Supporting Real-Time Video Over ATM Networks

A Masters Thesis By Benedict Shija
Supervised By Dr Daniel Mashao and Externally By Dr Bakhe Nleya

In Full Fulfillment Of The Requirements For The Masters Degree In Engineering (MSC ENG)

Acknowledgements

I gratefully acknowledge the support and assistance that I received from the following individuals/organizations

1. Dr Mashao (Supervisor)
2. Dr Nleya (External Supervisor)
3. Mr. Ventura (Head of Broadband Labs)
4. The guys and girls at the UCT Broadband Lab
5. Telkom and its sponsor partners (Siemens and THRIP)
6. My family and friends

Table of Contents

List of Figures

List of Tables

Abstract

The real time Variable Bit Rate (rt-VBR) service class defined by the ATM Forum is proposed to support continuous transmission of bursty traffics, such as compressed audio and video streams. Unfortunately, the characteristics of these traffics are difficult, if not impossible, to define. Consequently, excess resource, in terms of its peak cell rate and maximum burst size, must be reserved to accommodate the fluctuation in transmission rates in order that the Quality of Service (QoS) requirements can be adequately guaranteed. However, it is well known that temporal media of video is highly tolerable to short-term distortions (or breaks). Hence, it should be possible to maximise the video throughput by selectively discarding the least important video components of the video data streams while at the same time preserving the most important components. MPEG compressed video streams follow a strict hierarchical structure, which can be exploited by a selective discarding technique implemented at the ATM switch. The hierarchical structure comprises of the smallest encoding unit, Block, Macroblock, Slice, Frame, Group of Pictures, and Sequence.

By discarding at the slice level rather than at the cell level, freed resources can be used to provide enhanced support to the rest of the data stream without introducing significant distortion in the reconstructed sequence. By considering also the different importance of coded frames: I, P, or B, the impact of the distortion due to selective discards can further be reduced. Unlike the selective cell discarding mechanism based on the Cell Loss Priority (CLP), early slice discard discards remaining cells belonging to a slice where one or more cells have already been lost. This prevents the forwarding of cells belonging to corrupt video slice through the ATM network, thereby conserving bandwidth.

In this project, we propose and evaluate an approach to delimit and tag such independent video slice at the ATM layer for early discard. This involves the use of a tag cell differentiated from the rest of the data by its PTI value and a modified tag switch to facilitate the selective discarding of affected cells within each video slice as opposed to dropping of cells at random from multiple video frames. In addition a pre-emptive

selective discard algorithm to be implemented at the tag switch is proposed to carry out slice level discard operations based on slice priority, length and time to live.

Results obtained from simulations show a marked improvement in the performance of I-frames in terms of loss, throughput and delay compared to convectional transmission methods. Consequently, the performance of B-frames and (P-frames to an extent) shows a greater level of degradation.

1 Introduction

When a buffer at an ATM switch reaches overflow conditions, all incoming cells are automatically discarded, as they cannot be accommodated. More sophisticated techniques allow for low priority cells to be discarded first. VBR MPEG video is encapsulated into the payload of ATM cells and under congestion cells from any frame type can be discarded. When this occurs, reconstructing the original macroblocks of the frame becomes extremely difficult if not impossible. This can lead to error propagation as the subsequent macroblocks that were predicted from the lost one cannot be recovered. Quite clearly, it is much more preferable to lose predicted macroblock than to lose a macroblock from which subsequent macroblocks were predicted from. The ATM switch, however, is a layer 2 device and as such is incapable of interpreting application layer data. This means that when it discards arriving cells, and for whatever reason, it does so without determining first the relative importance of the cell in relation to the rest of the encoded video stream. The encoded video is highly hierarchical. This means that there are sections within the encoded stream that are much more important than others. The only option when attempting to protect this priority data is to reset the CLP bit on the cell's header for all cells that belong to base frames in order to depict their importance. A base frame is one from which one or more subsequent frames are predicted. This of course cannot always work, as the network is quite free to set this bit should conditions deteriorate.

The greatest advantage of ATM is its ability to transmit cells at high speed, mainly thanks to its constant sized small cell as well as its relatively low overheads. ATM cells are rapidly switched through based on their channel and path identifies only. It is these admirable properties of ATM that are proving problematic when compressed video and voice are involved. As explained earlier, the loss of base material in a video frame can result in propagating errors. This leads to a visible and annoying degradation of the content. It is our argument that where the loss of ATM cells must occur, great improvement in the received quality of the video can be had by having the ATM buffer controller discard those cells that belong to predicted frames, rather than those from base frames. We aim to show that the bad throughput of a video stream (measured at the application layer) can be greatly reduced as well as getting a marked improvement in QoS when the ATM switch adopts a pre-emptive selective slice discard algorithm (PSSD). The bad throughput is defined as the ratio of errered received slices as percentage of the total received. The PSSD discard algorithm discards cells at the slice level pre-emptively in an effort to maintain the queue level at a moderate utilization level. Slices arriving at or already in a queue are discarded based on their priority, length and time to live (TTL). We derived the TTL from user-specified maximum transfer delay.

Due to the encapsulations that are carried out by the various higher layer protocols and also due to the SAR process, the loss of a single cell can corrupt an entire packet or frame. If the receiver can carry out no corrective actions, then carrying these corrupted SDUs is a waste of precious network resources, as the end station will discard them. It is thus widely recognized that it makes greater sense to discard an entire higher layer SDU that has experienced some losses (and is thus corrupted) and therefore greatly enhance good throughput. The selective packet discard algorithm (SPD), as well as the early packet discard (EPD) algorithm addresses the challenges for IP over ATM transmissions. We propose extending these ideas to compressed MPEG video transmission by applying SPD or EPD at the slice level. It is widely recognized that the video slice forms the basic hierarchy of compression. This means that no decoding can occur until an entire video slice is first received. Based on the above logic, it therefore makes sense to drop a slice belonging to frame rather than a cell. Further it is well known that the MPEG hierarchy is composed of three types of frames each having a relative level of importance compared to the others. Thus we propose also to extend the selective and early slice discard algorithms so that they take into consideration the frame type when a cell is being considered for discard eligibility.

In order to carry a selective or early video slice discard policy at the ATM buffer, it is necessary that the ATM switch be enabled to allow it to recognize the boundaries of a video slice from a cell stream. We discuss some of the approaches that have been made to enable this feature, the majority of which rely on using the CLP bit in the cell header. The relative merits as well as shortcomings of these approaches are examined before we present our approach, which makes use of the resource management cell as a tag cell to accomplish the above objectives. Resource management cells are demarcated in ATM with a PTI value of 6 (110 binary) with the first byte in the payload being used to identify the type of RM cell. The tag cell specifies several parameters that identify the slice to the ATM tag switch including slice type, length and time to live. A pre-emptive slice discard algorithm applied to the tag switch results in improved buffer level performance for high priority video data at the expense of low priority data. The proposal is flexible in that it can be scaled up to tag frames and/or GoPs in addition to tagging slices. It is simple to implement since RM cells are already known to the ATM switch and all that is required is for additional routines to be added to the main routine that handles RM cells in the switch. These routines would force the ATM switch to recognize the tag cells and process them accordingly.

2 MPEG Video Compression Concepts

2.1 Introduction: Real time video applications

The large bit rates required for the transmission or storage of raw digital video means has until recently, the wide scale usage of video was restricted to the analogue domain. The introduction of the MPEG [7][28][29] compression standard has spawned a large number of digital video applications including DVD, CATV and HDTV. These can be divided into two categories; real time video applications and non-real time applications. Real time video applications include video conferencing, tele-school, broadcast or streamed video and instructional video. Instructional video includes the use of video to convey instructions to a lecture hall, or a team of doctors remotely located or communicating with the mars pathfinder to help it navigate the rocky terrain of a Martian planet. In all cases, real time video has stringent delay and jitter bounds and in some cases loss bounds. Certainly, the loss of vital compression headers on a video stream can lead to catastrophic consequences. Currently, three MPEG compression standards are available for efficiently compressing digital video. All of them are lossy algorithms and result in a somewhat degradation in quality level from the original video. MPEG 1 [41] is intended for the compression of non-interlaces video up to VHS quality levels [7]. MPEG 2 [44][7][10][15][16] followed up to accommodate higher quality level specifications (up to HDTV) and to support interlaced video. Both MPEG 1 and MPEG 2 have an average bit rate of 2 – 8 Mbits/s. MPEG-4 [8][41][42][30][38] was designed to support the compression of visual objects resulting in greater encoder side flexibility and with a bit rate that varies from as low as 60 Kbit/s to as much as 5 Mbits/s.

2.2 Video hierarchies

At the most fundamental level, a video stream is composed of pixels, each pixel having three octets, each one specifying the level of each of the primary colors (red, blue and green). A group of 8 x 8 pixels (2-D) is called block, and 2 x 2 blocks compose a macroblock. It is at the macroblock level that MPEG carries out spatio-temporal compression. Macroblocks may be lined up to form a slice. The MPEG standard [27] restricts a slice to a single line, although it may be composed of indeterminate number of macroblocks. Slices form a single frame and frames form a single GOP. Each hierarchy has an identifying header, which is useful to prevent large-scale propagation of errors through re-synchronization. As mentioned, compression is carried out at the macroblock level and this allows for great flexibility in fine-tuning the bit rate out put. Compression of video is carried out through the elimination of both spatial and temporal redundancy. Spatially, studies [7][28] have shown that the human eye is highly perceptive to changes in the image luminance levels and less so for changes in the chrominance levels. Furthermore, inter-frame correlation in any given video sequence is quite high so that a single frame looks very much like the one before and the one after it.

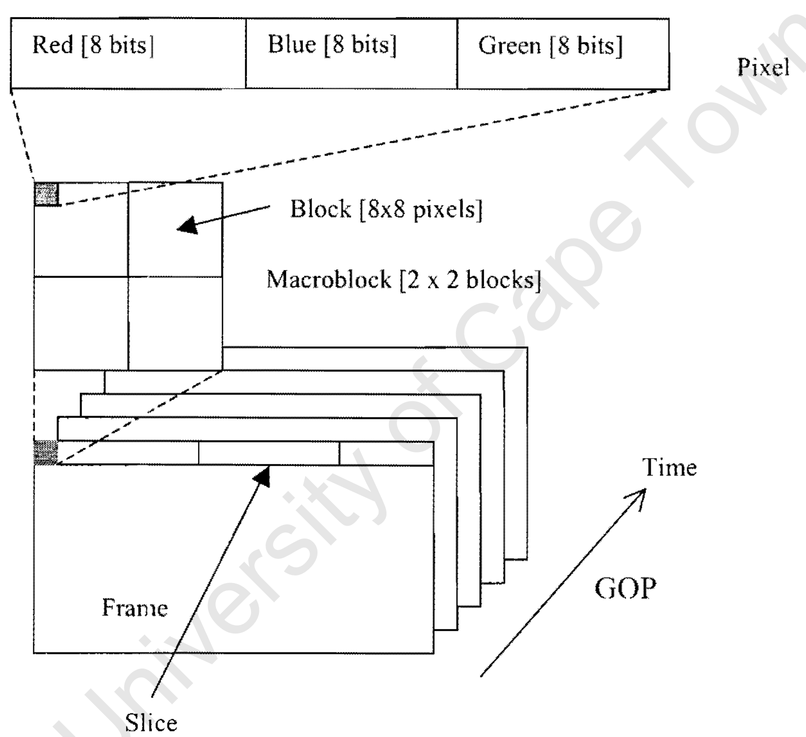


Figure 2.1: The hierarchies of raw digital video

2.3 MPEG 1 and MPEG 2

For spatial redundancy removal, the MPEG compression standards for MPEG 1 and MPEG 2 uses the DCT in conjunction with run length entropy codes such as the Huffman codes. The DCT is applied on each macroblock in a process that separates out the high frequency chrominance components from the low frequency luminance components. Further, a quantization process is carried out on the DCT coefficients before entropy coding is carried out. Quantization is carried out to eliminate the high frequency components by use of a quantization matrix whose entries are manipulated by a quantization factor. The so-called MQUANT [41].

Temporal redundancy removal is carried out using motion prediction vectors. Consider a macroblock on a frame i. In most cases, we can obtain this frame from the macroblock on the previous frame i - 1, displaced x units in the X direction and y units in the Y direction from a point of reference. Thus, it is only necessary to transmit (or store on a disk) the motion vectors for such a macroblock rather than the macroblock itself. To improve the quality of the decoded image, the error difference between the predicted and actual macroblock is also transmitted (after spatial compression has been carried out on it). We can predict a macroblock using only the previous frame or both the previous and next frame. It is therefore clear that three types frames are identifiable using the MPEG1/2 compression standard. These are

- *Intra coded (I) frames:* these frames are coded only in the spatial domain. They provide a valuable point for re-synchronization and therefore possibly stop propagating errors. I frames show the lowest compression efficiency.
- *Predicted (P) frames:* These are frames predicted from the previous frame. P frames have a lower bit rate that I frame. They do however; require the presence of the frame from which they were predicted from in order to be correctly decoded.
- *Bi-directional predicted (B) frames:* B frames predicted from both the previous and next frames and therefore show the highest compression efficiency. It is necessary to first receive both frames from which the B frame

was predicted from before it can be decoded. Thus, in the transmission of MPEG frames, B frames are transmitted after both P (or I or B) frames that are used for its prediction.

MPEG 1 and MPEG 2 achieve a compression ratio that is as high as 100: 1.

2.4 MPEG 4 Compression

The MPEG 4 video stream is hierarchically coded as follows:

- Visual object sequence (the entire stream)
- Video object (e.g. a face)
- Video object layer (each layer enhances the video object)
- Group of video object planes (similar to a GOP, but for a video object)
- Video object plane (similar to a slice but can be irregular shaped)
- Macroblocks (4:2:0 chroma format)

Every hierarchy offers a point at which the bit rate may be varied. The macroblock level is coded using the DCT. The level of compression is determined by the spatio quantization matrix whose magnitude can be varied. The video object plane uses shape coding to define the video plane object. In general, a rectangular shape is chosen except at the video object boundary. Research is ongoing for the optimal video object plane shape at the boundary of a video object. The video object plane may be intra-coded (I) or inter-coded (P or B). Video object planes make up a group of video object planes. Several groups of video object planes make up a video object layer. Video object layers enhance the resolution and overall quality of video objects. Several video objects combined make up the video sequence.

For example consider the following scenario: in a teleconferencing session, we have a speaker's face on a background. Thus two visual objects are identifiable, namely the face and the background. Techniques exist for deconstructing each frame into the various visual objects that make it up. The visual stream can thus be deconstructed into two visual objects, namely, a face object and a background object. For the background, we may choose to use only one layer and three layers for the face object. The first face video object layer contains only enough information to allow for accurate identification of the subject's face. The other two video object layers enhance the facial identification progressively. Furthermore, each layer (one for the background and three for the face) is composed of video object planes. Each video object plane has a regular shape (rectangular and we assume the same for the boundary case in order to simplify the example) and is coded (either intra- or extra-coded). The level at which a video object plane is coded can be controlled (spatial adjustments) this has an effect on the quality level of the video object layer's bit rate. In addition, for the face video object, we can control the number of face video object layers that are transmitted (temporal adjustments). Thus, MPEG 4 allows us to tune the final bit rate value in either the spatial or the temporal domain or both. MPEG 4 shows the widest variation in out put bit rate, from as little as 60 Kbits/s to as high as 5 Mbits/s.

2.5 Statistical Characteristics of encoded (MPEG) video

Statistically, MPEG coded video can be described in terms of the frame size (GOP size) distribution or in terms of the frame size (GOP size) auto-correlation function.
Table 2.1 shows some of the statistics gleaned from the empirical trace *"star wars"* from Bellcore [2][5] and figure 2.2 (a) shows the frame trace taken for the movie. The GOP pattern is also displayed in figure 2.2 (b). The movie is encoded using 12 frames per GOP so that there are just over 3000 GoP (40000 frames). Note the variable bit rate (VBR) nature of the sequence and that the peaks represent I frames. The trace is widely used in the field of MPEG traffic analysis and characterization. The average bit rate for this trace is 0.36 Mbps and runs for about two hours. In figure 2.3, we show the auto correlation function for GOPs of the above trace. We see that the fall off is not exponential and thus the trace has long-range dependencies.

