Network Bandwidth Allocation and Admission Control for a Continuous Media File Server

Dwight Makaroff, Gerald Neufeld, and Norman Hutchinson
{makaroff,neufeld,hutchinson}@cs.ubc.ca

Department of Computer Science, University of British Columbia
Vancouver, B.C. V6T 1Z4 Canada

Abstract

Resource reservation is required to guarantee delivery of continuous media data from a server across a network for continuous playback by a client. This paper addresses the characterization of the network bandwidth requirements of Variable Bit Rate data streams and the corresponding admission control mechanism at the server. We show that a characterization which sends data early, making intelligent use of client buffer space, reduces the amount of network bandwidth reserved per stream without creating any start-up latency. The results of performance experiments in a Continuous Media File Server find that operation with requests arriving over time can deliver up to 90% of the network bandwidth. The experiments also show that a system designer can configure a server so that the network and disk bandwidth can scale together.

Keywords: multimedia, file servers, variable bit rate, admission control, network transmission

1 Introduction

Continuous media file servers require that several system resources be reserved in order to guarantee timely delivery of the data to end-user clients. These resources include disk, network, and processor bandwidth. In a heterogeneous system accommodating variable bit-rate data streams, the amount of each resource differs for each stream and varies over time. A key component of determining the amount of a resource to reserve is characterizing each stream's bandwidth. Admission control is necessary to ensure adequate server resources for the duration of the playback requested by the user.

In this paper, we examine network bandwidth reservation from both the server's and the client's point of view. The provision of network bandwidth within the network between the server and the client is beyond the scope of this paper and has been addressed extensively in other work [6]. The server is only aware of problems with delivery through feedback from the client.

Two aspects of the network resource management issue are important: the bandwidth usage profile of each individual stream and the combined load on the network interface provided by the requests of all the clients of one server. The remainder of this paper is organized as follows. We begin with a description of the system model, then describe the comparative network allocation algorithms, followed by the network admission control algorithm. The description of the experimental model then provides a framework for the results. This is followed by a comparison of our approach with related work and finally, some conclusions and possible directions for future work.

2 System Model

This study takes place in the context of a Continuous Media File Server (CMFS). This system model is shown in Figure 1. The server is scalable in that multiple disks can be attached to each server node. Multiple server nodes can be configured with a single administrator node. The cumulative data traffic from the set of disks on a single server node provides the bandwidth that this paper characterizes.

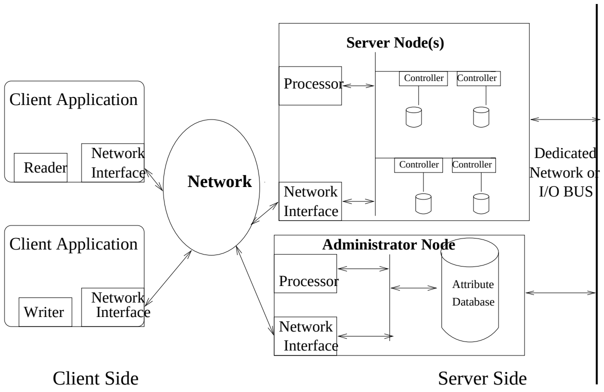

Figure 1: Organization of System

Client applications make requests for continuous media objects from the administrator node, which selects a copy of the object residing on one of the server nodes. A real-time data connection is then set up between the server node and the client to deliver the data at a rate that prevents the client application from starvation. If some small percentage of packets get corrupted or lost, the presentation can continue without loss of satisfaction from the user's point of view. Retransmissions can cause unacceptable latency [1].

Guaranteeing adequate bandwidth requires network resource reservation. This may be done in the form of a VBR connection in an ATM network, with statistical transmission guarantees. Cells may be lost due to transient overload. Such "capacity losses" (or "congestion losses") may invalidate the client's assumption on the expected error or loss rate, and may interfere with the client's ability to provide continuous playback [1][11]. We have selected instead to use CBR connections which can have bandwidth renegotiated, providing a small amount of overhead to the operation of the system.

If the network bandwidth cannot be maintained throughout stream delivery, some change to the delivery parameters is necessary. Unfortunately, the server does not know what adjustments would be appropriate for the client, nor if the client would be able to interpret the reduced amount of data that would be sent under the adjusted data rate. The client application requests a new delivery rate, which may have fewer frames per second, or involve skipping some sequences of the object. In this paper, we assume that the bandwidth of an individual server-client connection can always be maintained.

