ON NETWORK BANDWIDTH SHARING FOR TRANSPORTING RATE-ADAPTIVE PACKET VIDEO USING FEEDBACK

YIWEI THOMAS HOU[1*] SHIVENDRA S. PANWAR[2] ZHI-LI ZHANG[3]
HENRY TZENG[4] YA-QIN ZHANG[5]

[1] Fujitsu Laboratories of America, Sunnyvale, CA, USA
[2] Polytechnic University, Brooklyn, NY, USA
[3] University of Minnesota, Minneapolis, MN, USA
[4] Bell Laboratories, Holmdel, NJ, USA
[5] Microsoft Research, Beijing, China

SUMMARY

While existing research shows that feedback-based congestion control mechanisms are capable of providing better video quality and higher link utilization for rate-adaptive packet video, there has been relatively little study on how to share network bandwidth among competing rate-adaptive video connections, when feedback control is used in a fully distributed network. This paper addresses this issue by presenting a framework of network bandwidth sharing for transporting rate-adaptive packet video using feedback. We show how a weight-based bandwidth sharing policy can be used to allocate network bandwidth among competing video connections and design a feedback control algorithm using an available bit rate (ABR)-like flow control mechanism. A novel video source rate adaptation algorithm is also introduced to decouple a video source's actual transmission rate from the rate used for distributed protocol convergence. Our feedback control algorithm provides guaranteed convergence and smooth source rate adaptation to our weight-based bandwidth sharing policy under any network configuration and any set of link distances. Finally, we show the on-line minimum rate renegotiation and weight adjustment options in our feedback control algorithm, which offer further flexibility in network bandwidth sharing for video connections.

KEY WORDS: rate-adaptive video, minimum rate guarantee, rate allocation, feedback flow control, available bit rate (ABR), ATM

---

*Correspondence to: Y. T. Hou, Fujitsu Laboratories of America, 595 Lawrence Expressway, Sunnyvale, CA 94086-3922, USA. Tel. (408) 530-4529, Fax (408) 530-4515, Email: thou@fla.fujitsu.com.

1 INTRODUCTION

Most of the early work on the transport of real-time video over ATM networks relied on preventive congestion control mechanisms. Under such mechanisms, each user must declare a set of traffic parameters (or descriptors) during call set up time. Once admitted by the network, such connection must either conform to its traffic descriptors or the network will enforce such contract through usage parameter control (UPC) or policing. Such services, as defined by the ATM Forum Traffic Management Group, include the constant bit rate (CBR) service and variable bit rate (VBR) service. The primary advantage of using CBR for video is the simplicity of the connection management. However, such simplicity is achieved at the expense of both video quality and efficient network bandwidth utilization. The VBR service class attempts to provide better statistical multiplexing gain and the required quality of service [4, 7]. However, due to real-time video's often unpredictable nature, traffic descriptors are difficult, if not impossible in some cases, to predict accurately during call admission.

Due to the above problems with preventive congestion control techniques for packet video, there has been a shift toward the use of combination of preventive and reactive (*i.e.,* feedback-based) congestion control schemes [11, 14, 15]. All these mechanisms explore the rate-adaptive property of video by using some feedback signal from the network to control the video generation rate. Such mechanisms have shown to have two major benefits: 1) the quality of the video transmission is improved when the network is not congested or degrades gracefully when the network is congested; and 2) the network bandwidth is used efficiently.

Recently, a closed-loop flow control mechanism has been defined by the ATM Forum for the available bit rate (ABR) service class. It was originally assumed that data traffic would benefit from such flow control mechanism by taking advantage of any unused network bandwidth. However, most data traffic is bursty and small in size compared with the round-trip delay bandwidth product. Therefore, it turns out that a closed-loop flow control mechanism such as ABR may not be able to effectively control bursty data traffic in some cases. Consequently, a new service class, called unspecified bit rate (UBR) and without using any feedback control mechanism was introduced to transport bursty data.

Interestingly enough, at the same time, the ABR flow control protocol's feedback control mechanism may be ideal for transporting rate-adaptive video applications [12, 13, 18, 19]. Unlike bursty data traffic, video applications typically have much longer holding time and a feedback flow control mechanism can be very effective to control rate adaptive video. For example, a rate-adaptive video using multi-layer encoding has a high and a low priority stream. The high priority video stream can be supported by using the minimum cell rate (MCR) concept in ABR service, which can provide some minimum acceptable presentation quality. The low priority cell rate may be supported by the available bandwidth from the network since such low priority stream is intended to further enhance video quality. It has been shown in [12, 13, 19] that an ABR-like flow control mechanism is capable of providing sustainable bit rate for video with MCR guarantee and can further exploit any available bandwidth from the network.

A key performance issue associated with using such feedback control for video transmission is network bandwidth sharing among competing video connections. In particular, after guaranteeing each video connection its sustainable rate with MCR, how should the remaining network bandwidth be allocated among all video connections? Prior efforts such as [11, 14, 15, 18] did not address this issue. In [12, 13], an MCR-proportional max-min policy was proposed to allocate the remaining network bandwidth. But it was not clear what distributed feedback control algorithm should be employed to achieve such a network bandwidth sharing policy.

This paper addresses these issues by presenting a framework of network bandwidth sharing for rate-adaptive video using an ABR-like feedback control.

We first present a weight-based bandwidth sharing policy, also called *Weight-Proportional Max-Min (WPMM)* policy, to allocate network bandwidth among video connections. Unlike [12, 13] where the weight of a connection is set to its MCR, the weight associated with each connection in this paper is generic, *i.e.,* decoupled from (or independent of) its MCR. To achieve the WPMM policy in a fully distributed network, we employ an ABR-like mechanism and design a feedback control algorithm. Our algorithm is an extension of the *Consistent Marking* technique by Charny *et al.* in [2], which was designed to achieve the classical max-min rate allocation (without a minimum rate, peak rate, and weight for each connection). We show that our algorithm provides guaranteed convergence to the WPMM rate allocation policy for all video connections.

A unique feature associated with our feedback control algorithm is that it possesses the so-called *rate decoupling property.* That is, a source's actual transmission rate can be decoupled from the allowed cell rate (ACR) variable at the source. To take advantage of this property, we design a novel video source rate adaptation algorithm that provides a smooth (or 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 the potential available network bandwidth without undergoing the undesirable frequent fluctuations of feedback rate during transient periods.

Another contribution of this paper is that we have demonstrated the capabilities of on-line dynamic renegotiation of minimum rate (MCR) and weight assignment options in our feedback control algorithm. Such options are particularly important since the initial estimate of minimum rate requirement or weight may not accurately reflect the actual need of a particular video connec-

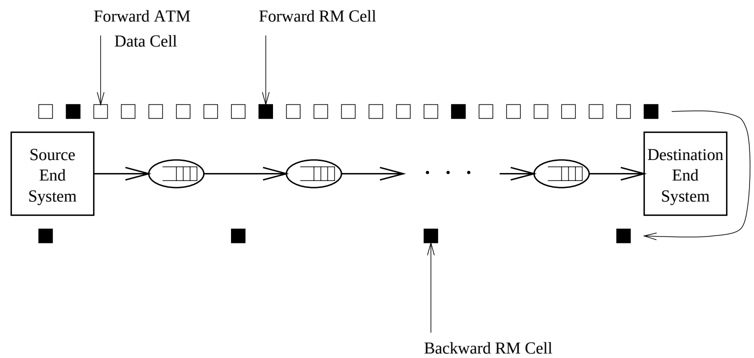

Figure 1: ABR feedback control mechanism.

tion. Without such renegotiation mechanisms, an accurate estimate of MCR is essential to support minimum video quality. We show that our feedback control algorithm can converge to a new rate vector for all video connections after some connection renegotiates its minimum rate or weight.