Table 2.1: statistical properties of the trace "star wars"

Frame	I	P	B	All	GOPs
Mean [bits]	60144	23192	7216	15599	187185
CoV	0.33	0.64	0.67	1.16	0.39
Peak / mean	3.1	7.6	8.7	11.9	0.39

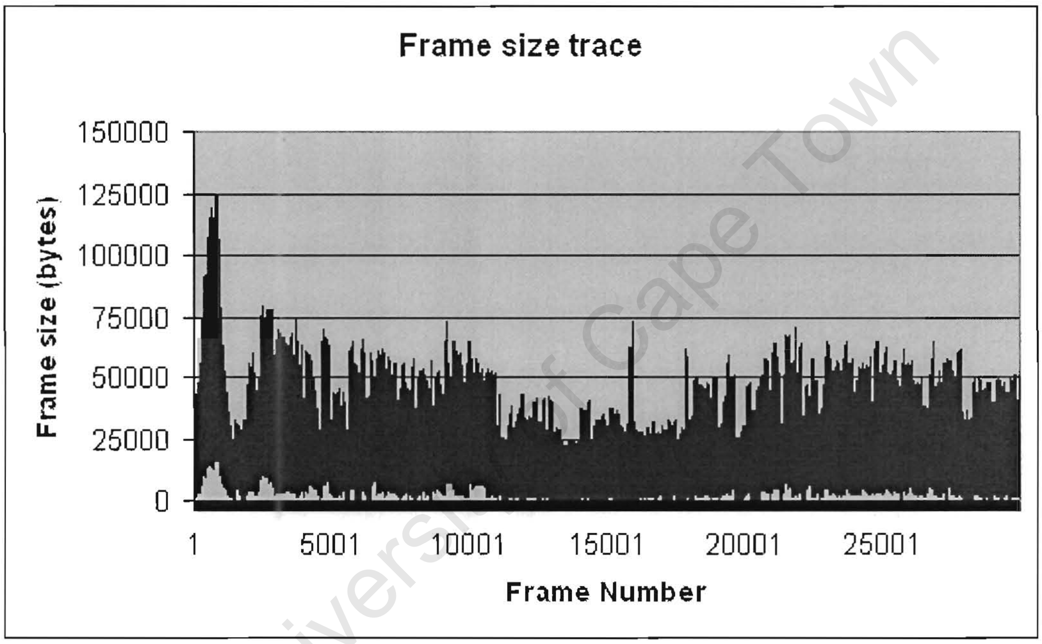


Figure 2.2: (a) Bit rate trace taken from the movie "Star wars".

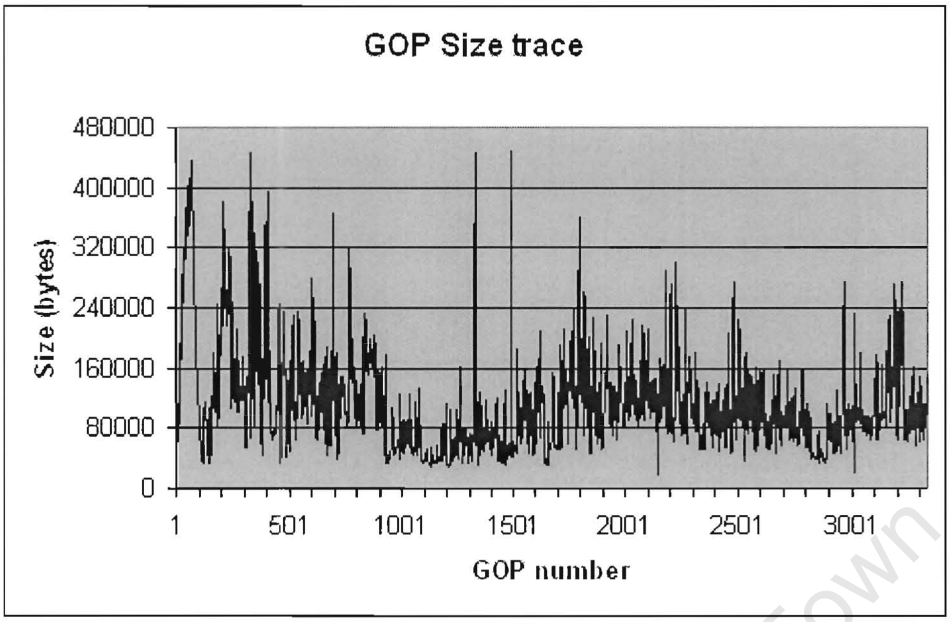


Figure 2.2: (b) The GOP trace of the same movie

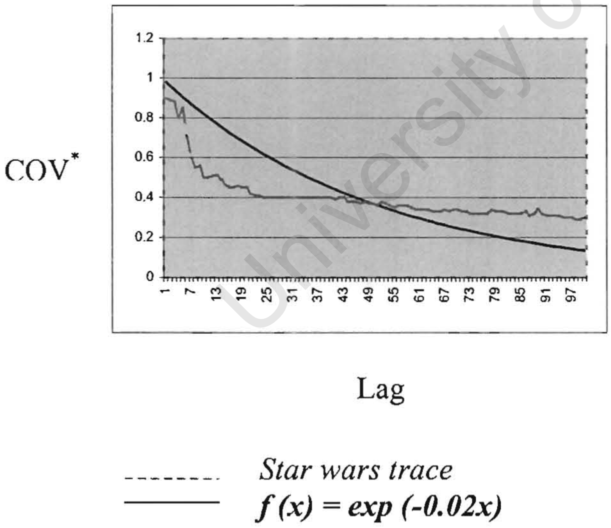


Figure 2.3: The "star wars" GOP correlation

2.5.1 Frame size measurements distribution

Figure 2.4(a) shows the histogram distribution of frame size for the I frame subsequence while figure 2.4(b) shows the histogram for the B frame subsequence. A similar histogram is obtainable for the P frame subsequence. We note that the distributions shown by each subsequence can be approximated by several mathematical distributions including the normal, lognormal and gamma distributions.

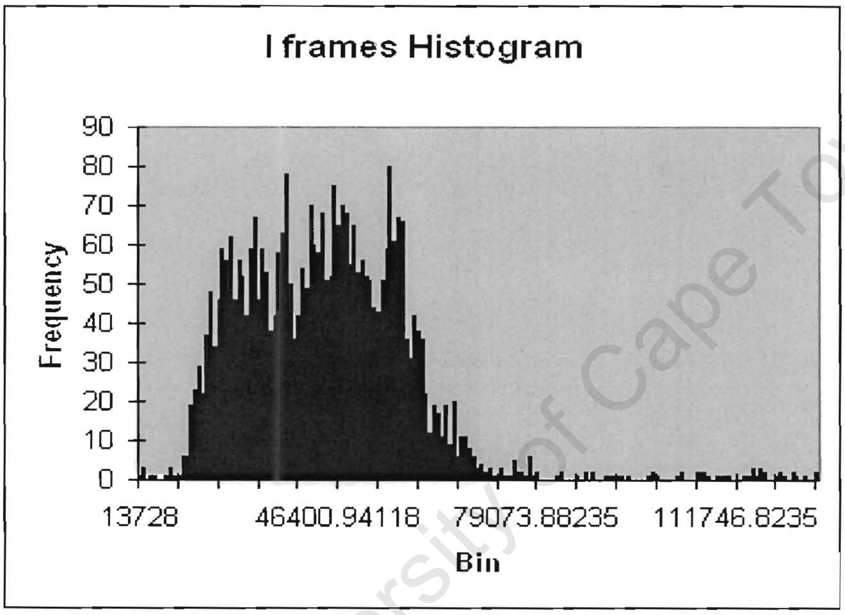


Figure 2.4: (a) I frame histogram subsequence

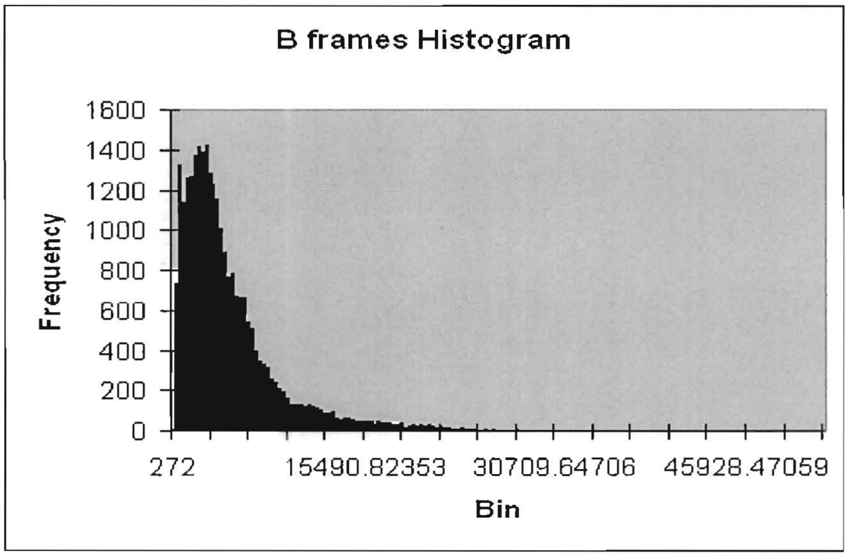


Figure 2.4: (b) B frame histogram subsequence

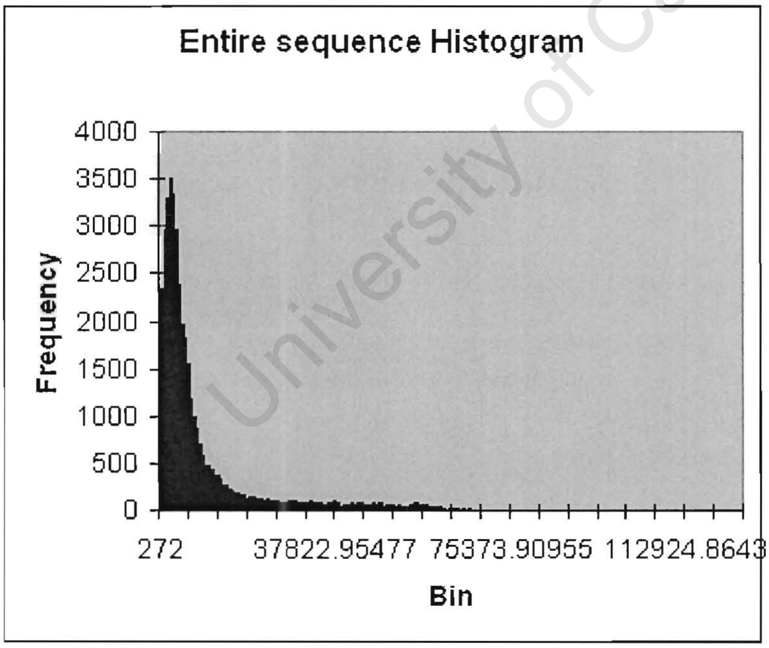


Figure 2.4: (c) The histogram distribution for the entire sequence. Note the similarities between this distribution and that of the P frames.

2.5.2 GOP size measurements distribution

The GOP size histogram is displayed in figure 2.5. It can be observed that the distribution is remarkably similar to that of the frame size histogram (and their subsequences). Thus, distribution can also be approximated by the same distributions as mentioned in section 3.4.1. The next section describes the modeling of MPEG GOP size distributions. GOP size models capture the long-range dependency of a video trace better than frame level models.

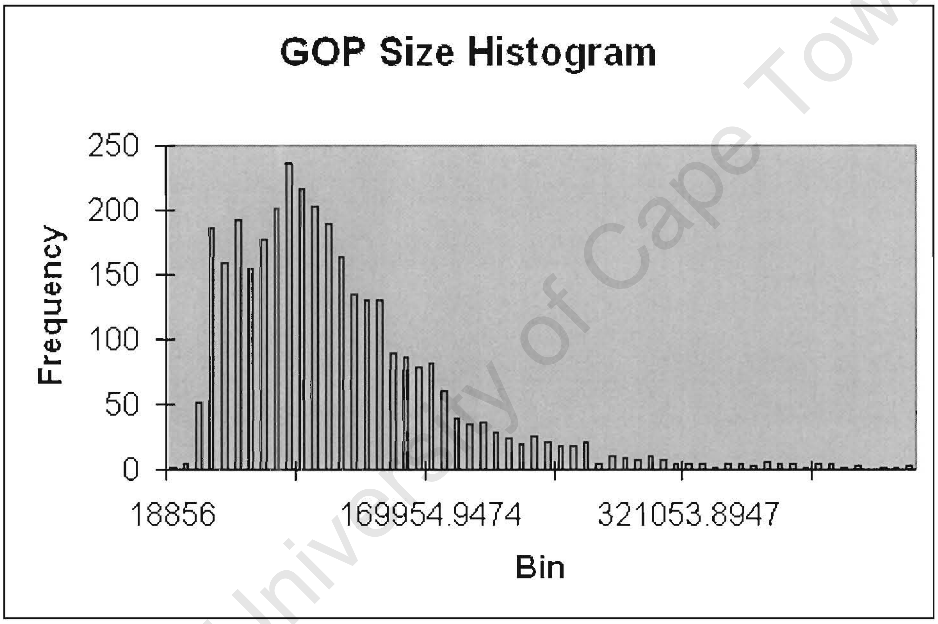


Figure 2.5: The GOP size histogram

2.6 Modeling MPEG Video

The behaviour of an MPEG video trace can be modeled mathematically in order to abstract certain aspects of the video trace for scrutiny. A mathematical model of a video trace is especially important in situations in which we may wish to study the behaviour of

MPEG video at a multiplexing buffer. Using a mathematic model, as opposed to using real trace could simplify the analysis process, while at the same maintaining a high level of accuracy. MPEG video models are designed to extract statistical properties of the video stream that have an impact on network performance. A second advantage for abstracting empirical video traces into simple models is a reduction in computational complexities and this allows us to test several network scenarios with minimum computing power, while maintaining the integrity of the results.
An MPEG video model also allows one to provide connection traffic descriptors, such as the PCR, SCR and MBS, for CAC/UPS purposes. Traffic models are generated by first obtaining fitting parameters from actual empirical data. This is done by matching certain statistical characteristics of an actual video footage and the model under consideration. It has been shown that the auto-correlation structure of a video sequence has the greatest impact on the queuing performance of a statistical multiplexer. It is therefore important that the selected model capture this property well.

There is a large variety of models that have been proposed in the literature, see for example [2][3][17][20]. But they can all be classified into three categories:

1. Markov chain based models
2. Auto regressive processes (ARP)
3. Fractal models

2 and 3 are beyond the scope of this study and will not be described further. See references [2][17] for further explanations of these types of models.

2.6.1 Multi-layered approach

In this approach, an attempt is made to capture the behaviour of MPEG video at various time scales. A typical trace will show quite a wide variation in statistical properties at the cell level frame level, GOP level and scene level. Table 2.2 shows the various time scales for the various layers of video.

Table 2.2: The time scales for various video layers

2.6.1.1.1 Layer	2.6.1.1.2 Time scale
Cell level	Several microseconds
Frame level	40 ms (PAL sequence)
GOP level	200 ms (12 frames per GOP)
Scene level	Several seconds (depends on nature of video)

Whichever layer is selected, the resulting model should be validated and it should have at least some the properties below

- Capture the long range dependencies
- Predict cell loss for reasonably sized buffers
- Predict other network performance parameter as accurately as possible

In the following sections, we describe three models that model the GOP size. All three models are markov chain based with the simplest them all being the histogram-based model. The first order markov chain is slightly more complex but captures the long dependencies better. Lastly, a scene based markov chain is described that has the ability to match the auto correlation features of the empirical trace much better than the other two.

