Network Bandwidth Sharing for Transporting Rate-Adaptive Packet Video Using Feedback *

Y. Thomas Hou[†] Shivendra S. Panwar[‡] Zhi-Li Zhang[§] Henry Tzeng[¶] Ya-Qin Zhang[||]

Abstract

This paper presents a framework for network bandwidth sharing to transport rate-adaptive packet video using feedback. We show how a weight-based bandwidth sharing policy can be used to allocated network bandwidth among competing video connections and design a feedback control algorithm using the available bit rate (ABR) flow control mechanism. A novel video source rate adaptation algorithm is introduced to decouple a video source's actual transmission rate from the rate used for the protocol convergence. We also demonstrate how an on-line minimum rate renegotiation and weight adjustment mechanisms can be employed to further enhance the flexibility of our feedback control protocol.

1 Introduction

The ABR protocol has been recently shown to be a viable feedback control mechanism for transporting rate-adaptive video [6, 7, 11]. A key performance issue associated with using feedback to control video transmission is network bandwidth sharing among competing video connections. Prior efforts such as [5, 8, 9, 11] did not address this issue. In [6, 7], a MCR-proportional max-min policy was proposed to support rate-adaptive video. But it is not clear what a distributed feedback control algorithm should be employed to achieve such network bandwidth sharing policy.

This paper presents a framework for network bandwidth sharing for rate-adaptive video connections using an ABR-like feedback control.

We first present a generic weight-based bandwidth sharing policy, also called Weight-Proportional Max-Min (WPMM) policy, to allocate network bandwidth among video connections. Unlike [6, 7] where the weight of a connection is its MCR, the weight associated with each connection in this paper is generic, i.e., decoupled (or independent) from its MCR. To achieve such policy in a distributed network, we design a feedback control algorithm employing ABR mechanism. We show that our algorithm provides guaranteed convergence to WPMM policy among video connections under any network configuration.

Our feedback control algorithm has the attractive property that a source's actual transmission rate can be decoupled from the rate information used for the convergence of flow control protocol. To take advantage of this property, we present a novel video source rate adaptation algorithm, which provides a smooth (i.e. infrequent) encoder rate adjustment according to its own time scale. We show that our video source rate adaptation algorithm is able to adjust a video source's rate gracefully to potential available network bandwidth without undergoing the undesirable frequent fluctuations of feedback rate.

Another major contribution of this paper is that we have demonstrated the feasibility of on-line dynamic renegotiation of sustainable rate (MCR) and weight assignment. Such flexibility is particular important since the initial estimate of minimum rate requirement (MCR) or weight may not be accurate to reflect the actual need of a particular video connection. Without such renegotiation mechanisms, an accurate estimate of MCR is essential to support minimum video quality. We show that by using on-line minimum rate renegotiation and weight adjustment mechanisms, each video connection can adapt to a new sustainable bandwidth (MCR) or a new weight assignment during the course of the connection.

The reminder of this paper is organized as follows. Section 2 examines the ABR mechanism and shows compelling motivations of using such mechanism to transport rate-adaptive video. Section 3 presents the weight-based bandwidth sharing policy. In Section 4, we show an ABR algorithm to achieve such bandwidth sharing policy. Section 5 presents a video source rate adaptation algorithm. In Section 6, we demonstrate an on-line MCR renegotiation and weight adjustment mechanism. Section 7 concludes this paper.

2 Supporting Rate-Adaptive Video Using an ABR-Like Mechanism

The ABR mechanism allows a source end system to adjust its information transfer rate based on the bandwidth availability in the network [1]. A generic ABR flow control mechanism for a connection is shown in Fig. 1. Despite the somewhat complex specifications for ABR in [1], the basic idea for ABR is, in fact, quite simple. Basically, ABR employs the cooperation between the sources and the network through the following two key components: 1) Information exchange: Special control packets (or Resource Management (RM)) cells are used to exchange information between the sources and the network; and 2) Source rate adaptation: A source adjusts its transmission rate based on the feedback information in the returning RM cells.

*This work was supported in part by the New York State Center for Advanced Technology in Telecommunications (CATT), Polytechnic University, Brooklyn, NY, USA.

[†] Fujitsu Laboratories of America, Sunnyvale, CA, USA.

[‡] Polytechnic University, Brooklyn, NY, USA.

[§] University of Minnesota, Minneapolis, MN, USA.

[¶] Bell Labs, Lucent Technologies, Holmdel, NJ, USA.

[||] Sarnoff Corporation, Princeton, NJ, USA.

0-7803-4984-9/98/$10.00 ©1998 IEEE.

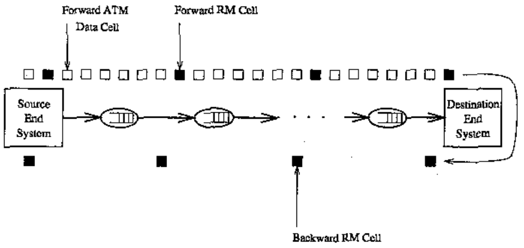

Figure 1: ABR mechanism.