The remainder of this paper is organized as follows. Section 2 examines the ABR flow control mechanism and shows the attractive features of using such a flow control mechanism for transporting rate-adaptive video. Section 3 defines the weight-based bandwidth sharing policy. In Section 4, we present an ABR-like feedback control algorithm to achieve our network-wide bandwidth sharing policy. Section 5 shows the rate decoupling property in our feedback control algorithm and presents a video source rate adaptation algorithm based on such rate decoupling property. In Section 6, we demonstrate the on-line MCR renegotiation and weight adjustment capabilities of our feedback control algorithm. Section 7 concludes this paper and points out future research directions.

2 SUPPORTING RATE-ADAPTIVE VIDEO USING ABR-LIKE MECHANISM

The ABR service defined by the ATM Forum [1] supports applications that allow a source end system to adjust its information transfer rate based on the bandwidth availability in the network. By the specifications in [1], on the establishment of an ABR connection, the user shall specify to the network both a minimum required bandwidth and a maximum bandwidth, designated as minimum cell rate (MCR) and peak cell rate (PCR), respectively, for the requested connection. The source starts to transmit at in initial cell rate (ICR), which is greater than or equal to its MCR, and may adjust its rate based on the congestion and bandwidth information from the network. Although the available bandwidth from the network may vary, the minimum rate (MCR) for each connection is always guaranteed.

A generic ABR flow control mechanism for a connection is shown in Fig. 1. Despite the somewhat complex specifications 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 three key components:

1. Information exchange: Special control packets called Resource Management (RM) cells are inserted among the data cells to convey information between the sources and the network. The source sets the fields in the forward RM cells to inform the network about the source's rate information (e.g. minimum rate, peak rate, current rate). The returning RM cells carry the available bandwidth information of the network to the source.

2. Rate calculation: The network (switches) perform rate calculation based on the information carried in the traversing RM cells and set appropriate fields in the RM cells.

3. Source rate adaptation: A source is capable of adjusting its transmission rate based on the feedback information in the returning RM cells.

For the video sources considered in this paper, we assume that each source employs adaptive, multi-layered encoding combined with feedback-based rate control mechanism that 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. The feedback mechanism controls 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 rate 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 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 (QoS) for video. In particular, the MCR concept in ABR comes naturally to provide such CBR-like service to ensure minimum video transmission rate and presentation quality. With feedback, the video encoder can adjust its transmission rate by modulating the quantization level and adapt to any additional available bandwidth from the network through the explicit rate information in the returning RM cell.

3 A NETWORK BANDWIDTH SHARING POLICY

Since there are many video connections sharing a network, each trying to exploit additional available bandwidth, it is essential that we have some rate allocation policy in place. In this section, we show how a particular network bandwidth sharing policy can be used for video connections. This policy guarantees each video connection some minimum required rate, and at the same time can efficiently allocate the remaining network bandwidth based on each video connection's weight, so that each video's presentation quality may be enhanced and the network utilization can be increased.

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 in 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 (either imposed by the application or the network 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. Clearly, for feasibility, we must have

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

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 share such remaining network bandwidth among video connections. We let each connection $s \in \mathcal{S}$ be associated with a weight (or priority) w_s. Such weight is assigned at call set up time. The remaining network bandwidth is allocated to each connection by using a 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 describes how such 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 (*i.e.*, peak cell rate, PCR), whichever comes first.

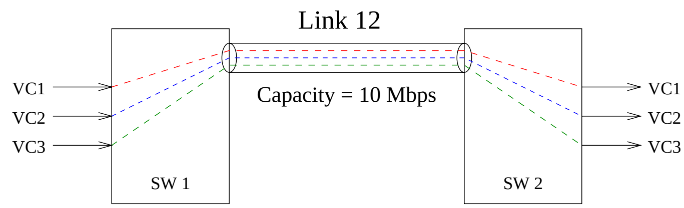

Figure 2: A peer-to-peer network.

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. □

Our network bandwidth sharing policy as characterized by Algorithm 1 extends the classical max-min policy with a minimum rate guarantee, peak rate constraint, and weight for each connection. Note that the weight of each connection is decoupled from (or independent of) its minimum rate, which adds considerably more flexibility than a MCR-proportional max-min policy used in [12]. To the best of our knowledge, the rate allocation policy introduced here, together with the MCR renegotiation and weight adjustment options presented in Section 6, offers the greatest flexibility in terms of network bandwidth sharing among all policies based on the classical max-min.

Our network bandwidth sharing policy may offer a pricing incentive for network service providers. More specifically, each connection may be charged a premium rate corresponding to the guaranteed bandwidth (*i.e.,* MCR). Beyond this rate, each connection is allowed to share any additional unguaranteed (or available) network bandwidth based on its weight (or priority). But the pricing issue associated with the weight of a connection is not straight forward and deserves further study.

We use the following simple example to illustrate how Algorithm 1 works to perform rate allocation.

Example 1 Peer-to-Peer Network

In this network (Fig. 2), there are three video connections at the input ports of switch 1 (SW1). An output port link of SW1 (Link12) is shared by these video connections and is the potential bottleneck link. We assume the capacity on Link12 for these video connections is 10 Mbps and that the minimum required rate, peak rate constraint, and weight for each connection are listed in Table 1. Table 2 shows the iterations of using Algorithm 1 to allocate network bandwidth for each video connection, which are explained as follows.

Table 1: 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

Table 2: Iterations to allocate rate for each connection under WPMM in the peer-to-peer network.

Iterations	Session{(MCR, PCR)(in Mbps), W}			Remaining Capacity
	VC1{(1.5, 10.0), 1}	VC2{(1.0, 3.0), 1}	VC3{(0.5, 5.0), 1}	Link 12
initialization	1.5	1.0	0.5	7.0
1st	3.5	3.0	2.5	1.0
2nd	4.0		3.0	0

- *Initialization:* We start the rate of each connection with its minimum rate requirement (MCR), *i.e.,* 1.5 Mbps, 1.0 Mbps, and 0.5 Mbps for VC1, VC2, and VC3, respectively. Since the capacity of Link12 is 10 Mbps, the remaining capacity of Link12 is then $10 - (1.5 + 1.0 + 0.5) = 7.0$ Mbps.

- *1st iteration:* We increase the rate of each connection with an increment proportional to its weight (equal weight for all connections in this simple example) until s_2 reaches its peak rate constraint of 3.0 Mbps. At this point, we have 3.5 Mbps, 3.0 Mbps, and 2.5 Mbps for VC1, VC2, and VC3, respectively, with a remaining capacity of 1.0 Mbps on Link12. Since the rate of VC2 has reached its peak rate constraint and cannot be increased further during future iterations, it is allocated this rate (3.0 Mbps) and is removed from future iterations.

- *2nd iteration:* Further increase the rates of the remaining connections VC1 and VC3 with an increment proportional to each connection's weight until there is no remaining capacity on Link12, *i.e.,* Link12 is saturated. The final rates for connections VC1, VC2, and VC3 are 4.0 Mbps, 3.0 Mbps, and 3.0 Mbps, respectively. □

The following example shows how Algorithm 1 works in a multi-node network.

