

An Efficient Communication Scheme for Media on-Demand Services with Hard QoS Guarantees

Guojun Lu and Chester Kang
Gippsland School of Computing and Information Technology
Monash University, Gippsland Campus
Churchill, Vic 3842
Fax: (03)99026842
Email: guojun.lu@fcit.monash.edu.au

Abstract

The critical issue of multimedia communications is to provide quality of service (QoS) guarantees while system resources are efficiently used. The system utilisation is usually low when hard QoS guarantees are required due to burstiness of multimedia traffic. We propose a scheme that provides hard QoS guarantees for media on-demand applications while fully utilizing system resources. The basic idea of the scheme is to convert variable bit rate streams into constant bit rate streams for transmission. The constant bit rate is equal to the average bit rate of the stream. This arrangement not only fully utilizes the system resources but also simplifies the server and network design. The scheme requires extra buffering at the receiver (client). We show how to determine the buffering delay and end-to-end delay. Our experimental results show that using our scheme the end-to-end delay is acceptable to most media on-demand applications and in the case of heavily loaded multi-hop networks may be lower than that experienced when no traffic smoothing is carried out.

1. Introduction

Multimedia communications require end-to-end quality of service (QoS) guarantees in terms of delay, delay jitter, error rate etc. For the system to provide QoS guarantees, the application must provide characteristics of the traffic to be communicated to the system. Due to the burstiness of compressed audio and video, it is difficult to characterise variable bit rate coded media streams. It was believed that system resources (such CPU cycles, memory, and network bandwidth) must be reserved based on the peak bit rate of a stream in order to provide hard (100%) performance guarantees to the stream. This strategy results in a very low system resource utilization, as some resources reserved are wasted when the stream is not at the peak bit rate. Therefore, the most important issue in multimedia communication is to provide QoS guarantees while using the system resources efficiently [1]. To address this issue, Knightly and Zhang proposed a traffic model called Deterministic-BIND which allows hard guarantees without reserving resources at the peak bit rate [2]. However, the achievable system utilization is still not very high. The reported bandwidth utilizations vary from 15% to 30% for MPEG compressed video streams, assuming that the bandwidth utilization is 100% when the average bit rates are used for all streams. Another technique to improve the resource utilization is to divide a video stream into a number of segments each of which is then smoothed into constant bit rate [3, 4, 5]. However, this technique is hard to implement and manage as QoS requirements change from segment to segment. Also, issues such as continuity across segments and buffering requirements at the receiver are not fully addressed. McManus and Ross proposed a constant-rate transmission and transport for video-on-demand applications [6]. They showed how to find transmission bit rates based on the buffer space available at

the receiver. However, the calculation of the required transmission bit rate is computationally intensive and the server may be stressed or overloaded as the transmission bit rate is calculated on-line when the stream is requested, leading to the difficulty in managing QoS. The end-to-end delay issue is also not dealt with thoroughly.

In this paper we propose a scheme which can achieve a bandwidth utilization of 100% while providing hard guarantees to media on-demand applications. It is generally difficult to characterise a variable bit rate coded stream a priori. In media on-demand applications, however, compressed media data are stored in servers in advance. We can study and characterise the media streams off-line. In addition, in media on-demand services, the playback startup delay is not critical although should be kept as low as possible, provided that the playback is smooth and synchronized once the playback is started. Our scheme achieves 100% bandwidth utilization by taking advantage of these characteristics of media on-demand systems. The basic idea of the scheme is to convert a variable bit rate stream into a constant bit rate (CBR) stream with the bit rate equal to the average bit rate of the original stream. With the CBR stream, the network resources can be used fully while admission control, scheduling and policing can be carried out easily. The only concerns are whether the startup delay and the buffer requirements at the transmitter and the receiver, are acceptable to media on-demand services and how these parameters can be obtained. We use MPEG compressed video streams to show that these parameters are easily obtainable and their values are acceptable to media on-demand applications.

The rest of the paper is organized as follows. The next section outlines the scheme. The success of the scheme depends on whether the smoothing delay and the end-to-end delay are acceptable to media on-demand applications. Thus in Section 3 we discuss the buffering delay bounds. . In Section 4 we discuss the effect of smoothing on the end-to-end delay. Section 5 reports experimental results obtained from 15 sample MPEG-1 video streams. Section 6 concludes the paper.

2. Overview of the Scheme

Most compressed bit streams are of variable bit rate (VBR) when compression is done in the most efficient way. However, it is hard for servers and networks to handle VBR streams and system resource utilisation is low when hard QoS guarantees are required for these VBR streams.

