

Dynamic Bandwidth Allocation for Real-time VBR Video Traffic in ATM Networks¹

Hon-Wai Chu

School of Science & Technology
The Open University of Hong Kong
30 Good Shepherd Street, Kowloon, Hong Kong.
e-mail: wchu@ouhk.edu.hk

Danny H. K. Tsang

Department of Electrical & Electronic Engineering
The Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
e-mail: eetsang@ee.ust.hk

Abstract

In this paper, we study the problem of dynamic bandwidth allocation for an ATM multiplexer loaded with real-time VBR video traffic. The proposed mechanism adjusts the allocated bandwidth at regular intervals based on measured QoS. Instead of actual measurement of small cell loss ratios, we combine virtual output buffer method with a regression method to shorten the time interval required for estimating the required bandwidth to support the specified QoS by prediction. Using numerical examples, we show that our proposed dynamic bandwidth allocation scheme is more efficient than the optimum static bandwidth allocation scheme. Our measurement-based scheme offers several advantages: 1) it removes the dependence on accurate traffic parameters to be declared by users; 2) it can be applied to a wide variety of traffic sources; 3) it adapts quickly to the variations of traffic with small measurement intervals; and 4) the complexity of the algorithm is small.

1 Introduction

With the large scale deployment of ATM-based Broadband ISDN, digital video is foreseen to become an increasingly important type of network traffic. Real-time video can be supported by using on-line measurement based approaches to dynamically allocate the required bandwidth to support the QoS. Some approaches [2][13] use the previous frame sizes to predict the size of the future frames and future bandwidth requirements. While these approaches which allocate bandwidth on a frame-by-frame basis are efficient in theory, they may not be feasible in practice since it may be difficult for the network to dynamically allocate bandwidth as frequently as each frame period (i.e. 24 or 30 times per second). Other approaches [3] use a larger time window (about 1 second) to reduce the frequency of bandwidth allocation. A problem

with almost all these approaches is that they are not easily scalable because they focus only on a single video source and require traffic monitoring on each individual source.

In [10][15], they have studied dynamic bandwidth allocation based on measured cell loss ratios (CLR). The measurement time interval of these methods is inversely proportional to the CLR. When the target CLR is small, a very large time window will be required to collect the statistics for CLR estimation. For example, measuring a CLR of 10^{-5} with meaningful statistics requires at least 10^7 cell arrivals to be monitored. However, it is not feasible to use very large time windows to dynamically allocate bandwidth since the traffic conditions may have changed significantly before accurate measurement can be obtained. Furthermore, their mechanisms do not control short-term QoS violation, which may lead to poor picture quality in video transmission for a short period of time.

In [4], we have shown that virtual output buffer (VOB) method [4][11] in combination with a performance estimation method can be used to shorten the time window required for monitoring small target CLRs such as 10^{-9} , which are difficult to measure directly. In this paper, we propose a measurement-based solution using the VOB method to dynamically allocate bandwidth to the aggregate traffic of several video sources. Our approach has several advantages. First, considering the aggregate traffic reduces the complexity associated with keeping track of the individual sources and also improves the accuracy of the bandwidth prediction. Second, bandwidth can be reallocated within a reasonable time window (2s to 10s). Third, resources are allocated based on measured QoS so that it does not require the source to specify precise traffic descriptors or traffic models.

This paper is organized as follows. Section 2 presents the motivation for studying dynamic bandwidth allocation for VBR video. Section 3 introduces our dynamic bandwidth allocation scheme. Section 4 presents two modified mechanisms to control short-term cell losses. Section 5 evaluates the performance of our dynamic schemes using numerical examples. Section 6 presents our conclusions.