Example 2 A Three Node Network

In this network (Fig. 3), there are four video connections and the output port links of SW1 (Link 12) and SW2 (Link 23) are potential bottleneck links for these connections. Assume that the capacity of Link12 and Link23 are both 10 Mbps, respectively, and the minimum rate requirement,

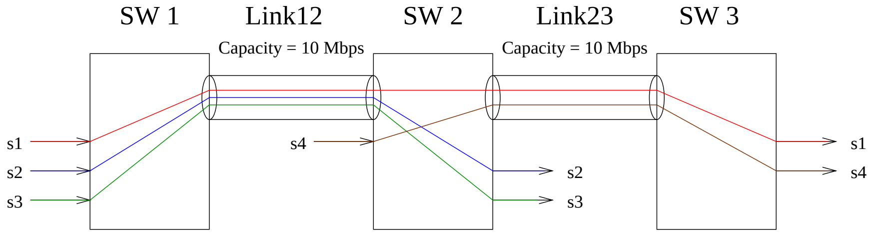

Figure 3: A three node network configuration.

Table 3: 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	0.5	1.5
VC2	1.5	9.0	1.5	4.5
VC3	2.0	4.0	2.0	4.0
VC4	1.0	10.0	1.0	8.5

peak rate constraint, and weight for each connection are listed in Table 3. Table 4 shows the iterations of using Algorithm 1 to allocate rate for each connection. □

We emphasize that the weight assignment of each connection can be arbitrary and the final rate allocation vector under Algorithm 1 is unique.

Also shown in the above examples is that the weight proportional rule is used only during the intermediate steps in Algorithm 1 and that the final rate allocated to each connection, after offsetting by its minimum rate, may not necessarily be uniformly proportional to its weight for all connections. In particular, a connection 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 a 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 a connection's weight is unguaranteed since they

Table 4: Iterations of rate allocation for each connection under WPMM in the three node network.

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

may be taken in the future by a newly joined video connection requiring some guaranteed minimum rate.

4 A FEEDBACK CONTROL ALGORITHM FOR VIDEO

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.

4.1 The Algorithm

Our feedback control algorithm includes a protocol for the end systems (source and destination) and an algorithm for the switches in the network. A source end system has the following parameters: Allowed Cell Rate (ACR), Initial Cell Rate (ICR), MCR, PCR, and Weight (W). Similarly, the following fields are used in an RM cell to exchange information between a source and the network: Current Cell Rate (CCR), which is set to the ACR at the source, MCR, Explicit Rate (ER), which is initially set to the PCR of the source and is adjusted or reduced by the switches along its traversing path, and Weight (W).

We first specify the behaviors of each connection's source and destination in our algorithm.

Algorithm 2 End System Behavior

Source Behavior[1]

The source starts to transmit at ACR := ICR, which is greater than or equal to its 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 simply returns every RM cell back towards the source upon receiving it. □

Note that since RM cells are periodically transmitted, once every N_{rm} data cells (e.g. 32), the overhead for carrying such flow control is, therefore, bounded with a fixed marginal percentage of network capacity.

Now we present the switch algorithm used in the network, which calculates the rate allocation for each connection. At each output port of a switch, we maintain a table and keep track of the state information of each traversing connection (also called per flow accounting). Specifically, for each forward RM cell at an output port of a link, the switch records the CCR, MCR, and W for such connection and performs rate calculation; for each backward RM cell, the switch updates its ER field with the calculated rate (see Algorithm 4). The following are some additional link parameters and variables used in the 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 connections $i \in \mathcal{S}_\ell$ at link ℓ.

b_ℓ^i: Bit used to mark connection $i \in \mathcal{S}_\ell$ at link ℓ.

$$b_\ell^i = \begin{cases} 1 & \text{if connection } i \in \mathcal{S}_\ell \text{ is } \textit{marked} \text{ at link } \ell; \\ 0 & \text{if connection } i \in \mathcal{S}_\ell \text{ is } \textit{unmarked} \text{ at link } \ell. \end{cases}$$

$\mathcal{M}_\ell$: 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 $\mathcal{M}_\ell \cup \mathcal{U}_\ell = \mathcal{S}_\ell$.

φ_ℓ: An auxiliary variable at link ℓ used to facilitate rate calculation, and is calculated as follows.

Algorithm 3 φ_ℓ **Calculation**

$$\varphi_\ell := \begin{cases} \infty & \text{if } n_\ell = 0;^2 \\ \frac{C_\ell - \sum_{i\in\mathcal{S}_\ell} r_\ell^i}{\sum_{i\in\mathcal{S}_\ell} w_i} + \max_{i\in\mathcal{S}_\ell} \frac{r_\ell^i - \mathrm{MCR}^i}{w_i} & \text{if } |\mathcal{M}_\ell| = n_\ell; \\ \frac{(C_\ell - \sum_{i\in\mathcal{S}_\ell} \mathrm{MCR}^i) - \sum_{i\in\mathcal{M}_\ell}(r_\ell^i - \mathrm{MCR}^i)}{\sum_{i\in\mathcal{U}_\ell} w_i} & \text{otherwise } (\textit{i.e.,}\ |\mathcal{M}_\ell| \neq n_\ell). \end{cases}$$

□

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

We give some intuition for the auxiliary variable φ_ℓ in Algorithm 3 under the special case when both $\mathrm{MCR}^i = 0$ and $w_i = 1$ for all $i \in \mathcal{S}$, *i.e.*, the classical max-min case. In this special case, the last expression becomes $\varphi_\ell := \frac{C_\ell - \sum_{i \in \mathcal{M}_\ell} r_\ell^i}{|\mathcal{U}_\ell|}$ when not all connections are marked. This is precisely the expression commonly used to calculate max-min rate allocation. The second expression for φ_ℓ shows what happens when all connections are marked, which would be the case when the distributed algorithm converges. In this case, $C_\ell = \sum_{i \in \mathcal{S}_\ell} r_\ell^i$ at a saturated link where all connections are marked and the second expression simply becomes $\varphi_\ell := \max_{i \in \mathcal{S}_\ell} r_\ell^i$, *i.e.*, the max-min bottleneck link. Such simple special case for max-min was done in [2]. By taking into account of the weight and minimum rate of each connection, our construction of φ_ℓ calculation in Algorithm 3 generalizes that in [2].

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

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[3]{
        $\mathcal{S}_\ell := \mathcal{S}_\ell - \{i\}$; /* Remove the terminating connection from the table. */
        $n_\ell := n_\ell - 1$;
        table_update();
        }

    if RM cell signals connection initiation {
        $\mathcal{S}_\ell := \mathcal{S}_\ell \cup \{i\}$; /* Add the newly initiated connection to the table. */
        $n_\ell := n_\ell + 1$;
        $r_\ell^i :=$ CCR; $\mathrm{MCR}^i :=$ MCR; $w_i :=$ W; $b_\ell^i := 0$;
        table_update();
        }

    else /* *i.e.,* RM cell belongs to an ongoing active connection. */ {
        $r_\ell^i :=$ CCR; $\mathrm{MCR}^i :=$ MCR; $w_i :=$ W;
        if ($\frac{r_\ell^i - \mathrm{MCR}^i}{w_i} \leq \varphi_\ell$) then $b_\ell^i := 1$;
        table_update();
        }

    Forward RM(CCR, MCR, ER, W) towards its destination;
}

[2]In fact, φ_ℓ can be set to any value when $n_\ell = 0$.

