On-line Bandwidth Allocation for MPEG Videos*

Errin W. Fulp† and Douglas S. Reeves‡
Departments of ECE and CSC
North Carolina State University

Abstract

Real-time and other performance guarantees for multimedia can be achieved through proper resource allocation. This paper evaluates an allocation method called Resource Efficient Quality of Service (REQS). REQS is an on-line algorithm that dynamically adjusts the resource allocation, based upon the measured quality of service (QoS). Advantages of REQS include efficient use of resources, reasonable implementation cost, and stringent QoS control. In this paper, we show how REQS dynamically allocates bandwidth to achieve a given loss rate for actual MPEG VBR videos. We also develope an enhancement for REQS to better handle these sources. Comparisons of performance and cost with a peak-rate allocation method are also presented.

1 Introduction

Multimedia traffic (voice and video) in networks requires control of such quality of service (QoS) parameters as cell transfer delay and cell loss probability. To provide the QoS requirements, conventional approaches require the application to describe its traffic, using statistics such as burstiness, peak rate, mean rate, etc. These statistics are often conservative estimates, and are statically declared. Based upon these statistics, calculations are made to determine the resource allocation (buffer space, network bandwidth, etc.) needed to provide the requested QoS. However more efficient allocation is needed as the contention for limited resources increases.

QoS guarantees for multimedia traffic are difficult for three reasons. First, when the traffic is processed in real-time (for interactive applications), it is impossible to predict precisely the future behavior. Second, compressed video can exhibit long-range dependency [4], which implies that the standard statistical traffic models are probably inadequate. Third, the number of allocation changes (renegotiations) should be as few as possible, since contention for more resources may exist. For these reasons efficient resource allocation, along with good control of QoS, is a challenging problem.

To date a variety of bandwidth allocation methods have been proposed. *Off-line methods* make allocation decisions before any traffic is transmitted, and assume that the traffic is available for analysis (such as in the case of stored video) before transmission. An off-line method may allocate one (static) resource level for the duration of the application, or may renegotiate the resource level at various times over this duration. An example of a static off-line method is the peak-rate allocation. This approach has the advantages of simplicity and predictability, but suffers from the problems noted above. Off-line, renegotiation methods can result in significantly better resource utilization [2], but require complete information about the source traffic. For video, this means a priori access to the entire video's traffic trace, which is not possible for interactive applications.

On-line methods periodically renegotiate the resource allocation based upon a prediction of the future traffic behavior. This prediction is derived from measurements of the traffic that have been observed so far, and the QoS that has been experienced. Such methods, including [1] [3] [6] [7] do not have the problems associated with off-line methods. One method, Resource Efficient Quality of Service (REQS), has demonstrated excellent performance with Markov Modulated Bernoulli Process (MMBP) traffic sources. However, to date no on-line method has demonstrated the ability to tightly control QoS for such difficult applications as transmission of actual compressed video. In addition, most methods suffer from a large number of renegotiations [6], and/or rely on a complex measurement and algorithm [3].

In this paper the performance of an existing on-line

*This work was supported by AFOSR grant F49620-96-1-0061. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFOSR or the U.S. Government.

†ewfulp@eos.ncsu.edu

‡reeves@eos.ncsu.edu

allocation method, REQS, is observed using MPEG-compressed traffic. This method was chosen since it has shown excellent performance with modeled traffic. An enhancement, interrupts, of the algorithm is then proposed and compared to peak rate allocation method.

2 Algorithm Description

2.1 System Model

REQS dynamically renegotiates the server rate (bandwidth) of a finite capacity queue in order to meet a desired QoS. A cell is the fixed-length unit of traffic storage and transmission. In this paper the QoS of interest is the cell loss probability (CLP) of a single source. Cell arrivals to and losses from this queue are monitored throughout the duration of the application. Rate changes are renegotiated at discrete instances of time. We denote the nth renegotiation instant as t_n, and the interval between renegotiation points t_n and t_{n+1} as the nth update interval, U_n. The service rate during U_n is constant and is denoted as μ_n.

During the nth interval, let the number of arrivals be represented by A_n and the number of losses as L_n. The CLP of the nth interval is then calculated as $P_n = L_n/A_n$. The cumulative CLP of all the intervals up to and including the nth is

$$P_{0\dots n} = \sum_{i=0}^{n} \frac{L_i}{A_i}$$

The CLP desired by the user is denoted Q_l. The goal of REQS is to adjust the server rate to a minimum value μ^*, while the CLP approaches Q_l. Secondary goals are simplicity of implementation, robustness, and minimizing the number of renegotiations.