On the other hand, if all data streams are of constant bit rates, network resources can be used fully and hard QoS guarantees can be easily achieved provided that the total bit rates of all streams are not higher than the network transfer capacity.

The basic idea of our scheme is to convert VBR streams into constant bit rate (CBR) streams using a smoothing buffer before the streams are transmitted to the network, or to design the server so that it reads and transmits stream data at CBR. The constant bit rate is equal to the average bit rate of the VBR stream. At the receiver side, the stream will arrive at CBR after removing the network delay jitter. This CBR stream will be buffered for a certain time to guarantee smooth decoding and display.

Thus the end-to-end delay consists of the following components:

- Buffering time at the transmitter (d_t): this time is required to convert the VBR stream into a CBR stream for network transmission. Once transmission has started, the smoothing buffer should never be empty for more than a packet transmission time in order to maintain continuity of data transmission.
- Network transmission delay (d_n): this is the time to transmit a packet from the transmitter (server) to the receiver (client).
- Buffering time at the receiver: this time is required to smooth out the network transmission delay jitter and to make sure the decoding and display is smooth. For ease of discussion we call the first part of buffering time d_j and the second part d_s . d_j is required so that d_n+d_j is constant and equal to the upper bound of network transmission delay. d_s is required so that there is no data starvation in the decoding and displaying process.

Thus the end-to-end delay D is determined by

$$D = d_t + d_n + d_j + d_s$$

The critical issues of our scheme are how to determine d_t and d_s and whether the end-to-end delay D is acceptable to the intended applications.

Note that for most information broadcasting and media on-demand applications, the buffer at the server (transmitter) is not required. Although the stored media have been compressed at VBR, the server can read data from the storage at a constant bit rate and transmit them at the constant bit rate to the network. This arrangement has two advantages. First, the end-to-end delay is reduced as the buffering time (d_t) at the server is not required. Second, the design of the server is much simpler since the server need only handle CBR and not VBR streams. With a CBR server it is easier to provide hard QoS guarantees while using the resources efficiently.

The CBR transmission at the average bit rate has many advantages. Firstly, it is simple and easy to implement. The server just needs to read and transmit data at the constant bit rate. Secondly, the CBR stream is easy to describe and police. A flow's characteristic is described simply by the average bit rate r and if a flow sends more than this rate, the flow is violating its promised traffic pattern. Thirdly, the total bandwidth (bit rate) of multiplexed CBR sources sharing the same buffer is equal to the sum of their individual bit rates. The additivity of bit rates among traffic sources facilitates the negotiation process between a network and its users. If users submit their required bit rates, the network can then deliver the required QoS by keeping the sum of bit rates of all users below or equal to its capacity. Fourthly, pricing can be simplified by charging based on the average bit rate.

Before we discuss the details of how to determine delays, let us summarize the operation procedure of media on-demand services using the proposed scheme. Each stored stream is analysed and its average bit rate, d_t and d_s are determined off-line and stored as header information of the stream. The retrieval process consists of the following steps:

1. The user selects the desired stream from a client's menu.

2. The client sends the request to the appropriate server.
3. The server sends the header information including the average bit rate and d_s of the stream to the client if the server can deliver the requested stream at the average bit rate.
4. The client requests a connection from the server to the client for the stream.
5. The network admits the connection if the sum of average bit rates of the existing connections and the new request is below its transfer capacity. Otherwise, the connection request is rejected.
6. If the connection is admitted, the network will inform the client the delay upper bound d_{nu} and lower bound d_{nl} .
7. The client reserves delay jitter removing buffer and smoothing buffer equivalent to $d_{nu}-d_{nl}$ and d_s respectively.
8. The client sends the “start” signal to the server which will start reading and transmitting at the stream’s average bit rate. The packet experiences the variable network transmission delay d_n . At the receiver, the packet is buffered for d_j so that $d_j+d_n = d_{nu}$ to remove the delay jitter. The packet is further buffered for d_s to guarantee the smooth decoding and display.
9. Assume the connection from the client to the server is dedicated. The time for the “start” signal to travel from the client to the server is fixed. We call this time t_s . The client should therefore start to decode and display the stream to the user $(t_s+d_{nu}+d_s)$ seconds after it has sent the “start” signal. In this way, the display will have guaranteed quality and the user pays the minimum cost based on the average bit rate of the stream.

In the following section, we discuss the determination of bounds for d_t and d_s for multimedia streams. We use MPEG video streams as examples in our discussion. If these bounds are acceptable to the intended applications, our scheme is viable. The effect of our scheme on end-to-end delay D will be discussed in Section 4.

