Dynamic Bandwidth Allocation Techniques*

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

Abstract

Performance guarantees for multimedia services can be achieved through proper resource allocation. Resources should be reserved in a manner which provides these guarantees as efficiently as possible. This paper presents an allocation method called the Dynamic Search Algorithm (DSA+). DSA+ is an on-line algorithm that dynamically adjusts the resource allocation, based upon the measured quality of service (QoS). Advantages of DSA+ include efficient use of resources, reasonable implementation cost, and stringent QoS control. In this paper, we show how DSA+ dynamically allocates bandwidth to achieve a given loss rate for MMBP generated traffic and actual MPEG VBR videos. Performance and cost of other allocation methods are compared, as well as the lack of dependence of DSA+ on initial parameter settings. DSA+ allocation for multiplexed traffic and multiple hop connections is also examined.

1 Introduction

As the use of multimedia applications increases, so are the demands for resources required to support them. Network resources such as bandwidth of each physical link, buffer space and processing time at each node, should be allocated in a cost-effective manner. Each application expects the network to provide a desired quality of service (QoS). QoS measurements include bounds on the cell loss probability, cell delay, etc. The service provider is interested in providing the desired QoS, but as efficiently as possible. For these reasons, allocation methods are needed to allocate resources while providing QoS guarantees.

Conventional approaches of resource allocation rely on predetermined traffic characteristics. The amount of resources required to provide the QoS is calculated using these values. These approaches experience the following fundamental problems. First, the source characteristics may not be known ahead of time. In the case of interactive video, the user must guess at these characteristics. Second, parameters may not adequately characterize the source. It has been shown for MPEG-compressed video that long-range dependencies occur, which implies that standard statistical models are probably inadequate [9]. Third, the number of parameters required should be kept small, so to reduce the complexity of the allocation method. A layered n-space Markov Model may adequately characterize a source, nevertheless the computation of allocation amounts may become intractable for real time applications [3].

Current allocation methods can be categorized as either *off-line* or *on-line*. Off-line methods predetermine allocation amounts before transmission begins. Such a method may allocate one resource level (static) for the duration of the application, or may renegotiate the resource level at various times. An example of off-line allocation is peak rate, which is used for most real time applications. This approach has several advantages including simplicity and predictability, but suffers from the problems noted above, as well as low resource utilization if the peak-to-mean ratio is high. Other off-line methods that renegotiate resource levels result in better utilization; on the other hand, they require complete control of the traffic source [4] [7]. For example, the off-line method developed by Feng, et al. determines the minimum number of renegotiations (increases or decreases in resource allocation) required for the playback of a previously-stored MPEG video [7]. The

*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

video is then transmitted at the calculated rates, so to prevent buffer overflow and underflow. For interactive applications, where the traffic source is not known nor directly controllable, these methods are not suitable.

On-line methods periodically renegotiate resource allocation based upon predicted traffic behavior [1] [5] [10] [16] [17] [19] [21]. Predictions are derived from measurements of the traffic and/or QoS observations. Such methods do not have the problems associated with off-line methods and may be implemented by filters [1], neural networks [5], dynamic algorithms [10] [16] or some other sampling procedure [17] [19] [21]. These methods also have the advantage of adjusting the resource allocation with respect to a desired QoS. To date no on-line method has the ability to tightly control the QoS for such difficult applications as transmission of compressed video. In addition, most methods suffer from a large number of renegotiations, and/or rely on a very complex measurement and allocation algorithm.

In this paper we present an on-line renegotiation method called the Dynamic Search Algorithm (DSA+) [8]. Dynamic algorithms (also referred to as adaptive algorithms) have been applied in various applications such as system identification, filtering and pattern recognition [2]. DSA+ encapsulates the general dynamic algorithm form with some new features to better handle non-stationary sources. DSA+ allocates resources so to meet a desired QoS. In this paper DSA+ is used to control the loss rate of simulated traffic or actual MPEG VBR videos by the appropriate allocation of bandwidth. Other off-line and on-line algorithms were also implemented and their performance is compared and contrasted. We are interested in efficient resource allocation, few renegotiations and easy implementation. Since ease of algorithm use is important, the robustness of DSA+ with respect to the initial parameter settings is demonstrated. Finally, DSA+ application for connection admission control and multiple hop allocation is investigated as well as some open questions concerning dynamic bandwidth allocation.

2 A New Method

2.1 System Model

DSA+ dynamically renegotiates the server rate (bandwidth) of a queue in order to meet a desired QoS. Cells, a fixed-length unit of traffic storage and transmission, arrive at a finite capacity queue and are serviced in a FIFO manner. Any cell arriving at a full queue is immediately lost. 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 and 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

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

The CLP desired by the user is denoted Q_l. The goal of DSA+ is to adjust the server rate, so to provide the desired CLP, Q_l, as efficiently as possible with few renegotiations. Secondary goals are simplicity of implementation and robustness.

2.2 An Algorithm for Dynamic Resource Allocation

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

$$\mu_{n+1} \leftarrow \mu_n + \frac{K}{\alpha} \times \ln\left(\frac{P_n}{Q_l}\right), \alpha = \begin{cases} 1 & if \quad P_n > Q_l \\ 2 & if \quad P_n \leq Q_l \end{cases} \tag{1}$$

This dynamic algorithm updates the server rate, μ_n, based on the observed CLP during the nth interval. This measurement along with the desired CLP value, Q_l, are then used in the error function $\ln(P_n/Q_l)$. This non-linear error function provides either a positive or negative feedback value based on the observation taken during the most recent interval. Note the feedback becomes smaller as the measured CLP approaches

```
1   curr_error ← ln(P_n/Q_l)
2   prev_error ← ln(P_{n-1}/Q_l)
3   if((P_{0...n} > Q_l) AND (P_n ≤ Q_l))then
4       U_{n+1} ← 2 × U_n
5   else
6       if (curr_error × prev_error ≤ 0)then
7           U_{n+1} ← 2 × U_n
8       endif
9       if(P_n > Q_l)then
10          μ_{n+1} ← μ_n + K × ln (P_n/Q_l)
11      else
12          μ_{n+1} ← μ_n + K/2 × ln (P_n/Q_l)
13      endif
14  endif
```

Figure 1: DSA+ algorithm. Numbers on the far left are for reference only.

the desired value, keeping the rate at a more stable value. The error function is also appropriate due to the very small loss rates that are normally desired. The constant K amplifies the response of the error function and this product ultimately determines how much the server rate can be increased or decreased. Parameter α allows the rate to be increased twice as fast as it can be decreased. This is done since, in an actual network, resources are more easily reduced than increased. Varying the gain is also beneficial if the source is non-stationary [2].

Figure 1 shows the complete algorithm at one renegotiation instant t_n. As seen in the figure, the renegotiation interval is lengthened (doubled) in two cases (lines 4 and 7). The interval is doubled on line 4 if the cumulative CLP ($P_{0...n}$) is worse than required and if the CLP during the most recent interval (P_n) is better than required. This will reduce the cumulative CLP towards the desired value, since the nth CLP is better than the desired CLP. The interval is doubled on line 7 when P_n and P_{n-1} are on different sides of (one greater than, the other less than) the desired CLP. Doubling the interval length also reduces the number of renegotiations required over time, a unique feature to both DSA+ and REQS [16].