3 Network Block Schedule Creation

The network resource usage of a particular stream may be characterized in many ways. The tightest upper bound is the empirical envelope [6], which has been a basis for much of the previous work in this area. It results in a conservative, piece-wise linear function, specified by a set of parameters, but requires $O(n^2)$ time to compute (where n is the number of frames in the stream). Approximations have been developed based on leaky bucket schemes, but the results have still utilized the entire stream to calculate the bandwidth profile off-line. In the system model of the CMFS, the schedule must be created at request delivery time, because each play request could select different portions of the object (slow motion, skipping sequences) as in [9].

It is possible to give a single value for bandwidth characterization (such as the average bandwidth), and let the network infrastructure deal with transient overloads in the network. Such an allocation algorithm faces two main problems: client starvation and server buffer space. Sending at the average rate for the entire duration of stream delivery does not ensure that enough data will be present in the client buffer to handle peaks in the bandwidth which occur early in the stream. It is possible to prefetch data, but this introduces start-up latency and requires a large client buffer. Parameterized variants of average bandwidth allocation with intelligent discarding[1] of data at the server have shown reasonably good results. Both client buffer size and start-up latency have been parameters in previous research [13] where reductions in either buffer space or latency can

[1] requiring server knowledge of encoding formats.

be achieved. An approach which utilizes the VBR profile is essential to reduce both of these values simultaneously. A study of the effect of packet loss over the Internet for MPEG streams [1] shows that enhanced error concealment and/or error resilience techniques in the stream can reduce the apparent loss of quality in a manner transparent to a server such as the CMFS.

In keeping with the philosophy of admission control and resource usage characterization in the UBC CMFS, we have chosen to divide the time period during which data is transmitted into *network slots*, and provide a detailed schedule of the bandwidth needed in terms of a *network block schedule*. A network slot is an even multiple of the disk slot time. This schedule allows the system to transmit data at a constant rate within a network slot, known in other literature as Piecewise Constant Rate Transmission and Transport [2][8]. The size of a network slot is significantly larger than a disk slot for two main reasons: overhead of renegotiation and smoothing capability. A renegotiation takes a non-trivial amount of time and should be effective for more than a disk slot time. As well, the ability to smooth out the data delivery by sending data earlier in the network slot than is absolutely required increases performance. This utilizes the available client buffer space. Other research has experimented with the size of network slots in the range of 10 seconds to 1 minute [4] [14]. Zhang and Knightly [14] suggest that renegotiations at 20 second intervals provide good performance. We have used 20 seconds as the size of the network slot for the initial experiments.

Our initial algorithm (hereafter called *Original*) considers only the number of bytes that are required to be sent in each network slot. The cumulative average number of bytes per disk slot is calculated for each disk slot in the network slot. The maximum value encountered in the current network slot is rounded up to the next highest number of disk blocks (64 KBytes). This method has the advantage of absorbing peaks in the disk block schedule by assuming that the server can send at the specified rate for the entire network slot. Peaks which occur late in the network slot have marginally less influence in the cumulative average and will be absorbed easily as shown in Figure 2. Here, the first three large peaks in disk bandwidth at slots 68, 94, and 136 do not increase the reservation. If a peak in disk bandwidth occurs early in a network slot, then the maximum cumulative average is near this peak (disk slots 201 and 241).

Our server-based flow control policy [10] takes advantage of the client buffer by sending data to the client as early as possible, without overflow. Since the value used in the network block schedule is the maximum cumulative average, it is likely that some data will be present in the client buffer at the beginning of the next network slot.

The second algorithm improves on the first by explicitly accounting for sending data early. In nearly all cases, there is sufficient excess bandwidth to fill the client buffer. This reduces the amount of bandwidth that must be reserved for each subsequent slot, smoothing the network block schedule, and thus, we call it the *Smoothed* algorithm. A peak in disk bandwidth that occurs very early in a network slot could be merged with the previous network slot. Figure 3 shows the smoothed network block schedule for the same stream taking into account