2.6.2 The histogram model

The histogram model is known as a zero order markovian chain. A markovian chain is a state transition diagram with a finite number of states. Each state has a transition probability. Thus, we can create a transition matrix **P** whose entries specify transition probabilities. The size of the matrix is bounded by the number of states that are specified by the markov chain.

The histogram model is used for the GOP size generation using a geometric distribution. From the empirical data, the number of states **M** is established. Each state ***i*** represents a range of GOP sizes. The size of each range is determined by the number of states there are for the given range. The higher the quantization interval that is used, the better the model will be at correlating the empirical data. A closer inspection of the GOP size distribution reveals that it closely matches a log-normal distribution [1]. A gamma distribution is suggested in [2].

The frame size generation process is as follows. The model generates a GOP size. We multiply this with a scaling factor λ. The scaling factor is obtained by dividing the mean frame size of the empirical data with the mean GOP size. As an example, consider the following.

A typical GOP has an "I" frame, three "P" frames and eight "B" frames. Now suppose the GOP has a mean value of K_{TR} bytes, then by observing the empirical trace, we can get an estimate of the mean size for each frame type in that GOP. We let these be ψ_I, ψ_B and ψ_P then it follows that

$$\lambda_I = \psi_I / K_{TR}$$

$$\lambda_P = \psi_P / K_{TR}$$

$$\lambda_B = \psi_B / K_{TR}$$

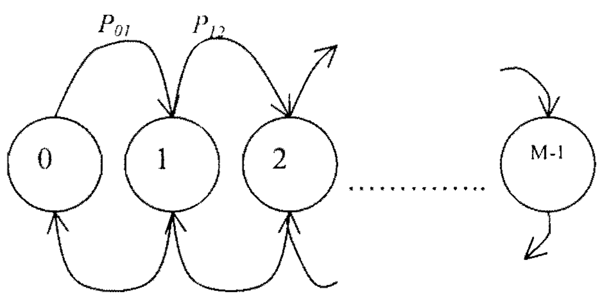
We then use these values on the GOP size generated by the model. Furthermore, this scheme has the added advantage that it is able to capture the periodic nature of the original trace without much effort.

Figure 2.6 shows a markov chain for generating the GOP size process. Below we also show the *1 X **M*** transition matrix that generates the transition probabilities. $\mathbf{P}_{ij}$ is calculated using the following equation

$$P_{ij} = n_i / N$$

Where n_i is the number of GOPs in the interval q_i, and

$$N = \Sigma n_i \quad (i = 0,1,2\ldots M\text{-}1)$$



$$\mathbf{P} = [\mathbf{P}_{(0)(1)}\ \mathbf{P}_{(1)(2)}\ \mathbf{P}_{(2)(3)}\_ \ldots \mathbf{P}_{(M-2)(M-1)}]$$

Figure 2.6: Transition diagram for the histogram model. The transition matrix *P* is shown also

Associated with each interval q_i is μ_{qi} that represents its mean value. For each transition from interval q_i to interval q_{i+1} the value, μ_{qi} is generated.

2.6.3 Higher order markovian chains

The histogram model, which in fact, is a zero order markovian, is sufficient for low lags. It cannot however capture the long-range correlation of the empirical data. This problem is tackled by using higher order markovian chains. Higher order markovians differ in the following respects to the histogram model.

- The number of states

- The estimation of the entries for the transition matrix

The number states or quantization intervals is calculated from the following equation

$$Q_i = (maxGOP - minGOP) / k$$

The value of k represents a quantization value. ***MinGOP* and *maxGOP*** are the minimum and maximumGOP values of the empirical trace respectively.

The transition matrix is now $M \times M$ size and its entries $\mathbf{P}_{ij}$ are estimeted as folows

$$\mathbf{P}_{ij} = n_{ij} / N_i$$

n_{ij} is the number of transition from state i to state j and

$$N_i = \Sigma n_{ik} \quad (k = 0,1\ldots M-1) \text{ for the interval } q_i$$

also note that $\Sigma P_{ij} = 1 \quad (j = 0,1\ldots M-1)$

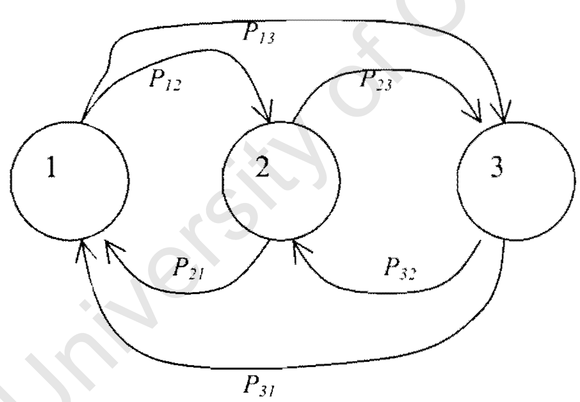


Figure 2.7: State diagram for a first order markov. The number of states is three

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P33 \end{bmatrix}$$

Figure 2.8: The entries for the transition diagram of fig 5

Figure 2.7 shows a three state markov chain. Note that in reality we need states of the order of 500 or more. Figure 2.8 shows the corresponding transition matrix. In the case of the higher order markovian, the current state depends on the previous state and not on its neighbouring states. This is in direct contrast to the histogram model where the current state depends on the previous state. The consequence of this is that with the higher order markovian it is not necessary that the next state be the neighbouring state.

As with the histogram model, for every transition from interval q_i a mean value μ_{qi} is generated.

Higher order markovian chains show a better auto correlation with the empirical data even at higher lags. The auto correlation function however, falls off exponentially and therefore the model fails to correlate at much higher lags. This model is therefore not suitable for estimation of say cell loss for large buffers. We may however, point out that for most real time traffic applications, the tight end-to-end delay bounds means that we may not use large buffers. The model is therefore adequate for the modeling of real time traffic.

2.6.4 Scene-based markovians

For the video sequences where the activity content is high, it is necessary that the model capture the various changes that go with scene variations. To accommodate this, we use a markov chain to generate the scene lengths and their variation in addition to either of the two GOP size-generating models described earlier. In [2] a method of capturing scene changes is proposed which we describe below.

Let $\boldsymbol{G_i}$ describe the $\boldsymbol{i^{th}}$ GOP size and $\boldsymbol{n}$ is the GOP number. Then

$$c_i = G_i / \sigma^2$$

where σ^2 is the standard deviation of the i^{th} GOP.

The scene boundary b_i is located where the inequality

$$|c_{new} - c_{old}| \, (n - b_{i-1} + 1) > \psi$$

holds.

c_{new} and c_{old} are the current and previous coefficients of variation (CoV) as calculated. ψ is a threshold value that we select.

We note that we can capture more scene changes if we lower ψ.

The above algorithm can be used to estimate the scene length S_l and the scene size $\boldsymbol{S_z}$. The scene size is obtained from summing up the GOPs for each scene and working the average value.

The scene length controlling markov will have $\boldsymbol{M}$ states and each interval q_i will represent a class of a scene. We can define the classes based on the amount of activity that we are willing to tolerate. This in turn is given by the threshold ψ.

The scene based markov chain captures the long dependencies much better than the other two models and is adequate for modeling high activity content such as live sport broadcasts and internet gaming.

3 ATM Networking Concepts

A network is a collection of nodes interconnected by links. The purpose of such an interconnection is to convey data from one node via an intermediate number of other nodes to a final destination node. The form and type of data often varies from one network to another, for example, a telephone network conveys voice signals whereas a computer network conveys packet signals. The connecting links take on different forms too, including fiber optic cables, twisted pair copper cables, radio wave signals and even satellite signals. Figure 3.1 shows a typical representation of a network.
The size of a network depends entirely on what the purpose of such a network is in the first place. In general, network sizes vary from a few metre squares to over a thousand kilometre squares.

A network that uses its various nodes to switch data to its destination is said to be a *switched network*. There are two common types of switched networks available today, these are *circuit switched networks* and *packet switched networks*. A good example of a circuit switched network is the current telephone network and a LAN is an example of a packet switched network. Circuit switched networks work by first establishing a direct connection to the end destination node before any attempt is made to send the data. This is clearly highlighted by telephone customers who must first dial the receiver's number, wait for the receiving party to connect the line by lifting up the handset, before any communications can take place. Indeed, until the 1980s, the circuit switched network was the only type of network widely accepted and therefore used.
On the other hand, packet switched networks work by breaking up data into discrete pieces known as packets (and sometimes frames or cells). The packets are then switched through the network by the various nodes on a store and forward basis. This means that the various nodes must first store the arriving packets while a suitable next hop route is found.

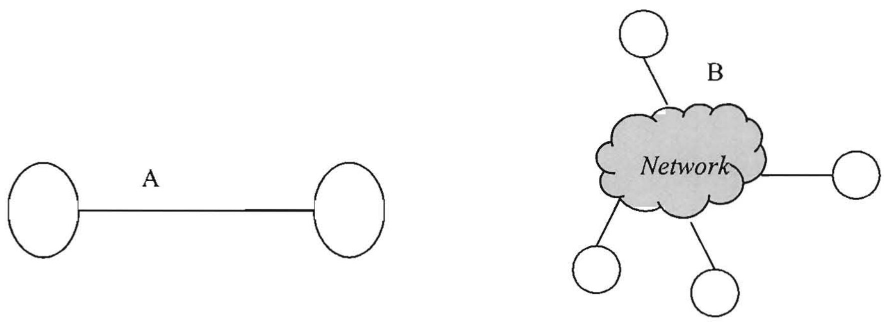


Figure 3.1: (a) shows a simple point to point network and (b) shows a complex one

This of course can be likened to the postal system where letters that arrive at a post office must first be sorted out before being send on their next leg of their journey. Moreover, just like the postal system, packets can be and often are misplaced, misrouted or simply lost due to various reasons. In addition, the act of looking for the next hop route takes time and thus delays packets to a varying extent. Packet switched networks are a new phenomena and are a direct result of the recent explosive growth of the computer industry and the Internet.

The major reason for the growth in packet switched networks is due to their efficiency in terms of bandwidth utilization. When two parties are connected in circuit switched network, the line that they use is unavailable to other connections, even during those periods that the line is idle. This, however, is not the case with packet switched networks since they are able to multiplex several connections over a single line. This type of multiplexing is known as statistical multiplexing and figure 3.2 highlights this very important concept.

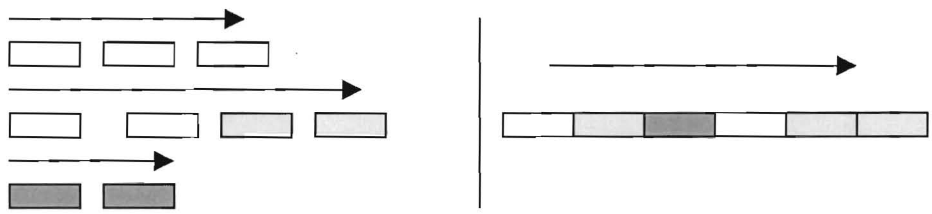


Figure 3.2: an example of statistical multiplexing results in higher link utilization

The key challenges facing statistical multiplexing are:

- Fairly allocating the link capacity to the different sources
- Congestion; Congestion occurs when the switch that is carrying out the multiplexing receives more packets than it can store in its available memory storage. Congestion leads to packet discarding. Also note that packets are delayed when they are stored in memory at a switch.

3.1 Designing a packet switched network: Common ground

In order to design an effective packet switched network, we must first understand the type of services such a network will offer. The modern network is composed of two basic components that communicate with one another to accomplish tasks. The first component makes request for service and is therefore known as the *client,* and the second component grants the service to the client and is therefore known as the *server.* This client/server process is shown by figure 3.3.

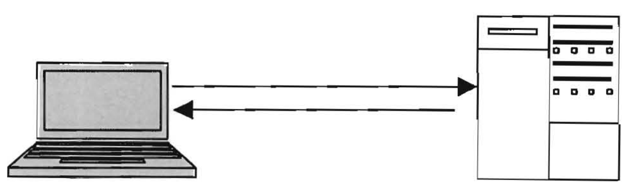


Figure 3.3: the client/server inter-communication processes is made up of requests and grants

The idea of server and client working together can be extended to the case of video conferencing. In this case both hosts must act as servers and both hosts must act as clients. Thus, video conferencing is characterized by high bit rates on both outgoing and incoming channels i.e. symmetrical transmission. Other examples of symmetrical transmissions include telephone calls and chat programs running on the Internet. Asymmetrical transmission include file downloads, video on demand, telemedicine etc. these are characterized by a higher bit rate on one channel than the other. Usually the channel with the higher transmission rate is referred to as the incoming channel.

3.2 Network abstraction: the layered stack

Designing a modern packet network is quite a complex and often difficult task. This is due the fact that modern networks are constantly changing and improving with time, with major modifications occurring quite frequently. The demands placed by the network users (applications) are also dynamic and their complexity is on the rise.

To deal with this network designers have developed a general network architecture that abstracts similar network functions into different layers. The most widely referenced of all network architectures is the OSI (*Open Systems Interconnection)* reference model [34][35][36]. This blue print of all network architectures has seven well-defined layers that encapsulate the basic functions of a packet switched network. Figure 3.4 shows this reference model.

Briefly speaking, the application layer defines those processes that directly interact with human users (in some cases other computers). This would include applications such as *File Transfer Protocol (FTP) and Hyper Text Transfer Protocol (HTTP).* These protocols define the different mechanisms for the transfer of different types of files, documents and other media. The presentation layer below this defines the different formats of the data that is to be transmitted. For example, the presentation layer has a set of protocols specifically designed to handle video traffic. Below this, is the *session* layer, which is used to tie together different traffic streams to the same session. For example in a conferencing session, this layer ties in the video frames to the sound stream.

The transport layer defines all the function that controls the flow of packets from one host to the destination host. The four layers defined above are located only at host nodes. They are not available at a network switch.

The network layer encapsulates those functions that are responsible for inter-nodal routing and traffic management. For ATM networks, the network layer would include the layer that manages and sets up *virtual channels* and *virtual paths*. The data link layer defines the actions responsible for the management of data at the frame level on link. This includes the actions taken to detect and correct bit errors and to manage link faults.

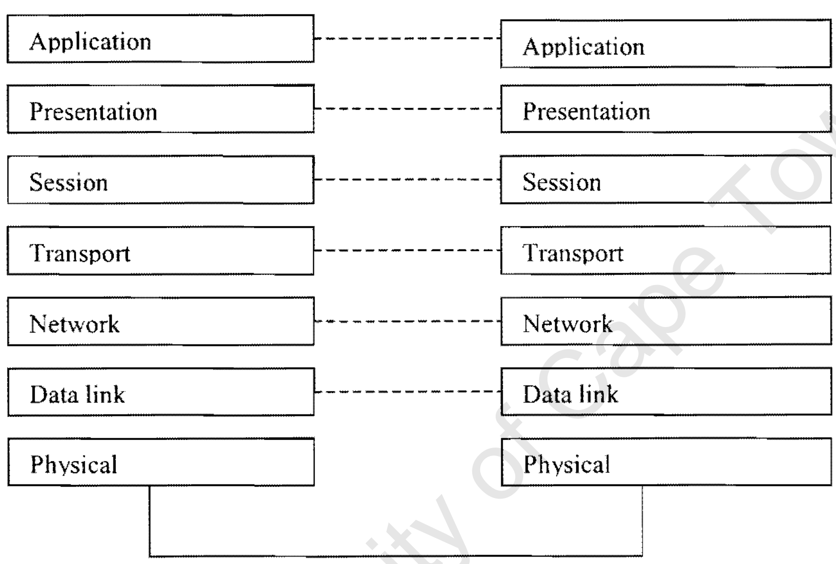


Figure 3.4: the OSI network reference model

The physical layer encompasses the medium of transmission itself and all the rules governing thereof. Some examples of physical media include optic fiber, twisted pair cable, radio frequency and co-axial cabling transmissions. Functions performed include switching and error detection and correction. Data at this level is handled as raw bit streams in form of electrical impulses (or electromagnetic radiation or light pulses).