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

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

table_update()
{
  rate_calculation_1: use Algorithm 3 to calculate φ_ℓ^1;

  Unmark any marked connection $i \in \mathcal{S}_\ell$ at link ℓ with $\frac{r_\ell^i - \text{MCR}^i}{w_i} > \varphi_\ell^1$;

  /* Update φ_ℓ after the above unmarking operation. */
  rate_calculation_2: use Algorithm 3 to calculate φ_ℓ;

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

As shown in the above end systems (source and destination) 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 each switch along its traversing path about the connection's current rate. The variable φ_ℓ at link $\ell \in \mathcal{L}$ serves the roles of estimating 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 "non-conforming" connections (denoted by set $\mathcal{U}_\ell$ at link ℓ) if their last seen CCR satisfies $\frac{\text{CCR}-\text{MCR}}{\text{W}} > \varphi_\ell$. Similarly, connections with $\frac{\text{CCR}-\text{MCR}}{\text{W}} \leq \varphi_\ell$ are called "conforming" connections (denoted by set $\mathcal{M}_\ell$ at link ℓ) and are therefore marked (b bit set to 1). The connections in the conforming set are assumed to converge to our rate allocation while connection in the non-conforming set are those that are still under transient iterations. During the iteration process, after each time φ_ℓ is updated, a connection previously belonging to $\mathcal{M}_\ell$ may be unmarked and become a connection in $\mathcal{U}_\ell$ (see table_update subroutine).

Theorem 1 After the number of video connections in the network stabilizes, the rate allocation for each connection by the distributed feedback control algorithm converges to the WPMM rate allocation. □

We refer interested readers to the appendix for a sketch of the proof.

4.2 Simulation Results

Theorem 1 gives us a theoretical guarantee that our feedback-based flow control algorithm converges to our bandwidth sharing policy under any network configuration and any set of link distances. In this sub-section, we use simulation results to show the convergence property of this feedback control 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 from 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 that the internal switching delay for a cell from an input port to an output port is 4 μs (not including the queuing delay at the output port). 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.,* $C_\ell = \frac{10}{0.95} \times 0.95 = 10$ Mbps. By setting a target link utilization strictly less than 1, we ensure that the potential packets build up at a bottleneck link during transient period will be emptied upon algorithm's 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 that 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.

The Peer-to-Peer Network

For this network (Fig. 2), 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 1.

Figure 4 shows the ACR behavior in our simulation run for VC1, VC2, and VC3, respectively. Each connection starts with 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 its respective rate listed in Table 1. 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.

Note that during the transient period, packets may build up at the bottleneck link (see Fig. 4 during 10 ms $<$ t $<$ 25 ms where $(5.0 + 4.0 + 3.0) = 12.0 > 10.526$ Mbps), we assume that adequate

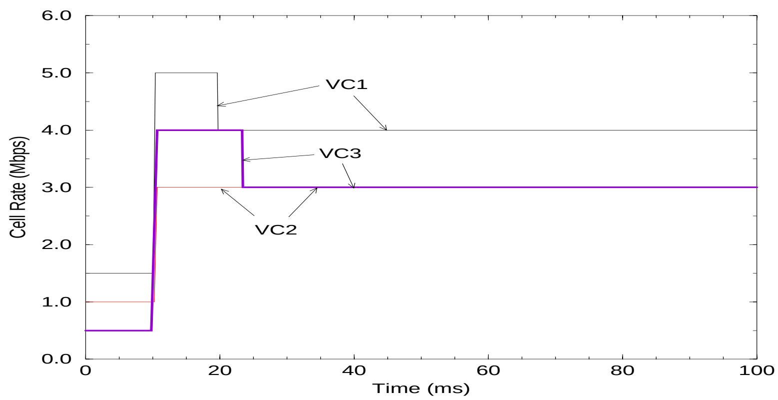

Figure 4: The ACR of all connections for the peer-to-peer network configuration.

buffer space has been reserved at each traversing link at call set up time. For example, we may use the per flow accounting based buffer management mechanism in [6] to reserve adequate buffer space for each flow.

The Three Node Network

For this network (Fig. 3), 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 3.

Figure 5 shows the ACR of each connection in our simulation run. 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 the respective rate listed in Table 3.

A Parking Lot Network

Figure 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 5 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 in our simulation run. Again, each connection starts to transmit at MCR and converges to the respective rate listed in Table 5.

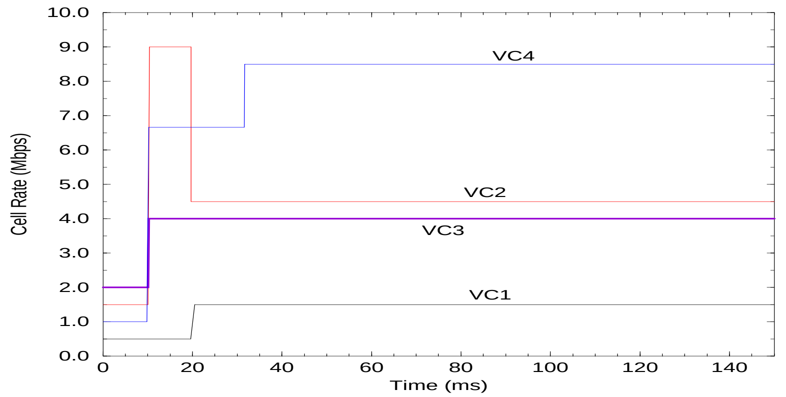

Figure 5: The ACR of all connections for the three node network configuration.

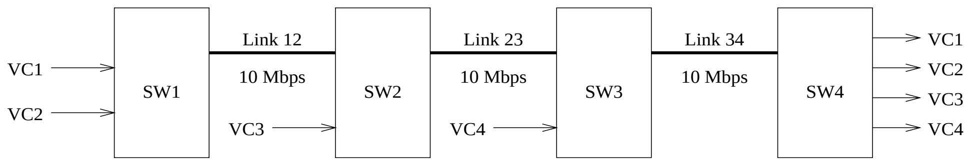

Figure 6: A parking lot network.

Table 5: 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

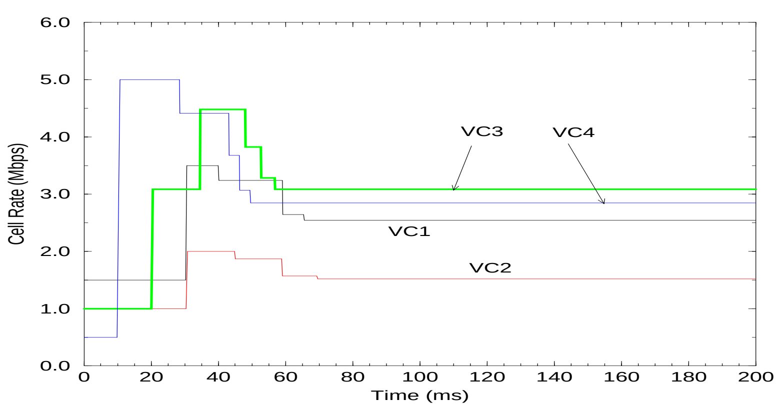

Figure 7: The ACR of all connections for the parking lot network configuration.

4.3 Minimum Rate Guarantee with Pushout Mechanism

Under our framework, a video connection is admitted into the network if it satisfies Eq. (1). That is, only when the addition of the MCR of the new connection to the existing MCRs of other connections in the network do not exceed link capacity along every link the new connection traverses will we admit the new video connection.