3. Determination of Bounds for d_t and d_s

3.1 General Requirements

We have mentioned that d_t may not be needed if the server can read and transmit data at CBR. Let us determine d_t assuming it is required. At the transmitter, a stream fills the buffer at its natural variable rate, while the data leaves the buffer into the network at the average bit rate of the stream. To ensure that the buffer has sufficient data to maintain a continuous output, the output should start some time after the start of input of source data.

Let $B(t)$ be accumulative amounts of data for the VBR stream up to time t and $A(t)$ be the accumulative amounts of data that leave the buffer at the average rate of the stream. We want to find a smallest t_0 so that

$$B(t) - A(t - t_0) \geq 0$$

It is easy to find the solution graphically (Figure 1), we shift the $A(t)$ line to the right so that all points on $B(t)$ are above or just on $A(t-t_0)$. $d_t = t_0$ is the buffering time at the transmitter required to guarantee a continuous transmission of the stream data at the average bit rate r .

The required bucket size B_0 without overflow is determined by $\text{Maximum}(B(t)-A(t-t_0))$.

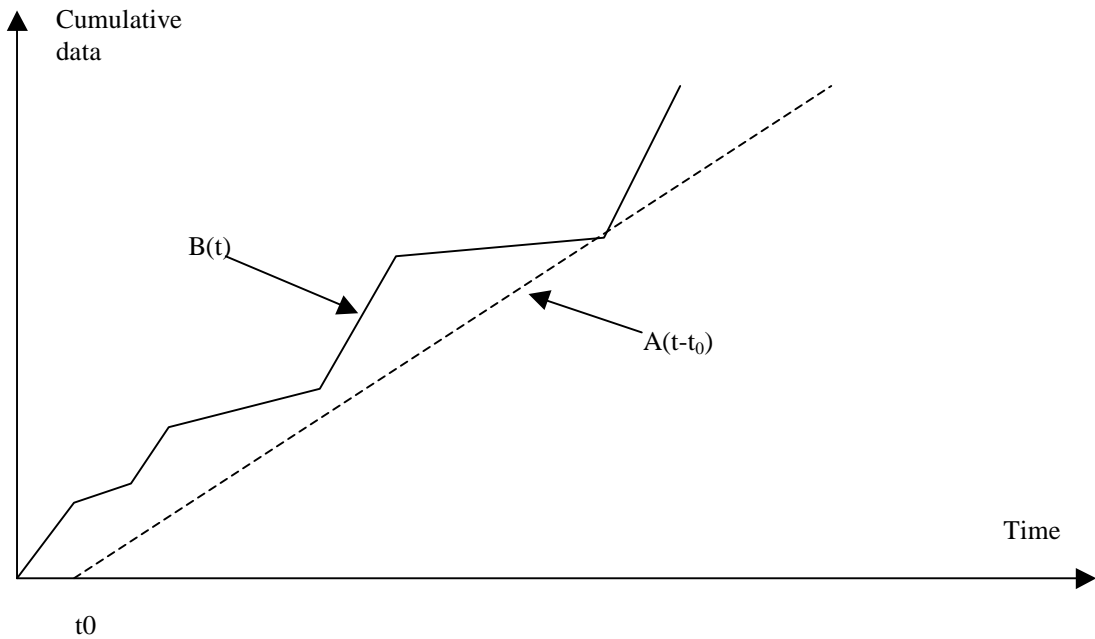


Figure 1 Relationship between $B(t)$ and $A(t-t_0)$ at the transmitter

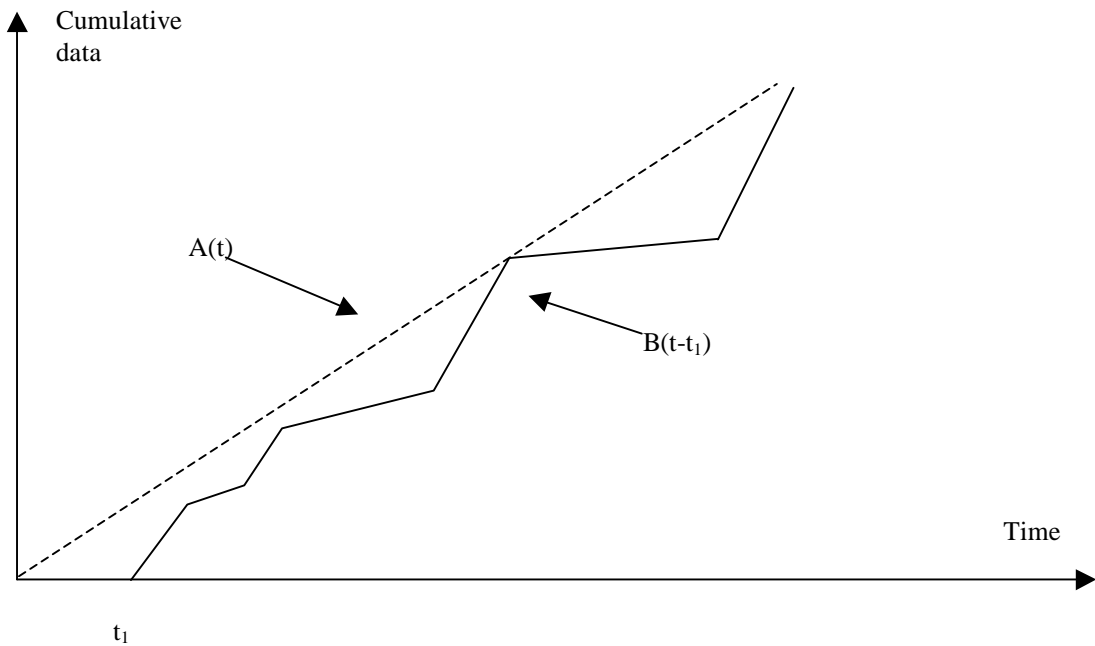


Figure 2 Relationship between $B(t-t_1)$ and $A(t)$ at the receiver

At the receiver, data arrive at the average bit rate (after the buffering of d_j) and the display process consumes data at the natural variable bit rate of the stream. The accumulative data arriving function is $A(t)$ and the accumulative data consumption function is $B(t)$. In order to ensure a smooth continuous playback, the receiver should start to play back some time (say, t_1 seconds) after the arrival of the first data packet, that is we must ensure