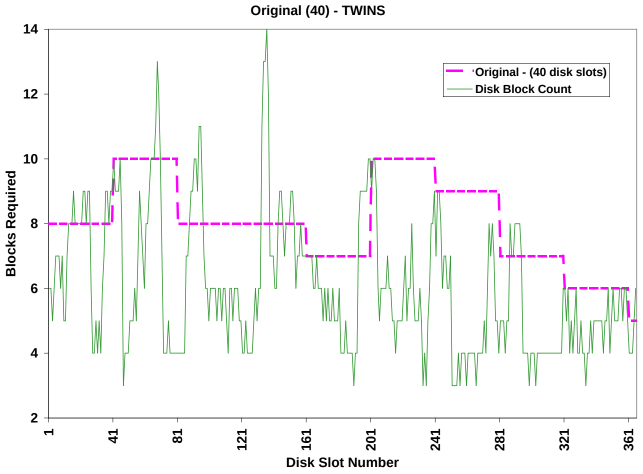

Figure 2: Network Block Schedule - Original

network send-ahead. Increased client buffer space enables smoothing to be more effective at reducing both the peaks and the overall bandwidth necessary [2].

A significant complication in the design is the assumption that the disk system has achieved sufficient read-ahead such that the buffers are available in memory for sending. The disk admission control algorithm utilized in the CMFS only guarantees that disk blocks will be available for sending at the end of the slot which they are required to be sent [9]. The disk subsystem guarantees a minimum bandwidth in every disk slot, for disk admission control. For a disk which is under heavy load, it is possible that the disk peaks which we have been trying to smooth at the network level will not be read off the disk when needed. If this is the case, the network bandwidth value must be increased above the cumulative average in order to transmit this peak amount when required.

The disk admission algorithm [9] guarantees that in steady state, the guaranteed bandwidth from the disk is always sufficient to service the accepted streams. In fact, the achieved disk bandwidth is greater than this value, because disk performance is variable and the average performance is somewhat above the guarantee. Thus, over time, all buffer space will be utilized by this aggressive read-ahead. The level of bandwidth for the accepted set of streams will always be lower than the capacity of the disk.

The issue of buffer space is slightly more complicated. In steady state, there are no buffers in which to read any blocks for the new stream, except those being returned to the system after being transmitted across the network. Buffers may be "stolen" from existing streams if the data is not needed until later than the deadline for the new stream. In the operation of the server, staggered request

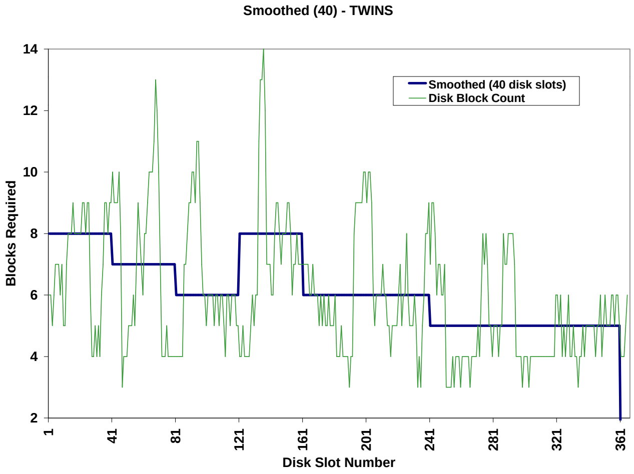

Figure 3: Network Block Schedule - Modified

arrivals and buffer stealing often results in a significant amount of contiguous reading when the new stream is accepted, increasing the bandwidth and the read-ahead achieved.

For example, consider video streams of approximately 4 Mbps, a typical value for average (TV) quality. If there are 5 currently accepted video streams and 64 MBytes of server buffer space, each stream would have approximately 12 MBytes of buffer space (or 24 seconds of video). If a new stream is accepted, there would be 10.6 MBytes per stream in steady state (or 20 seconds of video). This amount of data could accumulate from the disk in about 3 seconds, so that steady state is achieved rather quickly. The only time that the server would not have read ahead at least 20 seconds is during the first few slots of reading. With staggered arrival patterns, the server is reading from only one stream immediately after acceptance, and so the disk is substantially ahead after the first disk slot. The steady state will be reached soon enough that none of the borderline cases of buffer space and bandwidth will be encountered. Smoothing the bandwidth usage of each stream is a reasonable course of action, which reduces the resource reservation and potentially permits more simultaneous streams.