At the very top of this model, we have information represented in digital format. This can be anything that we wish to convey across to the receiving party. For example, in the case of a video conferencing session, the information will be frames recorded by a video camera. Each frame will have undergone a compression process in all likelihood for the simple reason that the uncompressed frame is too large! The compressed frame is then broken up into smaller pieces referred to as *messages*. Identifying headers would then be affixed to the end (or trailer) each of these messages before being passed down to the presentation layer. The presentation layer also carries out any processing required before affixing its own header to the received message before passing it on. At the end of it all the message segment, now called a packet would be switched through as a series of light pulses through a network to the destination node where at each layer, the corresponding header is stripped off, the packet processed and then passed on up. The final product being a complete frame being displayed on a display monitor. Of course this ignores a lot of the complexity involved in the entire process included such issues as error detection and recovery, audio visual synchronization and buffer management and control to name just a few.

Specifying a network as a series of interconnected layers allows network designers several advantages. These includes

- It breaks down the design process into smaller well known modules and
- Allows for the expansion of a network without the need to overhaul the entire network.
- Allow network administrators to specialize therefore improving productivity.

These features mean that networks are dynamic and certainly a lot simpler to design, implement and manage. The addition of extra modules to a certain layer need not affect the other layers. For example, when and if a network operator decides to migrate from twisted pair cable to fiber optic cabling, the higher layers are completely unaffected by this move. The following sections describe the operations of the current two most important network-switching technologies, the asynchronous transfer mode (ATM) and the Internet protocol (IP).

3.3 The ATM Model

ATM is a frame oriented transfer mode based on asynchronous time division multiplexing and the use of fixed length frames referred to as *cells*. Each cell has a five byte routing *header* and a 48-byte *payload* field used for information transfer. The header contains fields, which are used by intermitted nodes for the purpose of identifying cells belonging to the same *virtual channel* and/or *virtual paths*. In particular, the payload type indicator (PTI) is used to identify the type of payload that the cell is carrying. The first bit is used to indicate user or management data (0 – user 1- management), the second bit is used by ABR connections for BECN and FECN indication (0 – no congestion experienced 1 – congestion experienced) while the last bit is used by AAL 5 framing to indicate the last cell of an AAL 5 frame (bit set to 1). In addition, if the first bit is 1 (management) the second bit being 1 and the third bit being 0 indicates a resource management cell (RM). The first byte of the payload is used to further identify the type of RM cell that it is.

ATM transmission preserves cell integrity as well as sequencing. The information field of the ATM cell is carried transparently through the network. The ATM transport protocol does not carry out any processing, like error control, on the payload.

ATM allows for the transportation of different and separate classes of services by defining separate adaptation layers designed to meet the demands of the different classes. Each adaptation layer provides to its users service-specific functionality such as clock recovery, re-transmission requests and error recovery. All adaptation layer specific information forms part of the 48-byte payload.

ATM networks are connection-oriented. Each connection is assigned a specific identifier that is in effect for the entire duration of the call. Connection identifiers are translated at every switching station from source to destination. These are known as virtual paths and virtual channels. Virtual path connections (VPC) are designated for specific traffic classes.

The ATM forum defines five service classes to be supported on the network. The first two, constant bit rate (CBR) and real time variable bit rate (rt-VBR) are capable of supporting real time traffic whereas the remaining three support non-real time traffic. These are the non-real time variable bit rate (nrt-VBR), the available bit rate (ABR) and the unspecified bit rate (UBR) services. The rest of this project will only concern itself with CBR and rt-VBR services.

The CBR services reserves enough capacity on a VPC to support a given transmission at the peak cell rate (PCR). This offers a higher QoS level guarantee which, in fact, is similar to that offered by a circuit switched network. CBR connections can be wasteful in the sense that they fail to continually utilize the allocated bandwidth resulting in very low link utilization. To counter this problem, the ATM forum has also proposed the VBR class of service (both rt-VBR and nrt-VBR), which allows a user to reserve bandwidth at both the PCR and sustainable cell rate (SCR). The SCR is defined as being the upper bound on the average cell rate (ACR) taken over a sufficiently large period. Multiple VBR connections are set up under the assumption that the individual peak of a flow has a low probability of coinciding with that of any other flow. This has the very important implication that the sum of bandwidth requirements of all the flows is equal to the sum of the SCRs of all the flows as opposed to being equal to the sum of the individual PCRs. This leads to very high link utilizations at low cell loss occurrences. CBR and rt-VBR will be described further in section 2.2.4.

3.3.1 The ATM reference model

The ATM reference model is shown in figure 3.5.

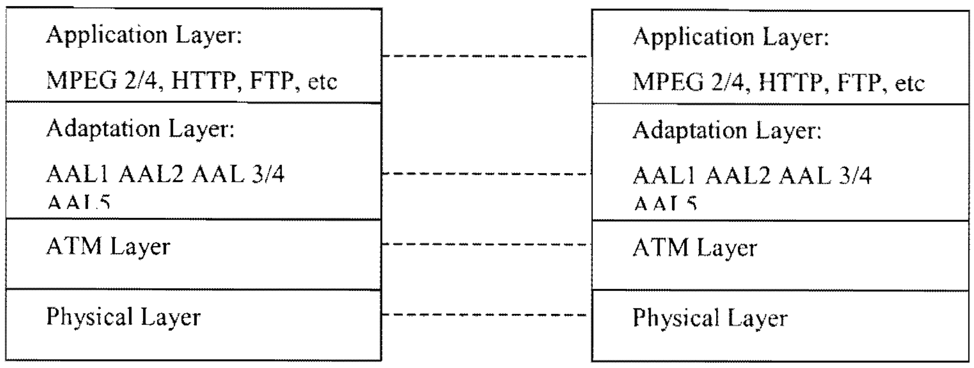


Figure 3.5: The ATM reference protocol

Applications such as an MPEG video stream reside over the adaptation layer. There are in fact different types of adaptation layers defined; each designed to support a separate traffic class with its own QoS requirements. AAL 1 is designated for CBR traffic classes because it maintains timing information as well some error detection and limited correction on the payload. AAL 2 was designed for rt-VBR traffic such as voice over ATM. AAL 5 was designed after AAL 3/4 were found have several short comings, for the transportation of framed data such as inter-LAN connections. One of the major problems that was found was the high payload overhead incurred by AAL 3/4. AAL 5 has been defined to have the least overhead and this makes it excellent for various higher layers such as IP and framed video. Adaptation layer protocols in addition are responsible for the segmentation and de-segmentation of the ATM payload. The protocol data unit (PDU) is passed to the ATM layer for switching purposes. The ATM layer in addition to switching through ATM cells (after affixing a payload header) also carries functions such as admission control as well as flow policing. The physical layer below the ATM layer carries out transmission of cells as raw signals (optical or electrical). It is the responsibility of the physical layer to maintain the integrity of the data.

3.3.2 More on the AAL 5

The AAL 5 was specifically designed for real time connection oriented traffic. The protocol overhead has been hugely minimized so that only the essential adaptation layer functionalities are maintained. This means that the protocol data unit is simpler to process resulting in minimal delay in the assembly/disassembly process. AAL 5 was initially conceived by the computer industry as direct response to the complexities and difficulties of implementation they had with AAL 3/4.More recently, AAL 5 has dominated as the adaptation layer of choice for the transmission of frame relay services and signaling messages. Like the other adaptation layer protocols, the AAL 5 has sub-layers that further differentiate the various functionalities of the adaptation layer. The service specific sub layer (SS) may be null and is responsible for providing service specific services to the applications above it. The common part (CP) sub layer is further divided into the common part convergence sub layer (CPCS) and the segmentation and reassembly (SAR) sub layer. The CP supports VBR traffic, both connection oriented and connection-less services. Figure 3.6 depicts the CPCS PDU for AAL 5. The payload has size range of 1: $2^{16} - 1$ (65535 bytes). Padding is provided by the PAD field so that the payload size is a multiple of 40. This, together with 8-byte header equals the size of one ATM payload. The user-to-user (UU) field provides a transparent information transfer mechanism between two AAL 5 users. The common part indicator (CPI) aligns the trailer to a 64-bit boundary. Other uses for the CPI are still up for study. The length field defines the CPCS PDU size and this can be used at the other end to remove the PAD field. The 32-bit CRC allows for the detection and correction of bit errors in the CPCS PDU payload.

CPCS-PDU [1 – 65535]	PAD [0-40]	UU [1]	CPI [1]	Length [2]	CRC [4]

Figure 3.6: AAL 5 CPCS sub layer PDU

The CPCS PDU is transferred to the SAR sub layer for segmentation/reassembly. The only overhead that the SAR sub layer makes use of is the payload type field in the cell payload. The last bit is used to designate the end of the CPCS-PDU so segmentation/reassembly can start[1]. The SAR-PDU is then transferred to the ATM layer where a cell header is affixed resulting in a 53 octet ATM PDU.

3.4 Support for real time video in ATM

ATM has been designed to offer support for real time traffic and this makes it ideal for the transmission of real time MPEG. Applications such a packetised video transport streams running over ATM are well suited to take advantage of the QoS guarantees that ATM offers. The adaptation layer of choice for the transmission of real time VBR video is the AAL 5. Its bare functionality provides the least complexity in the segmentation and reassembly process resulting in low latency for the video transport stream (TS). Figure 3.7 shows how the TS packets from the encoded video are encapsulated into the CPCS PDU. The SAR sub layer then assemblies ATM cell payloads.

The ATM forum has designated that only two TS packets are to be encapsulated into the CPCS service data unit (SDU). The reasoning behind this is to limit the potential jitter that can result from packaging too many TS packets [28]. The CPCS PDU has trailer appended to it (see section 2.2.2). The trailer has 32-bit CRC error detector as well as a 16-bit length indicator. This sets the maximum CPCS PDU size to $2^{16} - 1$ octets. Note that because the length indicator is for the CPCS-PDU, the loss of a single cell at the ATM layer results in the entire PDU being errored. It is not possible to determine which cell was lost. This is a great drawback for the AAL 5 adaptation layer. It is up to the application residing above the AAL 5 to provide for mechanisms that can handle a defective CPCS-SDU. For real time video transmissions, a re-transmission of the erred SDU is not viable. Error handling mechanisms such as error concealment may be used. Certainly, an erred CPCS-SDU is to be discarded.

[1] Note that first bit in the PTI field must be 0 to indicate user data type

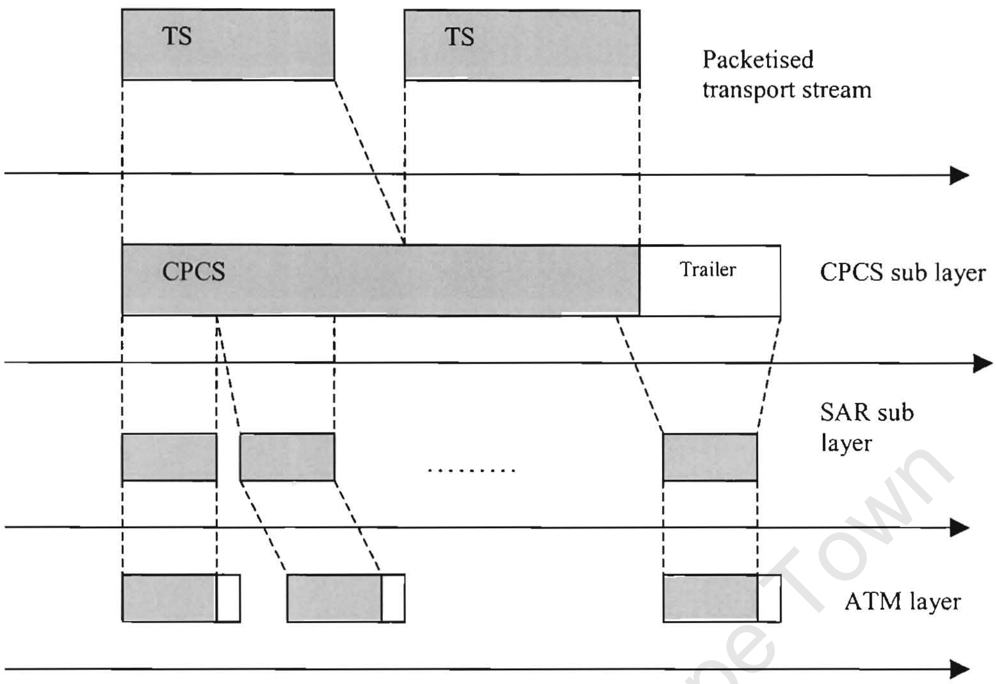


Figure 3.7: AAL 5 segmentation of MPEG transport streams

Despite this draw back, AAL 5 remains popular for supporting real time VBR MPEG over ATM.

3.4.1 QoS Issues in ATM

The ATM network provides QoS guarantees on the delay as well as loss probability for CBR and rt-VBR traffic. There is still a possibility, however, for cell loss events as well excessive cell delays to occur while in transit. The ATM forum [9] defines six QoS parameters, three of which can be negotiated at connection set-up. These are

- Peak-to-peak cell delay variation (peak-to-peak CDV)
- Maximum cell transfer delay (maxCTD)

- Cell loss ratio (CLR)
- Cell error ratio (CER)
- Severely errored cell block ratio (SECBR)
- Cell misinsertion rate (CMR)

The first three parameters are to be negotiated for at connection setup for CBR and rt-VBR traffic classes. Table 3.1 is further illustration of this.

Table 3.1: ATM QoS parameter specifications for the different traffic classes

Attribute	ATM layer service category				
	CBR	Rt-VBR	Nrt-VBR	UBR	ABR
Parameters					
Peak-peak CDV	Specified		Unspecified		
maxCTD	Specified		Unspecified		
CLR	Specified			Unspecified*	

* Specification of CLR for ABR services is network specific.

The network commits to guarantee safe passage to all traffic from a source, provided that the flow is in compliance with the negotiated parameters. CBR sources can emit flows at the peak-negotiated rate at any time and for any duration of time. The source is also allowed to emit at well below the peak rate and even be silent for extended periods of time. CBR connections are well suited, though not limited to, real time communications such as high quality video transfers and voice communications. A delay bound is set on CBR flow and cells exceeding it are considered valueless (or decreased value). CBR connections are characterized by low bandwidth utilization and therefore offer high cost per bandwidth unit.

Rt-VBR connections must, in addition to the CBR parameters, negotiate a sustainable cell rate (SCR) as well as a maximum burst size (MBS). The source is only allowed to emit cells at the peak rate for a maximum duration equivalent to the MBS otherwise cells must be emitted at a variable rate such that the average rate over a long period is equivalent to the SCR. There is also a delay bound on rt-VBR flows beyond which the usefulness of the data is diminished. Real-time VBR is designed to support applications requiring real time support, which transmit at a VBR. Rt-VBR connections can be statistical multiplexed for higher bandwidth utilization therefore offering a lower cost per unit of bandwidth than CBR connections.

ATM QoS parameters are guaranteed for the duration of the call provided that the source remains compliant. It should be noted that several factors limit a network's ability to guarantee the QoS parameters of a particular all of the time. These include

- The network cannot know the duration of a call at call set up. QoS service conditions can easily vary with time due to factors beyond the control of the network operator.
- Numerical QoS parameters may be given to precision beyond that which the network can support, measure or detect.

All this means that QoS parameters are supported by a network over the long term and over multiple connections with similar QoS parameters.

