3
Video coding and ATM statistical bit rate capability

*L. Casamayou, M. de Oliveira, J. Pechiar, M. Simon**
casa, mario, pechiar, msimon@iie.edu.uy
Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Montevideo, Uruguay

Abstract

This paper analyses the adequacy of the statistical bit rate (SBR) ATM transfer capability for supporting real time video services, where end to end delay is limited. For these type of services, mean data rates are normally low. However, if a reasonable image quality consistency is imposed, the generated traffic becomes bursty. This results in important bandwidth availability requirements.
A method is presented for ensuring that the generated traffic can be transmitted over an SBR connection while respecting the service requirements. Comparison with other transfer capabilities shows that SBR is outstanding in terms of application performance and network resources required.

Keywords

Video coding, ATM transfer capabilities, SBR

1 INTRODUCTION

It is widely accepted that ATM multiplexing and switching will be the way to integrate services on a common network. Video services are expected to attract the common public to the broad band. However, these services have special requirements, probably the most stringent concerning transmission. The high data rates involved, the need of synchronisation, and in interactive services the real time constraints are the main characteristics which make video services so demanding in terms of network resources.

Video services include video telephony or video conference, that have strong real time constraints, and digital television, information retrieval or others that are non real time services or have light constraints. As it will be explained in the next

* This work was supported by projects from the CONICYT (National Council for the Research and Development) and the CSIC (University's Commission for Research Support).

©IFIP 1997. Published by Chapman & Hall

section, video coders generate data at a rate which varies constantly depending mainly on what is called image complexity. This means that traffic generated by the coder is essentially variable bit rate (VBR).

When transmitting this data over ATM, a choice between several ATM transfer capabilities (ATC) is available for establishing the connection (I.371, 1995; Boyer, 1995; Roberts, 1995).

The first option is deterministic bit rate (DBR), where a fixed bandwidth is guaranteed (peak cell rate, *PCR*). This bandwidth can be set to the maximum bit rate generated by the coder, so data can be fed directly into the network. In this case, most of the available bandwidth would be misused since this maximum rate occurs rarely.

In order to use network resources efficiently, constant bit rate (CBR) traffic should be produced. This is done by storing the coded data into a smoothing buffer and therefore introducing a delay. If good video quality is expected then this buffer should be large enough to accommodate image transients (e.g. scene cuts), thus allowing for large transmission delays which could turn out unacceptable for real-time applications.

The use of a SBR connection seems more promising. A mean bit rate is guaranteed (which is essential for video transmission), but the transmission of bursts is allowed. In our context, emission of bursts could be used to transmit image transients with low delay.

A first problem to solve in this case is to find out the good connection parameters for transmitting different video services while satisfying some quality standards. These parameters are the sustainable cell rate which determines the maximum mean data rate, the peak cell rate and the maximum burst size.

Available bit rate (ABR) or other reservation protocols are not considered in this work since they require a user to network dialogue which is incompatible with real time constraints.

In an SBR connection, cells are tested for conformance at the usage parameter control (UPC) by means of the generic cell-rate algorithm (GCRA) (I.371, 1995). Letting the network discard non conforming cells would result in severe degradation of the reproduced image. On the other hand, reduction of data flow by an increase in quantisation error at the coder gives a much tolerable result. This is why only conforming traffic is considered as fed into the network, so only quantisation errors will be present at the decoded images.

The traffic emitted depends on the generation control algorithm and on the transmission policy. The former determines the data volume, while the latter decides how exactly this data is delivered.

The organisation of the paper is as follows. Section 2 outlines the principles of video coding, stressing on the control mechanisms. The control algorithm used in this work is described in section 3. Section 4 analyses the delay and quality requirements for real time video services and the limits they pose on the coding process. The values for the SBR parameters needed for video channels are

discussed in section 5. Some general bounds are found, on the basis of the fluid approach. They can be easily generalised for a discrete context. A general coding strategy is proposed, which is compatible with the service and the network constraints. An important point is that coding strategies may be designed independently of the transmission policies, as expressed by some simple analytical results. Some possible transmission policies are presented and compared in section 6. SBR and DBR scenarios are compared in section 7 by means of software simulation. Conclusions are summarised in section 8.

2 GENERAL OVERVIEW OF VIDEO CODING

Video coding is based on the removal of spatio temporal redundancy from the original signal. The MPEG compression standards (MPEG1, 1994; MPEG2, 1994) are widely accepted. For this work, a software version of a MPEG compatible coder was implemented. We now describe the basic ideas of MPEG and how the generated data rate can be controlled.

The input signal to the coder is a sequence of frames in a certain format (e.g. SIF, CCIR). A frame is considered the basic unit for coding and presentation.

Frames are logically divided into slices and macroblocks (MB). A macroblock is a small portion of the image whose size is normally 16x16 pixels, itself being formed by 8x8 pixel blocks. A slice is a horizontal strip of the frame, one macroblock wide.