4 Network Admission Control Algorithm

Once we have achieved a suitable network bandwidth characterization for each stream, the stream requests are submitted to a network admission control algorithm that determines if there is enough outgoing network bandwidth to support these requests. The network admission control algorithm used in the CMFS is relatively simple. The maximum number of bytes that the network interface can transmit per second is easily converted to the number of blocks per disk slot,[2] which we hereafter refer to as *maxXmit*. The algorithm is shown in Figure 4 and can be summarized as follows: for each network slot, the bandwidth values for each stream are added, and as long as the sum is less than *maxXmit*, the scenario is accepted.

Requests which arrive in the middle of a network slot are adjusted so that the network slot ends for each stream simultaneously. Thus, such a stream has less opportunity to fill the client buffer in that first network slot. In the sample streams this made very little difference in the overall bandwidth required for the network block schedule, although the initial shape did differ somewhat. It did not change the overall distribution of bandwidth.

```
NetworkAdmissionTest( newStream, networkSlotCount )
begin
   for netwSlot = 0 to networkSlotCount do
      sum = 0

      for i = firstConn to lastConn do
         sum = sum + NetBlocks[netwSlot]
         if (sum > maxXmit) then return (REJECT)
      end
      return (ACCEPT)
   end
end
```

Figure 4: Network Admission Control Algorithm

The network admission control algorithm is the same algorithm that was called the “Instantaneous Maximum” disk admission control algorithm in our previous work [7]. This algorithm was rejected in favour of the *vbrSim* algorithm that took advantage of aggressive read-ahead in the future at the guaranteed rate or aggressive read-ahead in the past at the achieved rate. The *vbrSim* algorithm could be considered for network admission control. The smoothing effect enabled by sending data early could further eliminate transient network bandwidth peaks. One major benefit of *vbrSim* is the ability to use the server buffer space to store the data which is read-ahead. This buffer space is shared by all the streams and thus, at any given time, one connection can use several Megabytes, while another may use only a small amount of buffer space. For scenarios with cumulative bandwidth approaching capacity, significant server buffer space is required to enable acceptance.

[2] For disk blocks of 64 KBytes and disk slots of 500 msec, 1 Mbps is approximately 1 Block/slot.

If the same relative amount of buffer space was available at each client, then network send-ahead could be effective. The server model only requires two disk slot's worth of buffer space, and so, very little send-ahead is possible. Even this amount of buffer space is large compared with the minimum required by a decoder. For example, according to the MPEG-2 specifications, space for as few as three or four frames is required.

5 Experimental Design

In order to examine the admission performance of our network admission control algorithm, we loaded a CMFS with several representative VBR video streams on several disks. Each disk contained 11 streams. Then we presented a number of stream request scenarios for streams which were located on the same disk to determine which of the scenarios could be accepted by the vbrSim disk admission control algorithm. The initial selection of the streams for each scenario was done choosing a permutation of streams in such a manner as to have the same number of requests for each stream for each size of scenario. Thus, there were 33 scenarios that contained 7 streams and each of the 11 streams was selected 33*7/11 = 21 times, and 33 scenarios that contained 6 streams. There were also 33 scenarios of 5 streams each and 44 of 4 streams each. When arrival times of the streams were staggered, the streams were requested in order of decreasing playback time to ensure that all streams in a scenario were active at some point in the delivery time. The scenarios for each disk were then combined with similar scenarios from other disks and the network admission control algorithm was used to determine whether or not the entire collection of streams could be accepted by a multi-disk server node. The admission control algorithm was not evaluated in a running CMFS, due to limitations in the measurement techniques employed.

A summary of the stream characteristics utilized in these experiments is given in Table 1. Each disk has a similar mix of streams that range from 40 seconds to 10 minutes with similar averages in variability, stream length, and average bandwidth. The variability measure reported is the coefficient of variation (Standard Deviation/Mean) of the number of blocks/slot.

6 Results

In this section, we compare the results of the Original algorithm with the Smoothed algorithm. The first observation that can be made is that the average bandwidth *reservation* is significantly greater than the average bandwidth *utilization*. When averaged over all scenarios, the Smoothed algorithm reserves significantly less bandwidth than the Original algorithm (113.3 Mbps versus 122.8 Mbps), both of which exceed the bandwidth utilization of 96.5 Mbps. Thus, it is reasonable to expect that the Smoothed algorithm will provide better admission performance results.