For the video sources considered in this paper, we assume that each video employs adaptive, multi-layered encoding combined with feedback-based rate control mechanism and can let its encoder match the explicit feedback rate in the returning RM cell. The adaptive multi-layered encoding divides the real-time video stream into high and low priority streams, and the feedback mechanism control the output rates of each of these streams to account for the congestion state of the network. The high priority cell rate can be adjusted to approximate to the amount of some guaranteed minimum bandwidth through reservation, while the low priority cell rate is adjusted to make use of any additional unguaranteed (or available) bandwidth. The control of the overall output rate of the video encoder requires the adjustment of the encoder's quantization (or coarseness) parameters.

When used to transport such video traffic in an integrated services network, an ABR-like flow control mechanism combines the best features of CBR and VBR traffic control without their major drawbacks. The admission control can make resource reservation for the lowest acceptable quality of service for video. In particular, the MCR concept in ABR comes naturally to provide such CBR-like service to ensure minimum video transmission and presentation quality. With feedback, the video encoder can still adjust its transmission rate by modulating the quantization level to aggressively adapt to any additional available bandwidth from the network through the explicit rate information in the returning RM cell. This is much simpler than having to make a prior assumption about the traffic statistics that a video source may have.

3 A Bandwidth Sharing Policy

Since there are many video connections in a network, each trying to exploit additional available bandwidth, ensuring fairness in bandwidth sharing is a challenging problem. This is due to the fact that a feedback-based control is distributed in nature and does not have a global view of the network. Therefore, when we design a feedback control algorithm, it is fundamental that such algorithm can achieve some rate allocation policy objective. In this section, we show how a particular network bandwidth sharing policy can be used for transporting rate-adaptive compressed video. This policy guarantees each video connection the required bandwidth for minimal acceptable presentation quality. At the same time, it efficiently and fairly allocates the remaining network bandwidth among video connections to further enhance their presentation quality.

In our model, a network $\mathcal{N}$ is characterized by interconnecting switches with a set of links $\mathcal{L}$. Let C_ℓ be the capacity of link $\ell \in \mathcal{L}$. A set of video connections $\mathcal{S}$ are supported by the network and share the network bandwidth. Each connection $s \in \mathcal{S}$ traverses one or more links in $\mathcal{L}$ and is allocated a specific rate. Let $\mathcal{S}_\ell$ denote the set of connections traversing link ℓ and MCR_s and PCR_s be the minimum required rate and peak rate constraint (usually imposed by the end system's port access speed) for each video connection $s \in \mathcal{S}$. In our policy, once a video connection is admitted into the network, its minimum rate (MCR) is always guaranteed. For feasibility, we must have

$$\sum_{s \in \mathcal{S}_\ell} \mathrm{MCR}_s \leq C_\ell \quad \text{for every } \ell \in \mathcal{L}. \tag{1}$$

This criterion is used by admission control at call setup time to determine whether or not to accept a new video connection.

From Eq. (1), we see that there may be excessive bandwidth available on link $\ell \in \mathcal{L}$ after first allocating each connection with its sustainable bandwidth (MCR). We employ the following policy to allocate the remaining network bandwidth. We let each connection $s \in \mathcal{S}$ be associated with a weight (or priority) w_s. Such weight is set by each user at call set up time. The remaining network bandwidth is allocated by using the weighted version of the max-min policy based on each connection's weight. The final bandwidth allocated to each connection is its minimum rate plus an additional "weighted" max-min share. The following algorithm shows how this rate allocation policy works.

Algorithm 1 Weight-Based Rate Allocation

1. Start the rate allocation of each connection with its minimum rate (MCR).

2. Increase the rate of each connection with an increment proportional to its weight until either some link becomes saturated or some connection reaches its peak rate constraint (PCR), whichever comes first.

3. Remove those connections that either traverse saturated links or have reached their PCRs and the capacity associated with such connections from the network.

4. If there is no connection left, the algorithm terminates; otherwise, go back to Step 2 for the remaining connections and remaining network capacity. □

Note that the weight of each connection is decoupled (i.e. independent) from its minimum rate. This adds considerable more flexibility than a MCR-proportional max-min policy used in [6]. We use the following example to illustrate how Algorithm 1 works.

Example 1 A Three-Node Network

In this network (Fig. 2), there are four video connections and the output port links of SW1 (Link 12)

Table 1: Minimum rate requirement, peak rate constraint, weight, and weight-based rate allocation for each connection in the three-node network.

VCI	MCR (Mbps)	PCR (Mbps)	Weight	Rate Allocation (Mbps)
VC1	0.5	7.5	1	1.5
VC2	1.5	9.0	3	4.5
VC3	2.0	4.0	4	4.0
VC4	1.0	10.0	2	8.5

Table 2: Iterations of rate allocation for each connection in the three-node network.