4 Transporting MPEG Video Over Packet Switched Networks: Issues and Challenges

The main requirement for the successful transmission of MPEG video over a network is a constant and predictable delay. This is necessary in order to recover timing information in the transport stream (TS). This in turn is required in order to synchronize the various program streams that are in the TS. The very nature of packet switched networks means that various network actions such as node buffering and multiplexing ensure that the delay will not be constant but variable. This introduces packet jitter, which makes it virtually impossible to carry out any clock recovery actions. For non-real-time video services, a holding buffer can be employed to smooth out the inter-arrival rate of the TS so that there is no jitter. Such a solution is however, problematic when it comes to real-time services since the holding buffer adds delay. A limited size buffer can still be used however, especially for the case where the real time service is non-interactive. Assigning a higher priority to real-time traffic is a viable solution for network jitter. In ATM, the CBR and rt-VBR classes of service are defined. The RTP protocol is defined for IP networks. In such a situation, a switch or router dedicates more resources to real-time traffic resulting in a lower delay (and low jitter).

Encoded MPEG can consist of an audio component and a video component. Both components have a timing model, which allows synchronization at the decoder.

A second feature of packet switched networks is their propensity to lose packets at switching points. A packet can be lost because it cannot be accommodated in a holding buffer. It can also be lost if there are errors in its routing headers or if the switching node misinterprets it. For real time traffic, packets that experience a delay beyond a limited bound are considered lost as they cease to be of any use to the end station.

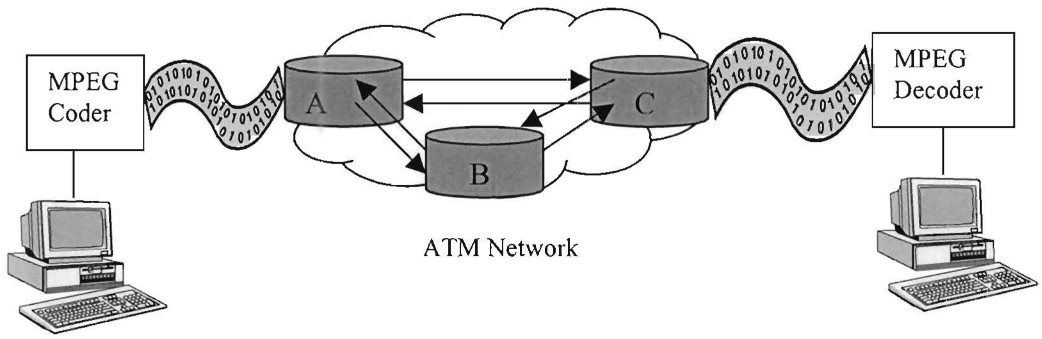


Figure 4.1: Encoded video across an ATM network

For the transmission of real time vide, the re-transmission of lost packets (cells) is not a viable option. In [49], a modified re-transmission scheme is proposed in which the re-transmitted packet is used for the correct decoding the next dependent frames. This is especially crucial in the case when an I frame packet is lost resulting in error propagation. If the packet is retransmitted, then the extent of the propagating errors is reduced. The scheme however does not scale well to large networks. A better approach for handling packet losses is the use of forward error concealment techniques (FEC) [43]. FEC works by inserting redundancy into the TS, which allows for the successful recovery of k packets from a total of n packets assuming n-k redundant FEC packets, are inserted. FEC however, adds computational complexity at the receiving terminal and in addition, losses due to packet burst can result in the loss of FEC packets and thus the scheme would fail in such an instance. FEC also adds redundancy to the encoded TS. A third approach works by attempting to conceal from the viewer the "glitches" that would be observed as a result of lost packets. Error concealment techniques are discussed in [45] and [48]. The basis of error concealment is the use of spatial and temporal neighbouring macroblocks to replace the affected macroblock. The use of error concealment suffers a set back when large-scale losses occur from a transmission burst. In this case, image inconsistencies become glaringly obvious to the viewer. A fourth approach is proposed in [43] (with further enhancements proposed in [44] by the same authors) in which encoder adds robustness in the TS in order to trap errors and therefore prevent them from propagating. This works by periodically inserting spatio-temporal re-synchronization points in the TS.

The price for robust TS is however, paid for by an added redundancy in the video code. Note that the chief reason for compressing video is the removal of spatio-temporal redundancy from the original trace. In most cases, the use of error concealment coupled with FEC in a video stream is sufficient to handle packet losses for real-time video. This is especially true for ATM networks where the bit error rate (BER) is expected to be of the order of 10^{-10}. The BER for IP networks is however considerable higher and in some cases intolerably so. In such instances, robust coding as well as delay-bounded re-transmission can be used with some improvements. See for example [5].

All the above approaches tackle the issue at the MPEG layer domain. This means that the ATM layer itself does not take part in any efforts to support video streams. Indeed, the ATM switch is designed to carry ATM cells transparently through the network in effort to reduce processing delays and hence increase the general throughput of the stream. The downside to this switching mechanism however, is that it becomes impossible to take advantage of some the underlying properties of video, which would certainly lead to a marked improvement in many facets of video transmission. Recently, proposals have been made to implement some form of tagged video switching which is aimed at allowing the switch to view some of the underlying properties of video. This would then allow us to implement a modified version of an early packet discard and selective packet discard algorithm on the video stream. These algorithms are implemented in IP over ATM transmissions to discard IP packets rather than ATM cells at the ATM layer. Recall that the PTI field is used to indicate the end of a CPCS-PDU. Thus when a frame (or packet) suffers some cell losses, it makes a lot of sense to discard the entire frame from the network. This reduces the bad-throughput. The selective and early packet discard algorithm carries out this function. Also in some instances, it may be required that some of the cells in a holding buffer be discarded. This could be to make way for higher priority cells. Once again, it makes a lot more sense to discard entire frames rather than single a cell. These discard algorithms can be modified for the transmission of MPEG over ATM. In this case, we have video slices, rather than IP frames and we must also take into consideration the special spatio-temporal properties of the video stream. Recall that the maximum size for an AAL 5 frame is just over 65Kbits and that the average size of a compressed video frame is around 2Mbits. This means that a video slice could easily fit into the payload of an AAL 5 frame. But recall that only two TS packets are allowed per AAL 5 frame and that a TS packet has fixed size of 188 bytes, which effectively means that slices are guaranteed to stretch over several ATM cells. Since we cannot know the exact size of each video frame slice beforehand, a tagging method is required to allow the switch to identify the EOB point for each video slice. The rest of this work describes how this can be achieved.

5 Video Tagging Methods

ATM networks are based on the transmission of small, 53 byte fixed sized cells. Most performance analysis studies have concentrated on cell level performance. However, in many cases, end users communicate using higher layer protocol data units, which are larger than cells and may be of variable size. Connectionless data, such as IP, use variable sized packets of up to several kilobytes. Digital video applications will send a slice as a single data unit, each slice also being several kilobytes long. The delay or loss of any one of a packet's cells affects the delay or loss of the entire packet. As such, what the user perceives is packet level performance, not cell level performance.

In order to carry out slice or frame level discarding at the switch buffer, it is necessary that the switch be able to clearly distinguish the demarcation points of each and every slice or frame of the video stream. The ability to do this will enable the switch to implement discard algorithms that carry out their operations at the slice level rather than at the cell level. Furthermore, to better take full advantage of the hierarchical nature of the encoded video stream, it would be required of the switch to also distinguish between the different frame type cells so that discard algorithms preserve the most important components of the encoded stream and discard the less important ones. These actions are aimed at improving the overall good throughput of the video stream while at the same time improving video quality and buffer performance levels. The good throughput is a measure of the number of received useful cells (to the decoder) compared to the total that was transmitted by the source. It is a measure of the number of error-free received slices (or frames). The video quality is enhanced, as I frames are mostly preserved by the network at the expense of B or even P frames. Buffer performance is improved as cells that should not be occupying buffer space are evicted before they cause deserving cells to be blocked. This decreases jitter as well as delay for the good throughput. Thus, the ability to identify slice level and even frame level boundaries as well as the ability to tell apart an I frame cell from say a B frame cell, are the main requirements of a switch in order to offer optimal performance to encoded video.

To enable the above requirements in the ATM switch, it is necessary that the transmitted cell stream containing the encoded video be tagged appropriately. Such a tagging method would allow switching software to recognize the structure and properties of the video that is being transmitted. This would then allow the buffering mechanisms to treat the video cells appropriately. Several methods have been proposed for tagging ATM cells and we discuss some of them

The most basic approach involves the use of the CLP bit being used as an I-frame cell marker. Indeed, this is one the functions for this bit, to indicate high priority traffic, and certainly, an I frame cell belongs to this category. Still, the bit has every chance of being set to low priority (CLP = 1) by the network resulting in I frame cells being treated as B (or P) frame cells by the switch and its buffer management algorithms. This greatly reduces this method's ability to tag encoded video. This method can be enhanced by using both the AAL 5 EOB marker bit in conjunction with the CLP bit [59][60]. This provides scope for four levels of tagging encoded video traffic and therefore increased flexibility.

In this implementation, the CLP bit is used to indicate the type of frame the cell belongs to. In this case only two possibilities exist; I or P/B frame or alternatively; I/P or B. The EOB marker is then used to indicate the end (or start) of a video slice. Notice that under light loads, the CLP bit can be set to vary between IP and B levels and under heavy loads between I and P/B levels.

But note that this method is still using the CLP bit and is still therefore error-prone. In addition, the AAL 5 EOB marker bit has reserved to indicate the end of an AAL 5 frame. This is to allow the SAR sub layer to begin the reassembly process. Further, this method cannot be used to tell us the length as well as the TTL of the tagged slice, which could be used by the switch buffering algorithms to determine the delay and losses that a particular slice has suffered while still in the network.

In this dissertation, we propose a tagging scheme that operates at the ATM layer but does not make use of the CLP bit.

The transmitted video is tagged using the RM cell as a tag. We call this type of RM cell a tag cell. This cell is defined using a PTI value of 6 (or binary 110). PTI value of 6 is used to define RM cells. RM cells are used in the ATM network to carry out various network management functions including flow control operations in ABR traffic. The tag cell is used to demarcate the slice boundaries of a video. They can also be used to identify the boundaries of frames and GoPs. The SAR function at the source end inserts a tag cell at the start of each video slice. This cell carries with it crucial information about the incoming slice including its length (in cells), the type of frame the slice came from as well as the time to live (TTL) for the slice. The TTL is used for jitter control. Intermediate switches that receive the tag cell use the information to determine the free buffer space that should be allocated to the incoming slice as well as to determine the amount of time that the slice can be allowed to spend in the buffer before it is deemed valueless and discarded. If the switch buffer does not have sufficient space to accommodate the entire slice, then the following would occur

- If the slice belongs to an I frame, the buffer manager attempts to free up sufficient space in the buffer by knocking out low priority slices that are already in the buffer
- If the slice belongs to a low priority slice, then it is simply discarded and the tag cell is marked accordingly and forwarded to the next switch. As a rule, tag cells are not to be discarded as they still serve a useful function to the end receiver i.e. they serve to inform the TS sequencer that data was lost as well provide the identity of the data.

5.1 Tag Cell set up

The tag cell to be used for video tagging purposes would be set out as shown below. The PTI field is set to 110_2 to indicate a resource management cell. The designation serves as a general mechanism for defining new communications mechanisms. RM cells are distinguished from each other using the first byte of the payload. In this case it would be set to indicate that the cell is an MPEG video tag cell. We suggest the value 0x02 (hex notation for 00000010_2) be used for this. The next byte would identify the name of the object being tagged by this RM cell. The object referred to can be a GoP, frame or slice. The remaining bytes in the payload would be set up as follows; The Object length would be used by the switch to verify the integrity of the transmitted object. For the case that the tagged object is a slice, then the switch can use the length field to determine if a slice has suffered losses. This it does by making a comparison between the value as given by the length field together with an actual cell count carried out on the cells. If the slice has suffered losses, then it is discarded. This pre-emptive action serves to free up buffer space and improve the overall good throughput of the video.

In normal transmissions, the object being referred to would be a video slice. Its end-to-end time to live would be indicated by the TTL field while the object sequence number would be used to indicate the slice sequence number for the current frame. In the case that the object is a frame, the object SN would indicate the number of frames received so far for this GoP. The object type would indicate the priority level of the current slice. In general, I frames would be given the highest priority while B frames would have the lowest. This however, is left to individual operators to set their own priority levels. The TTL is actually the maximum delay that a slice can have before the information it carries ceases to be of use to the end decoder. The TTL is calculated at call set up from the maximum transfer delay, as given by the user. At the sending host side, each slice is given an end-to-end TTL that it has across the entire network. When a switch receives the tag cell, it copies this TTL value to internal memory and it is used to initialize a timer. If the timer reaches zero, then the slice should be discarded as its play out time has expired. On the other hand, if the slice is de-queued before the expiration of the timer, the tag cell's TTL value is changed to the current value of the timer. Thus the TTL decreases as the tag cell is switched through the network. In order to improve the accuracy of this model, the link transfer delay should be factored in when setting the TTL value at the sending host. In most cases, the link transfer delay simply depends on the length of the link and, especially with fiber communication, can be safely assumed negligible. Nevertheless, to be safe, it is better to take this into consideration when setting the TTL.

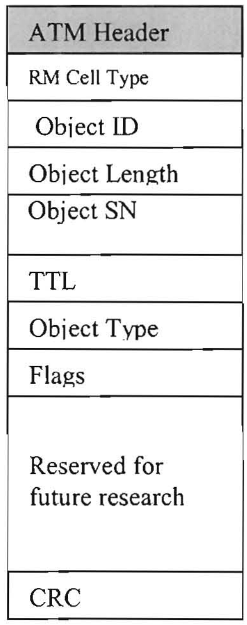


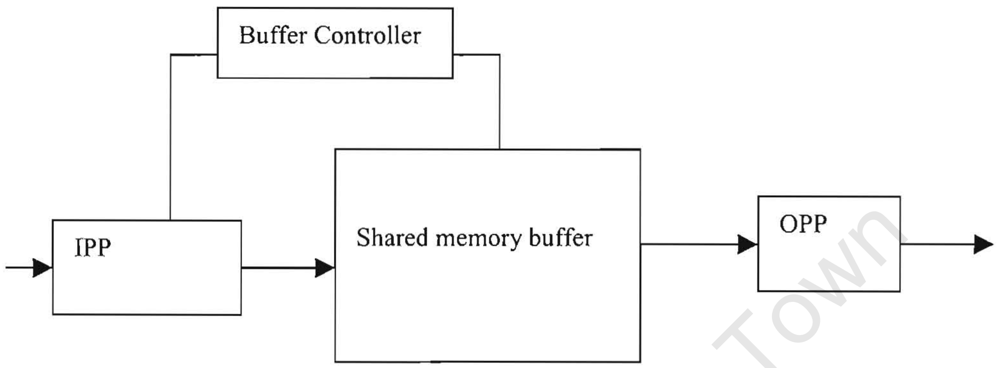
Figure 5.1: The structure of the MPEG tag RM cell

The first bit of the flags field tells the switch (or receiving host) whether the cells belonging to this slice have been discarded or not. The second bit is used to lock the slice while it is still in the queue and the switch is awaiting the rest of the cells that belong to this slice. This allows to the switch to perform an integrity check on the received slice. Currently, these are the only uses identifiable for the flags field and the rest of the bits are reserved for future use.

The remaining bytes in the payload are reserved for future use pending further research. The CRC is used to carry out limited bit-error detection and corrections on the payload.

5.2 Switch set up

A typical switch that would be able to interpret the above cell would be modified to function as described by figure 5.2. It would consist of a modified input port processor, buffer controller and output port processor. It would use a shared memory buffer to implement per VC queuing using a linked list structure. A linked list allows for random access into the queue. This is crucial since we would like the buffer controller to have the ability to remove cells that are already enqueued.