$$A(t) - B(t - t_1) \geq 0$$

Graphically, we shift $B(t)$ so that all points of $B(t-t_1)$ are below or just on the line $A(t)$, as shown in Figure 2. The buffer size B_1 required at the receiver is equal to $\text{Maximum}(A(t)-B(t-t_1))$.

The above are the general requirements to ensure continuous transmission and playback. In the following we derive t_0 , t_1 , B_0 and B_1 for MPEG compressed video streams.

Parameters for MPEG Compressed Streams

In MPEG, a video stream is organized into a number of groups of pictures (GOP) [7]. In principle, different GOPs can have different number of pictures, but in practice, each GOP commonly consists of the same number of pictures. Let G_{\max} be the maximum amount of data in a GOP and G_{\min} be the minimum amount of data in a GOP, and p be the ratio of G_{\max} and G_{\min} .

Assume that a video stream has n GOPs and the average bit rate is fixed, then t_0 is maximum when the first $n-1$ GOPs have G_{\min} amounts of data and the last GOP has G_{\max} amounts of data, as shown in Figure 3.

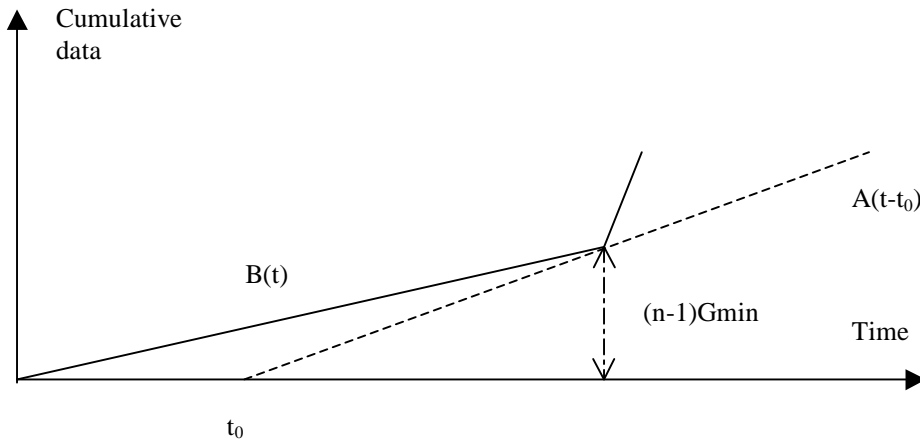


Figure 3 Situation when t_0 is at the maximum

From Figure 3, we have

$$r((n-1)T-t_0) = (n-1)G_{\min} \quad (1)$$

where r is the average bit rate of the stream and T is the duration of a GOP. Rewrite the above equation, we have

$$r = (n-1)G_{\min} / ((n-1)T-t_0) \quad (2)$$

According to the definition of r , we have

$$r = ((n-1)G_{\min} + G_{\max}) / (nT) = (n-1+p)G_{\min} / (nT) \quad (3)$$

Comparing equations (2) and (3), we have

$$t_0 = T(n-1 - \frac{n(n-1)}{n-1+p}) = \frac{T(n-1)(p-1)}{n-1+p} \quad (4)$$

It can be shown that when n is much larger than p , t_0 approaches uplimit of $T(p-1)$, which is independent of n . This is a very important property which indicates that t_0 is bounded, regardless of the length of the video.