Iterations	VC{(MCR, PCR)(in Mbps), W}				Remaining Capacity (Mbps)	
	VC1 {(0.5, 7.5), 1}	VC2 {(1.5, 9.0), 3}	VC3 {(2.0, 4.0), 4}	VC4 {(1.0, 10.0), 2}	Link 12	Link 23
initialization	0.5	1.5	2.0	1.0	6.0	8.5
1st	1.0	3.0	$\boxed{4.0}$	2.0	2.0	7.0
2nd	$\boxed{1.5}$	$\boxed{4.5}$		3.0	0	5.5
3rd				$\boxed{8.5}$		0

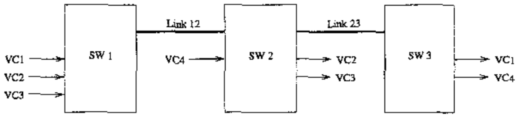

Figure 2: A three-node network configuration.

and SW2 (Link 23) are potential bottleneck links for these connections. Assuming the capacity of Link12 and Link23 are 10 Mbps, and the minimum bandwidth requirement, peak bandwidth constraint, and weight for each connection are listed in Table 1. Table 2 shows the iterations of using Algorithm 1 to achieve our network bandwidth sharing policy. □

Also shown in the above examples is that the weight proportional rule is used only during the intermediate steps in Algorithm 1 and the the final bandwidth allocated to each connection, after offsetted by its minimum rate, may not necessarily be proportional to its weight. A connection that traversing more hops (or bottleneck links) usually gets smaller proportion of bandwidth (with respect to its weight) than a connection with the same weight going through fewer number of hops. Another point worth mentioning is that only the MCR portion is intended to provide a CBR-like rate service and any additional bandwidth sharing from the remaining network bandwidth based on each connection's weight is unguaranteed since they may be taken by a newly joined video connection with some minimum rate requirement.

4 Feedback Control Algorithm

In this section, we show how an ABR-like flow control algorithm can be designed to achieve the weight-based bandwidth sharing policy for rate-adaptive video service. We first specify each connection's source and destination behaviors [1].

Algorithm 2 End System Behavior

- **Source Behavior:**[1]
  The source starts with ACR := ICR, ICR $\geq$ MCR;
  For every N_{rm} transmitted data cells, the source sends a forward RM(CCR, MCR, ER, W) cell with its fields initialized with CCR := ACR; MCR := MCR; ER := PCR; W := W;
  Upon the receipt of a backward RM(CCR, MCR, ER, W) cell from the destination, the ACR at the source is adjusted to: ACR := ER.

- **Destination Behavior:** The destination end system of a connection returns every RM cell back towards the source upon receiving it. □

Now we present the switch algorithm used in the network, which calculates the rate allocation for each connection. The following are the link parameters and variables used by our switch algorithm.

n_ℓ: Number of connections in $\mathcal{S}_\ell$, i.e., $n_\ell = |\mathcal{S}_\ell|$, $\ell \in \mathcal{L}$.
r_ℓ^i: CCR value of connection $i \in \mathcal{S}_\ell$ at link ℓ.
b_ℓ^i: Bit used to mark connection $i \in \mathcal{S}_\ell$ at link ℓ. $b_\ell^i = 1$ if connection $i \in \mathcal{S}_\ell$ is *marked* at link ℓ or 0 otherwise.
$\mathcal{M}_\ell$: Set of connections marked at link ℓ, i.e. $\mathcal{M}_\ell = \{i \,|\, i \in \mathcal{S}_\ell \text{ and } b_\ell^i = 1\}$.
$\mathcal{U}_\ell$: Set of connections unmarked at link ℓ, i.e. $\mathcal{U}_\ell = \{i \,|\, i \in \mathcal{S}_\ell \text{ and } b_\ell^i = 0\}$, and $\mathcal{M}_\ell \cup \mathcal{U}_\ell = \mathcal{S}_\ell$.
μ_ℓ: A variable at link ℓ used to facilitate rate calculation, which is calculated as follows.

Algorithm 3 μ_ℓ Calculation

```
if $n_\ell = 0$ then $\mu_\ell := \infty$;
```

[1] We use some unspecified field in the RM cell to carry the connection's weight.

else if $n_\ell = |\mathcal{M}_\ell|$ then

$$\mu_\ell := \frac{C_\ell - \sum_{i\in\mathcal{S}_\ell} r_\ell^i}{\sum_{i\in\mathcal{S}_\ell} w_i} + \max_{i\in\mathcal{S}_\ell} \frac{r_\ell^i - \text{MCR}^i}{w_i};$$

else

$$\mu_\ell := \frac{(C_\ell - \sum_{i\in\mathcal{S}_\ell} \text{MCR}^i) - \sum_{i\in\mathcal{M}_\ell}(r_\ell^i - \text{MCR}^i)}{\sum_{i\in\mathcal{U}_\ell} w_i}.$$

□

The following algorithm specifies our switch behavior at each output port, with the following initializations: $\mathcal{S}_\ell = \emptyset$; $n_\ell = 0$; and $\mu_\ell = \infty$.