As mentioned in the introduction, traffic sources such as MPEG-compressed video are complex due to their non-stationary behavior and long range dependencies. A potential problem with the algorithm as shown is that the traffic characteristics may change drastically during a renegotiation interval, while the server rate cannot be renegotiated. This can lead to excessive QoS violations. To reduce the severity of this problem, we introduce the use of interrupts. At fixed-length sub-intervals, called an interrupt interval I, an interrupt is generated if both P_n and $P_{0...n}$ are greater than the desired loss rate Q_l. The relationship between update and interrupt intervals can be seen in figure 2. In this case, the server rate is increased immediately according to equation 1, rather than waiting until the end of the renegotiation interval. The renegotiation interval itself, however, is not changed by an interrupt. The use of interrupts allows DSA+ to be more responsive to sudden, severe traffic changes, which should occur infrequently. This is a unique and key element of DSA+.

Initially the user must assign the following values: initial renegotiation interval (U_0), interrupt sub-interval (I), constant (K) and the initial server rate (μ_0). U_0, I, and K may depend on the source traffic, but their selection primarily impacts the number of renegotiations and the efficiency of the allocation. Initial variable selection is addressed later in this paper.

3 Numerical Results

In this section the performance of DSA+ and other allocation techniques is investigated using simulated traffic and actual MPEG-compressed traffic. For each experiment the system described in section 2.1 was

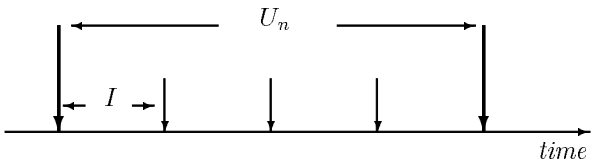

Figure 2: Time axis for updates and interrupts.

simulated. The desired QoS was a CLP of 1×10^{-3} and the queue capacity was 80 ATM cells (48 byte payload) [16]. While the targeted QoS was cell loss probability, the queue size was selected to provide a minimum cell delay as well. The minimum allowed bandwidth was 1 Mbps, therefore the maximum delay for any cell is 34 msec. This value was selected to provide a wide range of available bandwidths, however almost any combination of minimum bandwidth and queue size could have been chosen. A more restrictive bandwidth selection will only improve the performance of the algorithm.

We are interested in efficiently managing bandwidth, therefore two metrics are used. First, the number of renegotiations required for the entire simulation. This value is important since a large number of renegotiations will cause considerable strain on the signaling system of the network. Second, the number of bits reserved to transmit the video, or equivalently, the area under the allocation curve for the duration of the video. This measurement is important because we wish to transmit the video with as few bits as possible while maintaining the desired QoS. Minimizing the bits used can help increase the utilization of the network by providing more resources to other users.

3.1 MMBP Sources

A two state Markov Modulated Bernoulli Process (MMBP) was selected for simulated traffic. This model was chosen because it typifies burstiness and the correlation of interarrival times, two important characteristics of ATM traffic. For these experiments, the performance of DSA+ is compared to effective bandwidth and peak rate allocation. The effective bandwidth was calculated using the method described in [6], which was presented for MMPP sources. For this paper the method was extended for MMBP sources using techniques described in [11]. The effective bandwidth is an off-line calculation that yields the smallest rate which ensures a certain CLP. While the effective bandwidth technique requires complete source information and no renegotiations, its is presented here only as a possible lower bound allocation amount. This is certainly true if there are no renegotiations. Since DSA+ attempts to minimize the number of renegotiations, we expect its allocation to approach the effective bandwidth. The peak rate allocation is also given to provide an upper bound on the allocation amount. A dynamic technique should allocate less bandwidth than peak rate since peak rate will result in a zero losses. Consequently, the allocation amount of an efficient on-line method should fall below the upper bound and close to the lower bound.

The two state MMBP model is seen in figure 3, if the current state is $S0$ the probability of remaining there is p and the probability of changing state is $1-p$. If the current state is $S1$ the probability of remaining there is q and the probability of changing state is $1-q$. The cell arrival rates of state $S0$ and $S1$ are λ_0 and λ_1 respectively. More details about the MMBP model are presented in [14].

The parameters of the MMBP model were adjusted to vary two burstiness measurements. First, the rate of state $S0$ was increased to magnify the squared coefficient of variation, C^2. The parameter settings and resulting C^2 values for these MMBP sources are given in table 1. Second, the duration of state $S0$ was lengthen to magnify the peak to mean ratio. The parameter setting for these MMBP sources are presented in table 2. For each experiment, separate and independent simulations were executed to provide averages and 95% confidence intervals. Experiments ran for 6000 simulated seconds and each data point represents 100 simulations. The initial DSA+ parameters for all the MMBP experiments were; 40 cells/second for K, and 0.5 seconds for both U_0 and I and the peak rate of the source for μ_0.

The following experiments used the MMBP sources described in table 1 and were performed to show the

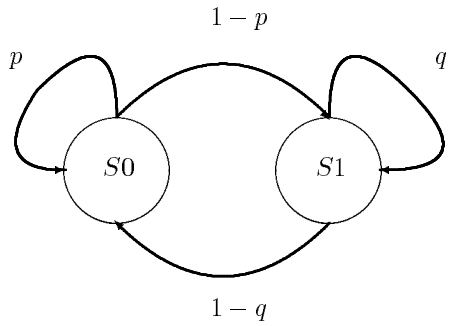

Figure 3: Two state MMBP.

Source	$S0$ State		$S1$ State		C^2
	Mean Rate (cells/second)	Duration (second)	Mean Rate (cells/second)	Duration (second)	
0	100	0.02	1×10^4	0.01	30
1	100	0.02	2×10^4	0.01	59
2	100	0.02	4×10^4	0.01	118
3	100	0.02	7×10^4	0.01	207

Table 1: MMBP parameter values for the C^2 experiments.

effect of an increasing C^2 value (30-207). Table 3 shows the results of the C^2 experiments and figure 4 shows the average bandwidth allocation and cumulative CLP graphs for source 1. As seen in the table and graphs, the peak rate reserves only 5% more bits for transmission than the effective bandwidth or DSA+. The DSA+ allocation for each experiment had slightly larger confidence intervals at the beginning, since the algorithm is searching for the appropriate value. As the algorithm approached an allocation level which provided the desired QoS, the allocated amount stabilized and the confidence intervals reduced in size. This was evident for all the burstiness experiments. Due to the small difference in allocation amounts, peak rate allocation can be considered an efficient method for these sources, only if its values is known in advance. DSA+ was able to closely match the allocation amount and QoS provided by the effective bandwidth method with no a priori information about the source.