IPP: Input port processor OPP: Output port processor

Figure 5.2: Switch implementation to handle tagged MPEG video

5.2.1 The Input port processor

The input port processor is responsible for receiving every cell that arrives at the switch. It is responsible for detecting cell boundaries as well as carrying out CRC checks on the header. CAC and UPC functionalities are also implemented at the IPP. At call setup time, the IPP sets up a VC in which tagged video is expected. The usage parameters are used to set up space for this VC on the shared memory buffer. During normal transmission, the IPP is also responsible for determining the integrity of the received slices (not cells). When a tag cell is received, the IPP determines, from the length field of the tag cell, the number of cells for this slice. A counter is then initiated and cells arriving are counted up to but excluding the next tag cell. The count value is compared to the value as specified in the length field. These two values must match otherwise the entire slice is discarded. A likely reason for a mismatch is the fact that one or more cells belonging to the slice was lost. Also, a tag cell could also be lost, in this case, in effect; two slices rather than one are being discarded. When a tag cell is received, its location in the queue is recorded and a counter is initialized. The length field value is then compared to the current queue occupancy level to determine if this slice can be fully accommodated. If the buffer controller reports that there is insufficient space in the queue, then the slice is discarded, provided that it is a low priority slice. The buffer controller frees up space if the slice is high priority and provided that there are slices in the queue of lower priority.

The second bit of the flags field is set to indicate that this slice is locked. A locked slice cannot be removed from the queue by the OPP. When the last cell has been received, a comparison is made between the slice size as given and the actual slice count. If these match, the bit is reset and the slice can now be processed by the OPP. If they don't match, the tag cell is still unlocked and all cells belonging to that slice are discarded. The first bit of the tag cell's flags field is set to indicate that the slice was discarded.

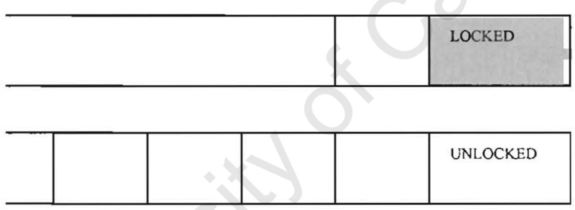


Figure 5.3: A slice is only unlocked when the last cell of the slice is received

5.2.2 The buffer controller

The buffer controller is responsible for the management of buffer resources. When a VC is setup, the buffer controller reserves a certain amount of space in the shared memory buffer exclusively for that VC. This space is reclaimed when the VC is torn down. The amount of space that is reserved for the VC depends on the usage parameters that were

entered into between the network and the user. For rt-VBR traffic, the maximum jitter, PCR and SCR are used.
Once the space for the VC has been reserved in the buffer, the controller, in conjunction with the IPP, is used to control and manage the admission of cells into their respective queues. Under stressing conditions the buffer controller can discard certain cells already in the queue that would meet certain defined criteria. This would be cells that have had a waiting time greater than the maximum allowed or in some cases, these would low priority cells that must make way for higher priority ones.

In order to control the TTL of each slice, the buffer controller regularly sweeps through the queue starting from the head to the tail. For each enqueued tag cell, its TTL is decremented by T_c which is the inter-sweep period as defined by the network operator. If as a result of this reduction, the tag cell's TTL is less than or equal to zero, the cells that were tagged by this tag cell are discarded from the queue. The tag cell itself is marked by setting the first bit in the flags field.

Another function of the buffer control would be to flush the queue when a high priority slice arrives and it cannot be enqueued due to lack of space. In this mode, the controller sweeps through the queue looking for slices with a slice discard eligibility (SDE) greater than that of the arrived slice. The discard eligibility is defined below. Every time a tag cell in the queue is found having a higher SDE, the cells belonging to that slice are discarded and the tag cell is marked. The process ends when either the end of the queue is reached, or enough space is created for the arrived slice. Note that if the end of the queue is reached, it can only mean that the controller was unable to free up sufficient space for the arrived slice. In this case, the tag cell of the arrived slice is unlocked and marked as discarded and all arriving cells are discarded until another tag cell is received.

5.2.3 The output port processor

The output port processor is responsible for transmitting cells to their next destination. It removes cells from their respective VCs and slots them into the correct outgoing VC port.

The VCs are serviced based on their priority levels. This means that low priority VCs are not serviced while the high priority ones still have cells in their queues. Rt-VBR traffic has the second highest priority below CBR connections. The OPP works as follows

- If there is any cell to be serviced at the head of the queue, a check is made to determine if it is a tag cell (PTI = (110) or ordinary data cell (PTI = 0xx)
- If it is a tag cell, the second bit of the flags field is checked to determine if this slice has been unlocked
- If it is locked, the OPP must wait for it to be unlocked. This adds a delay equal to the slice jitter
- If the tag cell is unlocked or if the cell is an ordinary data cell, it is removed from the queue and placed into the OPP's internal memory
- VPI/VCI values are obtained and translated using a translation table
- The cell is then transmitted through its outgoing interface

Because ATM uses a fixed size cell, the processing time per cell is identical for all cells at the OPP. Thus from this we can derive the queue waiting time per cell which only depends on the queue size at the time the cell is enqueued. In general the waiting time in a queue for the n^{th} cell is

$$W_n(t) = \sum_{i=0}^{n-1} W_i(t) + W_{CPU}$$

Where W_{CPU} is the processing time for each cell. Recall that this value is constant for ATM switches. $W_n(t)$ is variable for the following reasons

- At the time a cell arrives, the number of cells already in the queue is random
- Depending on the service policy of the OPP, service rate (not cell service time) can vary. Consider the scenario whereby the OPP must service two rt-VBR connections and one CBR connection. In this arrangement, the two rt-VBR channels are serviced in round robin fashion as long as the CBR connection has no traffic except that the OPP must reserve a number of clock cycles for the rt-VBR connections. From this we can see that the rt-VBR connections will have a

higher service rate when the CBR queue is empty and conversely a lower service rate when there is traffic in that queue.

The maximum jitter experienced by a cell is therefore fixed by the maximum queue size and the TTL, whichever one is smaller. If the TTL is low, cells can only experience a maximum delay equal to TTL seconds whereas if the queue is small, the maximum delay is likely to be suffered by the cell that is enqueued at the last slot in the queue. Using the equation above, $\boldsymbol{n}$ is equal to $\boldsymbol{Q_{max}}$.

5.3 Design issues

The sending host specifies that the call to be placed uses tagged video. For real time VBR video, the sending host also specifies the usual usage parameters as well as QoS requirements as required by a rt-VBR connection. The CAC then establishes a VC to the called host using a routing protocol such as PNNI. The user-specified maximum transfer delay is used as the VC's TTL. Each tag cell that is created has this value specified in its TTL field.

Once the call has been setup, the sending host begins to transmit the encoded video stream, placing a tag cell at every point that a slice starts. The tag cell fields are correctly specified and the cell is transmitted along with the slice cells that it is tagging. Note that the tag cell is transmitted using the same VC as the rest of the traffic. This is what is commonly known as in band signaling. In band signaling increases the user overhead.

A great concern of this scheme is what happens to the tag cell when it encounters a switch that does not recognize it. Since the tag cell has the format of an RM cell and RM cells are in most cases forwarded. This should not pose a great problem. However, if the switch were to interpret this cell as being erred and therefore discard it, then the entire scheme would fail. The next upstream switch would be waiting for a tag cell that will never arrive and hence all arriving cells would be delayed indefinitely. It cannot discard them since it has no way of knowing their TTL.

One way of solving this condition is to allow the tag switch to operate in two modes. The first mode is entered into by default and in this mode, tagged switching is assumed.

Assuming that the switch downstream is incapable of tag switching and also that the same switch will drop all tag cells with a probability of one, then no tag cell will ever reach the current switch. Upon receiving the first cell of the video stream, which will be an ordinary cell, the switch then goes into the second mode. In this mode, no tag switching is assumed and all incoming cells are treated as ordinary ATM cells. It is quite obvious that all upstream switches will also revert to tag-less switching modes too. This basically means that tagged switching is turned off for all switches from the switch that does not support tag switching. The switch remains in this mode until it receives a new tag cell from its downstream neighbour. Figure 5.4 illustrates this.

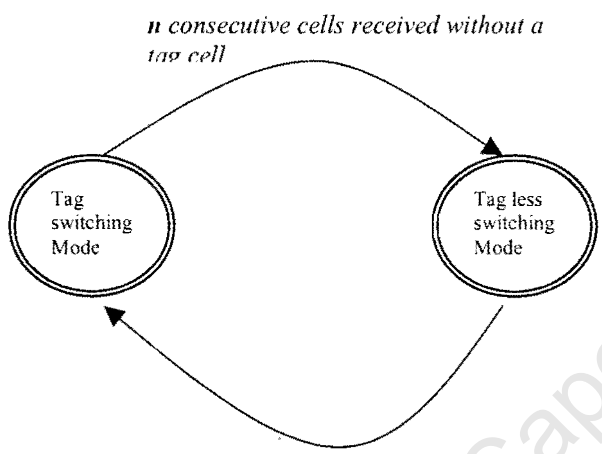


Figure 5.4: Switching modes to handle non-tag switches from downstream

Another design issue is the switching mode to be implement for the above switch. Recall that in our implementation above, the OPP must wait for the entire slice to be received first before transmitting it. This is regarded as store-and-forward switching. The entire slice must first be received and an integrity check is performed before the slice can be forwarded. This switching method incurs delay to the overall transmitted stream. It also has the unintended effect of shaping the traffic stream so that the traffic descriptors as defined by the user are altered. The main advantage to using a store-and-forward switching mode is to remove dead cells from the stream. Dead cells carry information that has no use to the receiver and therefore waste network bandwidth.

Alternatively, cut-through switching could be used in which the OPP begins to transmit a slice as soon as the first tag cell is received. In this mode, the slice is never locked. This removes the delay associated with waiting for the last cell of a slice to arrive and also maintains the traffic descriptors as defined. With cut-through switching, no length checking is performed on the slice and if one has suffered some cell losses, it will be transported right through the network. This has the effect of increasing the number of dead cells transported (bad-throughput).

A hybrid switching mode known as fragment free switching can be employed to maximise the advantages of cut-through switching (low latency) and store and forward switching (error free slices) while minimizing the disadvantages of both switching modes (carrying through errored slices and high latency). In this mode, the first $\boldsymbol{n}$ slices are switched through using the store and forward switching mode. After the $\boldsymbol{n}^{th}$ slice has been received and it is established that none of the previous $\boldsymbol{n}$ slices were errored, the switch then reverts to cut-through switching mode. This mode is maintained until a slice is found to have been errored. Note that an errored slice is one that has lost some of its cells. This switching mode maintains the advantages of both switching modes while minimizing the disadvantages.

A third design issue to be taken into account is the overhead incurred as a result of inserting tag cells into the video stream. Tag cells can be used to tag slices, frames or even GoPs if so desired. The overhead for tagging slices can be calculated as follows

Let the number of cells per slice be uniformly distributed with a mean β and variance γ. Thus the slice overhead ϑ is

$$\vartheta_s = 1 / (\beta + \gamma)$$

Assuming $\boldsymbol{n}$ slices per frame, the frame overhead ϑ_f is stated as

$$\vartheta_f = \frac{n}{\sum_{i=0}^{n} \beta_i + \gamma_i}$$

And per GoP = $12\vartheta_f$ assuming a GoP pattern of IBBPBBPBBPBBI...
Thus the frame overhead decreases for large sized slices per frame and vice versa.

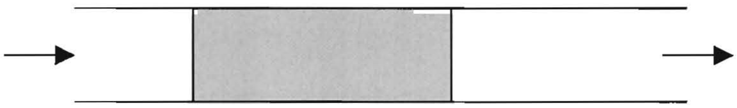
A tag switch on its own cannot result in greater enhancements for transmitted video. Equally critical is the need to have in place buffering mechanisms that will take advantage of the tagged video stream as described above. Thus in the second part of our work, we propose a pre-emptive selective slice discard algorithm that is designed to do just that.

6 The Pre-emptive Selective Slice Discard Scheme

The pre-emptive selective discard algorithm (PSSD) that we propose is designed to support MPEG video ATM cells at the slice level. This can however, be extended to support video at the frame and even GoP levels. The algorithm is designed to work hand in hand with a suitable video tagging design such as the one proposed above. The PSSD scheme is actually a set of cell discard tools applied to the ATM switch buffer in order to enhance the performance of MPEG video through a packet switched network. The PSSD recognizes and takes advantage of the fact that the delay, jitter and loss requirements of real time MPEG video are unique due to the compression algorithm employed. First, it is greatly preferable to lose cells that belong to B frames than those that belong to I frames. Recognizing this fact, the PSSD takes into consideration the frame type of each cell before discarding it. Secondly, real time video has finite fixed delay beyond which the usefulness of the data ceases to exist. The PSSD therefore also identifies cells in the video stream whose play out time has expired. Such cells are candidate for discard. This also improves the performance of those cells that are still within their play out time. The I frame is a point of reference for all subsequent frames in a single GoP and recognizing this fact, the PSSD does not discard slices that belong to I frames, even if their play out delay has exceeded the maximum. Lastly, the PSSD discards whole slices rather cells from the video stream. This has the effect of greatly enhancing buffer performance for the overall traffic.

6.1 Buffer operation states

Consider a queue buffer into which tagged video cells are being enqueued while awaiting transmission. For this queue, we define a lower and upper threshold level (L_T and U_T). L_T defines the queue level below which no pre-emptive buffering strategy is required. In this state all arriving slices can be accommodated into the queue and cells are enqueued regardless of their priority levels. It should be noted that the queue is being under utilized below this level.



H_T L_T

Figure 6.1: The low and high buffer thresholds under which the PSSD algorithm operates

For $L_T < q\ (t) < H_T$ where $q\ (t)$ is the queue level at time t, the PSSD tries to maintain $q\ (t)$ below H_T in order to avoid excessive cell losses. Every time a slice needs to be added to the queue, the buffer controller verifies that sufficient space exist in the queue to accommodate the entire slice. The buffer controller uses the slice length field in the tag cell to establish this. The slice is enqueued if there exist enough capacity in the queue. However, if it is established that the slice would not be able to fit without having sections of it being dropped, then the buffer controller attempts to free up space in the queue by discarding the low priority slices that are already in the queue. For the case that the arriving slice is low priority, the buffer controller will inform the IPP to discard it. The buffer controller calculates the slice discard eligibility (SDE) for each tag cell that is enqueued. This value is then periodically updated and is used to discard slices from the queue if a flush operation is required or the SDE for a slice is ∞, see sections 5.2.2.

For the case that $q\ (t) > H_T$ the buffer is in danger of overflowing and in this case, the PSSD must pre-emptively flush out the queue to bring it back into the range $L_T < q\ (t) < H_T$. This again is done by referring to the SDE of each tag cell to determine if there are any low priority slices in the queue. A discard eligibility matrix is then repeatedly applied and each time such a slice is found it is discarded. This continues until the above condition is met or the tail of the queue is reached. In that case, the slice cannot be accommodated.

A special situation arises when all the slices in the buffer are high priority. In this case, an attempt is made to enqueue just the tag cell while discarding the rest of the slice. The enqueued tag cell is then marked accordingly by setting the first flags bit. Note that for the case that the queue is completely full, the entire slice including its tag cell cannot be accommodated.

6.2 The discard eligibility matrix

This is a weighted index that is used to determine if a slice should be discarded. Currently, it is defined to use slice type, slice length and TTL to calculate the slice discard eligibility as follows

$$S.D.E = k0 / (TTL) + k1 * (ST) + k2 * (SL)$$

Where **S.D.E** is the slice discard eligibility, **k0** is the weight attached to the **TTL, k1** is the weight attached to the slice type (**ST**) and **k2** is the weight attached to the slice length (**SL**).

ST = 0 for I frames, 1 for P frames and 2 for B frames.

The slice with the highest **SDE** is eligible for discard. Notice that when TTL = 0, then **SDE** = **INFINITY** regardless of the value of **ST** and **SL. SDE** is directly proportional to **SL** implies that large slices are the first to be discarded, all other factors being equal. This results in marked increase in buffer space for the incoming slices.