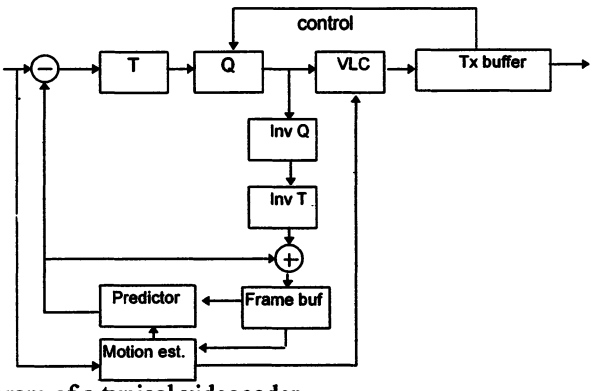

Figure 1 Diagram of a typical videocoder.

Figure 1 shows a simplified diagram of a controlled MPEG coder. The inner loop is a predictive loop, where temporal redundancy is eliminated. The frame to be coded is compared to a predicted frame. This predicted frame made up from the last (decoded) frames where each macroblock is affected by a local motion estimation, a technique known as 'motion compensation'. The difference frame, which normally contains mostly null values then goes through the extraction of spatial redundancy. Each block goes through a discrete cosine transform (DCT) in order to get the

spectral coefficients. These coefficients are then quantised with a granularity given by an external parameter *qp* (quantisation parameter). A low *qp* produces a large amount of data out of the coder, and a high quality (low noise) decoded image. A high *qp* produces little data, so the decoded image is noisy and the prediction loop takes several frames to stabilise. Data for coefficients, quantisation and motion compensation are then statistically compressed (lossless compression) by using variable length coding (VLC).

Temporal prediction can be used in 3 ways in MPEG.

Intra (I) frames use no temporal prediction. I frames are normally located periodically in the data stream for editing purposes and for limiting transmission error propagation. These frames use only spatial compression, so they produce a large amount of data. In the case of real-time services, I frames would require large transmission buffers for storage which would result in an unacceptable delay. Small I frames could be generated by using a high *qp*, but this would result in a severe image degradation on the decoded image.

Predictive (P) frames are based on preceding P or I frames.

Bidirectional or interpolated (B) frames depend both on past and future I or P frames, thus resulting in a higher compression rate. The use of B frames requires a several frame delay for coding-decoding, which becomes unacceptable in the case of real time services.

Therefore, when delay is a major restriction as in this article, only P frames are used. This strategy is part of the 'low delay profile' presented in Test Model 3, Appendix H, and all later versions (TM5, 1993). The effect of transmission errors, if they occur, is extinguished by a gradual refresh mechanism (e.g. periodically sweeping the image with Intra-coded macroblocks).

As all frames are predictive there is no generation pattern, as is the case in the Group of Pictures of the MPEG standard. Variations in generation reflect changes in the video input, not in the coding mode. The refreshment methods that avoid error permanence are assumed to produce only a slight overhead.

The outer feedback loop shown in the diagram is needed if some restrictions on the generated data should be met.

The data generated by each frame depends both on *qp* and on the 'image complexity'. Frames difficult to predict are complex images. These occur during scene transients, the best example being a scene cut, or when the scene contains lots of local movements. In the case of a scene cut, if *qp* is kept low, an important volume of data will be generated for this frame leading to a good decoded image. If, on the other hand, *qp* is high, no such burst will be produced, but it will take several frames for the prediction loop to establish around an acceptable quantisation noise level. If a whole portion of the sequence becomes complex, a new equilibrium between data generated and image quality will be reached.

Taking for example a CBR transmission and a real time service, a certain delay is assigned to the smoothing buffer (other delays are propagation, coding, decoding). This imposes a maximum size for this buffer. If the coder generates data at a rate

higher than the buffer's output rate, the buffer level will grow. A *qp* controller must be devised for the buffer never to overflow nor empty (in order not to waste bandwidth). This controller must calculate the *qp* to be used for the next frame according to the current buffer level.

In VBR operation, a generation control is still needed in order to honour the connection contract, even if restrictions are looser. The actual controller proposed in this work is introduced in the next section. It will be shown that the same controller can be used both for CBR and for VBR transmission.

3 BASIC CODING CONTROL

In general, the data rate produced by a coder can be adjusted by, for example, changing the frame rate, or altering the quantisation parameter (*qp*). The latter is the most usual technique.

This section describes the *qp* control algorithm used for the simulations. The controller must give a good trade off between quality consistency and generation stability: if the controller is too reactive to changes in the buffer level, *qp* will vary constantly thus lowering quality consistency, an effect considered annoying. If, on the other hand, *qp* is made too stable, then there is a high risk for the buffer to overflow or become empty due to unpredictable changes in image complexity. The variation of *qp* is inevitable in the case of a CBR transmission. For VBR, quality consistency may be improved as a burst can be emitted avoiding the filling of the transmission buffer.

The actual algorithm is based on the use of a fixed size buffer emptied at a constant rate (as is the case of CBR). It will be shown in section 5 that this algorithm ensures that the generated data can be transmitted over an SBR connection. The algorithm will observe a virtual buffer level instead of the actual transmission buffer level. All references to the 'transmission buffer' will be substituted with 'virtual buffer' for the case of SBR conforming transmission.