According to Theorem 1 and as shown in the above simulation results, our feedback flow control algorithm is capable of converging to the WPMM rate allocation as long as the number of video connections in the network stabilizes for a period of time. However, during transient convergence period, such as when a new connection is admitted into the network, the output buffer at a node may build up and overflow if buffer space has not been provisioned adequately. Under such buffer overflow scenario, the minimum rate of a connection may not be always guaranteed due to cell loss. To provide a minimum rate (MCR) guarantee to each connection at all time at each node, we introduce the following cell marking and buffer management mechanism in our framework.

We define two types of cells within each video connection; the Minimum Rate (MR) cells and the Additional Rate (AR) cells. The MR cells are supported by MCR and should have guaranteed delivery at all time. The AR cells carry traffic in excess of MCR and share any remaining available network bandwidth with other connections. We assume that each source end system is capable of marking its output cell stream into MR cells and AR cells. Upon the convergence of our flow control algorithm, the sum of MR and AR cells for each connection equal to the WPMM rate allocation for such connection. However, during transient convergence period, the sum of MR and AR may differ from WPMM rate allocation for a connection and buffer may build up and overflow.

We employ the so-called pushout cell discarding mechanism to guarantee the MCR of a video connection [3, 17] during transient congestion period. When a MR cell of a video connection arrives at a node and the buffer is full, an AR cell is discarded (pushed out) in the buffer to leave room for the incoming MR cell. Therefore, as far as MR cells are concerned, they have exclusive access of the output port buffer during congestion and are never starved out of buffer space because of AR cells. Since the sum of MCRs at a node is always less than the link capacity under our admission control (Eq. (1)), an incoming MR cell is always guaranteed to be served at a node with the pushout mechanism. Thus, the MCR of each connection is guaranteed at all time at each node, including transient convergence period.

5 VIDEO SOURCE RATE ADAPTATION ALGORITHM

Our feedback control algorithm based on ABR mechanism in the last section achieves our rate allocation policy through distributed and asynchronous iterations. The minimum rate of each video connection is always guaranteed throughout a connection's lifetime and any excess network bandwidth is shared among video connections according to the weight of each connection under the WPMM policy. Note that our distributed feedback control allows the joining of a new connection into the network and the termination of an existing connection. Based on the current set of connections in the network, our feedback control algorithm is always in the process of iterations with the aim of converging to our weight-based rate allocation. It can be shown that the convergence time is upper bounded by $2.5KD$, where K is the number of bottleneck rates in the network and D is the maximum round trip time among all connections [8].

A problem associated with an ABR-like feedback control algorithm is that during the transient 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 source behavior in Algorithm 2), the ACR variable at a source is also continually changing. For example, the simulation results in Figs. 4, 5, and particularly in Fig. 7 show that the ACR variable of a connection keeps undergoing fluctuations during the transient period. Such transient fluctuation of ACR is undesirable since a video encoder's quantization adjustment period may not be able to follow such frequent variations. Also, the rapid fluctuations of the video coding rate may adversely impact the video quality.

To alleviate the above frequent rate adaptation problem, a sophisticated source rate adaptation algorithm was introduced in [12]. It predicts the bit rate based on a smoothing average of ACR at a source and the encoder's buffer level. Its implementation is quite complex since the algorithm requires keeping track of the history of ACR (for smoothing average) and constant monitoring of encoder buffer content. The accuracy of such algorithm depends on careful system parameters tuning and there is no guarantee that it will offer consistent performance.

In this section, we present a novel video source rate adaptation algorithm that decouples a source's actual transmission rate from its ACR variable. It is both simple to implement and effective to adapt to the final rate allocation objective. Our video source rate adaptation algorithm is based on the following unique property in 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 a variable solely used for the purpose of distributed iterations for protocol convergence and a source's true transmission rate does not affect the final rate allocation. That is, a source's true transmission rate does not have to be identical to its ACR. 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 variable of each connection will converge to our rate allocation). We give a formal definition of this property as follows.

Since the ER calculation is highly dependent on the fields of an RM cell, it is essential to maintain the correctness of the fields of each RM cell in order for the flow control algorithm to converge to the desired rate allocation. The validity of these fields of an RM cell should be checked at network access point and appropriate error control mechanism should be in place inside the network. These issues are beyond the scope of this paper and we leave them for future study.

Definition 1 We say an ABR-like feedback control algorithm has the *rate decoupling* property if the actual transmission rate of a source can be decoupled from the ACR variable at the source without affecting the final convergence of ACR for rate allocation of all connections. □

We stress that such rate decoupling property is a consequence of our special design of switch algorithm where a table is used to keep track of the state information of each traversing connections, and the fact that the level of congestion status (e.g. buffer occupancy, load) does not play any role in the ER calculation. Feedback control algorithms such as [9, 10, 12, 16] are unable to offer such rate decoupling property since congestion status (e.g. buffer occupancy, load) is used in ER calculation and such congestion status is determined by the source's actual transmission rate.

Property 1 Our ABR-like feedback control algorithm as specified in Algorithm 4 has the rate decoupling property. □

Based on the unique rate decoupling property in our feedback control algorithm, we propose the following simple source rate adaptation algorithm for each video connection. 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 feedback control protocol. As before, a source keeps updating its ACR upon receiving each returning RM cell, but the adjustment of the actual transmission rate (*i.e.,* TCR) is only performed at a time interval, say I, which can be determined by each source encoder's physical property. One implementation to achieve such decoupling is to maintain a local timer at the source as a time reference and use a variable which we call *Rate Adjustment Time (RAT)* as an indication for the next time point that a source should adjust its true transmission rate. The details of such source rate adaptation algorithm is described as follows.

