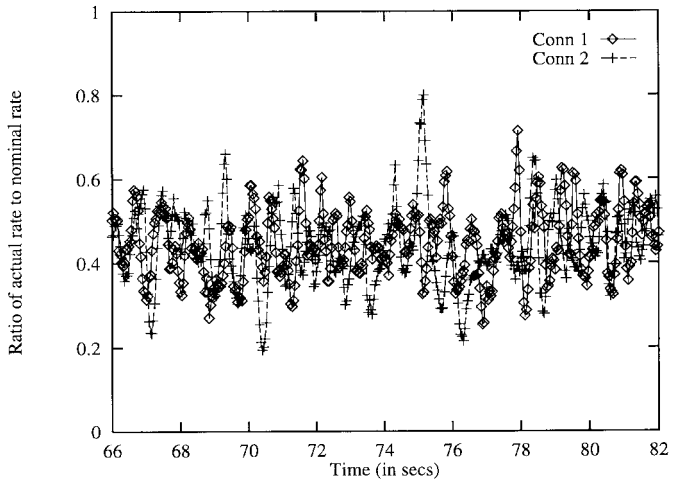

Fig. 8. Time-varying value of cropping ratio with weighted max–min fair switch allocation policy.

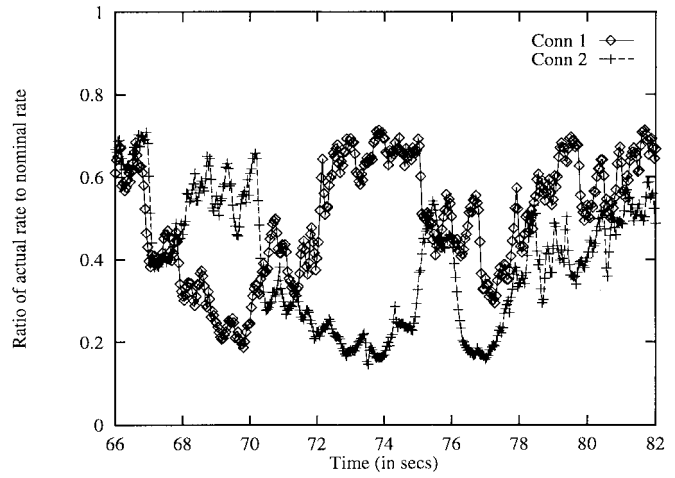

Fig. 9. Time-varying value of cropping ratio with unweighted max–min fair switch allocation policy.

almost identical and do not change over time. This implies that the degradation in quality for the two connections, which have significant differences in the "nominal" frame sizes over time, is disproportionate. Fig. 9 illustrates this behavior.

Thus, we believe that the use of the weighted max–min fair-allocation algorithm has the right characteristics for adapting to the varying demands from competing sources for the bottleneck link.

E. Multiple Bottleneck Links

We now examine the performance of the 24 connections going over the second configuration shown in Fig. 10. In this configuration, the link capacities are chosen so that there are three sets of bottleneck links, S1 → S2, S2 → S3, and S3 → H5 for the connections in Group 3, Group 2, and Group 1, respectively. Each group has eight connections, and the connections of each group share a common bottleneck. The connections in Group 1 are limited by the last 5-Mb/s link from S3 → H5, the connections in Group 2 are limited by the 30 Mb/s link from S2 → S3, and finally, connections in Group 3 are limited by the 155 Mb/s link S1 → S2.

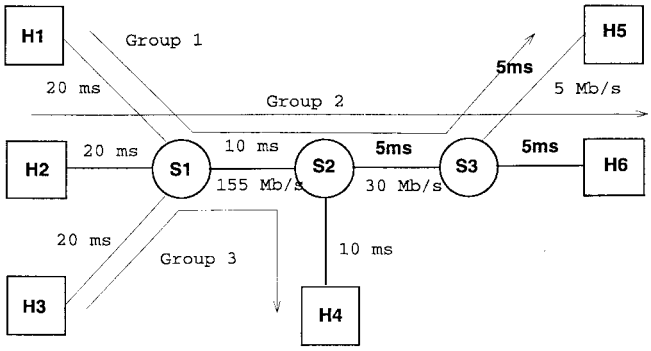

Fig. 10. Simulation configuration 2—multiple bottlenecks.

TABLE VI
PERFORMANCE OF WEIGHTED MAX–MIN FAIR-SHARE ALLOCATION SCHEME WITH MULTIPLE BOTTLENECK LINKS

	Avg. frame size (bits)	Bottleneck link utilization
Cns 1-8	42045.5	96% (S3->H5)
Cns 9-16	123519	94% (S2->S3)
Cns 17-24	151474	48% (S1->S2)

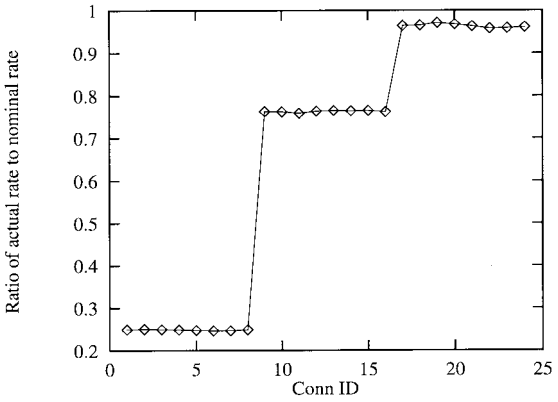

Fig. 11. Cropping ratio for Connections 1–24 with multiple bottleneck links.