1. Supported in part by the Hongkong Telecom Institute of Information Technology grant HKTIT93/94.EG01 and in part by the RGC Earmarked Grant HKUST780/96E

2 Motivation

In accordance with the ATM Forum specification, certain traffic parameters [1] are used to setup the traffic contract during the call admission phase. However, it is very difficult to characterize a long video sequence (say, 2 hours) with only a few traffic descriptors. Owing to the unpredictable nature of real-time video services, these traffic descriptors are expected to be imprecise. To take care of this problem, one can reserve resources for the worst case scenario to guarantee the QoS [5] but this results in low utilization of network bandwidth.

To improve the utilization of the network, the ABR service class [1] is proposed to utilize the leftover bandwidth between the VBR sources and the reserved bandwidth. Most of the studies in the literatures assumed that the ABR mechanism can find out this left-over bandwidth by measurement. Then the ABR service adjusts its transmission rate to use this bandwidth accordingly. Nevertheless, an accurate estimate of the leftover bandwidth is still an open research problem

To overcome this difficulty, we propose a new solution to utilize this left-over bandwidth by ABR services. The idea is to estimate the minimum required bandwidth to guarantee the QoS for VBR traffic at regular intervals. We then dynamically adjust the bandwidth allocation based on this estimated value in the next interval. That is, we have a piecewise CBR bandwidth allocation for VBR connections. Thus, the leftover bandwidth is simply the difference between the reserved bandwidth and the piecewise CBR allocation. That is, the leftover bandwidth is a constant value in each measurement period, which can be used by the ABR services until the value is adjusted in the next updated interval. This not only simplifies the ABR mechanism but also improves its response time.

3 Dynamic Bandwidth Allocation

The QoS parameters for real-time applications are generally specified by Cell Loss Ratio (CLR), maximum cell transfer delay (maxCTD) and cell delay variation (jitter) [1]. The issue of jitter is not considered in this work, since it can be controlled at the receiver at the expense of larger buffers and increased delays. Moreover, the maxCTD can be bounded by limiting the size of output buffer. Thus the QoS parameter we are most interested in is cell loss ratio.

3.1 Measurement-based Approach

We consider an ATM multiplexer loaded with N VBR video sources. The multiplexer has a finite buffer of size K cells and serves cells on a first-come-first-served basis at rate C cells per second. Since we are using a single FIFO buffer, we assume that the QoS (CLR) requirement for each video source is the same. Our objective is to allocate the

minimum service capacity which will meet the specified CLR ϵ . Bandwidth is reallocated at regular time intervals of duration Δ to accommodate the highly bursty nature of VBR video. Let $A(n)$ and $L(n)$ denote the number of cells that arrive at the buffer and the number of cells lost due to buffer overflow during the n^{th} measurement period. The CLR during the n^{th} measurement period is $P(n) = L(n)/A(n)$. The cumulative cell loss ratio for all intervals up to and including the n^{th} interval is denoted

$$P_{cul}(n) = \sum_{i=1}^n L(i) / \sum_{i=1}^n A(i) .$$

Similar measurement-based schemes are considered in [10] and [15] to allocate bandwidth dynamically according to observed CLR at regular intervals. A virtual buffer (i.e. a counter) is used to estimate the CLR with specified output capacity during each measurement period as follows. Let the size of the virtual buffer be equal to the size of the actual buffer (K cells). When a cell arrives at the actual buffer, it triggers a virtual cell to arrive at the virtual buffer (i.e., the counter increments by one). The virtual buffer is served with the same allocated bandwidth $C(n)$ as the real buffer at the n^{th} measurement period. The buffer occupancy of the virtual buffer is monitored for the duration of the measurement period. The number of lost virtual cells (i.e. cells which overflow the virtual buffer) are counted and the CLR $P(n)$ in the n^{th} measurement period is calculated. If $P(n)$ is larger (smaller) than the target CLR ϵ , then the allocated bandwidth $C(n+1)$ for the next period is increased (decreased). The required capacity to support the target CLR ϵ is allocated dynamically. However, the above methods require that the target CLR be large enough to be measured accurately within a measurement period. If the target CLR is small or the measurement period is short, the variance of the measured CLR in each measurement period can be very large. To reduce the variance, we apply the virtual output buffer method to estimate small CLR.