The following experiments used the MMBP sources described in table 2 and were performed to show the effect of an increasing peak to mean value (16-20). Table 4 shows the results of peak to mean experiments and figure 5 shows the average bandwidth allocation and cumulative CLP graphs for source 5. DSA+ was able to provide the desired QoS for each experiment. The number of bits reserved for transmission by DSA+ is consistently higher than the effective bandwidth allocated amount (an average of 51% more that effective bandwidth). This set of experiments shows that DSA+ is moderately affected by extreme peak to mean ratios. Although these amounts are higher than effective bandwidth, there is still a significant savings over peak rate allocation (an average of 88% less than peak rate allocation). Both sets of experiments indicated that DSA+ was able to efficiently manage the stationary traffic of various MMBP sources. The performance of DSA+ to non-station traffic is presented in the next section.

3.2 MPEG Traffic

In this section the performance of DSA+ is investigated using fifteen MPEG-compressed videos. All traces were obtained from Oliver Rose at the University of Würzburg, Germany [18][1]. Each trace is a thirty minute

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

Source	$S0$ State		$S1$ State		Peak/Mean
	Mean Rate (cells/second)	Duration (second)	Mean Rate (cells/second)	Duration (second)	
0	100	0.04	2.5×10^3	0.01	16
1	100	0.06	2.5×10^3	0.01	18
2	100	0.08	2.5×10^3	0.01	19
3	100	0.1	2.5×10^3	0.01	20

Table 2: MMBP paramter values for peak to mean experiments.

Source	DSA+	Effective Bandwidth	Peak Rate
	Avg. Bits Used ($\times 10^{10}$ bits)	Bits Used ($\times 10^{10}$ bits)	Bits Used ($\times 10^{10}$ bits)
0	2.26	2.27	2.54
1	4.78	4.80	5.09
2	9.81	9.89	10.2
3	17.2	17.5	17.8

Table 3: Allocation comparison for varying C^2 MMBP traffic.

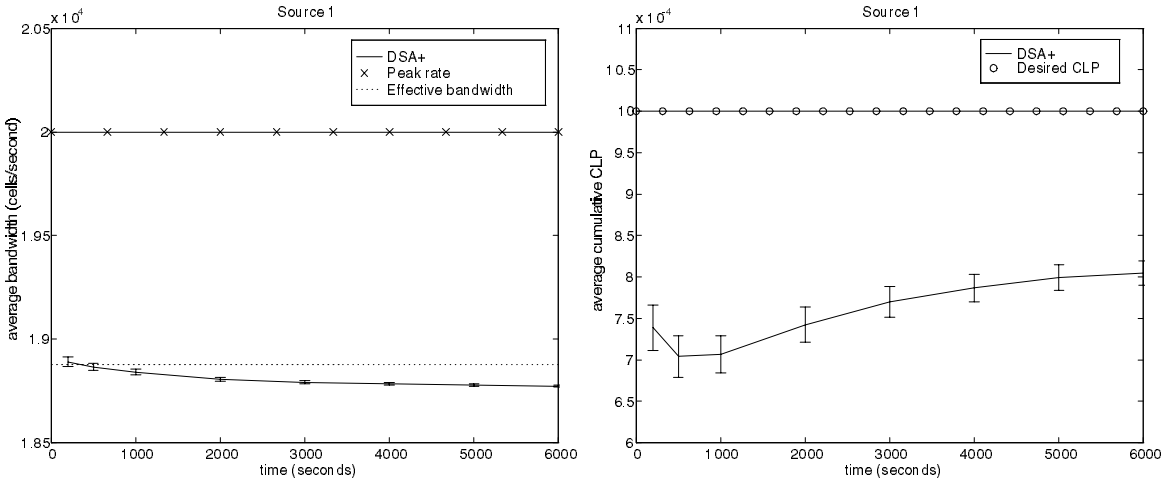

Figure 4: Average bandwidth allocation and cumulative CLP for MMBP source 1.

Source	DSA+	Effective Bandwidth	Peak Rate
	Avg. Bits Used ($\times 10^8$ bits)	Bits Used ($\times 10^8$ bits)	Bits Used ($\times 10^8$ bits)
4	8.22	4.38	63.6
5	7.84	3.79	63.6
6	7.21	3.49	63.6
7	7.17	3.31	63.6

Table 4: Allocation comparison for varying peak to mean MMBP traffic.

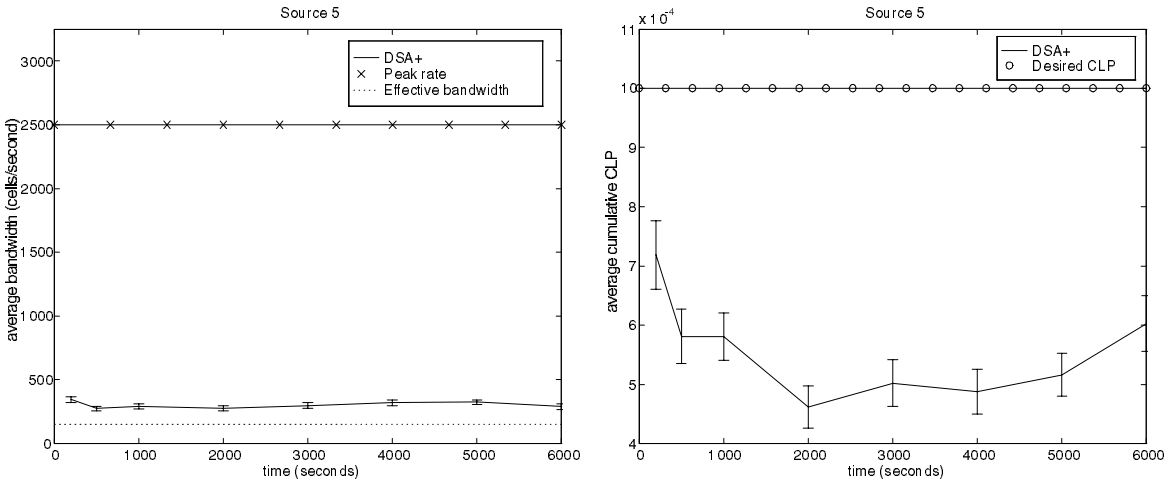

Figure 5: Average bandwidth allocation and cumulative CLP for MMBP source 5.

μ_0 (Mbps)	K (Kbps)	U_0 (second)	I (second)
1000	100	4	0.5

Table 5: Initial settings of DSA+ parameters for MPEG-compressed videos.

segment of the original video and each was encoded with constant quality using the same MPEG-1 encoder card. Relevant statistics of each video are presented in [8] and [18]. As reported in [18], the Hurst parameters indicate all videos exhibit long-range dependency, and significant peak-to-mean ratios ranging from 18.4 to 4.63 based on average frames. Therefore it is evident that these are very difficult sources to regulate, and to date there has been no successful attempt to efficiently manage them on-line.

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 "hard-case" test of any on-line allocation technique.