As an example, assuming p is 10 and each GOP has 12 pictures (i.e. $T = 12 \times 40 = 480$ ms), when $n = 100$, $t_0 = 3.9$ seconds, when n becomes very large, t_0 approaches 4.3 seconds.

At the receiver, the maximum buffering delay is required when the first GOP has G_{\max} and the rest GOPs have G_{\min} when r is fixed as shown in Figure 4.

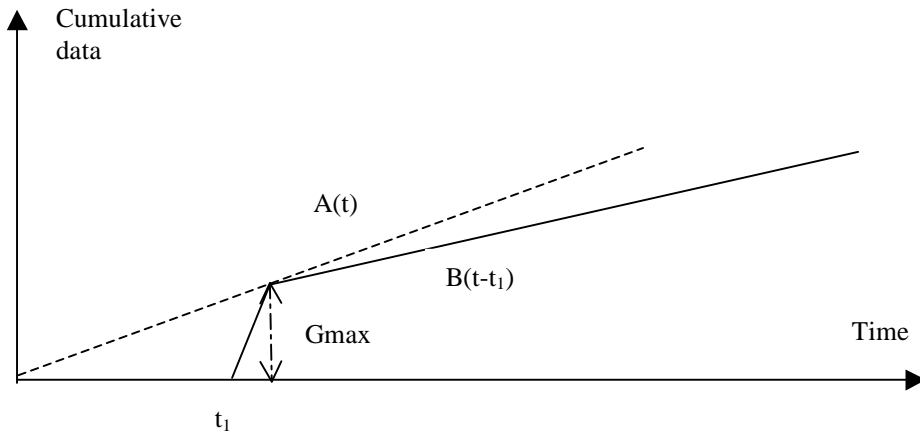


Figure 4 Situation when t_1 is at the maximum

From Figure 4, we have

$G_{\max} = (t_1 + T)r$, that is

$$r = G_{\max} / (t_1 + T) \quad (5)$$

According to the definition of r , we have

$$r = (G_{\max} + (n-1)G_{\min}) / nT \quad (6)$$

Comparing (5) and (6), we have

$$t_1 = \frac{T(n-1)(p-1)}{n-1+p} \quad (7)$$

which is the same as the uplimit of t_0 . But t_0 and t_1 approach their uplimit under different conditions. This means that a stream will not suffer worst delay in both the transmitter and the receiver.

With t_0 and t_1 , B_0 and B_1 can be found by multiplying the average bit rate. Since t_0 and t_1 are bounded, B_0 and B_1 are also bounded.

4. End-to-End Delay Bound

We so far discussed smoothing buffer delay at both the transmitter and receive sides. We now look at the effects of stream smoothing on the end-to-end delay. For simplicity, we use First Come First Served (FCFS) switching discipline as an example in our discussion.

Let D_p be the end-to-end delay bound when the peak bit rate is used for resource reservation (ie the case without smoothing), D_a be the end-to-end delay bound when the average bit rate is used for resource reservation (ie the case with stream smoothing using our scheme).

D_p is equal to the upper bound of network transmission time t_{np} , ie

$$D_p = t_{np}$$

If we assume that the server reads and transmits at the constant bit rate, ie, there is no buffering delay at the transmitter side, then D_a is equal to the upper bound of network transmission delay t_{na} plus the receiver smoothing upper bound t_1 , ie

$$D_a = t_{na} + t_1$$

We want to find t_{np} and t_{na} which are dependent on the switch service disciplines, network load and traffic characteristics.

Determination of t_{na}

Ignoring the propagation delay, t_{na} is equal to the sum of queuing delay at each switch. The admission policy of our scheme is to ensure that the sum of average bit rate of all admitted streams is less than or equal to the network transfer capacity. We have the upper bound t_{na} when the transfer capacity is reached. In the case of first come first served switch scheduling discipline, we have

$$t_{na} = SNP/l$$

where S is the number of switches along the connection, N is the maximum number of admitted connections, P is the maximum packet size, and l is the link speed. Note that we simplified discussion by assuming the traffic pattern remains constant along the path.