3.2 Virtual Output Buffer Approach

In [4], we introduced the Virtual Output Buffer (VOB) method for performance monitoring of small target CLR value. Here, the VOB uses a parallel set of counters to simulate virtual cells being multiplexed into several virtual buffers with different service capacities, as shown in Figure 1. It is conceptually similar to the parallel virtual buffers described in [10]. The major difference is that the target CLR in [10] must be large enough to be measured directly (e.g., 10^{-3}). On the contrary, we do not have this limitation in our case because the small target CLR (e.g., 10^{-5}) is obtained by prediction instead of direct measurement.

Let our VOB have M parallel virtual buffers with

services rates C_m , $m = 1, \dots, M$. Without loss of generality, we assume that $0 \leq C_1 < C_2 < \dots < C_M \leq C_{res}$, where C_{res} is the reserved capacity in the CAC mechanism. This reserved bandwidth is dedicated to the VBR video sources and cannot be used by other bandwidth guarantee services, like CBR and VBR. At the beginning of each measurement period, the buffer occupancy of each virtual buffer is set equal to the actual buffer occupancy. The number of cells that arrive $A(n)$ and cells that are lost $L_m(n)$ are measured for the m^{th} virtual buffer in the n^{th} measurement period. For the m^{th} virtual buffer with capacity C_m , we calculate the CLR $P_m(n) = L_m(n)/A(n)$. Those virtual buffers with smaller service capacities have higher virtual CLRs.

To relate the statistics of cell losses of these virtual buffers to those of physical buffer, we need to find a relationship between the service capacity and CLR. The simplest relationship in the literatures is the Gaussian approximation [7], which basically provides a quadratic relationship between $\log(\text{CLR})$ and service capacity C . Thus, we assume the following quadratic function to relate $\log(\text{CLR})$ and C :

$$\log(\text{CLR}) = a_0 C^2 + a_1 C + a_2, \quad (1)$$

where a_i 's are constants to be determined by minimizing the least squares error as follow:

$$\min \sum_{i=1}^{M'} [\log(P_i) - (a_0 C_i^2 + a_1 C_i + a_2)]^2, \quad (2)$$

where M' is an integer $\leq M$.

Once we have determined the coefficients a_i 's, we can use (1) to estimate the required bandwidth to support the target CLR value. The proposed algorithm is flexible since it makes no assumptions about the arrival process of the video sources. The regression method only requires to know the CLRs of the multiplexer when the service capacities are small. These are obtained from measurement using the set of parallel virtual buffers, i.e. (C_m, P_m) , $m = 1, \dots, M$.

The proposed bandwidth allocation mechanism operates as follows. For every measurement period, we apply the VOB method to obtain M pairs of (C_m, P_m) . These data are used to determine a_i 's in (1). Then (1) is used to estimate the bandwidth $C_e(n)$ that guarantees the target CLR (say, 10^{-5}) during the n^{th} measurement period. The allocated bandwidth for the $(n+1)^{\text{th}}$ measurement period is given by:

$$C_a(n+1) = \alpha C_a(n) + (1 - \alpha) C_e(n), \quad (3)$$

where $\alpha \in [0, 1]$ is the weight for the estimated

bandwidth, and $C_a(n)$ is the allocated capacity in the n^{th} measurement period. We set $C_a(1) = C_{res}$. Note that (3) is a simple exponential forecasting commonly used in time series prediction. Accordingly, the residual bandwidth that can be used by ABR service is simply equal to $C_{res} - C_a(n)$ in the n^{th} measurement period.

The advantages of this approach are two-fold. First, it is efficient and flexible since it attempts to allocate the minimum required bandwidth to satisfy the target QoS without assuming any particular traffic model. Second, it allows a much lower CLR to be specified while using a reasonable measurement interval.