As an example of the performance of DSA+, figure 6 shows the bandwidth allocation and cumulative CLP for the Simpsons video. In this experiment, the initial parameter settings are given in table 5. As seen in the figure, DSA+ quickly reduces the bandwidth allocated, until the cumulative CLP is approximately the desired value. Afterwards, when the measured CLP was worse than the desired value, the algorithm increased the rate with interrupts. The cumulative CLP graph shows that the algorithm is able to tightly control the bandwidth for the desired CLP. A total of 36 renegotiations were required, with approximately half occurring in the first 60 seconds. This is due to the high initial server rate; improvements can be obtained if the initial rate is less than the peak. Only sixteen of the renegotiations were for more bandwidth. This low number of renegotiations is due to doubling the update interval, as described in section 2.2.

4 Comparison with Other Methods

In this section DSA+ is compared to other allocation techniques: peak rate, Hsu's algorithm, and RED-VBR. The system described in section 2.1 was implemented and all fifteen MPEG videos were used as traffic sources. Again individual frames were split into ATM cells as described in the previous section. No smoothing, multiplexing, filtering or quantization changes of any kind were made to the videos. DSA+ initial parameters, the same for each video, are given in table 5.

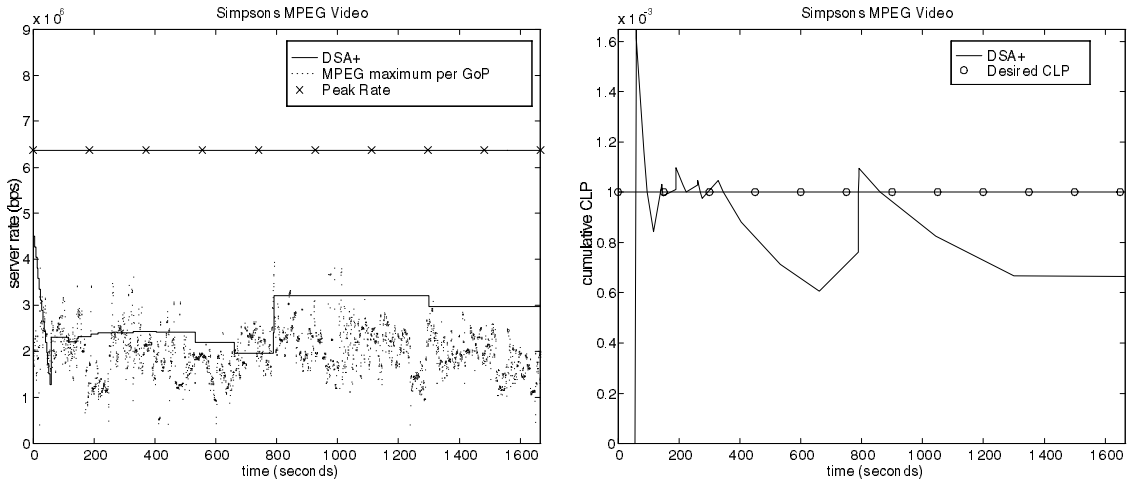

Figure 6: DSA+ bandwidth allocation and cumulative CLP for the Simpsons video.

Peak rate allocation was chosen since it is an accepted allocation method. To determine the exact peak rate requires the trace in advance. For that reason, this is an off-line method. In a sense this comparison is unfair to the remaining on-line methods. An on-line peak rate algorithm would require an overestimation of the traffic by a significant percentage to be cautious. Another difference is that peak rate allocation would result in zero losses, while other methods were targeted for a loss rate of 1×10^{-3}. Small but non-zero losses are considered to be acceptable for typical multimedia applications. Instead of a weakness, we consider the ability to manage QoS targets based upon the user's needs to be a strength of any on-line algorithm. Other off-line methods, such as PCBR [4] or Feng's algorithm [7], are not comparable, as they directly control the transmission of the source.

Hsu's method is a dynamic algorithm which has been proven to find the minimum bandwidth required for a stationary MMBP source [10]. This method was chosen since it is a simple on-line method that requires minimal source information. The algorithm renegotiates bandwidth over fixed length intervals, using previous loss measurements and a simple difference error function. Initial parameters for this algorithm were 4.5 Mbps for the initial server rate, 1 second for the interval and 1 Mbps for c the constant. No method of parameter selection was presented in the original paper. The values used were found to be the best from our experiments. It was also observed that the algorithm was very sensitive to parameter values. Small variations in the initial parameter values did result in over-allocation.

RED-VBR is a method for supporting VBR video, with an off-line or on-line allocation technique. For this comparison, the on-line version was implemented using a similar segmentation algorithm as presented in [21]. RED-VBR is based upon the D-BIND model [12]. This model consists of a set of rate-interval pairs, which characterize the source over various interval lengths. The allocation algorithm stores the currently reserved D-BIND parameters and calculates the D-BIND parameters for the last M frames. A renegotiation take place when a difference exists between the reserved and measured D-BIND parameters; more details are presented in [21]. RED-VBR does not use nor measure the QoS for allocation. QoS is an issue when sources are multiplexed together and is provided on a "per-segment" basis as described in [21]. As a comparison, only the renegotiation and allocation performance of this method will be considered. The initial parameters are given in table 6. The α and β values were taken from the original paper, while MIN_REGEN_INTERVAL, P and M were selected to reduce the number of renegotiations.

Table 7 shows the performance of all the algorithms for each individual MPEG video as a source. Figure 7 shows the bandwidth allocation and cumulative CLP of all the methods for the Asterix video.

DSA+ was able to provide the desired QoS for each video, with significantly fewer bits than the peak rate. Saving of 21 - 61% were observed over peak rate. The average number of renegotiations required was 36.2 and only 44% of the renegotiations were requests for more bandwidth. On average, increases were 189 Kbps.

Hsu's algorithm was not able to provide the desired QoS for the Goldfinger, News and Lambs videos.

α	β	MIN_RENEG_INTERVAL (seconds)	P	M
1.2	1.5	10	48	48

Table 6: RED-VBR parameters.

Video	DSA+		Hsu's		RED-VBR		Peak
	N.R.	Bits Used ($\times 10^9$ bits)	N.R.	Bits Used ($\times 10^9$ bits)	N.R.	Bits Used ($\times 10^9$ bits)	Bits Used ($\times 10^9$ bits)
Asterix	30	3.63	1666	3.50	305	5.77	6.51
ATP Tennis	30	5.68	1666	5.33	308	5.89	8.43
Formula 1 Race	25	5.87	1666	8.72	293	6.11	8.95
Goldfinger	47	5.11	1666	4.81	269	5.58	10.8
Jurassic Park	39	3.12	1666	2.75	296	3.89	5.28
Movie Review	35	4.38	1666	3.79	315	5.13	7.63
Mr. Bean	52	3.89	1666	13.1	269	5.03	10.1
MTV	44	6.72	1666	5.00	283	5.86	10.1
News	28	4.73	1313	9.73	220	4.64	8.60
Lambs	46	3.10	1666	2.46	279	2.82	5.93
Simpsons	36	4.56	1666	13.7	296	5.64	10.2
Soccer	25	6.53	1666	5.56	307	6.29	8.28
Super Bowl	34	4.25	1666	13.1	268	4.79	6.21
Talk	37	2.44	1666	14.6	261	3.97	4.73
Terminator	38	1.75	1666	1.44	293	2.86	3.53
Average	36.5	4.38	1666	7.17	284	4.95	7.72
Legend: **N.R.** = number of renegotiations							

Table 7: Algorithm comparison.

