RZ 2958 (# 93004) 09/22/97
Computer Science 23 pages

Research Report

Video Compression Adapted for Transmission on ATM Networks

P. Scotton

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
Phone : (+41) 724 89 48
e-mail : psc@zurich.ibm.com

C. Galand

Centre d'Etudes et Recherches IBM
Le Plan du Bois
06610 La Gaude
France
Phone : (+33) 4 92 11 48 26
e-mail : galand@vnet.ibm.com

J. Ménez

Université de Nice - Sophia Antipolis
URA CNRS 1376
41, bd Napoléon III
06041 Nice Cedex
France
Phone : (+33) 4 93 21 79 55
e-mail : jmenez@mimosa.unice.fr

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Video Compression Adapted for Transmission on ATM Networks

P. Scotton

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

C. Galand

Centre d'Etudes et Recherches IBM, Le Plan du Bois, 06610 La Gaude, France

J. Ménez

Université de Nice - Sophia Antipolis, URA CNRS 1376, 41, bd Napoléon III, 06041 Nice Cedex, France

Abstract

This paper focuses on the problem of video transmission over the emerging high-speed asynchronous networks. A video coder that achieves a good quality/throughput/complexity compromise is proposed. Moreover, this coder is hierarchized and has a variable bit rate, which provides good characteristics for transmission over asynchronous transfer mode networks.
After a brief discussion of flow control techniques for high-speed networks, a specific implementation of the leaky-bucket mechanism is introduced. An adaptive version of this algorithm is then presented. The behavior of the adaptive leaky bucket loaded with the output of a compressed video source is discussed. We show how a simple modification of the leaky bucket permits the use of this flow-control technique to regulate a compressed video source.

1 Introduction

With the rapid evolution taking place in the fields of image processing, networking, and multimedia technology, video will play an increasingly important role in the area of networking. Future networks will be based on the fast packet switching technology developed on asynchronous transfer mode (ATM). This transfer technique provides a flexible way to exploit the available bandwidth by means of statistical multiplexing and will yield a high-performance environment for video communications.

The transport of video sources on asynchronous high-speed networks addresses two major problems:

- Without compression, video sources require high bandwidths. These bandwidths must be reduced before transmission over networks can take place.
- In the ATM environment, it is important to take into account real time constraints imposed by video sources.

The first problem is solved by the application of numerical compression techniques. Nevertheless, it is important to develop coders that are well suited for ATM communication networks. Such coders should have the following features:

- The coder output bit rate must be variable to exploit statistical multiplexing.
- The coder must be hierarchized to prevent a significant degradation of quality during a network congestion period.

On the other hand, high-speed networks imply the use of preventive congestion-control techniques. These mechanisms often introduce a transmission delay jitter. It is therefore essential to design congestion-control mechanisms that reduce this jitter as much as possible. Moreover, the congestion-control strategy must be able to discard less significant parts of the information to be transmitted in case of network congestion. This implies that the coder must create two information streams: a fundamental information stream (to reconstruct a basic image) and a refinement stream (to improve the basic image quality).

The scope of this paper is to show a complete transmission chain of compressed video through a high-speed network based on asynchronous transport techniques. The paper is organized as follows. After a brief review of sub-band coding and perfect reconstruction filter banks, we propose a multiresolution quantization technique and its application to a hybrid video coder. The originality of this compression scheme relies on the use of sub-band spatial "energetic correlation". This allows us to achieve a good compression ratio with limited computational complexity. The third section is dedicated to the presentation of two congestion-control methods for high-speed networks, namely the leaky bucket and an adaptive version of this mechanism. We then propose a new window-based adaptive leaky-bucket mechanism. This new adaptation scheme increases the stability and reduces the overall computational complexity of the adaptive leaky bucket. In the fourth section we present simulation results obtained from the proposed compressed video source traffic regulation by the adaptive leaky-bucket mechanism. We show in particular how a modification of the leaky-bucket parameters permits the easy application of the adaptive leaky bucket to the regulation of compressed video sources.

2 Video hybrid coder using multiresolution quantization

It is well known [13, 27] that sub-band coding is a simple means to obtain a hierarchical coding scheme. The reason is that the digital signal is partitioned into several bands in the frequency domain, each of which contributes information that is more or less important for the quality of a reconstructed signal. This property can be used efficiently either for speech coding or for image and video compression.