Algorithm 5 Video Source Rate Adaptation

Initially, the source starts to transmit at ACR := ICR with ICR $\geq$ 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; /* Adjust video encoding rate. */
        RAT := RAT + I;
        }
```

□

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

It should be clear that by using such source rate adaptation algorithm (instead of the source algorithm in Algorithm 2) and the switch algorithm (Algorithm 4), our feedback control algorithm can still converge to the final weight-based rate allocation. Furthermore, the upper bound of $2.5KD$ for the convergence time still applies [8].

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

Figure 8 shows the source's true transmission rate (*i.e.,* TCR) of each connection for the peer-to-peer network (see Fig. 2 and Table 1) under the new source rate adaptation algorithm. The source rate adjustment intervals are 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 are allowed to adjust their transmission rate every 3 frames while source 3 is allowed to adjust its rate every 4 frames. The simulation for ACR variable of each connection is identical to those shown in Fig. 4. Comparing

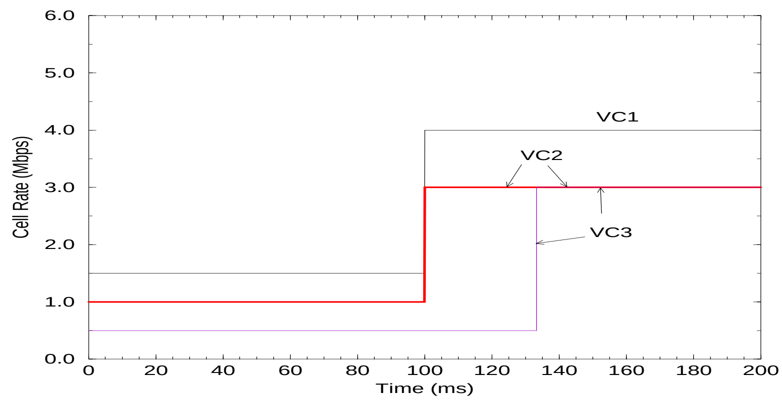

Figure 8: The TCR of all video connections for the peer-to-peer network configuration.

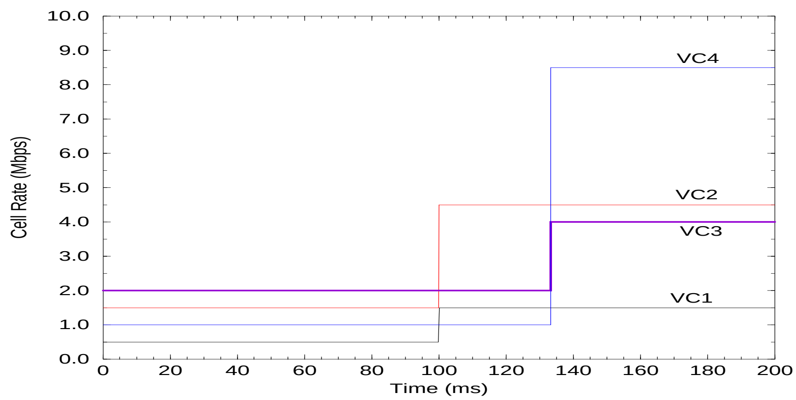

Figure 9: The TCR of all video connections for the three node network configuration.

Fig. 8 to Fig. 4, we find that our video source rate adaptation algorithm effectively shields a source's rate adaptation from the undesirable ACR variable fluctuations during transient periods.

Figure 9 shows the TCR of each connection for the three node network (see Fig. 3 and Table 3). 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 simulation run for ACR of each connection is identical to that 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 rate share without undergoing frequent ACR fluctuations during transient periods.

Figure 10 shows the simulation results of TCR for each connection in the parking lot network (see Fig. 6 and Table 5). 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 simulation run for ACR variables of all connections are identical to those shown in Fig. 7. Again, under our new source rate adaptation algorithm, the transmission rate of each

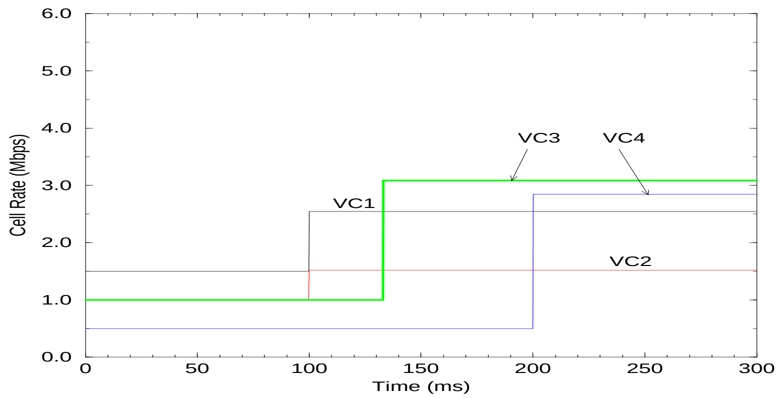

Figure 10: The TCR of all video connections for the parking lot network configuration.

connection in Fig. 10 adapts smoothly to its final rate share without undergoing the undesirable rate fluctuations in Fig. 7.

Remark 1 In Algorithm 5, a source's rate adaptation interval I is assumed to be a constant. 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 determined by the source. One advantage of using a variable time interval for rate adjustment is that we can adjust an 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 during transient period.[4] □

By employing the source rate adaptation mechanism (Algorithm 5), our feedback control protocol lets each video source operate in a piece-wise CBR-like mode (with infrequent quantization changes). The benefits of our scheme include: 1) The network is efficiently utilized and the quality of each video is further improved if there is available bandwidth from the network; and 2) The quality of video degrades gracefully (still above minimum presentation quality) when the network is congested.

6 DYNAMIC MCR RENEGOTIATION AND WEIGHT ADJUSTMENT

In our feedback control algorithm, each connection relies on MCR guarantee to support some minimum video quality and a weight to share any excess network bandwidth beyond its minimum

[4]Note that even during transient convergence period where congestion may occur, the minimum rate (MCR) of each video connection is always guaranteed due to the pushout mechanism at each output port (see Section 4.3). Therefore, packet loss may only occur to low priority (AR) packets.

Table 6: 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

rate. We have been implicitly assuming that each connection has a prior knowledge of the minimum rate requirement. 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. It is therefore desirable to offer a user the option of re-negotiating its MCR or adjusting its weight should the user feel necessary. This section demonstrates such capabilities in our feedback control algorithm.

6.1 MCR Renegotiation

Our feedback control algorithm is capable of providing the MCR renegotiation option. The only criterion that needs to be checked is that the sum of the new MCRs cannot exceed the link's capacity on any link in the network (see Eq. (1)). If Eq. (1) can be satisfied, 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 Algorithm 1. Theorem 1 guarantees that our distributed feedback control algorithm is able to reiterate and converge to this new rate allocation for all connections.

As an example, for the peer-to-peer network shown in Fig. 2, Table 1 shows the minimum rate, peak rate, weight, and rate allocation for each connection. Our feedback control algorithm is shown to converge to the optimal rate allocation in Table 1 for each connection (see Fig. 4). In Table 6, we let the minimum rate of VC3 change from 0.5 Mbps to 3.0 Mbps and show the rate allocation for each connection before and after such change. The simulation results of our feedback control algorithm before VC3's MCR change were 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 respective new rates of 3.0 Mbps and 2.5 Mbps (since $I_1 = I_2 = 100$ms); and at $t = 433.33$ms, VC3 adapts to its new rate of 4.5 Mbps (since $I_3 = 133.33$ms). The new rate allocation for each connection by our distributed feedback control algorithm match the respective rate allocation listed in Table 6.

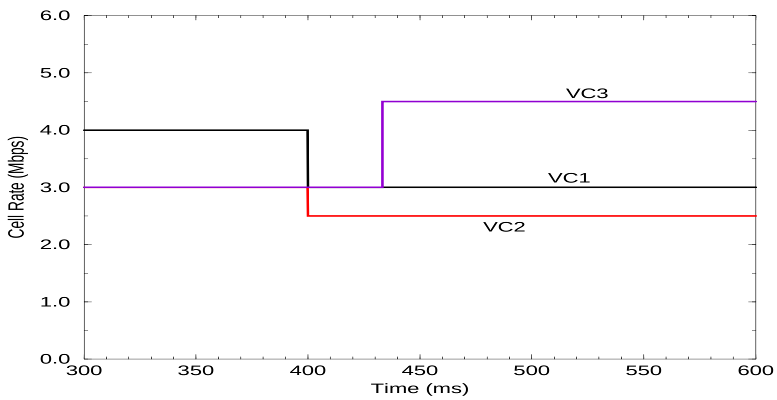

Figure 11: The TCRs of all video connections for the peer-to-peer network configuration. VC3 has renegotiated a new MCR requirement.

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 new rate allocation for all connections will change under Algorithm 1. Again, our feedback control algorithm is able to re-iterate and converge to the new rate vector.

As an example, for the three node network (Fig. 3), with minimum rate and peak rate for each connection being the same as those listed in Table 3, Table 7 shows the rate allocation (under Algorithm 1) for each connection before and after the weight of VC1 is adjusted from 0.5 to 4.0. In Fig. 12, we continue 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. At time $t = 400$ ms, the rates for VC1 and VC2 adapt to their new respective rates of 3.7 Mbps and 2.7 Mbps (since $I_1 = I_2 = 100$ ms); and at $t = 433.33$ ms, the rates of VC3 and VC4 also adapt to their respective rates of 3.6 Mbps and 6.3 Mbps (since $I_3 = I_4 = 133.33$ ms). Comparing with those rates listed in Table 7, we have demonstrated that our feedback control algorithm is capable of converging to the new rate allocation when the weight of a connection is changed to a new value.

Table 7: 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	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

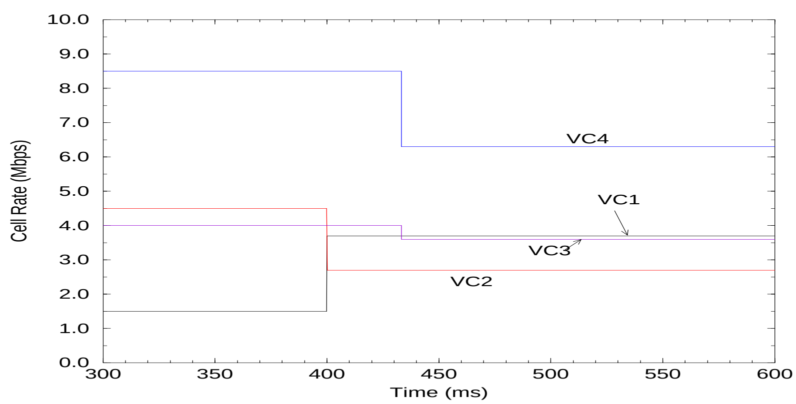

Figure 12: The TCRs of all video connections for the three node network configuration. VC1 changed its weight from 0.5 to 4.0.

7 CONCLUDING REMARKS