ST is set to zero for I frames to reduce as much as possible their possibility of being selected by this matrix. In some cases, it may be desirous to retain any I frame cells, regardless of its TTL. Recall that I frames provide a point a reference point for all frames of a GoP. Thus the frame can serve some use to the end decoder even though its play out time has expired. In which case, the above matrix would not be applied to I frame slices. A FIFO queuing strategy is assumed

6.3 Performance Issues

In order to determine the performance of the above proposal, we first study performance issues related to a single ATM switch buffer. From this, we can conservatively extrapolate to a network of n switches. Performance parameters that we wish to study at the switch buffer are the slice discard ratio (SDR) and the CLR for each frame type, the average latency and jitter per cell and slice as well as the good cell and slice throughput.

7 Simulation Scenario and Results Analysis

7.1 General Procedure

A switch buffer is partitioned into n queues each servicing a separate VC. In this scenario, each VC is assumed to be carrying tagged rt-VBR video content. A single processor operating at a fixed speed services each queue in a round-robin service strategy. The traffic used is derived from actual MPEG traces captured and freely distributed by the University Wurzbug, Germany via their ftp site [2][3]. The simulation procedure is as follows

- For each source, the next frame size in bytes is obtained from the trace file.
- From this, a slice distribution model is used to derive the number of slices in the frame as well as their size in bytes.
- The number of cells for each slice is then worked out assuming a 48-byte[2] AAL 5 payload. Bit stuffing is used to fill up the payload of the last cell if necessary.
- A tag cell is formatted with all the fields correctly filled for each slice that is generated.
- The source then transmits each slice starting with the tag cell.
- The source transmission rate is derived from the frame size based on the fact that 25 frames must be transmitted per second (PAL system). Thus the inter-frame transmission time is 40 ms and the inter-cell transmission time per frame is [40 / frame size] ms. Frame size is quoted in cells.
- A loss-delay model is applied to each cell to mimic network conditions. This is explained further below
- The IPP verifies the integrity of each slice before forwarding to the buffer controller
- The buffer controller enqueues slice and updates SDE regularly
- OPP transmits unlocked slices

[2] Note that all cells of an AAL 5 frame utilize the full 48 byte payload except the last cell which only uses 40 bytes and the remaining 8 bytes being taken up the AAL 5 frame trailer

The following statistics are collected from the above scenario

- The buffer size, which is varied for each scenario
- Average queue waiting time per cell
- Average queue waiting time per slice
- Number and type of cells dropped
- Number and type of slices dropped

Using the above statistics, we can then calculate CLR and SDR for each frame type as well as per GoP and for the entire stream. This allows us to derive the slice throughput for that single switch and hence derive a model for n switches. The cell and slice waiting time statistics are used to calculate latency and jitter for a single switch scenario. This can then be extrapolated to a network with n switches.

7.1.1 Slice Distribution Model

The input video trace file gives the size in bytes, of each frame for the video sequence being represented. It does not however tell us the sizes and therefore number of video slices there are for each frame. To get around this, we devised a slice distribution model that can be used to generate uniformly sized slices for a frame given its size. The process is illustrated below

- The frame size **F** is obtained.
- Using a uniformly distributed random function, average γ and variance β a slice is repeatedly created with size $S = \gamma + \beta$ and S is deducted from **F.**
- The process ends when $\mathbf{F} - S < 0$

7.1.2 The loss-delay model

The loss delay model is used to mimic network conditions by subjecting each cell to an exponentially distributed delay as well as a loss probability. The parameters for the loss probability as well as the exponential delay distribution are specified before running the simulation. In these tests we set the loss probability at 10^{-8} and the mean delay was set at 500 μs. Note that a delay of up to 500 ms is tolerable to a human user.

7.2 Methodology

The buffer is partitioned into $\boldsymbol{n}$ queues each having size equal to $\boldsymbol{Q}_{max}$. $\boldsymbol{Q}_{max}$ is varied from 10 cells to 10000 cells. The latter value approximates an infinite sized buffer. This allows us to test the algorithm's ability to remove cells with expired TTLs. We set the upper and lower thresholds $\boldsymbol{H_T}$ and $\boldsymbol{L_T}$ to 0.9 $\boldsymbol{Q}_{max}$ and 0.5 $\boldsymbol{Q}_{max}$ respectively. For each iteration, the queue size $\boldsymbol{Q}_{max}$ is set and each source transmits its video trace into its respective queue. We set the TTL for each stream to be 200 ms and the service rate for each queue to be 2.5 Mb/s. The simulation was allowed to run for 500000 iterations, during which time 118 frames representing 10 I-frames, 30 P-frames and 78 B-frames were transmitted.

The following statistics are calculated

$$CLR = \varepsilon_c \times 100$$

$$SDR = \varepsilon_s \times 100$$

Where ε_c is the ratio of lost cells to the total transmitted and ε_s is the ratio of lost slices to the total transmitted.

The CLR and SDR are obtained for each frame type.

Average cell waiting time = $E_i[\boldsymbol{T}_{out} - \boldsymbol{T}_{in}]$

Average slice waiting time = $E_i[\boldsymbol{T}_{last} - \boldsymbol{T}_{tag}]$

$\boldsymbol{T}_{out}$ and $\boldsymbol{T}_{in}$ are the cell de-queue and enqueue times for the sampled i^{th} cell respectively. $\boldsymbol{T}_{last}$ is the time at which the last cell of a slice is de-queued and $\boldsymbol{T}_{tag}$ is the time at which the tag cell of the same slice is enqueued. These are sampled values with a sample rate set at 20 μs.

7.3 Results Analysis

7.3.1 Cell Loss Performance

Figure 7.1 shows the CLR performance of the tag switch utilizing the PSSD buffer scheduler compared to that of a tagless switch. Notice that overall, the tagged switching method outperforms that tagless switching method in terms of CLR. The tagged switching method however, shows a poorer performance for buffer size less than 50 cells. This is to be expected since slices are only admitted into the queue if their size is less than the available space in the queue. It is to be expected that a most slices will exceed this requirement and hence be discarded. The tagless switching method however enqueues any arriving cells as long as there is space for it in the queue.

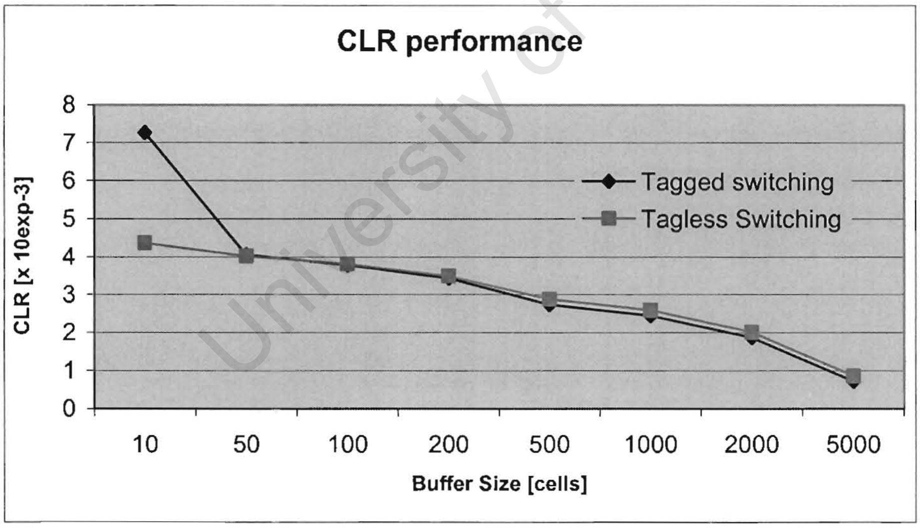


Figure 7.1: CLR performances for tagged switching against tagless switching

Figure 7.2 a shows the CLR performance for I frames using both switching methods.

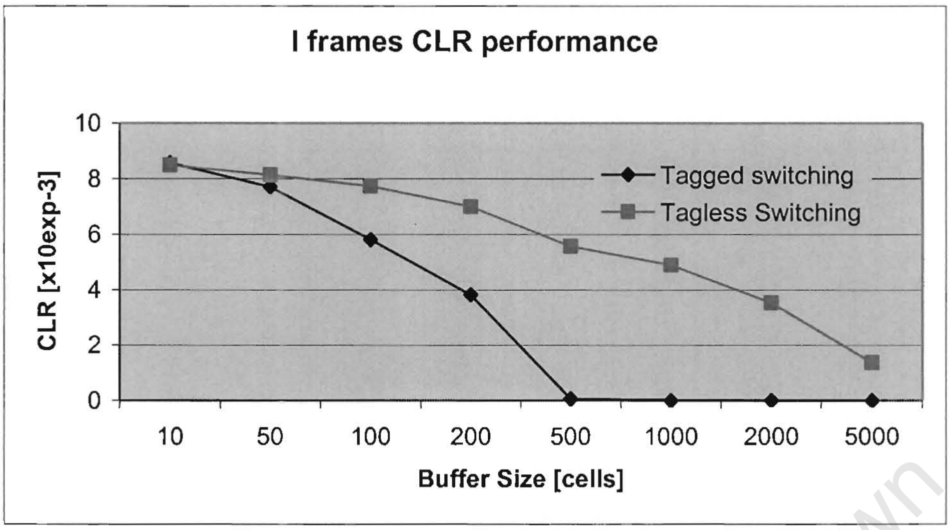


Figure7.2 a: CLR performance of I frame cells

In this case, we notice that the tagged switching method shows a vastly superior CLR performance where I-frame cells are concerned compared to the tagless switching mode. This can be explained by the fact that the tagless switching method discards cells indiscriminately when the queue overflows. This is in contrast to the tagged switching method in which the buffer control first attempts to remove low priority slices in the queue in order to make space for high priority I-frame slices. Thus the I-frame cells are afforded a very low CLR.

Again, notice that for buffer size less than 50 cells, the CLR performance is not very different from that of the tagless switching mode.

Figures 7.2a and 7.2b shows the CLR performance for P and B-frame cells for tagged switching vs. tagless switching modes. We can see that the tagged switching modes shows the poorer performance levels for both frame types and more so for B-frame cells. This is expected since the good CLR performance obtained for I-frame cells is at the expense of the low priority B and P cells.

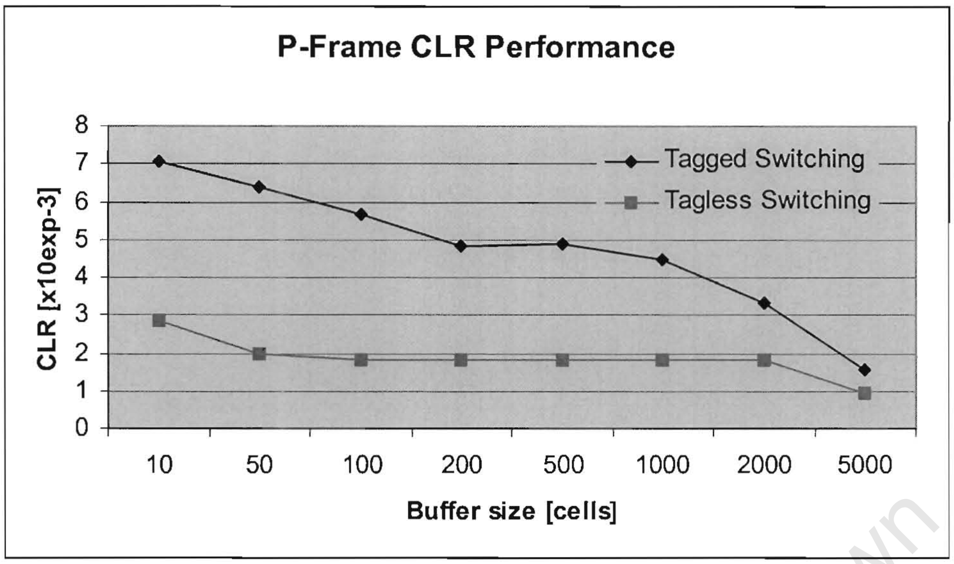


Figure 7.2 b: CLR performance for B-frame cells

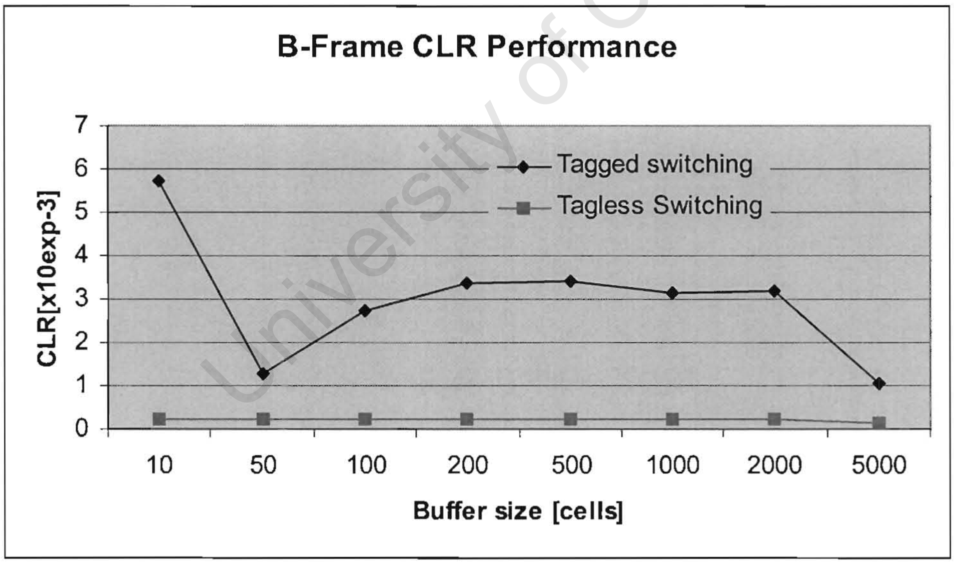


Figure 7.2 c: CLR performance for B-frame cells

Under the tagless switching mode, the low CLR performance experienced by the B-frame cells can be attributed to the low frame size of B frames. This implies that they are less likely to stress the queue when arrive. Indeed, this also holds true for the tagged switching mode. The high CLR performance displayed by the B-frames is caused by the arrival of P or I frame cells to the queue when it is about to overflow. In this case, instead of discarding the arrived cells, B-frame cells already in the queue are flushed out to make way.

7.3.2 Cell Delay performance

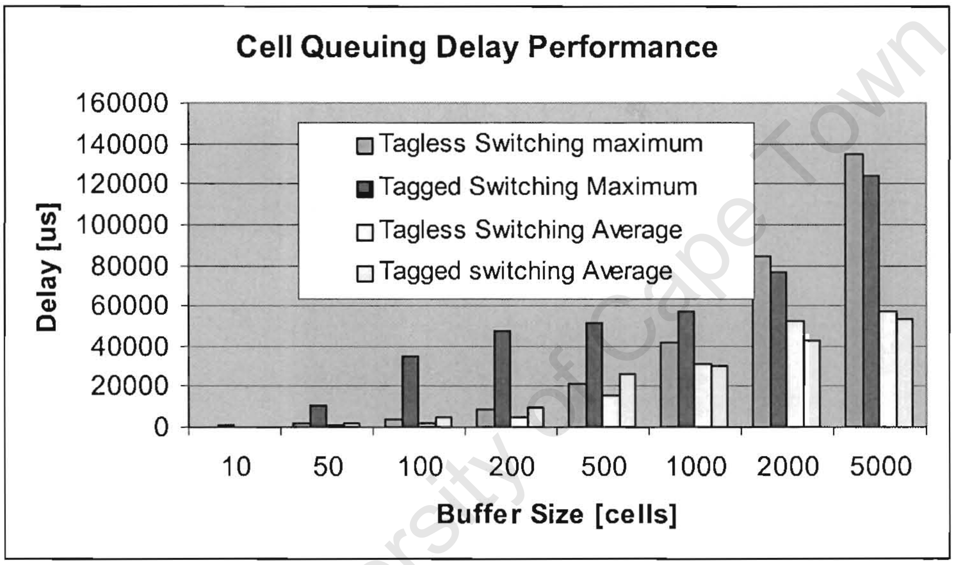


Figure 7.3 Cell queuing delay performance

Figure 7.3 shows the average and maximum queuing delays experienced by the cells for each type of switching mode. Initially, the tagged switching mode experiences the higher maximum and average delays compared to the tagless switching mode. This can be attributed to the fact that each arriving slice has to wait until all cells that belong to it have arrived before it can be switched through. In contrast, the tagless switching mode immediately begins to forward any arriving slice thereby cutting delay. This is at the expense of a high errored slice rate. But notice that the tagged switching mode outperforms the tagless switching mode for large buffer sizes (> 2000 cells).