Determination of t_{np}

To determine t_{np} , we have to consider two cases. The first case is when the network is lightly loaded and the second case is when the network is heavily loaded. A network is considered to be lightly loaded when the sum of peak bit rates of all admitted connections is less than or equal to the network bandwidth. Otherwise the network is considered heavily loaded.

When the network is lightly loaded, t_{np} can be calculated in the same way as t_{na} except that the number of admitted connections is smaller when the network bandwidth is the same.

When the network is heavily loaded, the upper bound of delay d at each switch (node) is calculated as follows [4, 5, 8]: Assume l is the link speed of the network, h is the total number of admitted connections, t is the interval over which the amount of data of stream j is bounded by $b_j(t)$, and t_p is the transmission time of the largest sized packet. During the time interval $[0, t]$, the maximum waiting time of any packet that arrives at time t is determined by the maximum backlog in the transmission queue $\text{Max}(\sum b_j(t) - lt)$ plus the largest remaining transmission time of any packet in transmission at t . Thus we have

$$d = \frac{1}{l} \text{Max}\left\{\sum_{j=1}^h b_j(t) - lt\right\} + t_p$$

When a connection is over S switches (ignoring propagation delay), we have

$$D_p = t_{np} = Sd$$

Effects on end-to-end delay

When the network is lightly loaded, D_a is larger than D_p by at least t_1 as t_{na} is slightly larger than t_{np} due to the fact that the number of connections supported is larger in our scheme than in the case when peak bit rate is used to reserve resources.

However, when the network is heavily loaded, ie when the sum of the peak bit rates of admitted connections is larger than the network capacity, d increases significantly. When the number of switches S reaches certain number t_{np} may be larger than t_{na} and D_p may be larger than D_a . This means that when we attempt to increase the resource utilization by increasing the admitted number of unsmoothed streams, the end-to-end delay may be larger than that when streams are smoothed. Note that the number of smoothed streams N admitted by the network is still much larger than the number of unsmoothed streams h admitted by the network of the same capacity. We use a numerical example to show this effect in the next section.

5. Experimental Results

We have implemented a software tool which can analyse MPEG-1 streams and produce the buffering bounds t_0 and t_1 , buffer requirement, the peak and average bit rate of the stream. The last two parameters are needed to compare resource utilisations between the proposed scheme and the scheme that reserves resources based on the peak bit rate.

We used two methods to calculate t_0 and t_1 . In the first method, we find G_{\min} for each stream and then calculate maximum size of a GOP possible for the stream G'_{\max} using the following relationship

$$(n-1)G_{\min} + G'_{\max} = \text{total amount of data of the stream}$$

From G_{\min} and G'_{\max} we find $p = G'_{\max}/G_{\min}$. With the p value we can calculate t_0 and t_1 using equations (4) and (7) respectively. We call t_0 and t_1 obtained using this method estimated t_0 and t_1 .

In the second method, we find minimum τ meeting the following requirements by tracing the actual stream (Figure 1):

$$B(t) - A(t - \tau) \geq 0$$

t_0 is then equal to the minimum τ .

Similarly, we find t_1 using the principle shown in Figure 2. We call t_0 and t_1 obtained using the second method actual t_0 and t_1 .

The estimated t_0 and t_1 are normally larger than actual t_0 and t_1 respectively. But it is faster to find the estimated t_0 and t_1 . This is because we have to repetitively increment τ by a picture interval (starting from 0) to find the minimum τ meeting the requirements.

In the following we present results obtained from 15 sample video streams. Some information of these 15 video streams is shown in Table 1.

Table 1 Sample video information

	Pictures Pattern in each GOP	No. of GOP
Sample 1	IBBPBBIBBPBBIBBPBBIBBPBBIBBPBB	7
Sample 2	IBBPBBIBBPBBIBBPBBIBBPBBIBBPBB	7
Sample 3	IBBPBB	44
Sample 4	IBBPBBPBBPBBPBBIBBPBBPBBPBBPBB	46
Sample 5	IBBPBBPBBPBBPBBIBBPBBPBBPBBPBB	43
Sample 6	IBPBIBPBIBPB	12
Sample 7	IBPBIBPBIBPB	12
Sample 8	IBPBIBPBIBPB	12
Sample 9	IBPBIBPBIBPB	12
Sample 10	IBBPBBPBBPBBPBBPBBPBBPBB	10
Sample 11	IBBPBBPBBPBB	47
Sample 12	IBBPBBPBBPBB	38
Sample 13	IBBBBBPBBBBPBBBBPBBBB	31
Sample 14	IBBPBBPBBPBB	75
Sample 15	IBBBPBBPBB	36

Delay Bounds

Table 2 shows estimated and actual t_0 and t_1 of these 15 streams. It can be seen that maximum actual delay bounds at both the transmitter and receiver are in the order of seconds which are acceptable for most media on-demand applications.