In the test models (TM5, 1993), the slice is used as the action time for the generation controller, choosing a *qp* for each slice. However, previous work (Simon, 1995) has shown that a per frame control gives a more consistent quality. This has been adopted as the basic operation for our control, only altering *qp* at slice or macroblock level after the triggering of alarm mechanisms which avoid buffer overflow. In the following, these two operation forms are described.

Normal Algorithm

1. Observe the buffer level and fix a desired frame generation volume. The transmission buffer (or virtual buffer in SBR) has a fixed target level (e.g. 1/4 of the buffer size). The deviation from this target level is used to calculate the amount of data which should be generated by each of the following *nf* frames in order to reach the target level. A low value for the parameter *nf* results in a very reactive

control strategy, whereas a high *nf* results in a looser control over the buffer level and a higher quality consistency.
2. Chose a *qp* for coding the next frame. Given the desired generation for the next frame, an adequate *qp* must be found. Due to real-time constraints, this cannot be done iteratively, but before the actual coding. Generation as a function of *qp* is not known a priori, so it is estimated based on previous frames as follows: the data generation and the corresponding *qp* for the last two frames are stored (*G1*,*G2*, *qp1* and *qp2*). If the desired generation lies between *G1* and *G2*, then a linear interpolation is used to find *qp*. Otherwise, a hyperbolic extrapolation is made, using only the nearest point (this comes from the fact that the product *qp generation* is approximately constant).

Alarm Mechanisms

A sudden increase in image complexity can cause the buffer to overflow. An alarm level is defined as a percentage of the buffer capacity (e.g. 95%). Two actions are taken:
1. At slice level, whenever the slope of the buffer level is high enough for the buffer to reach the alarm level in the current frame, *qp* is increased by 1 unit.
2. At macroblock level, if the buffer is full beyond the alarm level, *qp* is increased proportionally to the excess level.

The controlled system is stable. Alarms are triggered only in exceptional cases such as scene cuts.

4 REAL TIME VIDEO SERVICES

Video services with real time constraints are those where a low end to end delay must be satisfied. Examples of this type of services are videotelephony or videoconference. These are interactive services where acquisition to presentation delays must be kept below 150 ms in order not to confuse the users. With higher levels of delay, the fluidity of a conversation becomes affected.

When these services are carried over a data network, several factors add to total delay: acquisition, coding, transmission and reception buffering, propagation, switching and decoding. Therefore, a delay of 80 to 100 ms is assigned for buffering purposes. This delay is possible since only predictive (P) frames are coded (refer to section 2 for a discussion). Switching delays should be negligible since cell rates throughout the network are much higher than the application's cell rate. Propagation delays are low in the case of nation wide communications.

The image quality needed depends on the specific service. In real time video services the mean quality requirement is not critical. Users are interested in the expression of the speaker and not in having a clean, high definition image. Moreover, screen size is small in the case of video telephony so coding artefacts are

less apparent. Therefore, there is a quality threshold above which no further improvement is perceived.

The video signal quality has two components: mean quality and response to transients. For a fixed mean quality, the user expects the overall quality to be consistent.

The data rates for this kind of services is rather low, in the range 128 to 512 kbps. More complex signals (TV quality, for example) would require the default rate for MPEG1 (1.15 Mbps).

5 CODING CONTROL STRATEGY FOR SBR. WHAT PARAMETERS DOES VIDEO NEED?

When establishing an SBR connection, three parameters are specified: sustainable cell rate (*SCR*), peak cell rate (*PCR*) and maximum burst size (*MBS*).

PCR is the maximum rate at which cells can be emitted by the source. *SCR* is the maximum mean cell rate and *MBS* is the maximum size (in cells) a burst emitted at *PCR* can have.

Cell conformance is tested at the UPC by means of the generic cell rate algorithm (GCRA) with parameters $1/SCR$ for the interval and B for the limit. B is deduced from the other parameters as follows: $B=MBS(1-SCR/PCR)$ (I.371, 1995).

In order to obtain simpler and more intuitive results, a fluid model will be used throughout the paper for the traffic and the conformance algorithms. Exact results can be derived with no extra difficulty.

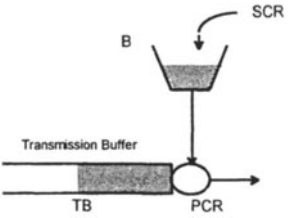

Figure 2 SBR transmission.

For the fluid model, conformance at the UPC is verified through a leaky bucket algorithm (Rathgeb, 1991), where the bucket depth is B and the mean rate *SCR*, or equivalently, input traffic should be of (σ,ρ) type with $\sigma=B$ and $\rho=SCR$ (Cruz, 1991). B and *SCR* are normally expressed in cells and cells per second respectively, σ in bits and ρ in bits per second.

From the source's point of view, traffic can only be emitted if enough credits are available. The source can be thought of as having a credit pool (Figure 2) which is filled at rate *SCR*, has a maximum of B credits and can be emptied at a maximum rate *PCR*. For each unit of data emitted, a credit unit has to be extracted from the credit pool. With this mechanism, the only restriction on the traffic produced by the source is that it is conforming at the UPC. Since the credit pool is very related to

the bucket in the leaky bucket algorithm, the term 'bucket' will be used from now on to refer to the credit pool at the source.