4 Control of Short-term Cell Losses

Real time video is highly bursty and difficult to predict. When there is a sudden jump in the arrival rate due to scene changes, we may under-allocate the required bandwidth in one particular measurement period. Excessive cells may be lost in this particular period. Although error correction technique, such as error concealment, can handle to certain degrees this kind of short-term cell losses, it may cause a sudden drop in picture quality for a short period of time. This may be totally unacceptable from the users' point of view. Furthermore, the cumulative CLR may also violate the guaranteed QoS due to this large short-term cell losses. Since we do not control such short-term cell losses in our proposed algorithm in Section 3, we call this algorithm to be *Uncontrolled Dynamic Algorithm (UCDA)*. In the following, we suggest two methods to control such short-term cell loss due to under-allocation of bandwidth.

The *Over-Allocation (OA)* method simply specifies ρ , a target bandwidth utilization, and allocates $C_a(n+1)/\rho$ instead of $C_a(n+1)$ as suggested in [2][3]. We find that setting ρ between 0.9~0.95 is sufficient to handle the effect of sudden surges in arrival rates for most cases. While this method is simple, it wastes bandwidth because the *UCDA* algorithm only occasionally under-allocates bandwidth (usually less than 5% of the time).

The *Re-Allocation (RA)* method avoids excessive cell loss by changing the bandwidth within a measurement period. For each measurement period, a target value of short-term cell loss L_{short} is specified. If insufficient bandwidth is allocated during the measurement period, more than L_{short} cells will be lost, which triggers the immediate increase of bandwidth to the reserved bandwidth C_{res} for the remaining interval. This mechanism is similar to the dynamic bandwidth allocation based on queue length threshold [12].

When the bandwidth allocation is accurate (i.e., the bandwidth allocation is approximately equal to the input rate), the expected number of cell losses allowed in a measurement period is $\varepsilon \Delta C_a(n)$, where ε is the target long-term CLR, $C_a(n)$ is the allocated capacity in n^{th} measurement period and Δ is the length of the measurement period. Suppose the UCDA algorithm fails τ percent of the time. To maintain the cumulative CLR to satisfy the target value ε , the maximum number of cells that are allowed to lose in a measurement period can be approximated by $\varepsilon \Delta C_a(n) / \tau$. Accordingly, we let $L_{short}(n)$ in the n^{th} measurement period be:

$$L_{short}(n) = \varepsilon \Delta C_a(n) / \tau \quad (4)$$

For example, using $\varepsilon = 10^{-5}$, $C_a(n) = 3.0 \times 10^5$ cells/s, $\Delta = 2$ s and $\tau = 5\%$, we have $L_{short}(n) = 120$ cells.

Generally speaking, larger τ gives smaller L_{short} and hence bandwidth reallocation is activated more frequently. This in turn gives smaller cumulative CLR but requires more bandwidth. The remaining problem is how to determine τ , the expected percentage of failure, which is not known in advance. We propose to estimate the value of τ on line as follows: we start with $\tau = 1$ (i.e., we require $L_{short}(1) = \varepsilon \Delta C_a(1) = \varepsilon \Delta C_{res}$). We then let the dynamic bandwidth allocation algorithm to run. If there is no reallocation up to the i^{th} interval, we will set $\tau = 1/(i+1)$ for the $(i+1)^{\text{th}}$ interval. Otherwise, if the first reallocation occurs at the i^{th} interval, we shall change τ to $2/(i+1)$ for the $(i+1)^{\text{th}}$ interval. Similarly, if the k^{th} reallocation occurs in the j^{th} interval, then we shall set $\tau = (k+1)/(j+1)$ for the $(j+1)^{\text{th}}$ interval. We can thus estimate τ using this on-line measurement. However, there may be no reallocation for a long period and τ will continue to decrease. Therefore we set a minimum value τ_{\min} to ensure that we control the maximum value of L_{short} . In general, the value of τ_{\min} depends on the number of cell arrivals in a measurement interval. If we have only a few number of cell arrivals, we may need to set a large τ_{\min} to get a reasonable L_{short} .