2.2 Algorithm for Dynamic Resource Allocation

At each renegotiation point REQS adjusts the server rate according to the following formula:

$$\mu_{n+1} \leftarrow \mu_n + K_n \times \ln\left(\frac{P_n}{Q_l}\right) \qquad (1)$$

The natural logarithm of the ratio of P_n to Q_l times the variable K_n is used to calculate the change in the server rate. The value of K_n and the length of the update interval U_n are determined using the algorithm in figure 1. Details concerning the algorithm are presented in [7]. REQS has shown good performance

/* K_0, K_∞, U_0 and Q_l assumed to be given */
/* initial values: inc_flag = **TRUE**, mode = 1 */
curr_error $\leftarrow \ln(P_n/Q_l)$
prev_error $\leftarrow \ln(P_{n-1}/Q_l)$

if(mode = 1)**then**
    $U_n \leftarrow U_{n-1}$
    if ((curr_error $\times$ prev_error > 0)**AND ...**
        (inc_flag = **TRUE**))**then**
        $K_n \leftarrow K_{n-1} + K_0$
    else
        inc_flag $\leftarrow$ **FALSE**
        $K_n \leftarrow K_{n-1}/2$
        if($K_n \leq K_\infty$)
            $K_n \leftarrow K_\infty$
            mode $\leftarrow 2$
        endif
    endif
else
    $K_n \leftarrow K_{n-1}$
    if (curr_error $\times$ prev_error < 0)**then**
        $U_n \leftarrow 2 \times U_{n-1}$
    else
        $U_n \leftarrow U_{n-1}$
    endif
endif

Figure 1: REQS algorithm, varying K_n and U_n

with MMBP traffic. In the next section REQS' performance with actual MPEG-compressed traces is investigated.

3 Numerical Results

As described in the introduction, MPEG-compressed video sources are among the most difficult to control due to their burst behavior. In this section the performance of REQS is investigated using MPEG-compressed videos. All traces were obtained from Oliver Rose at the University of Würzburg, Germany [9][1]. Each trace is a thirty minute segment of the original video and were encoded using the same MPEG-1 encoder card. Relevant statistics of each video are presented in [5] and [9]. As reported in [9], the Hurst parameters indicate all videos exhibit long-range dependency, and significant peak-to-mean ratios. Therefore it is evident that these are very difficult sources to regulate, and to date there has been

[1] Traces can be obtained from the ftp site ftp-info3.informatik.uni-wuerzburg.de in the directory /pub/MPEG

no successful attempt to efficiently manage them on-line.

For each experiment the system described in section 2.1 was simulated. The desired QoS was a CLP of 1×10^{-3} and the queue capacity was 80 ATM cells (48 byte payload) [7]. For each I, B or P MPEG frame, the equivalent number of ATM cells was determined. The cell arrival times were then uniformly distributed over the duration of the frame. This process was repeated for each frame until the end of the trace was reached. No smoothing, multiplexing, filtering or quantization changes of any kind were made to the videos. We consider these experiments to be a "hardest-case" test of REQS.

REQS performed poorly using the MPEG-compressed traces. Representative allocation and cumulative CLP graphs are given in figure 2. In each case the algorithm failed to provide the desired CLP for the duration of the trace. The poor performance was primarily due to mode transition. As described in [7], mode 1 is used to quickly converge to an approximate bandwidth. Once an approximate bandwidth is found, the algorithm moves to mode 2 where adjustments to the server rate are smaller and more infrequent (smaller K and larger U). The intent is to fine tune the bandwidth over longer intervals. However due to the nature of MPEG-compressed video, the algorithm may move prematurely to mode 2. If a large burst arrives at this point, the algorithm is not able to increase the bandwidth quickly enough to meet the changing source characteristics. This behavior can be seen in figure 2.

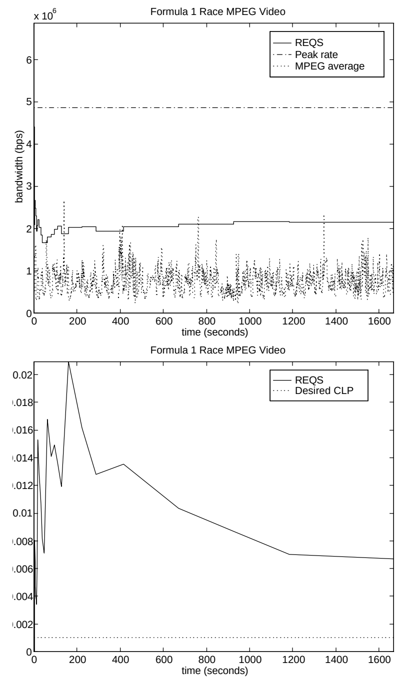

Figure 2: Original REQS allocation and cumulative CLP.