Algorithm 4 Switch Behavior

```
Upon the receipt of a forward RM(CCR, MCR, ER, W)
cell from the source of connection $i$ {
    if RM cell signals connection termination[2]{
        $\mathcal{S}_\ell := \mathcal{S}_\ell - \{i\}$;   $n_\ell := n_\ell - 1$;
        table_update();
        }
    if RM cell signals connection initiation {
        $\mathcal{S}_\ell := \mathcal{S}_\ell \cup \{i\}$;   $n_\ell := n_\ell + 1$;
        $r_\ell^i :=$ CCR;   MCR$^i :=$ MCR;   $w_i :=$ W;
        $b_\ell^i := 0$;
        table_update();
        }
    else {
        $r_\ell^i :=$ CCR;
        if ($\frac{r_\ell^i - \text{MCR}^i}{w_i} \leq \mu_\ell$) then $b_\ell^i := 1$;
        table_update();
        }
    Forward RM(CCR, MCR, ER, W) towards its
    destination;
}

Upon receiving a backward RM(CCR, MCR, ER, W)
cell from the destination of connection $i$ {
    ER := max{min{ER, ($\mu_\ell \cdot w_i +$ MCR$^i$)},  MCR$^i$};
    Forward RM(CCR, MCR, ER, W) towards its
    source;
}

table_update()
{
    rate_calculation_1: use Algorithm 3 to calculate $\mu_\ell^1$;
    Unmark any marked connection $i \in \mathcal{S}_\ell$ at link $\ell$
    with $\frac{r_\ell^i - \text{MCR}^i}{w_i} > \mu_\ell^1$;
    rate_calculation_2: use Algorithm 3 to calculate $\mu_\ell$;
    if ($\mu_\ell < \mu_\ell^1$), then {
        Unmark any marked connection $i \in \mathcal{S}_\ell$ at
        link $\ell$ with $\frac{r_\ell^i - \text{MCR}^i}{w_i} > \mu_\ell$;
        rate_calculation_3: use Algorithm 3 to
        calculate $\mu_\ell$ again;
        }
}
```

□

As shown in the above end systems and switch algorithms, each source is allowed to transmit at a rate of ACR and adjust its ACR to the ER rate upon receiving returning RM cell. The CCR field in the forward RM cell (set to ACR at source) informs the switch along its traversing path about the connection's current rate. The variable μ_ℓ at link $\ell \in \mathcal{L}$ estimates MCR-offsetted and weight-normalized max-min rate. The switches maintains a table at each output port to record all the traversing connections and their rate information. The set of connections are considered "un-conforming" (denoted by set $\mathcal{U}_\ell$ at link ℓ) if their last seen CCR satisfies $\frac{\text{CCR}-\text{MCR}}{\text{W}} > \mu_\ell$. Similarly, connections with $\frac{\text{CCR}-\text{MCR}}{\text{W}} \leq \mu_\ell$ are said to be "conforming" (denoted by set $\mathcal{M}_\ell$ at link ℓ) and are therefore marked with the b bit (set to 1). The connections in the conforming set are assumed satisfying our rate allocation while those in the un-conforming set are still under transient iterations. During the iteration process, after each time μ_ℓ is updated, a connection previously belonging to $\mathcal{M}_\ell$ may be unmarked and become a connection in $\mathcal{U}_\ell$. It has been shown that eventually all connections in the network become conforming and satisfy our bandwidth sharing policy [2].

Simulation Results

We use some simulation results to show the convergence property of our algorithm. For the networks in the simulation, all ATM switches are assumed to have output port buffering with sufficient internal switching capacity for the aggregate rates of all input ports. Each output port employs the simple first-in-first-out (FIFO) queuing discipline for all cells destined to that port. We assume the internal switching delay for a cell from an input port to an output port is 4 μs (does not include the queuing delay at the output port). In consistent with the link capacity used in the examples in Section 3, we set $C_\ell = 10$ Mbps at every link $\ell \in \mathcal{L}$ for explicit rate calculation (Algorithm 4). In actual simulation, we set the link capacity to 10.526 ($= \frac{10}{0.95}$) Mbps and a target link utilization of 0.95, i.e., $C_\ell = \frac{10}{0.95} \times 0.95 = 10$. By setting a target link utilization strictly less than 1, we ensure that the potential buffer build up during transient period will be eventually emptied upon algorithm convergence. The distance from an end system (source or destination) to the switch is 1 kilometer and the link distance between switches is 1000 kilometer (corresponding to a wide area network). We assume the propagation delay is 5 μs per kilometer.

At a source side, we set the initial transmission rate (i.e. ICR) to be the same as the minimum required rate (MCR) for the video. N_{rm} is set to 32 for all video connections.

A Peer-to-Peer Network

For this network (Fig. 3), there are three connections going to the same output port of SW1. The minimum rate requirement, peak rate constraint, weight, and rate allocation for each connection are listed in Table 3.

Figure 4 shows the ACR at source for VC1, VC2, and VC3, respectively. Each connection starts with

[2] This information is conveyed through some unspecified bits in the RM cell, which can be set either at the source or the UNI.

Table 3: Minimum rate requirement, peak rate constraint, weight, and weight-based rate allocation for each connection in the peer-to-peer network.

VCI	MCR (Mbps)	PCR (Mbps)	Weight	Rate Allocation (Mbps)
VC1	1.5	10.0	1	4.0
VC2	1.0	3.0	1	3.0
VC3	0.5	5.0	1	3.0

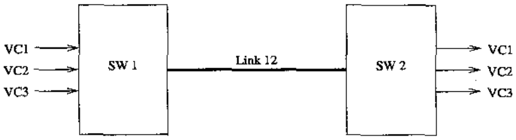

Figure 3: A peer-to-peer network.

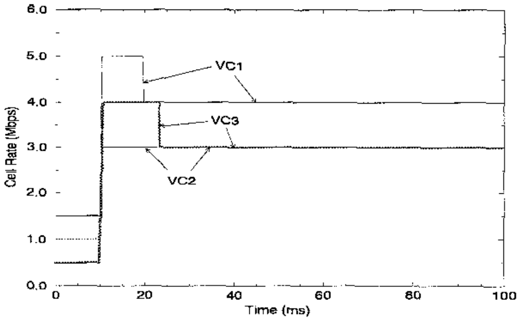

Figure 4: The ACR of all connections for the peer-to-peer network configuration.