This method also over-allocated bandwidth (more than the actual peak) for the Formula 1 Race, Mr. Bean, News, Simpsons, Super Bowl and Talk videos. This was a result of a over-allocation early in the trace, from which the algorithm was unable to reduce the bandwidth quickly enough. Placing bounds on the highest bandwidth allocated (peak) reduces this effect, but it requires the knowledge of the value *a priori*. Another difficulty with this method was the number of renegotiations. The algorithm uses constant intervals to renegotiate the bandwidth. Consequently, renegotiating every second would place a significant burden on the network's signaling system.

RED-VBR was able to provide the desired QoS for each video, with CLP values ranging from zero to 2×10^{-4}. Fewer bits than peak allocation (11 - 52% less) were used, but the algorithm required a large number of renegotiations. On average 284 renegotiations were performed, with 56% being for more bandwidth. As seen in figure 7, these increases were large, averaging 575 Kbps. The calculation of D-BIND parameters may also be problematic since it is done for each frame.

Overall DSA+ performed better than the other algorithms. It always required fewer bits for transmission than the peak, and on average less than the other on-line methods, while still providing the desired CLP. The significant savings was in the number of renegotiations. The algorithm required no more than 52 renegotiations and on average only 44% were for more resources. The number can be further reduced with a lower initial bandwidth value, as discussed in the next section. On average Hsu's algorithm required 47 times more renegotiations, while RED-VBR required 8 times as many. The magnitude of increases were relatively small, 189 Kbps, while RED-VBR increased three times as much. DSA+ also has the advantage of a simple algorithm that does not require large amounts of processing time.

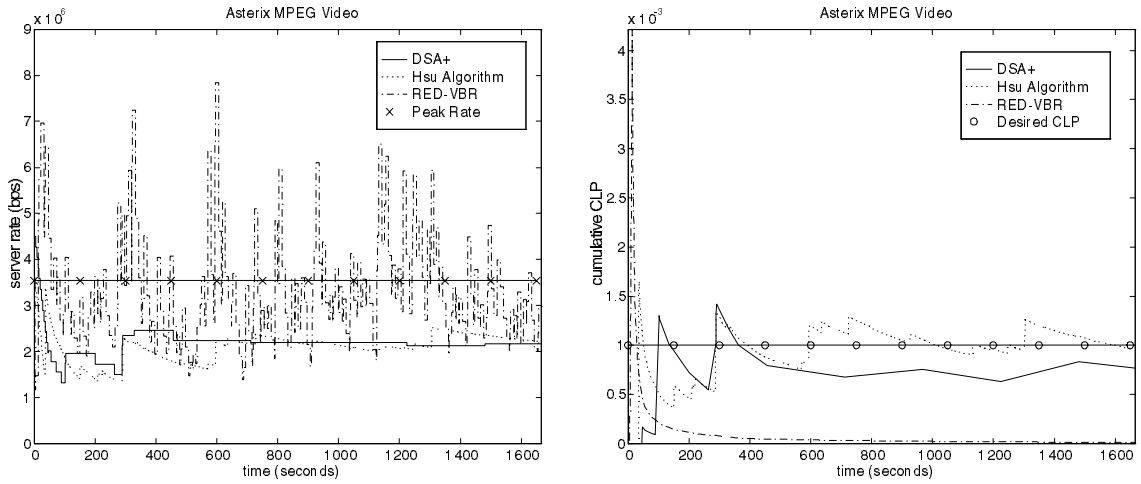

Figure 7: Bandwidth allocation and cumulative CLP for the Asterix video.

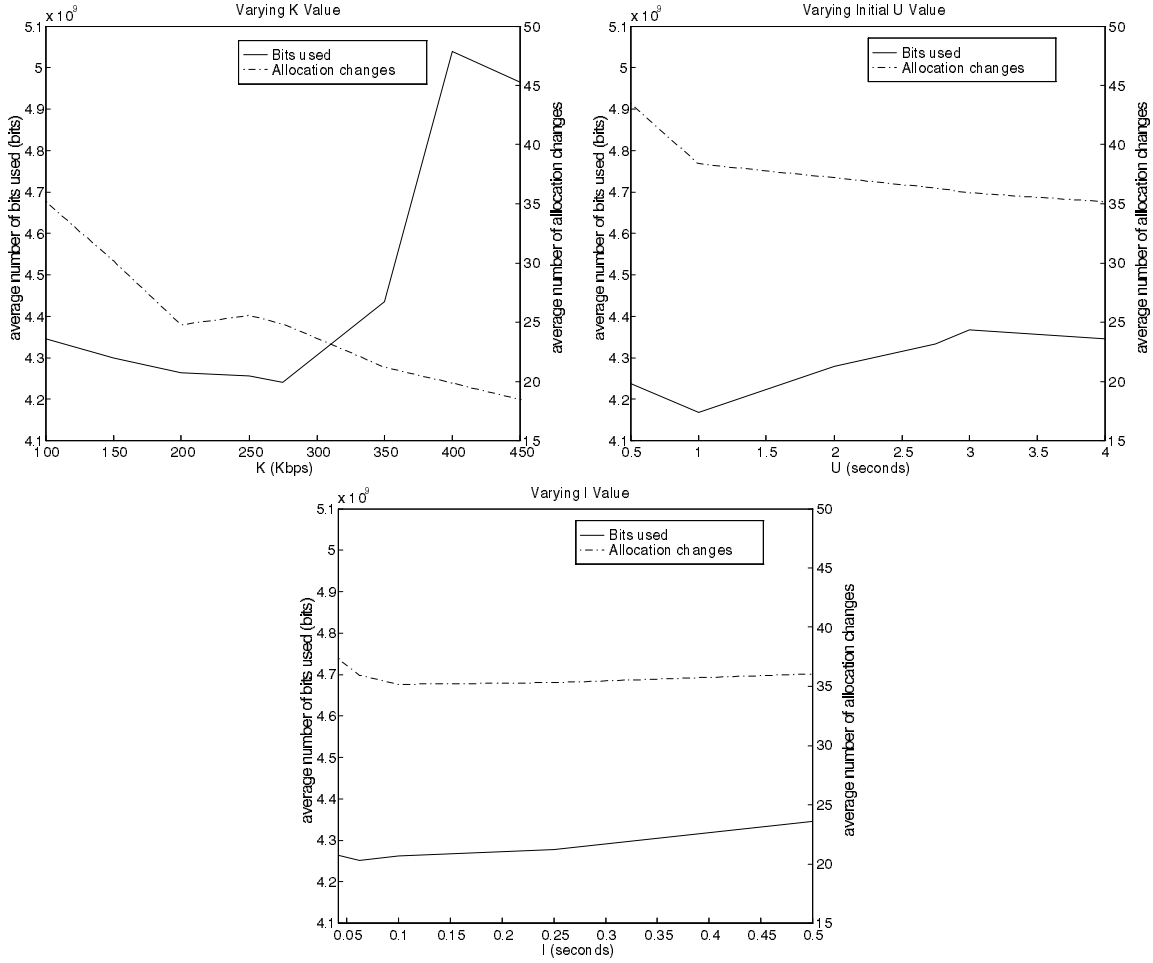

Figure 8: Varying initial parameters K, U_0 and I independently.