	Disk 1	**Disk 2**	**Disk 3**	**Disk 4**
Largest B/W	5.89 Mbps	6.03 Mbps	6.69 Mbps	7.28 Mbps
Smallest B/W	2.16 Mbps	3.33 Mbps	2.9 Mbps	1.71 Mbps
Average B/W	4.16 Mbps	4.89 Mbps	4.61 Mbps	4.61 Mbps
Std. Dev. B/W	1.15 Mbps	0.93 Mbps	1.07 Mbps	1.64 Mbps
Largest Variability	.43	.35	.354	.354
Smallest Variability	.184	.154	.185	.119
Average Variability	.266	.233	.251	.262
Longest Duration	574 secs	462 secs	625 secs	615 secs
Shortest Duration	95 secs	59 secs	52 secs	40 secs
Average Duration	260 secs	253 secs	311 secs	243 secs
Std. Dev of Duration	160 secs	139 secs	188 secs	181 secs

Table 1: Stream Characteristics

We grouped the scenarios with respect to the relative amount of disk bandwidth they request, by adding the average bandwidths of each stream and dividing by the bandwidth achieved on the particular run. The achieved bandwidth is affected by the placement of the blocks on the disk and the amount of contiguous reading that is possible.

In the first experiment, 193 scenarios were presented to a single-node CMFS configured with 4 disks. Each disk had a similar request pattern that issued requests for delivery of all the streams simultaneously. Table 2 gives a summary of admission performance with respect to number of scenarios in each request range that could be accepted by both the network admission algorithm and the disk admission algorithm on each disk, which were fewer than 193.

Pct	**Number of**	**Disk**	**Original**	**Smoothed**
Band	**Scenarios**	**Accepted**	**Accepted**	**Accepted**
95-100	0	0	0	0
90-94	8	0	0	0
85-89	6	0	0	0
80-84	9	3	0	0
75-79	26	7	0	3
70-74	21	15	2	14
65-69	34	33	18	33
60-64	25	25	23	25
55-59	10	10	10	10
50-54	2	2	2	2
Total	141	95	56	87

Table 2: Admission Performance: Simultaneous Arrivals (% of Disk)

The four disks were able to achieve between 110 and 120 Mbps. The scenario with the largest cumulative bandwidth that the Smoothed algorithm could accept was 93 Mbps, as compared with 87.4 Mbps for the Original algorithm. In this set of scenarios, the requested bandwidth varied from approximately 55% to 95% of the achievable disk bandwidth. The original algorithm accepts only a small percentage (2/15) of the scenarios within the 70-74% request range and approximately half the requests in the band immediately below. With the Smoothed algorithm, about half the requests in the 75-79% request range are accepted, and nearly all in the 70-74% range. The Smoothed algorithm increases network utilization by approximately 10 to 15%.

One major benefit of *vbrSim* is the ability to take advantage of read-ahead achieved when the disk bandwidth exceeded the minimum guarantee. This is enhanced when only some of the streams are actively reading off the disk, reducing the relative number of seeks, producing a significant change in admission results. The achieved bandwidth of the disk increases by approximately 10%, with only 9 of the 193 scenarios rejected by the disk system and the network block schedules are slightly different.

Pct Band	**Number of Scenarios**	**Disk Accepted**	**Original Accepted**	**Smoothed Accepted**
95-100	29	22	0	0
90-94	9	7	0	0
85-89	12	12	0	0
80-84	14	14	0	0
75-79	5	5	0	3
70-74	22	22	1	7
65-69	23	23	5	21
60-64	34	34	10	34
55-59	25	25	24	25
50-54	17	17	17	17
45-49	2	2	2	2
Total	193	184	59	103

Table 3: Admission Performance: Staggered Arrivals (% of Disk)

Table 3 shows admission decisions under staggered arrival. The Original algorithm performed significantly worse in terms of percentage of bandwidth requests that are accepted. As mentioned before, many of the scenarios move to a lower percentage request band, due to the increase in achieved bandwidth from the disk. This shows that the increase in disk bandwidth achieved due to stagger was greater than the increase in the amount of accepted network bandwidth. For the Smoothed algorithm, relative acceptance rates are unchanged. The ability to accept streams at the network level and at the disk level have kept up with the increase in achieved bandwidth off the disks.

Another experiment examined the percentage of the network bandwidth that can be accepted. The results of admission for the simultaneous arrivals and the staggered arrivals case are shown in Tables 4 and 5. We see that smoothing is an effective way to enhance the admission performance. A maximum of 80% of the network bandwidth can be accepted by the Original algorithm on simultaneous arrivals, although most of the scenarios in that range are accepted. The smoothing operation allows almost all scenarios below 80% to be accepted, along with a small number with greater bandwidth requests.