Non conforming traffic should be avoided in video transmission. Indeed, if the network discards a single cell, the decoder will have to discard all data until a synchronisation mark is received. This is because data transmitted is compressed being impossible to resynchronise at any point in the bitstream.

This SBR traffic generation scheme can be applied as a server for the video coder's transmission buffer (recall that for CBR transmission a constant rate server has to be used). As a basic rule, the server will be assumed to be conservative: no credits will be lost if there is data waiting for transmission. In other words, no bandwidth will be wasted by letting the bucket overflow while there is data to transmit.

Given an amount of data to be transmitted and a number of available credits, there is still a choice on how exactly will be the evolution of credit usage with time. This choice is not available on CBR. The way on how credits are used is decided according to a 'transmission policy'. For example, the server can wait for the bucket to fill and then transmit at *PCR*; or it can directly use all credits by transmitting at *PCR* and then continue at *SCR* (the bucket's filling rate).

As the speed at which the transmission buffer empties is not constant -as was the case for CBR- there is no direct relationship between buffer level and transmission delay. This implies a certain interaction between the transmission policy and the coder generation controller in order to satisfy the delay requirements. Therefore, a rather complex controller should be designed which takes into account not only the state of the transmission buffer but also of the traffic contract (the bucket level in this case) and the transmission policy. The problem of finding out an optimum controller becomes rather difficult.

However, it will be shown in the next paragraphs that the controller can be decoupled from the transmission policy. Also, the actual transmission buffer level and the credit bucket level can be put into a single variable called a 'virtual buffer level'. This virtual buffer level has a fixed maximum value which ensures that the delay requirements are met. The controller described in the preceding sections can be used in this case, by observing the virtual buffer level instead of the actual transmission buffer level.

In order to prove the above assertions, the terminology used in the remaining of the section is as follows:

- *PCR* is the peak cell rate. *br*(*PCR*) means the bit rate corresponding to *PCR*.
- *SCR* is the sustainable cell rate. *br*(*SCR*) is the corresponding bit rate.
- *TB* is the transmission buffer level in bits, *MTB* is its maximum.
- *B* is the bucket depth in bits.
- *C* are the used credits, in bits. Therefore, *B*-*C* are the available credits. The bucket is referred to as 'full' when $C = 0$.
- *MBS* is the maximum burst size. $MBS \cdot [1 - SCR/PCR] = B$ holds for the fluid approach.

- *Delay* is the portion of total delay assigned for buffering (e.g. 80 ms). Other delays correspond to coding, propagation, etc. and will not be considered in the following sections.

Consider the data generated by the coder as being fed simultaneously into two transmission buffers: one buffer served by the SBR mechanism with its credit bucket, and another buffer served at a constant rate *SCR* (Figure 3). Call this second buffer the 'virtual buffer', whose level is *VB*.

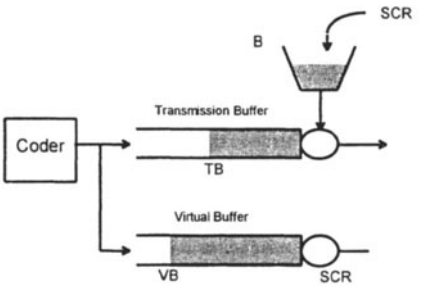

Figure 3 Real and virtual buffers.

Since both servers are conservative, there is a simple relation between both buffer levels: $VB = TB + C$.

One way to see this is assuming that data arrives to both buffers at rate *SCR*. *VB* will be stable. *TB* will be stable if data is transmitted at *SCR* and therefore the bucket level and *C* are constant. If *C* increases by 1, it means that 1 bit from the buffer has been transmitted in addition to the *SCR* traffic, so *TB* will be reduced by 1. Notice also that the virtual buffer empties only when the bucket gets full.

The following lemmas will give more meaning to the variable VB.

Lemma 1: Maximum transmission buffer

Transmission buffering delay is bounded by *Delay*. Given the actual number of credits used (*C*), the maximum amount of data that can be transmitted in *Delay* is the number of credits available at that moment plus the number of credits which will arrive during this time. This gives the maximum level the transmission buffer can have at any moment: $\max(TB) = MTB = B - C + Delay \cdot br(SCR)$

Notice that both *MTB* and *C* are functions of time.

Therefore, if *TB* never reaches *MTB*, transmission is possible in a time less than *Delay*♦

Assuming that the coder puts data into the buffer at a speed much higher than actual transmission speeds, the controller may safely allow the coder to generate up to (*MTB-TB*) bits. This is a rather conservative approach because if the coder is

known to produce data at a maximum rate, data in the buffer can be actually transmitted while the coder is producing new data. In this case, these bound on generation could be made tighter (i.e. generate more than simply *MTB-TB*).