2.1 Sub-band coding

One of the most important issues in sub-band coding is the design of analysis/synthesis filter banks that yield perfectly reconstructed output signals. A two-channel analysis/synthesis filter bank provides the starting point for the study of sub-band coding systems, Fig. 1. The necessary and sufficient conditions for perfect reconstruction in the case of an equally spaced two-sub-band decomposition of a one-dimensional signal were first established by A. Croisier, D. Esteban, and C. Galand [3, 4]. They have shown that, to achieve perfect reconstruction, two conditions must be satisfied. In the Z-transform domain the reconstructed signal is given by

$$\begin{aligned}\tilde{X}(Z) &= \frac{1}{2} X(Z)\,[H_0(Z)\,G_0(Z) + H_1(Z)\,G_1(Z)] \\ &\quad + \frac{1}{2} X(-Z)\,[H_0(-Z)\,G_0(Z) + H_1(-Z)\,G_1(Z)] \quad , \qquad (1)\end{aligned}$$

which can be written

$$\tilde{X}(Z) = T(Z)\,X(Z) + A(Z)\,X(-Z) \quad , \qquad (2)$$

where the term $A(Z)\,X(-Z)$ represents aliasing due to sub-sampling. Consequently, a perfect reconstruction requires the following two conditions:

1. On aliasing cancelation:

$$[H_0(-Z)\,G_0(Z) + H_1(-Z)\,G_1(Z)] = 0 \quad . \qquad (3)$$

2. On error free from amplitude/phase distortion:

$$[H_0(Z)\,G_0(Z) + H_1(Z)\,G_1(Z)] = 2\,Z^{-k} \;\; , \; k \in \mathcal{N} \quad . \qquad (4)$$

The first class of filters that attempted to satisfy these conditions was the *Quadrature Mirror Filters* (QMF) family [3]. However, the only QMF filters that lead to an exact reconstruction are

- the *Haar filters* based on the Haar wavelet [10], and
- low-pass and high-pass ideal half-band filters, which are nonphysically realizable.

In 1984, M.T.J. Smith and T.P. Barnwell [21] proposed another finite-impulse response (FIR) solution called the *Conjugate Quadrature Filters* (CQF) family that yields a perfect reconstruction and easily designable filters. Nevertheless, the linear phase requirement is relaxed in their solution. A new FIR solution called the *Polynomial Filters* family was introduced by M. Vetterli [28] in 1986. These filters have a linear phase, but they are not in quadrature. Other solutions have been proposed, including infinite impulse response (IIR) filters [26], paraunitary and lossless lattice filters [25], and more recently the wavelet-based approach [29].

2.2 Multiresolution quantization

One way to improve decorrelation in the time domain is to introduce linear predictive techniques instead of sub-band decomposition. This leads to a hybrid video [?17], the principle of which is depicted in Fig. 2.

A multiresolution quantization technique is applied to the coder presented in Fig. 2. In this coder, spatial decorrelation is achieved by a tree-structured decomposition in seven sub-bands. The filters used are the Haar wavelet filters, i.e. $H_0(Z) = (1+Z^{-1})/2$ and $H_1(Z) = (1-Z^{-1})/2$. A dynamical bit allocation over all sub-bands improves their respective performance. Haar filters have been used to limit computational complexity as much as possible.

This coder allows better performance than the Karlsson–Vetterli coder [12]. The latter is based on a three-dimensional sub-band coding for video (one dimension for the "time domain" and two dimensions for the "spatial domain"). The performance improvement (in terms of image quality and computational complexity) achieved by our hybrid coder is due to the use of time-domain differential pulse code modulation (DPCM), which allows a better signal decorrelation. However, our coder is of course much more sensitive than the Karlsson–Vetterli coder to transmission errors. Both coders do not have an "intrinsically" variable bit rate.

It is worth noting that in those two coders the quantization technique does not take into account a sub-band signal model. We propose therefore a new multiresolution quantization scheme that provides a further hierarchization, an intrinsically variable bit rate, and better overall performance.

Let us consider again the hybrid coder. When the time prediction error signal is decomposed into sub-bands, two kinds of areas can be distinguished:

- High-energy areas appear in the image as "contours".
- Low-energy areas appear in the image as flat, homogeneous regions.