As mentioned previously, a fixed amount of bandwidth C_{res} has already been reserved for the VBR connections in the call admission phase. In the n^{th} measurement period, we allocate $C_a(n)$ to the VBR traffic and the residual bandwidth $C_{res} - C_a(n)$ is released to support ABR service. If more than L_{short} cells are lost in a measurement period, reallocation of bandwidth will be activated. Since ABR traffic is not sensitive to delay, it can be suspended temporarily to provide the extra bandwidth

$C_{res} - C_a(n)$ back to the video. Although we cannot adjust bandwidth instantaneously due to practical network management limitations, it normally takes a short time to finish the adjustment. It is shown in [17] that readjustment of ABR bandwidth allocation in a LAN network is feasible within 20 ms. If the bandwidth readjustment is infrequent, this approach is potentially more efficient than the OA approach.

5 Numerical Results

In this paper, real video traces are used to study the performance of our bandwidth allocation mechanism. We use both JPEG and MPEG-1 [6][16] encoded sequences as the video sources. The video traffic characteristics are summarized in Table 1.

5.1 Generation of Video Sequences

In this section, we simulate the performance of our proposed dynamic bandwidth allocation mechanism using video bit stream traces. We consider an ATM multiplexer with a finite buffer of size K . There are N video sources multiplexed to form the incoming traffic. Simple FIFO policy is used and cells which arrive to a full buffer are simply dropped. We assume that each source sends $1/T$ frames per second and cell arrivals are equally spaced within each frame interval. Furthermore, we chose the arrival instants of frames from different sources to be uniformly spaced in the $1/T$ interframe interval. This can reduce the *source-periodicity effect* [9], which causes different sources to experience very different CLRs in a FIFO queue.

We start our studies with a JPEG encoded video sequence which have characteristics shown in Table 1. The original sequence is 2 hours long and contains 171,000 frames. The frame rate is 24 frames/s and the cell size is assumed to be 48 bytes. Since we have only a limited number of real video traces, we use the method proposed in [6][9] to generate multiple video sources from a single trace. We consider the VBR trace as a circular list and then randomly pick a starting point in the list as the first frame of a video sequence. The starting point of each trace is chosen to be at least 4000 frames apart from each of the others. Since the videos have long range dependence, the cross-correlation between sources may be significant even for such large lags, i.e., the generated sources are not truly independent. To improve this undesirable behaviour, we only use the first 60,000 frames ($= 2500$ s) from each source. Therefore, the characteristics of two generated video sources can be different.

To simulate N multiplexed sources, we randomly generate N such sources which are then multiplexed together. The experiment is repeated with different random

generations of the N sources.

5.2 Simulation Results

Suppose we have $N = 20$ video sources which are multiplexed into a buffer of size 500 cells with call duration 2500 seconds. Let the target cell loss ratio be 10^{-5} . As argued in Section 3, the measurement-based algorithms proposed in [10][15] fail to estimate the required bandwidth to support such a small CLR value in a short measurement period. For instance, it takes at least 50 seconds to estimate a CLR of 10^{-5} with acceptable accuracy using direct measurement with 20 multiplexed sources. However, the traffic conditions may have already changed well before their algorithms can take corrective action by adjusting the bandwidth allocation. We thus apply the algorithm we proposed in Section 3 to allocate bandwidth dynamically using a much smaller interval Δ (say 2s).

In the simulation, the weight α is set to 0.9 and the measurement period Δ is 2 seconds. We also set the parameters $\rho = 0.95$, and $\tau_{\min} = 2\%$ for the two methods respectively. Due to the unpredictable nature of real-time VBR video, the required bandwidth is often overestimated in the CAC procedure. Let the initial bandwidth allocation $C_a(0) = C_{res} = 3.36 \times 10^5$ cells/s. For comparison, the required bandwidth to support 20 JPEG sources with traffic characteristics in Table 1 using the stationary Gaussian approximation [7] is 3.42×10^5 cells/s. We shall show that the proposed dynamic bandwidth allocation schemes can improve the bandwidth utilization by learning the required bandwidth on-line.