5 Robustness, Appropriate Parameter Selection

As described in section 2.2, four initial parameters must be specified in order to use DSA+: the initial server rate (μ_0), the first renegotiation interval (U_0), the rate adjustment coefficient (K), and the interrupt subinterval (I). A capable dynamic allocation method should be relatively insensitive to the initial parameter values, so that user insight into these parameters is not a requirement. We ran four experiments to investigate whether this was the case. For each experiment, we set the parameters to a particular value and simulated the behavior of DSA+, for all 15 videos. Then, the average number of bits and number of renegotiations per video were calculated. The system described in section 3 was simulated and the desired CLP for each experiment was 1×10^{-3}. We are interested in minimizing the number of renegotiations as well as the amount of bandwidth allocated. Yet, improving one value may result in a negative effect on the other. For example, a higher number of renegotiations can result in a more efficient allocation, since the algorithm can closely follow the arrival pattern of the source.

Figure 8 shows the effect of varying the initial parameters individually. It was observed that the coefficient K, did not significantly impact the number of renegotiations, but values over 350 Kbps did increase the number of bits used. In general, for MPEG videos smaller K values (100 - 350 Kbps) are better, since the additional number of renegotiations is not significant compared to the savings in bandwidth. The initial server rate does effect both the number of renegotiations and the bits used. Small initial server rates (less than half peak rate) have fewer renegotiations, but allocate more bits. This is a result of interrupts occurring early in the trace. Large initial server rates (more than 75% peak rate) cause more renegotiations but fewer bits used. The higher number of renegotiations is the effect of reducing the rate more slowly than it can be increased (see equation 1). Generally, larger initial bandwidths are better due to the savings in bits used. Like any renegotiation method however, the better the initial rate guess, the better the performance. The intervals, U_0 and I, have some effect on performance; however their values should be set in accordance to the desired CLP. For example, a more stringent CLP would need a smaller interrupt interval to prevent excessive losses during an interval. Overall DSA+ is robust to initial parameter settings. It can accept a variety of values and still provide the desired QoS. Like any method of resource allocation, *a priori* information can help guide the initial parameter selection; however it is not a necessity.

6 Multiplexing

In this section the effects of multiplexed MPEG videos is examined. The motivation for this experiment is the use of DSA+ for connection admission control. An admission control technique could simply use DSA+ to predict bandwidth usage of the current sessions. Subtracting this value from the total amount would provide a quick and accurate measure of the remaining resources available. For this experiment, DSA+ was used to manage a multiplex stream of fifteen different MPEG videos. The QoS provided to the multiplexed stream is an aggregate value, thus individual guarantees are not provided. The performance was then compared to the summation of bandwidth from controlling each video independently, as in section 2.3. Each multiplexed video was randomly started at frame x, where x was uniformly distributed between the start and end frame. The frames of each video were then added together to create one multiplexed video stream. This multiplexed stream had a peak rate of 17.2 Mbps and mean rate of 7.61 Mbps, yielding a peak to mean ratio of 2.26. The stream was then broken into ATM cells in the same manner as described in section 3. The cells then arrived at a FIFO queue (80 cell capacity), where any cell encountering a full queue was immediately lost. The desired CLP was 1×10^{-3}. Multiplexing is expected to reduce burstiness and LRD behavior [15]; thus a multiplexed source should be easier to manage.

Figure 9 shows the allocation and cumulative CLP of the DSA+ managed multiplexed stream and the summed individual DSA+ managed streams. As noted in the figure, significant savings occurs from managing the multiplexed stream. Savings of 15% over peak and 66% over the summed individual streams were observed. The primary source of savings (over controlling each video individually) is from multiplexing. Multiplexing smooths the stream, reducing burstiness and LRD. The result is a source that is "well-behaved" as compared to the individual videos. The number of renegotiations for the multiplexed stream is 67 as compared to 546 total renegotiations for the individual videos. Over half of the renegotiations for the multiplexed stream occurred in the first two minutes.

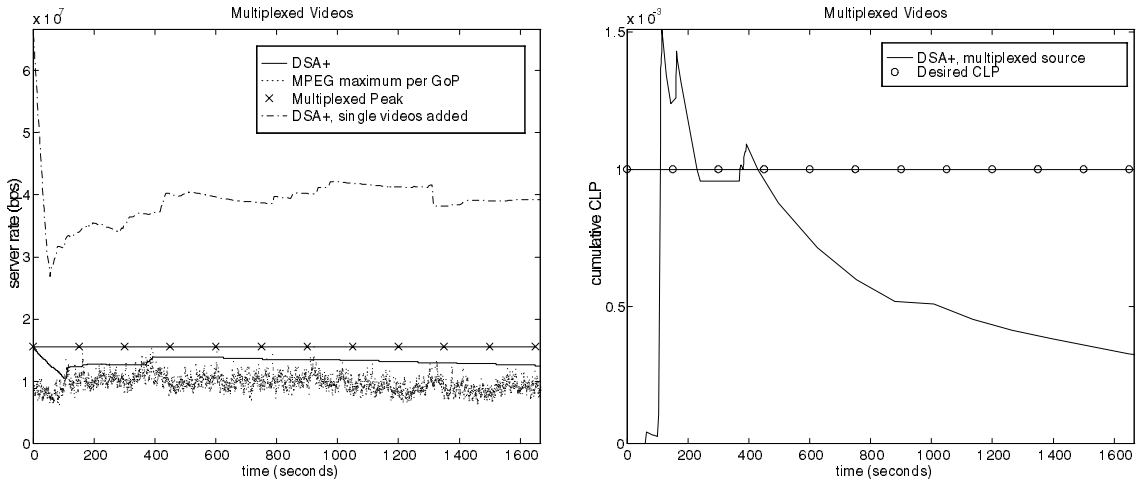

Figure 9: Bandwidth allocation and cumulative CLP for the multiplexed stream.

7 Network Allocation

We have only applied DSA+ for controlling the QoS of a single hop. In this section we investigate the application of DSA+ for a multiple hop connection. For these experiments four nodes are connected in series. The nodes are interconnected with 155 Mbps links, each measuring 50 meters in length. Each node consists of an adjustable rate server and finite capacity FIFO queue (80 ATM cells) as described in section 2. Each MPEG-video was segmented as described in section 3. The stream entered the network at node zero and proceeded forward until node three was reached. For these experiments we are interested in providing an end-to-end cell lost probability of 1×10^{-3}. Two implementations of DSA+ were investigated: *each-node* and *first-node*.