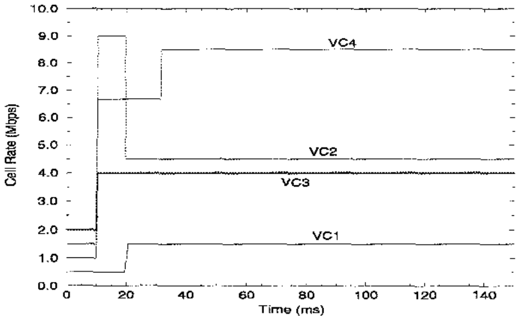

Figure 5: The ACR of all connections for the three-node network configuration.

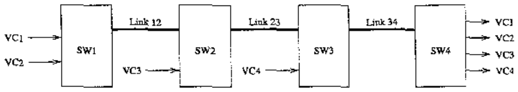

Figure 6: A parking lot network.

its minimum rate. The first RM cell of each connection returns to its source after one round trip time (RTT), or 10 ms. After a few iterations, we see that the cell rate of each connection converges to the rate listed in Table 3. Also, we find that during the course of iterations, the ACR of each connection is bounded between its minimum rate and peak rate, i.e., MCR $\leq$ ACR $\leq$ PCR.

The Three-Node Network

For this network (Fig. 2), there are four connections and the output port links of SW1 (Link12) and SW2 (Link23) are potential bottleneck links. The minimum required rate, peak rate constraint, weight, and rate allocation for each connection are listed in Table 1.

Figure 5 shows the ACR of each connection under our feedback control algorithm. Again, each connection starts with its minimum rate. The ACR of each connection is always bounded between its MCR and PCR during the course of the connection. Upon convergence, the rate allocation for each connection matches its rate listed in Table 1.

A Parking Lot Network

Fig. 6 shows a parking lot configuration, where connections VC1 and VC2 start from the first switch and go to the last switch; and connections VC3 and VC4 start from SW2 and SW3, respectively, and terminate at the last switch.

Table 4 lists the minimum rate requirement, peak rate constraint, weight, and our rate allocation under Algorithm 1 for each connection.

Figure 7 shows the ACR of each connection under our feedback control algorithm. Again, each connection's rate starts to transmit at MCR and converges to the optimal rates listed in Table 4.

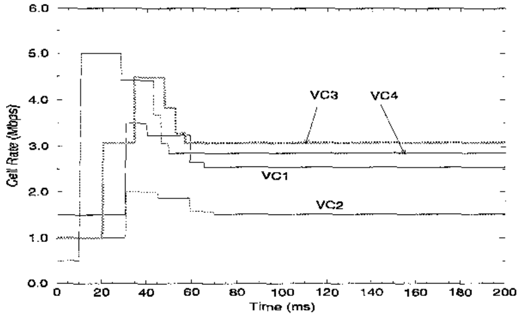

Figure 7: The ACR of all connections for the parking lot network configuration.

Table 4: Minimum rate requirement, peak rate constraint, weight, and weight-based rate allocation for each connection under the parking lot network.

VCI	MCR (Mbps)	PCR (Mbps)	Weight	Rate Allocation (Mbps)
VC1	1.5	3.5	4	2.543
VC2	1.0	2.0	2	1.522
VC3	1.0	5.0	8	3.087
VC4	0.5	5.0	9	2.848

5 Video Source Rate Adaptation Algorithm

A problem associated with our ABR-like feedback control algorithm is that during the transient convergence period when the algorithm is attempting to converge to the final rate allocation, the ER value in the returning RM cells is continually changing. Since the ACR of a source is adjusted to ER immediately upon receiving a returning RM cell (see Algorithm 2), ACR for the source is also continually changing. For example, in the simulation results in Figs. 4, 5, and particularly in Fig. 7, the ACR variable of a connection keeps undergoing fluctuations during the convergence period. Such transient fluctuation of ACR is undesirable since the video encoder's quantization adjustment period may not be able to follow such frequent variations of ER value in the returning RM cells. Also, the rapid fluctuations of the video coding rate may adversely impact the video quality.

In this section, we present a novel video source rate adaptation algorithm that separates a source's actual transmission rate from its ACR variable. It is both simple to implement and effective to adapt to the final optimal bandwidth share. Our video source rate adaptation algorithm is based on the following fundamental property of our feedback control algorithm.

Note that in our source algorithm (Algorithm 2), the ACR of a source is adjusted immediately upon receiving a returning RM cell. A closer look at the mechanics of our switch algorithm (Algorithm 4) reveals that the ACR variable at a source (recorded as CCR in the forward RM cell) is used as a variable solely for the purpose of distributed protocol convergence iterations and a source's true transmission rate does not affect the convergence property. That is, a source's true transmission rate does not have to be identical to its ACR at all time. For example, as long as a source's true transmission rate is between its MCR and ACR, the overall feedback control protocol can still operate properly (i.e. the ACR for each connection will converge to our optimal rate allocation). Such decoupling property is a consequence of our special design of switch algorithm where a table is used to keep track of the traversing connections and their rate information, and the fact that a source's true transmission rate does not play any role in the ER calculation. Feedback control algorithms such as [3, 4, 6, 10] are unable to offer such rate decoupling property since a source's true transmission rate is used in ER calculation.