The ABR flow control mechanism offers attractive features for transporting rate-adaptive video. Such feedback control maintains the simplicity of admission control for CBR with minimum rate guarantee and at the same time, exploits any available network bandwidth through feedback. This paper sets up a framework for network bandwidth sharing among video connections using an ABR-like feedback control. The main contributions in this paper are listed as follows.

- We introduced a weight-based rate allocation for video application. This rate allocation supports the minimum rate requirement and peak rate constraint of each connection and associate each connection with a weight to share any excessive network bandwidth. To the best of our knowledge, such weight-based rate allocation policy, combined with the minimum rate renegotiation and weight adjustment options, offers the greatest flexibility in terms of bandwidth sharing among all rate allocation policies based on the classical max-min.
- We presented a feedback control algorithm using an ABR-like mechanism. This algorithm specifies the behavior at each switch as well as at the source and destination of each connection. Our feedback control algorithm was shown to provide guaranteed convergence to our rate allocation policy. Furthermore, by incorporating the pushout mechanism at each node, the MCR of each connection is guaranteed at all time, including transient convergence period.
- Our feedback control algorithm possesses the unique property that a source's actual transmission rate can be decoupled from the ACR variable used for protocol convergence. We stress that such rate decoupling property is a consequence of our special design of switch algorithm where a table is used to keep track of the state information of each traversing connections (per flow accounting), as well as the fact that the level of congestion status (e.g. buffer occupancy, load) does not play any role in the ER calculation. Other ABR algorithms that rely on congestion status (e.g. buffer occupancy, load) in ER calculation are unable to offer such rate decoupling property.
- The rate decoupling property in our switch algorithm enabled us to design a novel source rate adaptation algorithm to avoid the undesirable rate fluctuations during transient periods. We demonstrated that with the new source rate adaptation algorithm, our overall feedback control algorithm converges smoothly to the final rate allocation without frequent fluctuations during transient periods.
- We demonstrated simple MCR renegotiation and weight adjustment options for each video connection and showed that our feedback control algorithm is capable of supporting such

options. The MCR renegotiation maintains the simplicity of connection management for guaranteed minimum rate and encompasses the Re-negotiated CBR (RCBR) feature proposed in [5], while the weight adjustment option offers further flexibility in sharing any excess network bandwidth beyond a connection's minimum rate. We believe that such dynamic on-line renegotiation options offer useful flexibility in bandwidth allocation for video applications, whose durations are typically on the order of tens of minutes.

Our future work will focus on other issues in our feedback control algorithm. One challenging issue for us is to reduce the storage and computational complexity of our algorithm (currently $O(N)$) and yet retain the guaranteed convergence property. Another issue is to study the pricing policy associated with the weight assignment and adjustment in our rate allocation algorithm. Our work in this paper has demonstrated the technical capability of offering such flexible weight adjustment option, but the pricing issue from such an option deserves further investigation.

ACKNOWLEDGMENTS

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

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] ATM Forum Technical Committee, "Traffic management specification, version 4.0," *ATM Forum Contribution, AF-TM 96-0056.00,* April 1996.

[2] A. Charny, D. Clark and R. Jain, "Congestion control with explicit rate indication," *Proc. IEEE International Conference on Communications,* pp. 1954–1963, June 1995.

[3] I. Cidon, L. Georgiadis, R. Guérin and A. Khamisy, "Optimal buffer sharing," *IEEE Journal on Selected Areas in Communications,* **13**, (7), 1229–1240, (1995).

[4] 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 Journal on Selected Areas in Communications,* **13**, (6), 1004–1016, (1995).

[5] M. Grossglauser, S. Keshav and D. Tse, "RCBR: a simple and efficient service for multiple time-scale traffic," *IEEE/ACM Trans. on Networking,* **5**, (6), 741–755, (1997).

[6] R. Guérin, S. Kamat, V. Peris and R. Rajan, "Scalable QoS provision through buffer management," *Proc. ACM SIGCOMM'98,* Sept. 1998.

[7] D. Heyman, A. Tabatabai and T. V. Lakshman, "Statistical analysis and simulation study of VBR video teleconference traffic in ATM networks," *IEEE Trans. on Circuits and Systems for Video Technology,* **2**, (1), 49–59, (1992).

[8] 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, 1997.

[9] R. Jain, *et al.*, "ERICA switch algorithm: a complete description," *ATM Forum Contribution, AF-TM 96-1172,* August 1996.

[10] 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. IFIP International Conference on High Performance Networking,* pp. 143–154, Sept. 1995.

[11] H. Kanakia, P. P. Mishra and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport," *IEEE/ACM Trans. on Networking,* **3**, (6), 671–682, (1995).

[12] 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, April 1997.

[13] P. P. Mishra, "Fair bandwidth sharing for video traffic sources using distributed feedback control," *Proc. IEEE Global Telecommunications Conference,* pp. 1102–1108, Nov. 1995.

[14] Y. Omori, T. Suda, G. Lin and Y. Kosugi, "Feedback-based congestion control for VBR video in ATM networks," *Proc. IEEE International Workshop on Packet Video,* Sept. 1994.

[15] C. M. Sharon, M. Devetsikiotis, I. Lambadaris and A. R. Kaye, "Rate control of VBR H.261 video on frame relay networks," *Proc. IEEE International Conference on Communications,* pp. 1443–1447, June 1995.

[16] K.-Y. Siu and H.-Y. Tzeng, "Intelligent congestion control for ABR service in ATM networks," *ACM SIGCOMM Computer Communication Review,* **24**, (5), 81–106, (1994).

[17] L. Tassiulas, Y. C. Hung and S. S. Panwar, "Optimal buffer control during congestion in an ATM network node," *IEEE/ACM Trans. on Networking,* **2**, (4), 374–386, (1994).

[18] B. J. Vickers and T. Suda, "An ATM service architecture for the transport of adaptive encoded live video," *Proc. International Conference on Computer Communications and Networks,* October 1996.

[19] B. J. Vickers, M. Lee and T. Suda, "Feedback control mechanisms for real-time multipoint video services," *IEEE Journal on Selected Areas in Communications,* **15**, (3), 512–530, (1997).

APPENDIX: CONVERGENCE OF FEEDBACK CONTROL ALGORITHM

The key concept used in the convergence proof of our distributed algorithm is the notion of *marking consistent*, which is defined as follows.

Definition 2 Let $\mathcal{M}_\ell$ be the set of connections that are marked at link $\ell \in \mathcal{L}$ and μ_ℓ be calculated according to Algorithm 3. The marking of connections at link $\ell \in \mathcal{L}$ is *marking-consistent* if

$$\frac{r_\ell^i - \text{MCR}^i}{w_i} \leq \mu_\ell$$

for every connection $i \in \mathcal{M}_\ell$. □