Table 2 summarizes the performance of the *OA* method, the *RA* method, and the *UCDA* method. We find that the worst CLR (e.g. 4.1×10^{-2} in *UCDA*) in one particular measurement period can make the cumulative CLR violate the QoS requirement. However, both of the controlled algorithms can guarantee the target CLR since they limit the worst case CLR (in the order of 10^{-3}). We also use simulation to find the minimum bandwidth which guarantees the target CLR using static bandwidth allocation. In general, precise traffic parameters and models are required to get this minimum value. Nevertheless, accurate traffic parameters for real-time video is highly questionable and VBR video modeling is still an open research problem. Therefore, the minimum bandwidth provided in Table 2 is only an ideal static allocation to be compared with our dynamic scheme. Table 2 shows that the *RA*, *OA* and ideal static schemes are all smaller than the reserved bandwidth C_{res} .

In summary, the *RA* scheme is the most efficient scheme that can guarantee the target QoS. However, this scheme requires adjustment of bandwidth allocation within a measurement interval. Although the reallocation is

infrequent (e.g. τ is around 2% for our simulation), this may be infeasible in wide area networks because of the long delay in adjustment of bandwidth. The *OA* schemes therefore may be useful in this case. Furthermore, Table 2 indicates that *OA* schemes is very competitive with the ideal static schemes since the difference in bandwidth requirement is less than 1%. As a result, *OA* scheme is not only a feasible but also an efficient choice in practice. Note that the exact amount of savings depends on the accuracy of reserved bandwidth in CAC. In other words, if we have a loose estimation of required bandwidth in the CAC, which is the case in real-time VBR video, the dynamic schemes will significantly improve the network efficiency.

5.3 MPEG Sequences

In this subsection, we evaluate the performance of our algorithms using MPEG frame size traces. As suggested in [8], MPEG streams are smoothed in a buffer before sending to the ATM multiplexer. This can improve the utilization of the network by removing the burstiness due to periodic occurrence of *I/B/P* frames.

We begin with the movie “Star Wars” and use it to generate multiple video sequences as in the JPEG case. We consider 10 such MPEG streams being multiplexed into a buffer of 500 cells with target CLR of 5×10^{-5} . Now the aggregate mean rate of the multiplexed stream is only 1.1×10^4 cells/s and hence an accurate measurement of CLR in a small windows seems to be difficult.

We apply the *RA* scheme with $\alpha = 0.9$ and $\Delta = 5$ seconds to a typical multiplexed sources. Since the number of cells which arrive in a 5 seconds interval is relatively small, we set $\tau_{\min} = 10\%$ to allow a bigger L_{short} (around 30 cells). The results are shown in Table 3 which shows again that ideal static scheme requires more bandwidth than the dynamic scheme (1.26×10^4 cells/s vs. 1.22×10^4 cells/s) to support a CLR requirement of 5×10^{-5} .

In all the previous discussions, we have applied the same sequence to generate multiple sources and the traffic characteristics of each source may be similar to each other. Therefore, we would like to test the scheme with heterogeneous traffic. We take two MPEG-1 encoded sequences, namely “Jurassic Park” and “MTV” [16] as our traffic source. We use each sequence to generate 7 sources respectively and then multiplex 14 sources into a buffer of 500 cells. Since the original sequences are relatively short (only 40,000 frames each), the simulation is done only for the first 1500s. The reallocation scheme is applied to the aggregate sources with $\alpha = 0.9$ and $\Delta = 5$ s. Table 3 shows the results for a typical video trace and the findings are similar to the homogeneous case. That is, dynamic schemes outperform the static schemes. Therefore, our proposed algorithm can be applied to heterogeneous traffic as well.