Lemma 2: The virtual buffer (*VB*) and its maximum (*MVB*)

The variable *VB* introduced above has the intuitive interpretation of being the level of a CBR smoothing buffer, and is actually easy to keep track of without even knowing the bucket level: *VB* increases with coder data generation and decreases at a constant rate *br*(*SCR*). The coding system easily keeps track of time since the time between frames or slices is constant. So recalculating *VB* at each slice interval is a trivial task.

The usefulness of *VB* is now presented.

From Lemma 1, it is seen that $TB + C \leq Delay \cdot br(SCR) + B$ for the delay bounds to be met. Therefore, introducing the constant 'maximum virtual buffer' $MVB = Delay\ br(SCR) + B$, the delay bounds are summarised in the simple expression:

$VB \leq MVB$ ♦

So, if the controller described in section 3 is given *VB* as a variable to control rather than *TB*, with a maximum of *MVB* rather than *MTB*, data can be delivered on time, independently of the transmission policy used.

When $VB=0$, the bucket is full of credits and there is no data to transmit. $VB=MVB$ indicates for example that *MVB* bits are present in the buffer and all credits are available; or that no credits are available and there are $Delay \cdot br(SCR)$ bits to transmit.

Recalling that for a CBR transmission at rate *SCR*, the buffer was limited to $Delay \cdot br(SCR)$ bits, the ability to send bursts is equivalent to an increase of *B* bits in the buffer size, even when the mean data rate (and mean quality) remains the same. A larger buffer allows for a better quality consistency: scene transients can be coded at a lower *qp* without risk of buffer overflow.

With respect to the connection parameters, $MVB = Delay \cdot br(SCR) + MBS \cdot (1 - SCR/PCR)$. So virtual buffering capacity decreases for lower *PCR* values.

Restriction 1: The Delay imposes a *PCR - MBS* relationship

The value for *PCR* is not arbitrary, even if this parameter does not appear explicitly in Lemma 1. An underlying assumption is that *TB* can effectively be transmitted in a time less than *Delay*, given the finite value of *PCR*. Therefore, the condition $TB/br(PCR) < Delay$ should be valid for any possible level of *TB* given in Lemma 1: $Delay \cdot br(SCR) + B - C \leq Delay \cdot br(PCR)$. This restriction depends on the used credits *C*. Its worst case gives $B \leq Delay \cdot [br(PCR) - br(SCR)]$, or equivalently, $MBS \leq Delay \cdot br(PCR)$.

This means that given a value for *PCR*, the useful burst size (or bucket depth) is limited. A bigger bucket is useless because extra credits will be never evacuated in time. The useful region lies below the straight lines shown in Figure 4, which are drawn for two different delay values (80 and 120 ms). Higher delay values correspond to higher slopes.

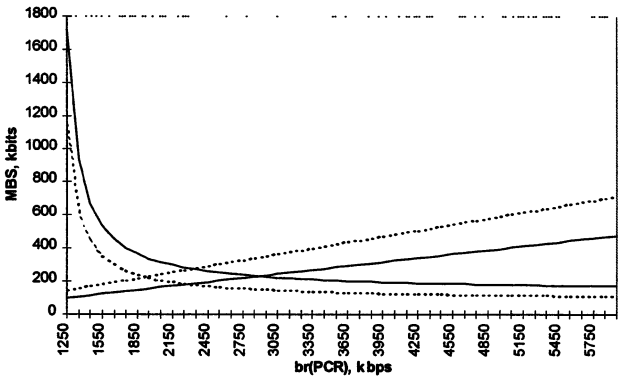

Figure 4 *PCR - MBS* relationship.

Restriction 2: A fixed video quality imposes a *PCR - MBS* relationship

The response to transients is better if the controller allows a big virtual buffer excursion. This is tightly related to *MVB*, which is the maximum of this buffer. Restriction 2 shows the relation between *MBS* and *PCR* that correspond to a given video quality.

From Lemma 2, $MVB = Delay \cdot br(SCR) + MBS \cdot (1 - SCR/PCR)$.

Given *MVB* and *SCR*, different (*MBS*, *PCR*) couples will give the same transient quality, provided that $MBS \cdot (1 - SCR/PCR) = \text{constant} = MVB - Delay \cdot br(SCR)$.

The hyperbolas in Figure 4 are constant quality curves. The minimum *PCR* and maximum *MBS* are obtained at the intersection of the line and the hyperbola corresponding to a certain *Delay* value. In this case, $MBSmax = MVB$ (independent of *Delay*), and $br(PCR)\text{min} = MVB / Delay$.

Only the low right part of each hyperbola is of interest, where the *MBS - PCR* balance is meaningful. Note that an increase in *PCR* provides only a very slight decrease in the maximum burst size.

6 TRANSMISSION POLICIES: HANDLING OF BUFFER AND BUCKET

The main problem in video transmission is having the data stored on time at the reception buffer for when it is required by the decoder. An SBR type transmission

can be thought of as a CBR transmission with the ability of sending data in advance to the reception buffer. The amount of data sent in advance is exactly the value of C used previously. Therefore, during periods of low generation rate, credits can be accumulated thus lowering C, so when a high generation rate transient appears, up to B bits can be sent 'in advance' to the receiver letting the remaining data for transmission at the normal rate (*SCR*). This gives a rather intuitive idea of how SBR can be more performing in terms of quality consistency when low delays are required.