Table 2 Delay Bounds

	Estimated t0 (s)	Actual t0 (s)	Estimated t1 (s)	Actual t1 (s)
Sample 1	0.632	0.375	0.632	0.625
Sample 2	1.147	0.375	1.147	1.125
Sample 3	2.320	0.133	2.320	0.167
Sample 4	22.630	0.700	22.630	1.167
Sample 5	9.472	0.000	9.472	4.533
Sample 6	0.527	0.133	0.527	0.067
Sample 7	0.346	0.033	0.346	0.100
Sample 8	0.343	0.033	0.343	0.100
Sample 9	0.467	0.133	0.467	0.067
Sample10	0.898	0.080	0.898	0.320
Sample11	13.297	2.080	13.297	0.480
Sample12	5.358	0.240	5.358	1.080
Sample13	4.138	0.968	4.138	0.133
Sample14	8.964	0.480	8.964	1.400
Sample15	0.097	0.080	0.097	0.080

Buffer Requirements

We have discussed determination of bounds of required buffering delay. The buffer space requirement is equal to the buffering delay times the average bit rate of the stream.

Table 3 shows the actual buffer requirements at the transmitter and receiver sides for the 15 sample video streams. It can be seen the buffer requirement is very realistic and can be provided by most of workstations.

Table 3 Buffering requirements

	Buffer Requirements (KB)	
	Transmitter	Receiver
Sample 1	52	86
Sample 2	81	243
Sample 3	16	21
Sample 4	26	43
Sample 5	0	248
Sample 6	54	27
Sample 7	14	41
Sample 8	11	34
Sample 9	48	24
Sample10	11	45
Sample11	134	31
Sample12	15	69
Sample13	114	16
Sample14	14	41
Sample15	11	11

Resource Utilization

In our proposed scheme, the resources are reserved based on the average bit rate of the stream, resulting in 100% resource utilisation. Table 4 shows that the resource utilisation ranges from 13% to 48% if resources are reserved based on the peak bit rate. Our results show that there is a significant improvement in resource utilisation when resources are reserved based on the average bit rate. The resource utilisation based on the peak bit rate obtained from other studies is similar to or lower than our results [3, 4, 9].

Table 4 Resource utilisation

	Peak rate (bps)	Average rate (bps)	Utilisation based on the peak rate (%)
Sample 1	2,362,944	1,124,171	48
Sample 2	3,699,840	1,770,241	48
Sample 3	2,436,240	1,009,478	41
Sample 4	2,396,400	304,852	13
Sample 5	2,471,280	447,572	18
Sample 6	10,626,960	3,338,113	31
Sample 7	10,705,920	3,351,203	31
Sample 8	8,810,640	2,779,639	32
Sample 9	8,916,000	2,927,208	33
Sample 10	3,894,000	1,151,255	30
Sample 11	2,356,000	529,339	23
Sample 12	1,696,800	525,933	31
Sample 13	3,139,200	963,037	31
Sample 14	1,007,200	242,387	24
Sample 15	4,103,100	1,098,813	27

End-to-End Delay

We have discussed end-to-end delay calculation. In the following we use a numerical example to compare end-to-end delay between our scheme and when streams are not smoothed.

Example

We assume the network used is an ATM network with S switches and a link speed of 155 Mbps. All streams are of the same characteristics as sample 8 in Table 3, ie, $t = 40$ ms (picture interval), $b_j(t) = 8810640 \times 0.04 = 352$ kbits (the largest picture size), $t_1 = 100$ ms and the average bit rate is 2.780 Mbps.

Then the network can admit about 56 ($155/2.78$) smoothed streams with end-to-end

$$D_a = t_1 + S \times 56 \times 48 / 155 = 100 + 2.68S / 155 \text{ (ms)}$$

When the streams are not smoothed (the peak bit rate is 8.81 Mbps) and the network is lightly loaded, the number of connections that can be admitted is about 18 ($155/8.81$). The end-to-end delay

$$D_p = S \times 18 \times 48 / 155 = 0.86S / 155 \text{ (ms)}$$

This means that our scheme adds about 100 ms end-to-end delay while increases the number of connections from 18 to 56.