It can be shown that by using the three-step rate calculation for μ_ℓ in the "table_update()" subroutine of Algorithm 4, the marking of all connections at a link satisfies the marking-consistent property after the switch algorithm is performed for each RM cell traversing this link [8].

Denote M the total number of iterations needed to execute Algorithm 1. It can be shown that $M \leq |\mathcal{S}|$, where $|\mathcal{S}|$ is the total number of connections in the network [8]. Let $\mathcal{S}_i$, $1 \leq i \leq M$ be the set of connections being removed at the end of the ith iteration, i.e. connections in $\mathcal{S}_i$ have either reached their WPMM-bottleneck link rate or their PCRs during the ith iteration of Algorithm 1. Let τ_i, $1 \leq i \leq M$ be defined as follows:

$$\tau_i = \frac{r^s - \text{MCR}^s}{w_s} \quad \text{for every } s \in \mathcal{S}_i, \;\; 1 \leq i \leq M,$$

where r^s is the final WPMM rate allocation for connection s by Algorithm 1. By the operation of Algorithm 1, for a connection $p \in \mathcal{S}$ which has not yet gone through a saturated link or reached its PCR, its $\frac{r^p - \text{MCR}^p}{w_p}$ increases at each iteration. Therefore, we have $\tau_1 < \tau_2 < \ldots < \tau_M$.

It can be shown that after some finite time T_1, the set of connections in $s \in \mathcal{S}_1$ will either reach their WPMM-bottleneck link rate or their PCR constraints. These connections will be allocated with their optimal rates permanently and are marked at every link they traverse. By the operation of our rate calculation in the switch algorithm, such marked connections (as well as their associated bandwidth) can be used as the base case of an induction argument for the convergence of the second level WPMM rate allocation (i.e. $s \in \mathcal{S}_2$). Using the same token (i.e. induction), it can be shown that eventually all connections in the network will reach their WPMM rate allocation and will be marked at every link they traverses [8]. For a complete formal proof, we refer interested readers to [8].

Authors' biographies

Yiwei Thomas Hou obtained his B.E. degree (*Summa Cum Laude*) from the City College of New York in 1991, the M.S. degree from Columbia University in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, New York, in 1997, all in Electrical Engineering. He was awarded a National Science Foundation Graduate Research Traineeship for pursuing Ph.D. degree in high speed networks, and was recipient of Alexander Hessel award for outstanding Ph.D. dissertation in 1998 from Polytechnic University.

While a graduate student, he worked at AT&T Bell Labs, Murray Hill, New Jersey, during the summers of 1994 and 1995, on implementations of IP and ATM internetworking; he also worked at Bell Labs, Lucent Technologies, Holmdel, New Jersey, during the summer of 1996, on network traffic management. Since September 1997, Dr. Hou has been a Research Staff Member at Fujitsu Laboratories of America, Sunnyvale, California. His current research interests are in the areas of next generation Internet architecture, protocols, and implementations for differentiated and integrated services. Dr. Hou is a member of the IEEE, ACM, and Sigma Xi.

Shivendra S. Panwar received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kanpur, in 1981, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Massachusetts, Amherst, in 1983 and 1986, respectively. He joined the Department of Electrical Engineering at the Polytechnic Institute of New York, Brooklyn (now Polytechnic University), where he is now an Associate Professor. Since 1996, he has served as the Director of the New York State Center for Advanced Technology in Telecommunications (CATT).

Dr. Panwar has worked with NYNEX Science and Technology, Fujitsu Network Communications, Sumitomo Electric Industries, Lucent Technologies, Chase Manhattan Bank, Securities Industries Automation Corporation, Frequency Electronics, LNR, Inc., and the US Department of State, on a variety of projects. His research interests include the performance analysis and design of high speed networks, which has been supported by the National Science Foundation. Dr. Panwar is a member of Tau Beta Pi and Sigma Xi and a Senior Member of the IEEE.

Zhi-Li Zhang received the B.S. degree in Computer Science from Nanjing University, China, in 1986 and his M.S. and Ph.D. degrees in Computer Science from the University of Massachusetts, Amherst, in 1992 and 1997, respectively.

In 1997 he joined the Computer Science and Engineering faculty at the University of Minnesota, where he is currently an Assistant Professor. From 1987 to 1990, he conducted research in Computer Science Department at Århus University, Denmark, under a fellowship from the Chinese

National Committee for Education. Dr. Zhang's research interests include computer communication and networks, especially the QoS guarantee issues in high-speed networks, multimedia and real-time systems, performance evaluation, queueing theory, applied probability theory, theory of computation and algorithms. He is a member of IEEE, ACM and INFORMS Telecommunication Section. Dr. Zhang received the National Science Foundation CAREER Award in 1997.

Henry Tzeng received his B.S. degree from the Tatung Institute of Technologies, Taiwan, Republic of China, in 1988. He received his M.S. and Ph.D. degrees in Electrical Engineering from the University of California, Irvine, in 1993 and 1995, respectively. He was a recipient of the University of California Regent's Dissertation Fellowship in 1995 and the 1997 IEEE Browder J. Thompson Memorial Prize Award.

Since 1995, Dr. Tzeng has been a Member of Technical Staff at Bell Labs, Lucent Technologies, Holmdel, New Jersey. His research interests are in the area of algorithm design for high-speed networks and fault-tolerant distributed systems. In particular, he has been working on the issues related to reliable multicast, congestion control protocols for unicast/multicast ATM ABR services, TCP performance over ATM networks. He is currently working on the high performance routing technologies for the Internet.

Ya-Qin Zhang was born in Taiyuan, China in 1966. He received his B.S. and M.S. in Electrical Engineering from the University of Science and Technology of China (USTC) in 1983 and 1985, respectively. He received his Ph.D in Electrical Engineering from George Washington University, Washington D.C. in 1989. Dr. Zhang joined Microsoft Research in China in January 1999. He was previously Director of Multimedia Technology Laboratory at Sarnoff Corporation in Princeton, NJ (formerly David Sarnoff Research Center). He was with GTE Laboratories Inc. in Waltham, MA and Contel Technology Center in Virginia from 1989 to 1994.

Dr. Zhang has authored and co-authored over 150 journal and conference papers and 30 US patents granted or pending in digital video, Internet, multimedia, wireless and satellite communications. He currently serves as the Editor-in-Chief for the *IEEE Transactions on Circuits and Systems for Video Technology* from July 1997 to July 1999. He was a Guest Editor for the *Proceedings of the IEEE* special issue on Advances in Image and Video Compression in February 1995. Dr. Zhang also serves on the editorial boards of seven other international journals and over a dozen of conference committees. He has been an active contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia. Dr. Zhang received numerous awards, including several industry technical achievement awards and IEEE awards. He is a Fellow of IEEE.