A transmission policy can be defined as a set of rules for delivering data into the network. This includes the management of the available credits and of the transmission buffer.

It has been shown in previous sections that the coding process could work independently of the transmission policy used. However, this policy cannot be chosen arbitrarily. For example, a policy could be to transmit at *SCR*, never using extra credits for emitting bursts. This CBR type policy is conservative, but certainly some data will not be delivered in time (this occurs to all data which find the transmission buffer level above *Delay·br*(*SCR*)). Therefore, a well behaved policy should emit bursts according to the transmission buffer level to ensure in time delivery of data into the network.

Two examples of possible transmission policies will be presented below, each producing traffic with different burstiness characteristics.

In the remaining of the section, the following terminology will be used:

- N is the total delay measured in video units (for instance, slices or frames).
- When coding the nth unit, the (n-N)th unit is to be presented at the receiver.
- *RB* and *TB* are the transmission and reception buffer levels when n has just been coded.
- C are the used credits from the bucket, and B the bucket depth ($0 < C < B$).
- *bpu*(*SCR*) and *bpu*(*PCR*) are bits per video unit at mean and peak rates.
- $g(n)$ are the bits generated by nth unit.

Note that for any transmission scheme, the number of video units stored at the transmission and reception buffers add to a constant value (N). Video units are assumed to be stored and extracted from the buffers instantaneously by the coder and decoder.

Transmission policy 1- Save as many credits as possible

With this policy, the steady state situation corresponds to a full credit bucket (C=0) and the transmission buffer at the target level. So transmission is normally done at *SCR* until *TB* reaches the level *Delay·br*(*SCR*). Whenever *TB* goes beyond this level, transmission at a higher rate should be triggered. The precise implementation

of this policy is quite complex regarding when to resume transmission at *SCR* after a burst is sent.

When the coder's generation rate is lower than *SCR*, first the bucket is allowed to fill, then data are transmitted in advance in order not to waste credits.

Assume that the (*n-N-1*)th unit has just been decoded. Therefore the reception buffer, if $RB > 0$, contains some part of the (*n-N*)th unit. The remaining of the $g(n\text{-}N)$ bits should be present in the reception buffer in the next time unit. Therefore, $g(n\text{-}N) - RB$ bits have to be transmitted by this time (if this value is positive). As the policy is to save as many credits as possible, the volume to be transmitted is $\max[\, g(n\text{-}N) - RB\,,\, bpu(SCR) - C\,,\, 0\,]$.

The second term corresponds to transmission being conservative: credits that cannot be used to further fill the bucket are used to transmit data in advance. This policy will produce rare but considerable bursts. In fact, as the bucket is normally full, the potential burst size is the maximum allowed by the leaky bucket size.

Transmission policy 2 - Transmit as early as possible

Now the target situation is to have a certain number of used credits and the transmission buffer empty. The reception buffer will contain in average at least the controller's *VB* target level.

This policy is very simple to implement: each time there are credits available, data is transmitted at *PCR*. It will produce frequent short bursts (tens of cells) much like the coder's output. Some credits are normally used, so even the largest bursts will be smaller than *MBS*.

These small bursts should be multiplexed more effectively at the network level with others of the same type. So this policy is easier to implement, and not aggressive to the network (bursts are smaller).

The transmitted amount during one time unit is $\min[bpu(PCR),\, TB(n) - g(n),\, bpu(SCR) + MBS - C]$.

This means that the transmitter is delivering as much as it can, limited by the most restrictive of: *PCR*, the total amount of information present in the transmission buffer, or the credits present in the bucket.

Recall that in all cases the transmitted signal is the same and the real time requirements are met.

Figure 5 shows the traffic in cells per second when the second policy is used corresponding to mean rates $br(SCR)$ of 512 kbps and 320 kbps, $PCR = 3 \cdot SCR$, and $B = 167$ cells (64 000 bits). The test sequence has a scene cut in frame 79. After the cut the sequence images are more complex at the bottom. Indeed, the generation pattern during each frame exhibits a peak around the bottom slices of each frame. During the steady state, as transmission is immediate, the transmission rate coincides with the generation rate. Note that in this phase bursts are very short. In fact, they correspond to a video slice time, which is 40 ms / 18 = 2 ms and contain

about 7 cells per slice when $br(SCR)$ = 512 kbps, or 4 cells when $br(SCR)$ = 320 kbps. It is expected that a set of sources of this kind could be multiplexed using moderate buffers in the network.