Table VI and Fig. 11 show the overall performance achieved by the connections in each of the groups. The average size of transmitted frames is smallest for the 1st group of connections (1–8), as they share the bottleneck link with the lowest available capacity. For these connections, the average value of the cropping ratio is only about 0.25. This is because the bottleneck link is running at saturation (96% utilization), and the sources have to be throttled back substantially. If we had an admission-control policy active, we might have admitted fewer than the eight connections for this group, based on the limited capacity of the S3 → H5 link.

The next group of connections (9–16) share the capacity on their bottleneck (25 Mb/s) that remains after satisfying all of the Group 1 connections. These connections are able to drive the utilization of the link to 94%. The reduction of the target rate is less than that for the first group, receiving nearly 75% of their "nominal" desired rate.

Finally, Group 3 connections (17–24) which are potentially limited by the 155 Mb/s link between S1 and S2, get nearly all their "nominal rate" because the link is only 48% utilized on the average. The DEMAND from all of the 24 connections together is only about half the capacity of the 155-Mb/s link,

thus resulting in no substantial degradation for the Group 3 connections.

We thus observe that the weighted max–min fair-share algorithm does the "right thing" in providing a fair allocation of the bottleneck bandwidth to those connections limited by it. The resulting performance impact for the video source is commensurate with the bandwidth share it receives.

VIII. Concluding Remarks

We proposed a scheme that uses an explicit-rate based feedback mechanism, similar to that used for ATM's ABR service class, for transporting compressed video traffic. The key features of our scheme are the following.

- We use the inherent negotiation in ABR using RM cells to allow sources to indicate their desired rate over very short intervals. This desired rate is generated using forecasts which exploit the high short-term correlation in video. The source adapts its rate to the rate communicated back by the network, whenever necessary. The source enhances its adaptation by using information about the source buffer occupancy and a recent set of allowed rates.
- A source uses the MCR guarantee of ABR to ensure that the transported video stream get an acceptable service quality. This use of the MCR distinguishes our scheme from completely rate-adaptive video, such as those used in the Internet video tools (NV, VIC).
- We expect a separation of the video flows, that are admission-controlled, from bursty data that may not be admission-controlled. Since explicit-rate-based ABR maintains small network queues by minimizing burstiness (at the cell level) and ensures that the capacity of the links in the network are not over-allocated, we can achieve acceptably low delays for the video flows.
- We propose a new rate-allocation mechanism in the network based on a weighted max–min fairness criterion. This enhanced allocation mechanism allows the network to treat flows unequally (in proportion to their weights), with regard to the rate allocations at their bottleneck. With this mechanism, higher rate sources whose quality is more likely to be affected are treated preferentially (instead of all sources experiencing the same rate reduction).

We presented simulation results showing the efficacy of our proposed scheme, with a long video-teleconferencing trace. We showed that the weighted max–min fair-allocation scheme evenly degrades quality across multiple video sources when the network is unable to grant their requested rate. We also showed that even when the bottleneck link is being utilized near saturation, the total end-to-end frame delay is within acceptable levels (less than 300 ms) over a WAN with a propagation delay of 80 ms. We also make observations on the consequences for admission control when an ABR service is used to transport compressed video. In particular, we observe that it may be necessary to use a rate that is a factor larger than the MCR as the basis for admitting a connection, rather than then the MCR itself.

We believe that transporting video using the enhanced explicit-rate-based feedback control, as proposed in this paper, has the potential to combine the best features of VBR, CBR, and RCBR video without some of their primary drawbacks.

References

[1] D. Bertsekas and R. Gallagher, *Data Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1992, ch. 6.

[2] F. Bonomi and K. Fendick, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network,* vol. 9, pp. 25–39, Mar./Apr. 1995.

[3] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *Proc. ICC'95,* pp. 10–15.

[4] A. Charny, K. K. Ramakrishnan, and A. Lauck, "Time scale analysis and scalability issues for explicit rate allocation in ATM networks," *IEEE/ACM Trans. Networking,* vol. 4, pp. 569–581, Aug. 1996.

[5] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. ACM SIGCOMM'89 Conf.,* pp. 174–187.

[6] D. Tse and M. Grossglauser, "Measurement-based call admission control: Analysis and simulation," in *Proc. IEEE INFOCOM,* Apr. 1997, pp. 237–248.