The each-node implementation requires each node to run DSA+ separately and independently, as seen in figure 10. The end-to-end CLP was divided evenly among the nodes resulting in a target CLP of 2.5×10^{-4} per node. If each node provides this CLP, the end-to-end CLP would be the desired 1×10^{-3}. The end-to-end QoS could have been divided differently, perhaps based on the current condition of the individual nodes. The remaining DSA+ initial parameters were identical for each node and are given in table 5. One primary advantage to the strategy is that no inter-node algorithm communication is necessary, thus eliminating any need for algorithm control packets. First-node implementation only requires the first node of the connection to run DSA+, as seen in figure 11. The initial DSA+ parameters are given in table 5. The first node controls the bandwidth for all the remaining downstream nodes. The first node has a CLP of 1×10^{-3}, therefore the remaining nodes can have zero losses. When a renegotiation occurs at the first node a control packet, containing the new bandwidth value, is sent downstream. Once a downstream nodes receives the bandwidth control packet it must immediately renegotiate to this value then forward it downstream. We assumed that the control packets are sent on another reliable connection, as done in many communication protocols [20]. Only transmission and propagation delays were factored for the control packets.

Table 8 shows the total number of bits (summation of the bits reserved for the four nodes) reserved by each method. Figures 12 and 13 show the allocation and observed CLP for the Talk video of each-node and first-node respectively. For the first-node method, downstream nodes required less bandwidth. This was evident for all the each-node experiments performed. This is primarily due to a reshaping effect each node has on the traffic. As the traffic passes through a node some fluctuations in the arrival stream are removed due to the storage and transmission, resulting in a less bursty departure stream. Downstream nodes benefit from this effect, resulting in a lower bandwidth allocation (1 - 47% less than the first node). However, the first node implementation consistently reserved fewer total bits, as seen in table 8; yet this implementation requires the overhead of inter-node algorithm communication.

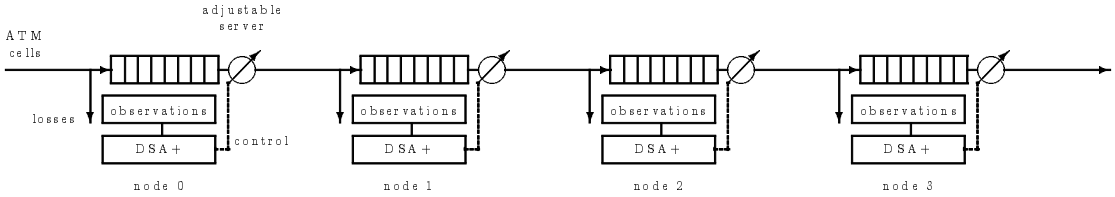

Figure 10: Multiple hop connection with each-node implementation.

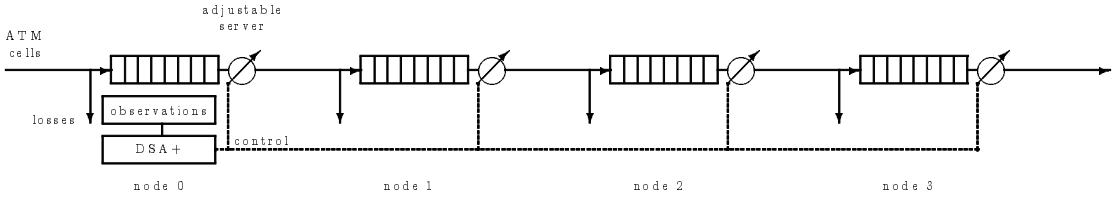

Figure 11: Multiple hop connection with first-node implementation.

Either implementation of DSA+ for end-to-end QoS showed promising results. The each-node arrangement provided the end-to-end QoS with no inter-node algorithm communication overhead, yet there are a few disadvantages. One disadvantage is that this method requires more bits as seen in the table. This is primarily due to the more stringent QoS required at each node, especially the first node. Another disadvantage is dividing the QoS among the individual nodes. This may be problematic if either there is a large number of nodes and/or if the end-to-end QoS is very stringent. For example, if the end-to-end CLP was 1×10^{-6} in for a 10 node connection, each node would have an individual CLP of 1×10^{-7}. This individual CLP value may be too small for any type of on-line method. One possible solution could include the use of importance sampling or restart [13]. First-node implementation does not have the disadvantage of individual QoS requirements, but it does require more support for inter-node algorithm communication. It is possible that both methods could be combined to lessen the effects of both disadvantages.

8 Conclusions

This paper presented an on-line algorithm, DSA+, which efficiently allocates resources to provide a required QoS. DSA+ was used to manage the bandwidth of MMBP generated traffic and MPEG-compressed video traces with a specified allowable cell loss probability. We were interested in minimizing both the bandwidth allocated and the number of renegotiations. MMBP traffic experiments showed that DSA+ efficiently allocated bandwidth close to the predicted effective bandwidth value, without prior knowledge of statistics of the underlying traffic generation process. For the MPEG experiments, fifteen actual MPEG traces were collected and used. As compared to an off-line peak-rate allocation, DSA+ saved 13–58% in bandwidth. On average 36 renegotiations were required, but only 44% were for more bandwidth, which seems acceptably low. Other methods which were compared, either over-allocated bandwidth or required up to 47 times more renegotiations. The effect of multiplexing was investigated and showed DSA+ has no problem guaranteeing QoS to such a traffic source. The inclusion of DSA+ in connection admission control was also discussed.

Our experiments also indicate the algorithm is fairly insensitive to the choice of initial parameter values. For all the experiments performed the same initial parameters were used and showed excellent results. DSA+ requires limited information about the source, however any a priori information can, of course, benefit the performance of any on-line algorithm.

Video	Each-node	First-node
	Total Bits Used ($\times 10^{10}$ bits)	Total Bits Used ($\times 10^{10}$ bits)
Asterix	1.54	1.35
ATP Tennis	2.03	1.58
Formula 1 Race	2.31	2.06
Goldfinger	1.73	1.28
Jurassic Park	1.12	0.86
Movie Review	1.56	1.15
Mr. Bean	1.31	1.07
MTV	2.21	1.78
News	1.51	1.17
Lambs	1.14	0.78
Simpsons	1.56	1.27
Soccer	2.11	1.64
Super Bowl	1.39	1.06
Talk	0.85	0.71
Terminator	0.76	0.66
Average	1.54	1.23

Table 8: Multiple-hop allocation comparison.