Next let us consider the case when the network is heavily loaded with the number of admitted unsmoothed streams to be 35. In this case,

$$D_p = S(352000 \times 35 - 155,000,000 \times 0.04) / 155,000,000 + 720S / 155,000,000 = 39S + 0.72S / 155 \text{ (ms)}$$

When S is equal to 3, D_p is about 118 ms. On the other hand, D_a is still about 100 ms. This means that our scheme improves end-to-end delay by about 18 ms while increases the number of admitted streams from 35 to 56. When S increases, D_p will increase substantially.

The above example shows that although we add some buffering delay at the receiver side, the network transmission delay is reduced due to the fact that the streams are smoother. Thus the overall end-to-end delay using the smoothing technique may not be much longer, or may be shorter than that experienced without smoothing depending on the network load and the number of switches the packets go through.

6. Discussion and Conclusion

We described a scheme that smooths compressed media streams into CBR streams to fully utilise system resources while providing hard QoS guarantees. The scheme uses extra buffer delay to trade resource utilisation. Using MPEG-1 compressed streams, we described how to calculate buffering delay bounds. The experimental results show that the buffering delay and buffer space requirements are acceptable for most applications and platforms. We further analyse the end-to-end delay and show that our smoothing scheme adds very small end-to-end delay on single hop networks and may improve the end-to-end delay on a multi-hop congested network.

We have so far stated that an initial delay of a few seconds is acceptable to most media on-demand applications. Now let us discuss issues related to end-to-end delay and acceptability to applications.

- First, d_s is small when the original stream is not very bursty thus our scheme will not add much to the end-to-end delay even in the single hop-network.
- Second, d_s will be large when the original stream is very bursty. But if our scheme is not used (i.e. resources are reserved based on the peak bit rate) the system utilization will be very low when streams are very bursty.
- Third, the initial delay and cost must be balanced. If the application can tolerate some initial delay, our scheme should be used which provides full resource utilization and is cheapest in terms of transmission cost. If fast response time is a must, unsmoothed streams with resources reserved using the peak bit rate should be used. Note that to achieve low end-to-end delay the sum of the peak bit rates of admitted streams should be less than or equal to the network bandwidth. Otherwise, the end-to-end delay may increase significantly when streams are not smoothed. In this case, the more bursty the streams the lower the utilization.
- Fourth, in order to specify the maximum required buffer and delay at the client, the information provider can take certain measures (such as stuffing and altering quantization in the encoding process) to restrict p below a certain value. In this case, the client meeting the specification can retrieve and display all streams.

Overall, our proposed scheme is useful for media on-demand applications due to the following advantages:

- It simplifies network and server design.
- It fully utilises system resources.
- It provides hard QoS guarantees.
- It does not add much extra delay on single hop networks.
- It may improve end-to-end delay on multi-hop heavily loaded networks.

References

- [1] G. Lu, *Communication and Computing for Distributed Multimedia Systems*, Artech House 1996.
- [2] E. Knightly and H. Zhang, "Traffic characterization and switch utilization using deterministic bounding interval dependent traffic models", *Proceedings of IEEE INFOCOM'95*, Boston, MA, USA, April 1995, pp. 1137-1145.
- [3] E. W. Knightly and P. Rossaro, "Effects of smoothing on end-to-end performance guarantees for VBR video", *Proceedings of the 1995 International Symposium on Multimedia Communications and video coding*.
- [4] E. W. Knightly and P. Rossaro, "Improving QoS through traffic smoothing", *Proceedings of IFIP IWQoS'96*, Paris, France.
- [5] J. D. Salehi et al , "Supporting stored video: reducing rate variability and end-to-end resource requirements through optimum smoothing", *Proc. ACM SIGMETRICS*, May 1996.
- [6] J. M. McManus and K. W. Ross, "Video-on-demand over ATM: constant-rate transmission and transport", *IEEE Journal on Selected Areas in Communications*, Vol.14, No.6, August 1996, pp. 1087-1098.
- [7] ISO/IEC International Standard IS 11172, *Information Technology - Coding of Moving Pictures and Associated Digital Storage Media at up to about 1.5 Mbits/s*. 1993.
- [8] D. E Wrege et al, "Deterministic Delay Bounds for VBR Video in Packet-Switching Networks: Fundamental Limits and Practical Tradeoffs", *IEEE/ACM Transactions on Networking*, June 1996.
- [9] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modelling in ATM systems", *Research Report No. 101*, University of Wurzburg, Institute of Compute Science, Germany, 1995.