When there are no scene changes, camera movements, or zooms, the difference between two consecutive images will show several regions in which pixel values are very close to zero and only a few regions that have significant amplitudes.[1] Except for the base-band, all sub-bands have been high-pass filtered at least once, and their high-energy areas have been emphasized as the low-frequency area energy is reduced. Of course, in the lowest sub-band, low-frequency areas are emphasized.

The main idea of multiresolution quantization is to encode high-energy areas more finely. The problem to be solved is how to detect such areas, which will henceforth be called *active areas*. To do so, an activity detection algorithm has been used as follows.

- Sub-bands are decomposed into square blocks of N pixels side length. For high sub-bands (i.e. sub-bands 5 to 7), $2N \times 2N$ blocks have been considered in order to take the sub-sampling factor into account.

[1] These regions correspond to movement areas.

- The pixels of each block are scanned. If a pixel has a value greater than a given threshold T_1, then the pixel is said to be active.

- If a block has at least T_2 active pixels, then the block is active.

Upon completion of the algorithm, active blocks represent active areas.

Some simplifications can be made by taking into account the *"energetic correlation"* that exists between sub-bands. The simplified activity-detection algorithm is described as follows.

- An activity detection routine is performed on sub-band 1. Results are used only for this sub-band.

- A joint activity detection routine is performed on sub-bands 2 to 4 and it is used to encode these sub-bands. That is, if a block is active in at least one of these sub-bands, then it is considered active for all three sub-bands 2 to 4.

- No activity detection routine is performed on sub-bands 5 and 6. These sub-bands are encoded by reporting the location of active blocks in sub-bands 2 to 4 and scaling the results by a factor of 2.

- Sub-band 7 is not encoded.

Sub-bands are encoded with a fixed-level Laplacian scalar quantizer. For active-block quantization, different quantities of binary resources are applied to each sub-band. Low-frequency sub-bands are more finely encoded than high-frequency sub-bands. Non-active blocks are not encoded for sub-bands 5 and 6. They are quantized with only one bit in sub-bands 1 to 4. A more complete description of this technique can be found in [19, 20].

In order to reduce computational complexity, another hybrid structure has been proposed. The idea is to perform sub-band decomposition outside the DPCM loop. The advantage is that no sub-band reconstruction is required inside the loop. Somehow, this structure leads to a "multi-DPCM" coder: one DPCM for each sub-band. Simulation results have shown that the same reconstructed image quality can be obtained by moving sub-band decomposition outside the DPCM loop. The proposed coder synopsis is shown in Fig. 3.

2.3 Simulation results

All the coders presented here have been tested on a "Head & Shoulders" sequence (a *Miss America* sequence) and on a "Full Motion" sequence (a *Table Tennis* sequence), see Table 1. The sequences contain 8-bit encoded black-and-white images. The image size is 256×256 pixels.

For these coders, results have been obtained without a loss-less coding technique. In the case of the proposed coder, we have estimated that the compression ratio can be improved by about 30% using Rissanen's context algorithm [18] in conjunction with arithmetic coding [15].

Table 1 clearly shows that our hybrid coder achieves a significant performance improvement over that of the Karlsson–Vetterli coder. This is especially true for the *Table Tennis* sequence. On the other hand, multiresolution quantization improves the hybrid coder results by about 3 dB. Moreover, in the case of the *Miss America* sequence, it increases the compression ratio by about 0.1 bit/pel. The complexity of these four coders is compared in Table 2.

Figure 4 is an example of original and reconstructed sequences. Three consecutive images representing a scene change are shown. The quality of the first and the third images is very good, but it is degraded in the second image (because of the scene change). However, at a rate of 25 images per second, this quality loss is imperceptible.

Figure 5 shows the output bit rate and the peak signal-to-noise ratio (SNR) for a sequence of 50 images including two scene changes. The images presented in Figure 4 correspond to images 19, 20, and 21. The mean bit rate of the 50 images is 1.545 Mbits/s. The two scene changes at images 20 and 30 generate a peak of approximately 3.8 Mbits/s.