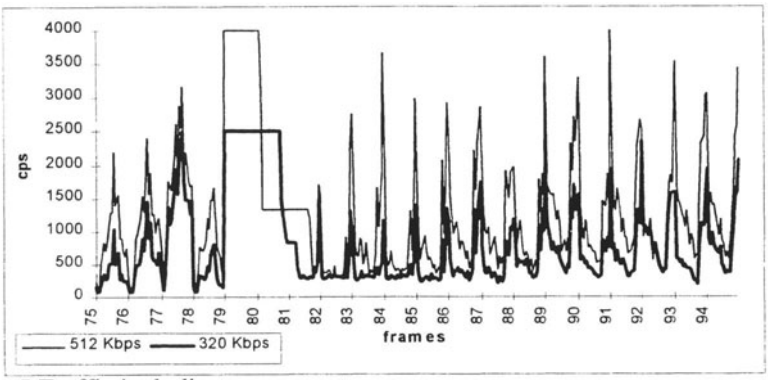

Figure 5 Traffic including a scene cut.

During the scene cut the cell rate rises to *PCR* until the credits are exhausted. At this moment transmission continues rate *SCR* until the large units are delivered. Note that the *SCR* phase is longer in the 512 kbps case. This is because the common *B* value (167 cells) is relatively much higher when $br(SCR)$ = 320 kbps, therefore resulting almost sufficient for emitting all of the transient information units.

7 SBR - DBR COMPARISON

Simulations were performed on 25 frame per second sequences using mean rates in the range 320 to 512 kbps for the video signal, speeds which are adequate for video conference or high quality video telephony. The delay assigned for transmission buffering was set to 80 ms, equivalent to two frames, leaving sufficient margin for other delays (coding, propagation, switching, image scanning) which are not considered here.

The DBR option is compared with SBR regarding several aspects of the global service quality.

Same signal, same mean rate ⇒ Delay?

For the same video quality and mean rate, if the DBR capability is used, transmission should be done at CBR (with $PCR = br(SCR)$). Here the maximum delay is given by the size of the smoothing buffer, which has been shown to be *MVB* (section 5). Numerical values are given in Table 1. The CBR figures are too high for a real time service, since buffering delay is only part of the global delay.

Table 1 Same signal, same mean rate

Mean rate (kbps)	*Delay SBR (ms)*	*Delay CBR (ms)*
320	80	213
512	80	163

Same signal, same delay ⇒ Negotiated peak rate?

If the same coded signal, and therefore the same mean data rate, and the same buffering delays are used, a DBR connection can be used but with a higher rate. In this case, since this rate is rarely used, statistical multiplexing is compromised. Using the data obtained in the working examples, the minimum DBR to be negotiated is about 2 times the mean rate for 512 kbps, and 2.66 times for 320 kbps, giving a very low utilisation factor. The rates to be negotiated in SBR and DBR in this case are shown in Table 2. A complete comparison should consider the number of DBR and SBR channels that may be multiplexed.

Table 2 Same signal, same delay

Mean rate (kbps)	*Negotiated mean, SBR (kbps)*	*Negotiated rate, DBR (kbps)*
320	320	853
512	512	1043

Same mean rate, same delay ⇒ Signal quality?

Finally, when mean rate and delay are maintained, the signal quality is compared giving error images and luminance signal to noise ratio (LSNR) during the sequence. In this case, for the DBR channel the signal is coded at mean rate *SCR*, using a short buffer to achieve the delay.

Global image quality is affected. Figure 6 shows the LSNR for DBR and SBR. A scene cut occurs in frame 79. In steady state, both DBR and SBR at 512 kbps give similar results. The pictures in Figure 7 are, from left to right, the original frame 79, and the error images obtained for DBR and SBR transmission strategies. Error images were obtained by subtracting original and decoded images, taking absolute values, and magnifying by a factor 8. The parameters used were *Delay* = 80 ms (i.e. 2 frames), *br*(*SCR*) = 512 kbps, *PCR* = 3·*SCR* and *MBS* = 177 cells. Other parameters are summarised in Table 3. The expected burst is calculated as *MBS*·(1 -

SCR/PCR) - *target*, because transmission policy 2 was used, so *target* credits are normally used. Note that the traffic is not very bursty. In fact, the expected bursts are not large.

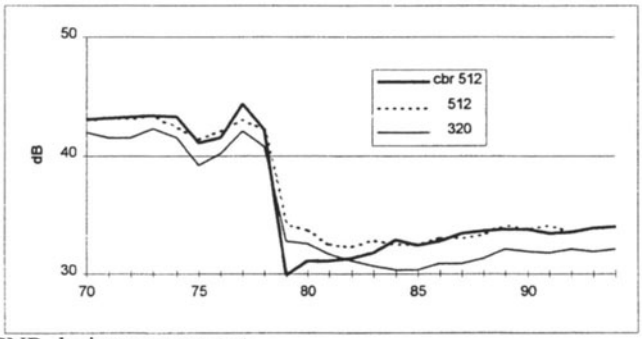

Figure 6 LSNR during a scene cut.

Figure 7 Original and error images after scene cut.

Table 3 Same delay and mean rate. Simulation values overview

br(SCR) (kbps)	*PCR (kbps)*	*MVB (bits)*	*control target (bits)*	*expected burst (cells)*
320	3 · SCR	68 267	13 653	76
512	3 · SCR	83 626	16 725	67
512 (CBR)	512	40 960	8196	—

A simulation using *br*(*SCR*) = 320 kbps is also included. In these conditions, a CBR connection is unable to respect the imposed delay, even with the coarsest quantisation level, whereas the SBR image is fairly good. Naturally, the steady state LSNR is lower when the mean rate is *br*(*SCR*) = 320 kbps. Subjective evaluation of the coded sequences show that the transient of the CBR signal (at 512 kbps) is quite

noticeable, whereas the SBR signal, even at 320 kbps, has consistent quality and the distortion is not annoying.