We propose the following simple source rate adaptation algorithm for each video, which utilizes our unique rate decoupling property in our feedback control algorithm. Instead of setting a video source's actual transmission rate directly to its ACR, we introduce a new parameter at the source, called True Cell Rate (TCR), to decouple the direct relationship between a source's actual transmission rate and the ACR variable. The TCR will be the true transmission rate of a video source and the ACR will only be used as a reference variable by the source for the convergence of flow control protocol. A source keeps updating its ACR upon receiving each returning RM cell but only adjusts its transmission rate (i.e. TCR) at a time interval, say I, which can be set flexibly according to each source encoder's physical property. This is achieved by keeping a local clock at the source as a time reference and use a variable called Rate Adjustment Time (RAT) as a reminder for the next time point that a source should adjust its transmission rate. The details of a video source's rate adaptation algorithm is given in Algorithm 5.

Algorithm 5 Video Source Rate Adaptation

```
• Initially, the source starts to transmit at
  ACR := ICR with ICR ≥ MCR and sets
  RAT = time + I.

• For every $N_{rm}$ transmitted data cells, the source
  sends a forward RM(CCR, MCR, ER, W) cell
  with its fields initialized with CCR := ACR;
  MCR := MCR; ER := PCR; W := W.

• Upon the receipt of a backward RM(CCR, MCR,
  ER, W) cell from the destination
    ACR := ER;
    if (time ≥ RAT) {
        TCR := ACR;
        RAT := RAT + I;
  }                                              □
```

Note that the rate adjustment interval I is a local parameter that can be set by each source's encoder based on its physical property. Thus, each source may have different time interval I for its rate adjustment.

It should be clear that by using such source rate adaptation algorithm and the switch algorithm (Algorithm 4), our feedback control algorithm will still converge to the final weight-based rate allocation. The only difference is that the convergence time may be different under the new source algorithm.

To demonstrate the performance of our new source algorithm, we rerun the simulations in the last section. Each video source is assumed to have a frame rate of 30 frames/sec, or 33.33 ms per frame.

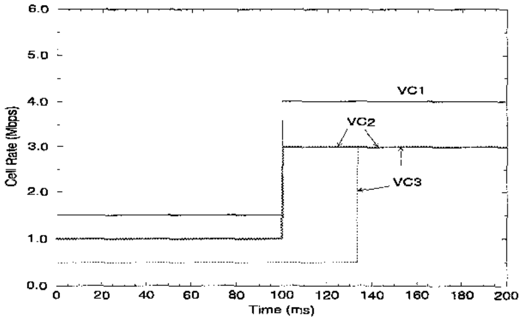

Figure 8: The TCR of all video connections for the peer-to-peer network configuration.

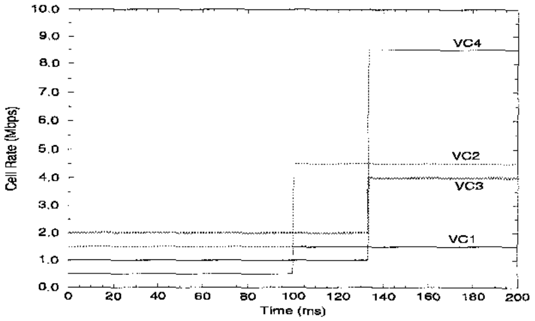

Figure 9: The TCR of all video connections for the three-node network configuration.

Figure 8 shows the source's true transmission rate (i.e. TCR) of each connection for the peer-to-peer network (see Fig. 3 and Table 3). With the source rate adjustment interval set to $I_1 = I_2 = 100$ ms (or 3 frames) for sources 1 and 2, and $I_3 = 133.33$ ms (or 4 frames) for source 3. That is, sources 1 and 2 adjusts their transmission rate every 3 frames while source 3 adjusts its rate every 4 frames. The simulation for ACR parameter for each connection is identical to those shown in Fig. 4. Comparing Fig. 8 with Fig. 4, we find that our video source rate adaptation algorithm shields effectively the source's rate adjustment from the undesirable ACR rate fluctuations during iterations while retaining the ability to adapt to the same optimal rate upon convergence.

Figure 9 shows the TCR of each connection for the three-node network (see Fig. 2 and Table 1). The source rate adjustment intervals are $I_1 = I_2 = 100$ ms (or 3 frames) for sources 1 and 2, and $I_3 = I_4 = 133.33$ ms (or 4 frames) for sources 3 and 4. The ACR simulation of each connection is identical to the those shown in Fig. 5. Again, we find that under our new source rate adaptation algorithm, each video is able to adapt smoothly to its final optimal rate without undergoing the frequent ACR fluctuations during convergence period.

Figure 10 shows the simulation results of TCR for each connection in the parking lot network (see Fig. 6 and Table 4). The source rate adjustment intervals are $I_1 = I_2 = 100$ ms (or 3 frames) for sources 1 and 2, $I_3 = 133.33$ ms (or 4 frames) for source 3, and $I_4 = 200$ ms (or 6 frames) for source 4. The ACR for all connections in this simulation are identical to those shown in Fig. 7. Again, under our new source rate adaptation algorithm, the transmission rate of each connection adapts smoothly to its final optimal bandwidth share without undergoing undesirable rate fluctuations as shown in Fig. 7.