The experiments are repeated for different combinations of sources and similar conclusions are obtained. As a result, our algorithm does not appear to be restricted to any particular source characteristics or traffic mixes. However, the major restriction of our method is to require the same QoS for the all sources because of the simple FIFO queue. If we require different QoS for different traffic, more complicated scheduling schemes such as Generalized Processor Sharing [14] are required.

6 Conclusions

In this paper, we introduce a dynamic bandwidth allocation mechanism based on measured QoS. Instead of adjusting the allocated bandwidth based on direct measurement of cell losses, which occur rarely (say, 10^{-5}), we propose a bandwidth estimation mechanism that combines the Virtual Output Buffer method with a regression algorithm. This approach shortens the time window required for estimating the required bandwidth to support the specified QoS by prediction.

We also discuss the potential problem of short-term cell losses with our *UCDA* scheme and then introduce two solutions, namely the *OA* and *RA* schemes, to control cell losses. Through simulation, we show that the *RA* scheme can be more efficient than the *OA* scheme if dynamic adjustment of bandwidth within a measurement window is allowed.

Using numerical examples, we show that our proposed dynamic bandwidth scheme is more efficient than the ideal static bandwidth allocation scheme, which requires accurate traffic information. Finally, we examine the effects of different parameters on the performance of our proposed algorithms.

In summary, the major advantages of our proposed measurement-based schemes are: 1) they do not require the user to specify accurate traffic parameters; 2) they can be applied to a wide variety of traffic sources; 3) they adapt quickly to the traffic variation within small measurement intervals; and 4) the complexity of the algorithms is small.

Acknowledgment

We would like to thank Mark. W. Garrett, Bellcore and Martin Vetterli, Univ. California, Berkeley and Oliver Rose, Univ. of Wuerzburg, for generating the video traces used in this paper and making them available for ftp in the public domain.

ftp site: thumper.bellcore.com/pub/vbr.video.trace
ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG

References

- [1] The ATM Forum, Traffic Management Specification Version 4.0, March 1996.
- [2] A. Adas, "Supporting real time VBR video using dynamic reservation based on linear prediction", Proc. IEEE INFOCOM'96, pp. 1476-1483, 1996

- [3] S. Chong and S.-Q. Li, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM", IEEE JSAC, vol. 13, no. 1, pp. 12-23, Jan. 1995.
- [4] H-W Chu and D.H.K. Tsang, "Performance monitoring using regression approach in ATM networks", Proc. ITC 15, Washington, D.C., June 1997.
- [5] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation", IEEE Trans. on Info. theory, vol. 37, no. 1, pp. 114-141, Jan. 1991.
- [6] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic", Proc. SIGCOMM 94, pp. 269-280, 1994.
- [7] R. Guerin, H. Ahmadi, and M. Naghshineh, "Effective capacity and its application to bandwidth allocation in high-speed networks", IEEE J. Select. Areas Commun., vol. 9, no. 7 pp. 968-981, 1991.
- [8] Z. Guo, J. Ni and T. Yang, "Traffic smoothing and bandwidth allocation for VBR MPEG-2 video connection in ATM network", Proc. International Conference on Computer Communications and Networks (ICCCN'95), pp. 554-561, 1995.
- [9] D.P. Heyman, A. Tabatabai, and T.V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks", IEEE Trans. on Circuits and Systems for Video Technology vol. 2, no. 1, pp. 49-59, 1992.
- [10] I. Hsu and J. Walrand, "Dynamic bandwidth allocation for ATM switches", Journal of Applied Probability, vol. 33, no. 3, pp. 758-71, Sept. 1996.
- [11] S.T.C. Lam, B. Bensaou and D.H.K. Tsang, "Efficient estimation of cell loss probability in ATM multiplexers with a fuzzy logic system", Proc. IFIP Conf. on Broadband Communications 96, Montreal, Chapman & Hall, pp. 306-317, 1996.
- [12] P. Mishra and S. Tripathi, "Dynamic bandwidth allocation in high speed integrated service networks", Proc. IEEE INFOCOM'93, pp. 1298-1305, 1993.
- [13] P. Pancha and M. El Zarki, "Bandwidth-allocation schemes for variable-bit-rate MPEG source in ATM networks", IEEE Trans. on Circuits and Systems for Video Technology, vol.3, no.3, pp.190-8, June 1993.
- [14] A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case", IEEE/ACM Trans. on Network., vol.1, no.3 (June 1993) pp. 344-357.
- [15] S. Rampal, D. S. Reeves and I. Viniotis, "Dynamic resource allocation based on measured QoS", Technical Report TR 96-2, North Carolina State University, available on the ftp site: ftp.csc.ncsu.edu.
- [16] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems", University of Wuerzburg. Institute of Computer Science Research Report Series. Report No. 101. Feb. 1995.
- [17] D.H.K. Tsang and W.K.F. Wong, "A new rate-based switch algorithm for ABR traffic to achieve max-min fairness with analytical approximation delay adjustment", Proc. IEEE INFOCOM'96, pp. 1174-1181, 1996.