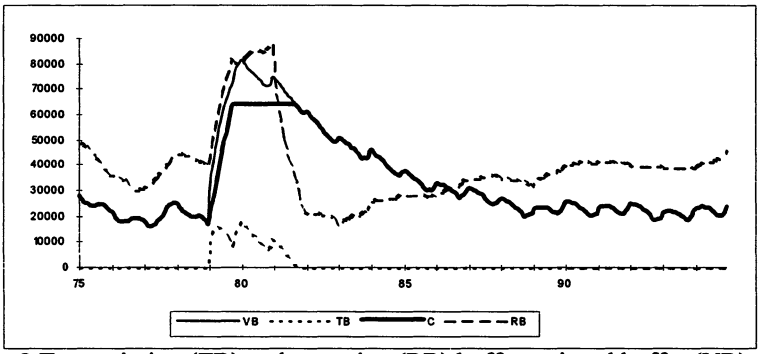

Figure 8 Transmission (TB) and reception (RB) buffers, virtual buffer (VB) and used credits (C) evolution.

The behaviour of buffers and bucket in the SBR scenario, corresponding to the given *SCR*, *PCR* and *MBS* and to the transmission policy 2 is shown in Figure 8. In steady state the reception buffer contains an amount of data corresponding to the delay and some credits are normally used. The transmission buffer is usually empty.

During the scene cut credit usage grows at *PCR*, and stays at *B* during some frames. This is where the transmission buffer (*TB*) accumulates data. *TB* first grows because generation is faster than *PCR*. Then it grows as no more credits are left and transmission is limited to *SCR*. The burst transmitted during the scene cut causes an increase in the reception buffer level. Then this level goes below the steady state level due to the decoding of the huge frame 79. Anyway, this buffer is always far from starvation.

8 CONCLUSIONS AND PERSPECTIVES

A general coding strategy was presented for real time video services to be transmitted using the SBR ATM transfer capability. It was shown that the controller takes into account the transmission parameters and limitations, which appear as a unique bound, the maximum virtual control buffer. A strong consequence is that the same control strategies developed for CBR operation are valid for SBR if the real buffer capacity is replaced by this bound.

The concept of a transmission policy is introduced. Some policies are presented to show the possibility of transmitting the generated bit stream while respecting network and service constraints, and to analyse buffer and bucket evolution. The control algorithm was shown not to depend on the transmission policy used.

The simulation results show that the SBR capability, compared with DBR, allows for a better quality when other conditions (mean rate, delay) remain equal. An important conclusion is that video telephony and video conference services can be implemented over SBR channels with moderate *SCR*, *PCR* and *MBS* values, and no extra coder complexity. A more precise characterisation of the SBR parameters needed to support the different video services is to be done.

Values for the SBR parameters are analysed for video applications. In general, when lower mean rates and delays are required, the advantages of VBR transmission over CBR are more dramatic. For example, if propagation delay becomes important as in a long distance connection, the delay assigned for buffering must be kept very limited.

A next step in this subject will be to quantify SBR multiplexing gain compared to DBR. In fact, the comparison between the DBR and SBR cases for the same delay and quality will be completed by establishing how many SBR channels can be multiplexed.

9 REFERENCES

Boyer, J., Gravey, A. and Sevilla, K. (1995) Resource allocation for worst case traffic in ATM networks. *WATM'95*, Paris, Dec. 1995.

Cruz, R.L. (1991) A Calculus for Network Delay, Part I: Network Elements in isolation. *IEEE Trans. on Information Theory*. Vol. 37, Num. 1.

Cruz, R.L. (1991) A Calculus for Network Delay, Part II: Network Analysis. *IEEE Trans. on Information Theory*. Vol. 37, Num. 1.

I.371 (1995) Recommendation I.371: Traffic control and congestion control in B-ISDN. *ITU-T Study Group 13 meeting*, Geneva.

MPEG1 (1994) ISO/IEC 11 172: Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s. ISO/IEC JTC1/SC29/WG11.

MPEG2 (1994) ISO/IEC 13 818: Generic coding of moving pictures and associated audio. ISO/IEC JTC1/SC29/WG11, Singapore.

Rathgeb, E. (1991) Modelling and performance comparison of policing mechanisms for ATM networks. *IEEE JSAC, April 1991*.

Roberts, J.W. (1995) What ATM transfer capabilities for the B-ISDN? *WATM'95*, Paris, Dec. 1995.

Simon, M., Casamayou, L., Villegas, P. and Roser, M. "Improved quality video coding for CBR transmission: Bit production control and pre-analysis", Midwest Symposium on Circuits and Systems, Rio de Janeiro, Brazil, August 1995.

TM5 (1993) ISO/IEC JTC1/SC29/WG11 Test Model Editing Committee: Test Model 5.