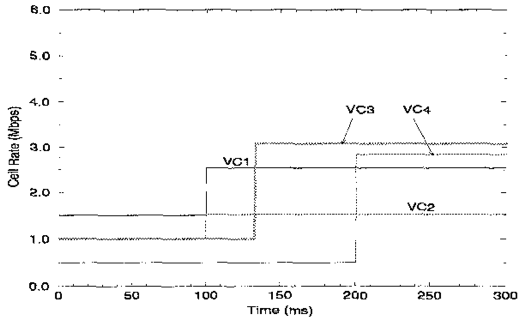

Figure 10: The TCR of all video connections for the parking lot network configuration.

In Algorithm 5, a source's rate adaptation interval I is a constant set by its encoder. Such fixed timing requirement may be further relaxed in our overall feedback control algorithm. That is, each source may adjust its rate at a variable time interval I. One of the significant benefits by using variable time interval for rate adjustment is that we can adjust the encoder's rate as soon as possible if the returning ER value is less than the current TCR. This will help to reduce network buffer requirements and alleviate network congestion substantially during transient period.

6 Dynamic MCR Renegotiation and Weight Adjustment Mechanisms

In our feedback control algorithm, each connection relies on MCR guarantee to support minimum video quality and a weight to share any excessive network bandwidth beyond its minimum rate. We have been implicitly assuming that each connection has prior knowledge of such requirements. However, it is sometimes difficult for each connection to have an accurate estimate of its minimum required rate, let alone to specify how much weight to be requested. Therefore, it will be very useful that a user can renegotiate its MCR or adjust its weight should the user feels necessary. This section demonstrates such capability in our feedback control algorithm.

6.1 MCR Renegotiation

Our feedback control algorithm is able to provide MCR renegotiation. The only criterion that needs to be checked is that the sum of the new set of MCRs cannot exceed the link's capacity on any link it traverses (see Eq. (1)). If Eq. (1) is satisfied, then the newly negotiated minimum rate may be granted, otherwise, the request is rejected.

It should be clear that each time when a connection changes its minimum rate, the optimal rate allocation for all connections in the network will change under

Table 5: Weight-based rate allocation for each connection in the peer-to-peer network before and after VC3 has renegotiated a new MCR.

VCI	MCR (Mbps)		PCR (Mbps)	Weight	Rate Allocation (Mbps)	
	before	after			before	after
VC1	1.5	1.5	10.0	1	4.0	3.0
VC2	1.0	1.0	3.0	1	3.0	2.5
VC3	0.5	3.0	5.0	1	3.0	4.5

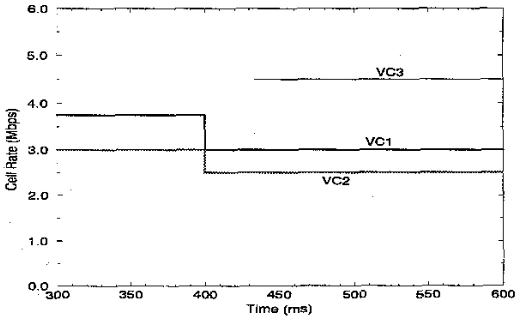

Figure 11: The TCR of all video connections for the peer-to-peer network configuration. VC3 has renegotiated a new MCR requirement.

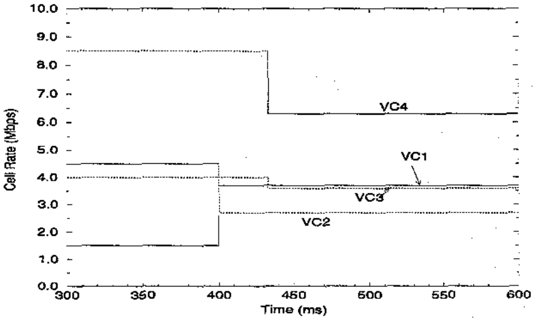

Figure 12: The TCR of all video connections for the three-node network configuration. VC1 changed its weight from 0.5 to 4.0.

Algorithm 1. We claim that our distributed feedback control algorithm is able to reiterate and converge to such new optimal rate allocation for all connections. This is because the convergence property of our switch behavior is independent from MCR renegotiation.

As an example, for the peer-to-peer network shown in Fig. 3, Table 3 shows the minimum rate, peak rate, weight, and optimal rate allocation for each connection. Our feedback control algorithm is shown to converge to the optimal rate allocation in Table 3 for each connection (see Fig. 4). Table 5 shows the the rate allocation for each connection when the minimum rate for VC3 is changed from 0.5 Mbps to 3.0 Mbps. The simulation results of our distributed algorithm before VC3's MCR change has been shown in Fig. 4. In Fig. 11, we continue the same simulation run in Fig. 4 and at time $t = 300$ ms, we change VC3's MCR requirement from 0.5 Mbps to 3.0 Mbps. At time $t = 400$ms, VC1 and VC2 adapt to their new bandwidth of 3.0 Mbps and 2.5 Mbps, respectively (since $I_1 = I_2 = 100$ms); at $t = 433.33$ms, VC3 adapts to its new optimal rate of 4.5 Mbps (since $I = 133.33$ms). The new rate allocation for each connection by our feedback control algorithm match the optimal rates listed in Table 5.