Bit stream	Bit stream statistics (bytes / frame)				Sequence length (frames)
	Max	Min	Mean	Std. Dev.	
Jurassic Park (MPEG-1)	14954	110	1635	1844	40,000
MTV (MPEG-1)	31426	60	2473	2682	40,000
Star Wars (MPEG-1)	23158	60	1950	2271	174,136
Star Wars (JPEG)	78459	8622	27791	6254	171,000

Table 1: Traffic characteristic of video sources.

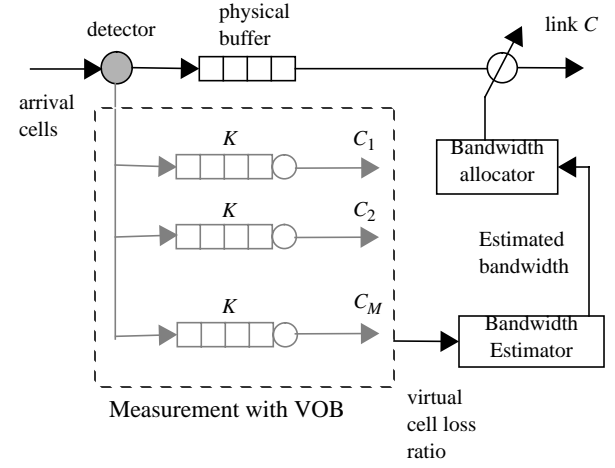


Figure 1: Dynamic bandwidth allocation based on real-time traffic measurement.

Bandwidth allocation methods	Allocated capacity (cells /s)		Cumulative CLR	Worst CLR in a measurement period
	Mean	Std. Dev.		
<i>UCDA</i>	3.01×10^5	6.77×10^3	1.1×10^{-4}	4.1×10^{-2}
<i>OA</i> (5%)	3.16×10^5	6.79×10^3	5.1×10^{-6}	4.6×10^{-3}
<i>RA</i>	3.01×10^5	7.15×10^3	9.4×10^{-6}	1.1×10^{-3}
Ideal Static	3.13×10^5	0	1.0×10^{-5}	1.0×10^{-2}

Table 2: Bandwidth allocation for 20 video sources with $\alpha = 0.9$, $\rho = 0.95$, $\tau_{\min} = 2\%$ and $\Delta = 2s$. Target CLR = 1.0×10^{-5} .

Source	Allocated capacity (cells /s)		Cumulative CLR with dynamic scheme	Static bandwidth required (cells/s)
	Mean	Std. Dev.		
10 Star Wars	1.22×10^4	400	3.0×10^{-5}	1.26×10^4
7 Jurassic Park + 7 MTV	2.27×10^4	1100	4.8×10^{-5}	2.49×10^4

Table 3 : Bandwidth allocation for MPEG sources with target CLR = 5×10^{-5} .