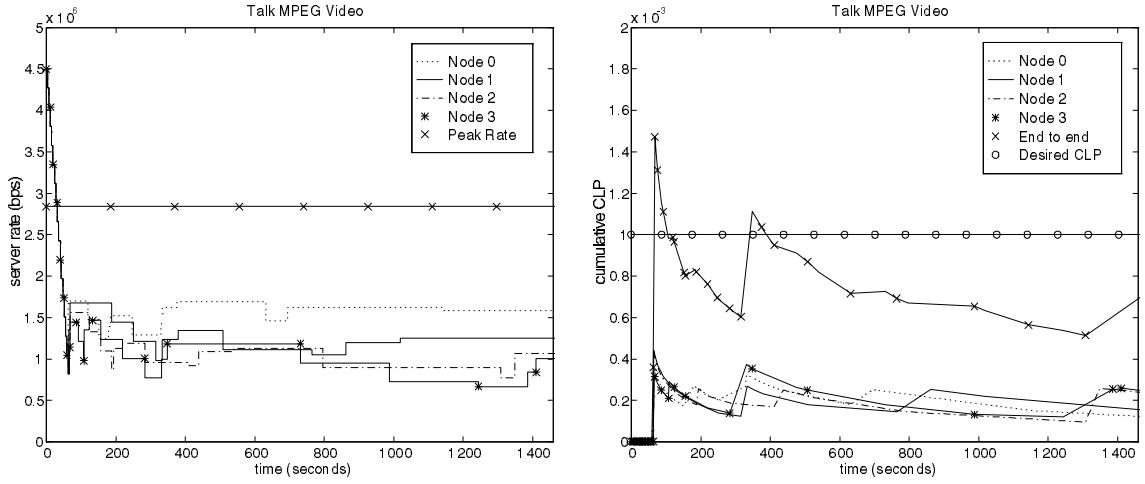

Figure 12: Four node connection, each running DSA+ independently (each-node implementation).

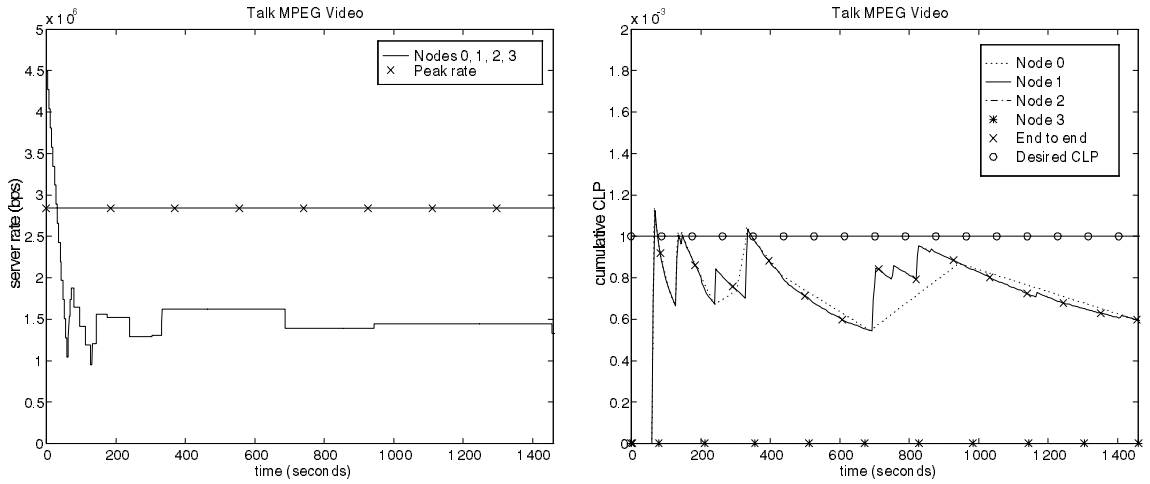

Figure 13: Four node connection, only first node running DSA+ and controlling all allocation (first-node implementation).

Multiple hop connection allocation was also addressed. In this case a connection of four nodes was simulated to evaluate the performance of DSA+ for end-to-end CLP. Two implementations were investigated; each-node and first-node. Both methods were able to provide the end-to-end QoS, however each method may suffers from some possible disadvantages. More details about our work on dynamic resource allocation, including individual MPEG, allocation and CLP graphs, can be found at the web site

`ftp://ftp.csc.ncsu.edu/pub/rtcomm/rtcomm.html`

While the focus of this paper was the bandwidth allocation, DSA+ may be useful for other real-time applications. Examples include CPU scheduling and disk bandwidth management. In both cases the central idea is to provide guaranteed service to variable traffic, with the minimum amount of resources and user input.

This paper assumed no limits on the availability of resources. When any allocation method renegotiated for more resources, they were instantly granted. However in actual implementation this assumption can not be made. In the case of network overload, where contention for more resources is high, resources may not be available. This was the primary purpose for reducing the number of renegotiations for more resource as low as possible. Nevertheless if more resources are required yet not available the users QoS will suffer. If the QoS manager has access to the MPEG compress rate, the shortage of resources can be compensated by altering the Q factor of the compression [17]. The result is a loss of picture quality, until resource are available.

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] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations.* Springer-Verlag, 1990.

[3] C.-S. Chang and J. A. Thomas. Effective Bandwidth in High-Speed Digital Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1091–1099, Aug. 1995.

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

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

[6] A. I. Elwalid and D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, June 1993.

[7] W.-C. Feng, F. Jahanian, and S. Sechrest. An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video. To appear in *ACM/Springer-Verlag Multimedia Systems Journal.*

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

[9] M. W. Garret and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *SIGCOMM'94*, pages 269–280, London, 1994.

[10] I. Hsu and J. Walrand. Dynamic Bandwidth Allocation for ATM Switches. *Journal of Applied Probability*, 33(3):758–771, 1996.

[11] L. Kleinrock. *Queueing Systems, Volume I: Theory.* John Wiley & Sons, 1975.

[12] E. W. Knightly and H. Zhang. Traffic Characterization and Switch Utilization using a Deterministic Bounding Interval Dependent Traffic Model. In *Proceedings of IEEE INFOCOM'95*, pages 1137–1145, Boston, MA, Apr. 1995.

[13] J. F. Kurose and H. T. Mouftah. Computer-Aided Modeling, Analysis, and Design of Communication Networks. *IEEE Journal on Selected Areas in Communications*, 6(1):130 – 145, Jan 1988.

[14] D. Park, H. G. Perros, and H. Yamashita. Approximate Analysis of Discrete-time Tandem Queueing Networks with Bursty and Correlated Input Traffic and Customer Loss. *Operations Research Letters*, (15):95 – 104, 1994.

[15] C. Partridge. *Gigabit Networking.* Addison-Wesley Publ., 1994.

[16] S. Rampal, D. Reeves, Y. Viniotis, and D. Argrawal. Dynamic Resource Allocation Based on Measured QoS. In *The Fifth ICCCN-International Conference On Computer Communications and Networks*, pages 24–27, 1996.

[17] D. Reininger, G. Ramamurthy, and D. Raychaudhuri. VBR MPEG Video Coding with Dynamic Bandwidth Renegotiation. In *IEEE International Conference on Communications*, pages 1773–1777, 1995.

[18] 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.

[19] H. Saito. Measurement Driven Traffic Technologies in ATM Networks. *NTT Review*, 8(1):51–55, Jan. 1996.

[20] W. Stallings. *ISDN and Broadband ISDN with Frame Relay and ATM: Third Edition.* Prentice Hall, 1995.

[21] H. Zhang and E. W. Knightly. RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video. To appear in *ACM/Springer-Verlag Multimedia Systems Journal*, May 1997.