[7] A. Elwalid, D. Heyman, T. V. Lakshman, D. Mitra, and A. Weiss, "Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing," *IEEE J. Select. Areas Commun.,* vol. 13, pp. 1004–1016, Aug. 1995.

[8] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A simple and efficient service for multiple time-scale traffic," in *Proc. ACM SIGCOMM'95 Conf.,* pp. 219–230.

[9] E. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE J. Select. Areas Commun.,* vol. 9, pp. 1024–1039, Sept. 1991.

[10] V. Jacobson, "Multimedia conferencing on the Internet," *Conference Tutorial 4,* presented at the ACM SIGCOMM, Aug. 1994.

[11] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Vishwanathan, "ERICA switch algorithm: A complete description," *ATM Forum Contribution, AF-TM 96-11721,* ATM Forum Traffic Management Working Group, Aug. 1996.

[12] H. Kanakia, P. P. Mishra, and A. R. Reibman, "An adaptive congestion control scheme for real-time packet video transport," in *Proc. ACM SIGCOMM'93 Conf.,* pp. 20–31.

[13] ———, "Packet video transport in ATM networks with single-bit feedback," presented at the 6th Int. Workshop on Packet Video, Portland, OR, Sept. 1994.

[14] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "An efficient rate allocation algorithm for ATM networks providing max-min fairness," presented at the 6th IFIP Int. Conf. High Performance Networking, Spain, Sept. 11–15, 1995.

[15] D. Heyman, "The GBAR source model for VBR videoconferences," *IEEE/ACM Trans. Networking,* vol. 5, pp. 554–560, Aug. 1997.

[16] D. Heyman and T. V. Lakshman, "Source models of broadcast-video traffic," *IEEE/ACM Trans. Networking,* vol. 4, pp. 40–48, Feb. 1996.

[17] D. Heyman, T. V. Lakshman, A. Tabatabai, and H. Heeke, "Modeling teleconference traffic from VBR video coders," in *Proc. ICC'94,* pp. 1744–1748.

[18] D. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of VBR video teleconference traffic in ATM networks," *IEEE Trans. Circuits Syst. Video Technol.,* pp. 49–59, Mar. 1992.

[19] T. J. Ott, T. V. Lakshman, and A. Tabatabai, "A scheme for smoothing delay sensitive traffic offered to ATM networks," in *Proc. IEEE INFOCOM,* May 1992, pp. 776–785.

[20] S. Keshav, "A control theoretic approach to flow control," in *Proc. ACM SIGCOMM,* Sept. 1991.

[21] P. P. Mishra and H. Kanakia, "A hop by hop rate based congestion control scheme," in *Proc. ACM SIGCOMM,* Aug. 1992, pp. 112–123.

[22] P. P. Mishra, "Fair bandwidth sharing for video traffic sources using distributed feedback control," presented at the IEEE GLOBECOM, Singapore City, Singapore, Nov. 1995.

[23] S. Mukherjee, D. Reininger, and B. Sengupta, "An adaptive connection admission control policy for VBR+ service class," in *Proc. IEEE INFOCOM,* San Francisco, CA, Mar./Apr. 1998, p. 849.

[24] K. K. Ramakrishnan, R. Jain, and D.-M. Chiu, "Congestion avoidance in computer networks with a connection less network layer—Part IV: A selective binary feedback scheme for general topologies," DEC-TR-510, Digital Equipment Corp., 1987.

[25] L. Roberts, "Enhanced PRCA (proportional rate control algorithm)," *ATM Forum Contribution,* AF-TM 94-0735R1, Aug. 1994.

[26] "ATM forum traffic management specification version 4.0," *Draft Specification ATM Forum 95-0013R11,* Mar. 1996.

T. V. Lakshman (S'84–M'85–SM'98), for photograph and biography, see this issue, p. 615.

Partho P. Mishra received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, in 1988, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 1991 and 1993, respectively, all in computer science.

He is currently a Principal Member Technical Staff in the Networking and Distributed System Research Center, AT&T Labs-Research, Florham Park, NJ. His research interests include traffic management, wireless networking and packet telephony, and video services.

K. K. Ramakrishnan (S'76–A'83) received the B.S. degree in electrical engineering from Bangalore University, India, in 1976, the M.S. degree in automation from the Indian Institute of Science in 1978, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1983.

He is a currently a Technology Leader with AT&T Labs-Research in Florham Park, NJ. He was a Consulting Engineer with Digital Equipment Corporation until 1994. His research interests include the design and performance of algorithms for computer networks and distributed systems. He has worked and published papers in the areas of congestion control and avoidance, algorithms for Ethernet, FDDI and ATM, load balancing, distributed systems performance, packet telephony, and issues relating to network I/O. He has several patents issued in these areas.

Dr. Ramakrishnan is an Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING, *IEEE Network Magazine*, and is on the editorial board for the *Computer Communications Journal*. He is a member of the End-to-End Research Group, as part of the Internet Research Task Force and participates in the IETF and the ATM Forum.