Pct Band	**Number of Scenarios**	**Original Accepted**	**Smoothed Accepted**
95-100	0	0	0
90-94	5	0	0
85-89	4	0	2
80-84	18	1	17
75-79	32	19	32
70-74	19	18	19
65-69	11	11	11
60-64	2	2	2
Total	91	51	85

Table 4: Admission Performance: Simultaneous Arrivals (% of Network)

Pct Band	**Number of Scenarios**	**Original Accepted**	**Smoothed Accepted**
95-100	5	0	0
90-94	19	0	2
85-89	15	2	14
80-84	27	3	27
75-79	29	18	29
70-74	22	22	22
65-69	11	11	11
60-64	2	2	2
Total	131	59	106

Table 5: Admission Performance: Staggered Arrivals (% of Network)

In Table 5, we see that the maximum bandwidth range requested and accepted by the disk subsystem approaches 100 Mbps. None of these high bandwidth scenarios are accepted by either network admission algorithm. A few scenarios between 80% and 90% can be accepted with the Original algorithm. The Smoothed algorithm accepts nearly all requests below 90% of the network bandwidth, due to the fact that a smaller number of streams are reading and transmitting the first network slot at the same time. With staggered arrivals, all streams but the most recently accepted stream are sending at smoothed rates, meaning lower peaks for the entire scenario.

The results of these experiments enable an additional aspect of the CMFS design to be evaluated: scalability. It is desirable that the disk and network bandwidth scale together. In the configuration tested, 4 disks (with *minRead* = 23) provided 96 Mbps of guaranteed bandwidth with a network interface of 100 Mbps. At this level of analysis, it would seem a perfect match, but the tests with simultaneous arrivals did not support this conjecture. A system configured with guaranteed cumulative disk bandwidth approximately equal to nominal network bandwidth was unable to accept enough streams at the disk in order to use the network resource fully. There were no scenarios accepted by the disk that requested more than 94% of the network bandwidth. In Table 4, there are only 4 scenarios in the 85-89% request range, that were accepted by the disk system. In Table 5, there were 15 such scenarios. This increase is only due to the staggered arrivals as the same streams were requested in the same order.

With staggered arrivals, the network admission control became the performance limitation, as more of the scenarios were accepted by the disk. There were no scenarios that requested less than 100 Mbps that were rejected by the disk. This arrival pattern would be the common case in the operation of a CMFS. Thus, equating disk bandwidth with network bandwidth is an appropriate design point which maximizes resource usage for moderate bandwidth video streams of short duration if the requests arrive staggered in time.

7 Related Work

The problem of characterizing the network resource requirements of Variable Bit Rate audio/video transmission has been studied extensively. Zhang and Knightly [14] provide a brief taxonomy of the approaches from conservative peak-rate allocation to probabilistic allocation using VBR channels of networks such as ATM.

The empirical envelope is the tightest upper bound on the network utilization for VBR streams, as proven in Knightly et al. [6], but it is computationally expensive. This characterization has inspired other approximations [3] which are less accurate, less expensive to compute, but still provide useful predictions of network traffic.

Traffic shaping has been introduced to reduce the peaks and variability of network utilization for inherently bursty traffic. Graf [3] examines live and stored video and provides traffic descriptors and a traffic shaper based on multiple leaky-buckets. Traffic can be smoothed in an optimal fashion [12], but requires a-priori calculation of the entire stream. If only certain portions of the streams are retrieved (i.e. I-frames only for a fast-motion low B/W MPEG stream delivery), the bandwidth profile of the stream is greatly modified.

Four different methods of smoothing bandwidth are compared by Feng and Rexford [2], with particular cost-performance tradeoffs. The algorithms they used attempt to minimize the number of bandwidth changes, and the variability in network bandwidth, as well as the computation required to construct the schedule. They do not integrate this with particular admission strategies other than peak-rate allocation. This bandwidth smoothing can be utilized in a system that uses either variable bit rate network channels or constant bit rate channels. Recent work in the literature has shifted the focus away from true VBR on the network towards variations of Constant Bit-Rate Transmission [8],[14]. Since the resource requirements vary over time, renegotiation of the bandwidth [4] is needed in most cases to police the network. This method is used by Kamiyama and Li [5] in a Video-On-Demand system. McManus and Ross [8] analyze a system of delivery that prefetches enough of the data stream to allow end-to-end constant bit rate transmission of the remainder without starvation or overflow at the client, but at the expense of substantial latency in start-up. Indications are that minimum buffer utilization can be realized with a latency of between 30 seconds and 1 minute [13]. For short playback times (less than 5 minutes) that may be appropriate for news-on-demand, such a delay would be unacceptable.