3.1 Algorithm Enhancement

To reduce the effect of changing source characteristics during an interval interrupts are introduced. REQS now monitors the system and may update the server rate via an interrupt at any time, if both the cumulative and nth CLP are greater than the desired value, Q_l. The server rate is increased by applying equation 1. Representative allocation and cumulative CLP graphs are presented in figure 3. Here the desired CLP is provided for the video. Interrupts are used to increase the bandwidth once the desired CLP is not met. Similar results were found for all fifteen videos and are presented in table 1. One metric for comparison used is the number of bits used to transmit the video, or equivalently, the area under the allocation curve for the duration of the video. For each video, REQS was able to provide the desired CLP with fewer bits than peak rate allocation. Savings of REQS with interrupts ranged from 7% to 51% over peak rate. Allocation changes ranged from 56 to 185,

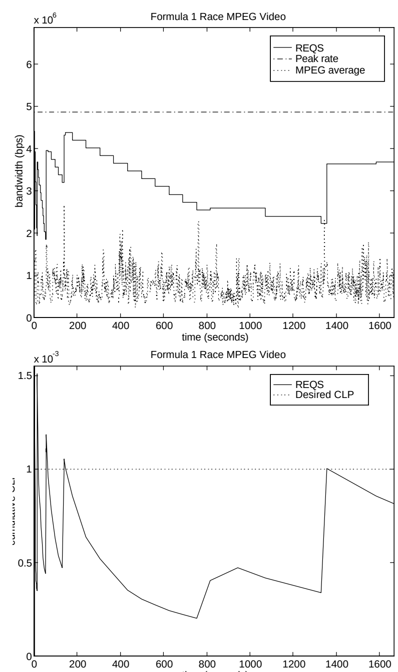

Figure 3: REQS with interrupts allocation and cumulative CLP.

Video			Peak	REQS with Interrupts				
	Bits ($\times 10^9$)	Peak/ Mean	Bits Used ($\times 10^9$)	Number of R.U.	Number of I.	Bits Used ($\times 10^9$)	CLP ($\times 10^{-3}$)	% Savings over Peak
Asterix	0.894	6.59	5.90	37	42	3.76	0.444	36
ATP Tennis	0.876	8.72	7.63	49	58	6.10	0.645	20
F1 Race	1.230	6.58	8.10	47	66	5.25	0.814	35
Goldfinger	0.972	10.1	9.79	101	48	5.32	0.677	46
Jurassic Park	0.523	9.15	4.78	39	66	3.23	0.651	32
Movie Review	0.572	12.1	6.90	84	87	4.66	0.419	32
Mr. Bean	0.706	13.0	9.17	43	125	6.81	0.275	26
MTV	0.984	9.31	9.17	48	129	6.64	0.347	26
News	0.651	9.01	7.78	42	88	4.59	0.393	41
Lambs	0.292	18.4	5.37	66	119	3.01	0.695	44
Simpsons	0.742	12.9	9.62	49	49	4.73	0.618	51
Soccer	1.090	6.9	7.48	42	94	5.84	0.544	22
Super Bowl	0.940	5.99	5.63	40	40	5.22	0.278	7
Talk	0.581	7.34	4.27	32	24	2.43	0.550	43
Terminator	0.436	7.29	3.18	39	36	2.11	0.395	37
Legend: **R.U.** = regular updates, **I.** = interrupts								
Note: Each video consisted of 40000 frames								

Table 1: Algorithm performance.

which is less than other on-line algorithms.

4 Conclusions

This paper presented an enhancement for an on-line algorithm, REQS, to efficiently allocate resources to provide a desired QoS. The original REQS algorithm was combined with interrupts to manage the bandwidth of MPEG traces with a specified allowable cell loss probability. Fifteen actual MPEG traces were collected and used in the experiments to measure the performance of the original algorithm and the algorithm with interrupts. The original REQS was unable to provide the desired QoS due to its modes of operation. Interrupts allowed the algorithm change the bandwidth when the source characteristics had changed. Savings of REQS with interrupts ranged from 7% to 51% over peak rate allocation, while only requiring 56 to 185 renegotiations.

References

[1] A. Adas. Supporting Real Time VBR Video Using Dynamic Reservation Based on Linear Prediction. In *IEEE INFOCOM'96*, pages 1467–1483, 1996.

[2] K. Chang and H. T. Kung. Efficient Time-Domain Bandwidth Allocation in A Video-on-Demand System. In *The Fifth ICCCN Conference*, pages 172–178, Oct. 1996.

[3] S. Chong, S.-Q. Li, and J. Ghosh. Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM. *IEEE J. on Selected Areas in Communication*, 13(1):12–23, January 1995.

[4] A. Erramilli, O. Narayan, and W. Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet Traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, Apr. 1996.

[5] E. W. Fulp, D. S. Reeves, and Y. Viniotis. Dynamic Bandwidth Allocation for VBR Sources. Technical Report TR-96/45, NCSU Department of Electrical and Computer Engineering, Oct. 1996.

[6] I. Hsu and J. Walrand. Dynamic Bandwidth Allocation for ATM Switches. To appear in the *Journal of Applied Probability*, Sept. 1996.

[7] S. Rampal, D. Reeves, and I. Viniotis. Dynamic Resource Allocation Based on Measured QoS. In *The Fifth ICCCN Conference*, pages 24–27, 1996.

[8] S. Rampal, D. Reeves, Y. Viniotis, and D. Argrawal. Dynamic Resource Allocation Based on Measured QoS. Technical Report TR-96/02, NCSU Department of Computer Science, 1995.

[9] O. Rose. Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems. Technical Report 101, University of Würzburg Institute of Computer Science, Feb. 1995.