7.3.3 Slice Jitter Performance

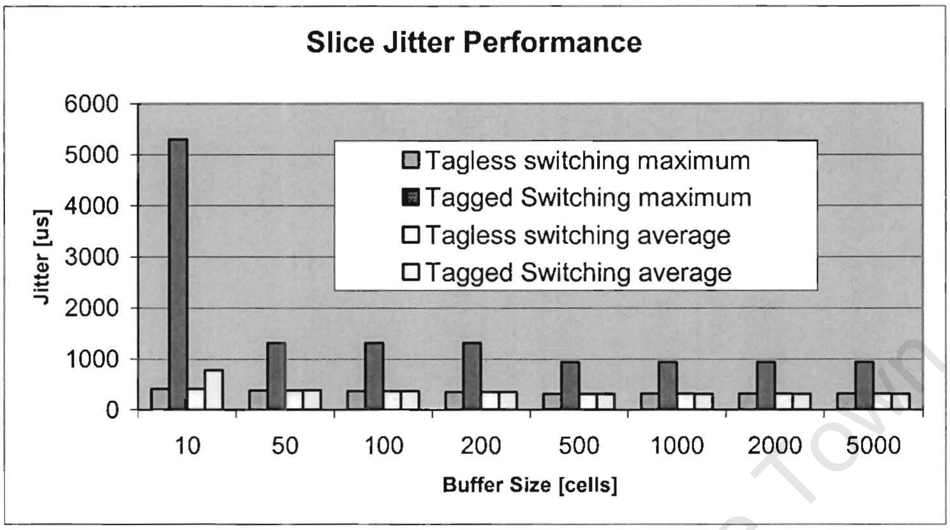


Figure 7.4: Slice jitter performance

Figure 7.4 shows the average and maximum slice jitter performance experienced for both tagged and tagless switching modes. The high peak for experienced for tagged switching when queue size is 10 is due mostly to discarded slices that cannot fit into the queue. in these experiments, we defined slice jitter as the inter-tag cell queuing delays. Again we notice that the tagged switching mode consistently has the higher maximum jitter for all buffer sizes. This is due to the slice integrity checks that must be carried out on each slice. Firstly, arriving slices must have their length integrity verified and secondly, while enqueued, all tag cells must have their TTL values updated and verified against the maximum. All this processing adds an appreciable delay. The advantage, of course, is the great reduction in bad through put as well as enhanced support for high priority video cells

8 Conclusions

This work has presented a unique method of supporting compressed real time video over ATM networks. The method works by taking special advantage of the hierarchical and structural properties of compressed MPEG. A key feature of which is the fact that some components of the video stream are more important than others. Recognizing this important feature allows us to propose a method of tagging video so that the ATM switch can recognize the different components of the transmitted video.

Tagging is accomplished using a variant of the RM cell that we define. This tagging method is more flexible than other proposed methods because it allows us to tag at various layers of the video stream. The only modifications that it requires to be carried out on the ATM switch is in software so that no hardware changes are required. Although some delay is introduced to the video stream, the overall end user QoS levels are improved since the cell loss performance of high priority cells is greatly improved at the expense of low priority cells resulting in an overall improvement in the end quality levels of the received video.

We also propose a buffer management algorithm that selectively discards slices rather than cells when buffer occupancy reaches certain pre-defined levels. To accomplish this, a slice discard eligibility vector is calculated for every slice that is enqueued using the slice's TTL value as well as its priority and length. Enqueued slices have this value periodically updated and when a slice arrives at the queue and it cannot be accommodated, the slice with higher slice discard eligibility value is candidate for discard to make way for the arrived slice. To enhance the support given to real time video, each slice is also assigned an end-to-end time to live corresponding to the maximum transfer delay allowed for the video stream. This is decreased at every node that the slice traverses. The decrease is equal to the slice waiting time at the node. If this value reaches zero, the slice is discarded. This ensures that dead cells are not occupying valuable network resources.

Simulation studies confirm that the end quality, as measured from the CLR suffered by the high priority frames as well as the overall buffer performance show a marked improvement over convectional CLP based tagging methods.

All in all, the proposed video tagging method shows great promise for the effective support of real time VBR MPEG over ATM networks and further research using a small sized network is called for.

9 References

[1] Sang-Jo YOO, Sung-Hoon HONG and Seong-Dae KIM
"Traffic Descriptor Dimensioning for VBR MPEG video sources over ATM networks"
IEICE TRANS COMMUN.. VOL E83-B No 1 JAN 2000

[2] O.Rose
"Simple and efficient Models for Variable Bit Rate MPEG video Traffic"
UNIVERSITY OF WÜRZBURG, Institute of Computer Science Research Reports Series. Report #16
JULY 1995

[3] O.Rose
"Statistical Properties of MPEG Video Traffic and Their Impact on traffic modeling in ATM systems.
UNIVERSITY OF WÜRZBURG, Institute of Computer Science Research Reports Series. Report #16
FEB 1995

[4] Çağlan M. Aras, James F. Kurose, Douglas S. Reeves and Hening Schulzrinne
"Real-Time Communication in packet switched Networks"
Department of Electrical and Computer Engineering, North Carolina State University, Ralegh NC

[5] DAPENG WU, YIWEI THOMAS HOU YA-QIN ZHANG
"Transporting Real-Time Video over the Internet: Challenges and approaches"
PROCEEDINGS OF THE IEEE, VOL. 88, NO. 12 DEC 2000

[6] Pao-Chi CHANG Jong-Tzy WANG and Yu-Cheng LIN
"Adaptive Video Quality Control Based on Connection Status over ATM networks"
IEICE TRANS. COMMUN.. VIL. E82-B NO.2 SEPT 1999

[7] John Wiseman
"An Introduction to MPEG Video Compression"
http://members.aol.com/symbandgrl/

[8] R.Shäfer
"MPEG 4: A Multimedia Compression Standard for interactive applications and services"
ELECTRONICS & COMMUNICATIONS ENGINEERING JOURNAL, DEC 1998

[9] The ATM Forum technical Committee
"Traffic Management Specification Version
4.0" af-tm-0056.000, APRIL 1996

[10] Gunnar karlsson
"Video over ATM networks"
Swidish institute of Computer Science

[11] Anna Charny, David D. Clark and Raj Jain
"Congestion Control with explicit rate indication"
Digital Equipment Corporation

[12] R.Jain
"Myths about congestion management in high-speed networks"
Digital Equipment Corporation

[13] Raj Jain
"Congetsion Control and Traffic Management in ATM Networks: Recent Advances and a

Survey"
Department of Computer and Information Science, The Ohio State University

[14] Apostolos Dailianas, Andreas Bovopoulos
'Congetsion Control in ATM Networks"
Washington University and Chipcom Corporation

[15] A. Mehaoua, R. Boutaba
"The Impact of Errors anf Delays on the Performance of MPEG-2 Video Communications"
University of Cambridge and University of Toronto, ECE dept.

[16] Pascal Fossard, Olivier Verscheure (Assistant)
"MPEG-2 over lossy packet Switched Networks QoS Analysis and Improvement"
Swiss Federal Institute of Technology Lausanne
JULY 1997

[17] Mark W. Garrett, Walter Willinger
"Analysis, modeling and generation of Self- Similar VBR Video Traffic"
Bellcore

[18] Majdi Ashibani, D Mashao and B. Nleya
"Renegotiation-based Quality of service Approach for Scalable Compressed Video over ATM Networks"
University of Cape Town

[19] S.McCainne, V.Jacobson and M. Vetterli
"Receiver-Driven Layered Multi-cast"
Proc. ACM SIGCOMM
AUGUST 1996

[20] Marwan Krunz, Herman Hughes
"A traffic Model for MPEG-Coded VBR Streams"
Michigan State University

[21] Yu Gong, Ian F. Akyildiz
"Dynamic Traffic Using Feedback and Traffic prediction in ATM Networks"
IEEE Infocom Proceedings Vol 1
JUNE 1994

[22] Nanying Yin, Michael G. Hluchyj
"On Closed-Loop Rate Control for ATM Cell Relay Networks"
IEEE Infocom Proceedings Vol 1
JUNE 1994

[23] G.Ramamurthy, R.S Dighe
"Performance Analysis of Multilevel Congestion Controls in BISDN"
IEEE Infocom Proceedings Vol 1
JUNE 1994

[24] Song Chong, San-qi Li and Joydeep Ghosh
"Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM"
IEEE Infocom Proceedings Vol 1
JUNE 1994

[25] Christoph Kuhmunch and Gerald Kuhne

"Efficient Video Transport over lossy networks"
Universitat Mannheim, Praktische Informatik IV

[26] Ismo Anttila, Markku Paakkunainen
"Transferring real-time video on the Internet"
Helsinki University of technology, Telecommunications Software and Multimedia laboratory.

[27] Darren L. Spohn
"Data Network Design" 2nd Edition
McGraw-Hill Companies 1997

[28] Pascal Frossard, Olivier Verscheure(assit)
"MPEG-2 over lossy packet networks QoS Analysis and improvement"
Swiss Federal Institute of technology-Lausanne. July 1997

[29] Fernando Pereira
"MPEG-4: Why, What, How and When?"
Instituto Superior Tecnico – Instituto de Telecommicacoes

[30] Olivier Avaro, Ganesh Rajan, Alexandros Eleftheriadis, Carsten herpel, Liam Ward
"MPEG-4 Systems: Overview"
ISO/IEC 14496 (MPEG-4 Systems)

[31] UC Davis
"Introduction to TCP/IP"
gopher://gopher -
chem.ucdavis.edu/00%2fIndex%2fInternet_aw%2fIntro_the_Internet%2fintro.to.ip%2f01_What_is_TCP_I
P

[32] Ingo Busse, Bernd Deffner, Henning Schulzrinne
"Dynamic QoS Control of Multimedia Applications based on RTP"
http://www.fokus.gmd.de/step/acontrol/ac.html

[33] Polyphemus: Project Book; Biel School of Engineering and Architecture
"Multimedia over the Internet"
http://www.hta-bi.bfh.ch/Projects/polyphem/documents/projectwork-techreport-mediainternet.html

[34] Mentor technologies
"Introduction to MPLS"
http://www.ccci.com/learn/welcher/papers/mplsintro.html

[35] *"ISO/OSI Reference model*
http://www.it.kth.se/edu/gru/Telesys/96P3_Telesystem/HTML/Module4/ISO-9.html

[36] CSE 81: University of California San Diego
"OSI Reference model "
http://www-cse.ucsd.edu/users/polyzos/CSE81/home.96/osi.html

[37]. Y. Yemini, J. Hellerstein
"Introduction: The OSI reference model"
http://www.cs.columbia.edu/dcc/classes/E6998-025-F96/Oct10/directory16.htm
1995

[38] G. Franceshini
"Delivery layer in MPEG-4"
CSELT Centro Studi e Laboratori Telecommunicazioni S.p.A

[39] Anonymous
"Broadband battle pits ATM against IP"
HP Chronicle
January 2000

[40] Sandra Wermer
"TCP/IP will overtake ATM soon"
http://www.Hoise.com/articles/SW-PR-03-98-20.html
March 1998

[41] Touradj Ebrahimi, Caspar Horne
"MPEG-4 Natural Video Coding- An OverView"
Signal Processing labolatory, Swiss federal Institute of technology

[42] A. Murat Tekalp and Jorn Ostermann
"Face and 2-D mesh Animation in MPEG-4"
http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/8-SNHC_visual_paper/8-SNHC_visual_paper.htm]

[43] Kihong Park, Wei Wang
"QoS-Sensitive Transport of Real-Time MPEG video using Adaptive Forward Error Correction"
Network systems lab, Dept of Computer science, Pardue University

[44] Pascal Frossard, Olivier Verscheure
"Content-based MPEG-2 Structuring and protection"
Signal processing Laboratory, Institute for Computer communications and Applications, Swiss federal Institute of technology

[45] Eduardo Asbun, Edward J. Delp
"Real-time error concealment in Compressed Digital Video Streams"
Video and Image processing laboratory (VIPER),School of Electrical and Computer Engineering, Pardue University.

[46] Ahmed Bashandy, Edwin Chong, Arif Ghafoor
"Jitter Control and Dynamic Resource management for multimedia communication over Broadband Network"
School of Electrical and Computer Engineering, Pardue Univesity.

[47] Dinesh C. Verma, Hui Zhang, Domenico Ferrari
"Delay and Jitter control for real-time communication in a packet switched network"
Dept. Electrical Engineering and Computer Science, University of California at Berkeley

[48] Benjamin W. wah, Xiao Su, Dong Lin
"A Survey of Error-Concealment Schemes for real-Time Audio and Video Transmission over the Internet"
Dept. Electrical and Computer Engineering, University of Illinois at Urbana-Champaign

[49] Injong Rhee, Srinath R. Joshi
"Error recovery using FEC and re-Transmission for Interactive Video Transmission"
Dept Computer Science, North Carolina State University.

[50] Hui Xie, John S. Baras
"Performance Analysis of PNNI Routing in ATM networks: Hierarchical Reduced Load approximation"
Dept. Electrical Engineering, Institute for Systems Research, University of Maryland

[51]. Mika Loukola
"Private Network-to-Network Interface"
Helsinki University of Technology

[52] M.R. Frater, O. Rose
"A Comparison of Models for VBR Video Traffic sources in B-ISDN"
Institute of Computer science, University of Wurzburg

[53] R. Prashant Chandra, Hiroshi Murata
"Signaling Support for Dynamic Rerouting of Connections in an ATM Network"
School of Computer Science, Carnegie-Mellon university

[54] Colin J. Parris, Anindo Banerjea
"An Investigation into Fault Recovery in Guaranteed Performance Service connections"
Tenet Group, Computer Science Division, UC Berkeley

[55] Bay Networks
"Reference Guide for the Centillion 50/100 and Model 5000BH Switches"
© 1999 Bay Networks

[56] The ATM Forum
"ATM: User-Network Interface Specification v3.0"
PTR Prentice Hall Publishing
© 1993

[57] OPNET Modeler Guides
"Volume 1-12"
MIL 3 Inc
© 1989-1997

[58] Nortel Networks
"Asynchronous Transfer Mode (ATM) Fundamentals

[59] Zhi-Li Zhang, Srihari Nelakuditi, Rahul Aggarwal and Rose P. Tsang
"Efficient Selective Frame Discard Algorithms for Stored Video Delivery across Resource Constrained Networks"
http://www.cs.umn.edu/Research/CNMRG/Papers/Zhan98:Efficient.pdf

[60] Ahmed Mehaoua*, Raouf Boutaba** and Youssef Iraqi***
"Partial versus Early Packet Video Discard"
http://www.prism.uvsq.fr/~mea/files/ahmed_globecom98.pdf

[61] Ahmed Mehaoua
Guaranteed Slice Rate: A Novel ATM Service for Best Effort Video Communications
University of Cambridge Centre for Communication Systems Research

[62] Ahmed Mehaoua and Raouf Boutaba
Performance Analysis of a Partial Video Slice-Level Discard Scheme (Adaptive-PSD) for MPEG Connections over UBR+ Service
Computer Science Research Institute of Montréal, Canada

[63] Qiyong Bian, Kohei Shiomoto, Jonathan Turner
Dynamic Flow Switching: A New Communication Service for ATM Networks
Proceedings of Infocom
March 1998