8 Conclusions and Further Work

In this paper, we have presented a network bandwidth characterization scheme for Variable Bit Rate continuous media objects which provides a detailed network block schedule indicating the bandwidth needed for each time slot. This schedule can be utilized to police the bandwidth allocated for each network channel via sender-based rate control, or network-based renegotiation.

We observed that the Original algorithm was susceptible to disk bandwidth peaks at the beginning of network slots. The Smoothed algorithm was introduced, taking advantage of client buffer space and excess network bandwidth that must be reserved, for a reduced overall reservation.

The network admission algorithm provides a deterministic guarantee of data transmission, ensuring that no network slot has a cumulative bandwidth peak over the network interface bandwidth. Scenarios with simultaneous arrivals were limited by the disk subsystem. The disk admission control method [7] used in the CMFS, when combined with staggered arrivals, showed that the same disk configuration shifted the bottleneck to the network side. The network admission control algorithm and the smoothed network bandwidth stream characterization combined to provide an environment where scenarios that request up to 90% of the network interface can be supported.

These experiments utilized a single value for the size of the network slot and a single granularity for the block size. Extensions to this work could include comparing admission results with different values for these two parameters.

References

[1] Jill M. Boyce and Robert D. Gaglianello. Packet Loss Effects on MPEG Video Sent Over the Public Internet. In *ACM Multimedia*, Bristol, England, September 1998.

[2] Wu-Chi Feng and Jennifer Rexford. A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video. In *IEEE Infocomm*, pages 58–66, Los Angeles, CA, June 1997.

[3] Marcel Graf. VBR Video over ATM: Reducing Network Requirements through Endsystem Traffic Shaping. In *IEEE Infocomm*, pages 48–57, Los Angeles, CA, June 1997.

[4] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic. In *ACM SIGCOMM*, pages 219–230, Boston, MA, August 1995.

[5] N. Kamiyama and V. Li. Renegotiated CBR Transmission in Interactive Video-on-Demand Systems. In *IEEE Multimedia*, pages 12–19, Ottawa, Canada, June 1997.

[6] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang. Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic. In *ACM SIGMETRICS '95*. ACM, 1995.

[7] D. Makaroff, G. Neufeld, and N. Hutchinson. An Evaluation of VBR Admission Algorithms for Continuous Media File Servers. In *ACM Multimedia*, pages 143–154, Seattle, WA, November 1997.

[8] J. M. McManus and K. W. Ross. Video on Demand over ATM: Constant-Rate Transmission and Transport. In *IEEE InfoComm*, pages 1357–1362, San Francisco, CA, October 1996.

[9] G. Neufeld, D. Makaroff, and N. Hutchinson. Design of a Variable Bit Rate Continuous Media File Server for an ATM Network. In *IST/SPIE Multimedia Computing and Networking*, pages 370–380, San Jose, CA, January 1996.

[10] G. Neufeld, D. Makaroff, and N. Hutchinson. Server-Based Flow Control in a Continuous Media File System. In *6th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 29–35, Zushi, Japan, 1996.

[11] Ranga Ramanujan, Atiq Ahamad, and Ken Thurber. Traffic Control Mechanism to Support Video Multicast Over IP Networks. In *IEEE Multimedia*, pages 85–94, Ottawa, Canada, June 1997.

[12] J. D. Salehi, Z. Zhang, J. F. Kurose, and D. Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. In *ACM SIGMETRICS*, May 1996.

[13] Subrahata Sen, Jayanta Dey, James Kurose, John Stankovic, and Don Towsley. CBR Transmission of VBR Stored Video. In *SPIE Symposium on Voice Video and Data Communications: Multimedia Networks: Security, Displays, Terminals, Gateways*, Dallas, TX, November 1997.

[14] H. Zhang and E. W. Knightly. A New Approach to Support Delay-Sensitive VBR Video in Packet-Switched Networks. In *5th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 381–397, Durham NH, April 1995.