6.2 Weight Adjustment

Unlike MCR renegotiation, where a connection's MCR adjustment may be denied if such negotiation violates Eq. (1), the adjustment of a connection's weight is always achievable. This is because the minimum rate of a connection corresponds to a guaranteed rate and offers a CBR-like service, while the weight of a connection is used to share any unguaranteed (or available) network bandwidth in addition to its minimum rate. Similar to the case in MCR renegotiation, once a connection adjusts its weight, the optimal bandwidth allocation for all connections will change under Algorithm 1. Again, our feedback control algorithm is able to converge to the new optimal rate vector through distributed iterations.

As an example, for the three-node network (Fig. 2), with minimum rate and peak rate for each connection being the same as those listed in Table 1, Table 6 shows the optimal rate allocation (under Algorithm 1) for each connection before and after the weight of VC1 is adjusted from 0.5 to 4.0. Figure 12 continues the same simulation run in Fig. 9 and at time $t = 300$ ms, the weight of VC1 is adjusted from 0.5 to 4.0 in its forward RM cells. At time $t = 400$ ms, the rates for VC1 and VC2 have adapted to their new optimal rates of 3.7 Mbps and 2.7 Mbps, respectively (since $I_1 = I_2 = 100$ ms); and at $t = 433.33$ ms, the rates of VC3 and VC4 have also each adapted to their optimal rates of 3.6 Mbps and 6.3 Mbps (since $I_3 = I_4 = 133.33$ ms). Comparing with those rates listed in Table 6, we have demonstrated that our distributed feedback control algorithm is able to adapt to the new optimal rate vector under weight adjustment of a connection.

7 Concluding Remarks

The ABR flow control mechanism offers attractive features for transporting rate-adaptive video. Such feedback control keeps the simplicity of admission control for CBR to guarantee the minimum video quality and exploits any available network bandwidth through feedback. This paper sets up a framework for network bandwidth sharing among rate-adaptive

Table 6: Weight-based rate allocation for each connection in the three-node network after VC1 changes its weight.

VCI	MCR (Mbps)	PCR (Mbps)	Weight		Rate Allocation (Mbps)	
			before	after	before	after
VC1	0.5	7.5	0.5	$\boxed{4.0}$	1.5	3.7
VC2	1.5	9.0	1.5	1.5	4.5	2.7
VC3	2.0	4.0	2.0	2.0	4.0	3.6
VC4	1.0	10.0	1.0	1.0	8.5	6.3

video using the ABR-like feedback control. The main contributions in this paper are listed as follows. We showed how a weight-based network bandwidth sharing policy can be used for video connections and presented an ABR-like feedback control algorithm to achieve this policy in a distributed network. Our feedback control algorithm is designed with the unique property that a source's actual transmission rate can be decoupled from the ACR variable used for protocol convergence. By taking advantage of such property, we proposed a novel source rate adaptation algorithm and demonstrated that our overall feedback control algorithm is capable to converge smoothly to the final optimal rates without frequent fluctuations during transient period. Furthermore, we showed how simple MCR renegotiation and weight adjustment mechanisms can be employed to further enhance our distributed flow control algorithm for video traffic.

References

[1] ATM Forum Technical Committee, "Traffic Management Specification, Version 4.0," *ATM Forum Contribution, AF-TM 96-0056.00*, April 1996.

[2] Y. T. Hou, H. Tzeng, and S. S. Panwar, "A Generic Weight Based Network Capacity Sharing Policy for ATM ABR Service," *Tech. Report CATT 97-118*, Center for Advanced Technology in Telecommunications (CATT), Polytechnic University, Brooklyn, NY, July 1, 1997.

[3] R. Jain, *et al.*, "ERICA Switch Algorithm: A Complete Description," *ATM Forum Contribution, AF-TM 96-1172*, August 1996.

[4] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Dynamics of an Explicit Rate Allocation Algorithm for Available Bit Rate (ABR) Service in ATM Networks," *Proc. 6th IFIP Int. Conf. High Performance Networking (HPN '95)*, pp. 143-154, Sept. 1995.

[5] H. Kanakia, P. P. Mishra, and A. Reibman, "An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport," *IEEE/ACM Trans. on Networking*, vol. 3, no. 6, pp. 671–682, Dec. 1995.

[6] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan, "Transporting Compressed Video over ATM Networks with Explicit Rate Feedback Control," *Proc. IEEE INFOCOM'97*, pp. 38–47, Kobe, Japan, April 7–12, 1997.

[7] P. P. Mishra, "Fair Bandwidth Sharing for Video Traffic Sources Using Distributed Feedback Control," *Proc. IEEE GLOBECOM'95*, pp. 1102–1108, Singapore, Nov. 14–16, 1995.

[8] Y. Omori, T. Suda, and G. Lin, "Feedback-Based Congestion Control for VBR Video in ATM Networks," *Proc. 6th International Workshop on Packet Video*, Portland, OR, Sept. 26–27, 1994.

[9] C. M. Sharon, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye, "Rate Control of VBR H.261 Video on Frame Relay Networks," *Proc. IEEE ICC'95*, pp. 1443–1447, Seattle, WA, June 18–22, 1995.

[10] K.-Y. Siu and H.-Y. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 81-106, October 1994.

[11] B. J. Vickers, M. Lee, and T. Suda, "Feedback Control Mechanisms for Real-Time Multipoint Video Services," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 512–530, April 1997.