3 Preventive congestion-control mechanisms

3.1 Introduction

It is well known that future broadband integrated networks will be based on ATM technology. The principle of this technique lies in the fast switching of fixed-length cells (53 bytes). This yields very high transmission rates and a good means of bandwidth management by the use of statistical multiplexing.

To achieve statistical multiplexing, it is essential for the sources to have variable bit rates. Suppose that N sources are multiplexed on a common trunk having a capacity of C. If R_i and m_i are, respectively, the peak and the mean bit rates of source i, the following relation must be satisfied to achieve statistical multiplexity:

$$\sum_{i=0}^{N} m_i < C < \sum_{i=0}^{N} R_i \quad . \tag{5}$$

Of course, if $\sum_{i=0}^{N} R_i = C$, the system will be in a deterministic multiplexing situation.

When the network is working in statistical multiplexing mode, it is clear that, the achievable quality of service (QoS), i.e. cell losses, depends on one's knowledge of the source traffic's behavior. As it is difficult to have a good model of source, flow control mechanisms must be used to ensure a given QoS to a connection.

In a telecommunications network, two major problems must be solved to achieve "efficient" flow control as transmission speed increases:

- In order to reach high speeds (of the order of several hundred megabits per second), protocols have to be simplified. In fact, even if the performance of the constituent processors increases continuously, the available computational power is still too little for high-speed networking. Let us take the example of a packet-switched network, each packet having a fixed length of 53 bytes. Suppose that the transmission rate is 10 Mbits/s. If the processor in a mode has a power of 20 Mips (million instructions per second), 850 instructions can be dedicated to a single packet. If the transmission ratio is now 100 Mbits/s, only 85 instructions might be used to process a single packet! It is therefore essential to reduce the complexity of the communication protocols as much as possible.

- Another key problem is that high-speed networks have a high *bandwidth–transmission delay product*. This can be illustrated by a simple example. Suppose that the roundtrip transmission delay is 100 ms. If the transmission rate is 16 Mbits/s, 800 kbits will be transmitted in the time it takes for information to go from the receiver to the source.[2] If the transmission rate is now 155 Mbits/s, the latent information quantity increases to 7.75 Mbits. It is obvious that in such a situation traditional "handshaking" flow-control

[2]This is true if the roundtrip delay is symmetrical. Of course, if the delay is not symmetrical the quantity of information transmitted may decrease or, in the worst case, increase.

techniques are very difficult to use. This is especially true in the case of wide-area networks where transmission delay is long.

Therefore, in most cases, only preventive congestion-control mechanisms can be used for wide-area high-speed networks [2, 9, 20]. Reactive control mechanisms can be adopted in some cases if special control techniques are used [11].

A preventive flow-control technique is the *leaky-bucket* mechanism, several implementations of which have been proposed [7, 24]. The main idea is to provide to the packets transmitted by a given source a grant for sending them through the network. This grant is given to a packet only if the source traffic respects a pre-established contract.

3.2 Leaky-bucket mechanism

As stated in the previous section, several types of leaky-bucket mechanisms have been proposed. We will consider in this paper only the queuing leaky bucket used in the experimental network plaNET [2, 6, 7]. A schematic view of this mechanism is given in Fig. 6. The source sends the packets to a finite-length queuing buffer. Each packet needs a given number of green tokens[3] before it can be sent to the network. The green tokens are accumulated in a green token pool containing M_g tokens. If *all* the tokens are available, the packet is sent immediately. Otherwise it must wait until the tokens are available. Green tokens are generated at rate of γ tok/s. Suppose that a token is equivalent to a grant to send a byte to the network. For packets having a fixed length of L bytes, it can easily be seen that, if the source mean rate m is such that $m \leq \gamma/L$, the leaky-bucket mechanism will be transparent. If the source increases its mean rate and $m > \gamma/L$, the leaky bucket will run out of green tokens and the packets will be delayed in the queuing buffer. In this case the leaky bucket exhibits a traffic regulating behavior.

To accept brief traffic increases, the leaky bucket has been designed with a red token pool of M_r tokens. If a packet is in the buffer and there are not enough green tokens, the packet can be sent with red tokens (if enough of them are available). In that case, however, the packet is sent with a violation tag (labeling it a "red packet") to indicate that the source has overdrawn its traffic contract. If a congestion situation occurs in a node, red packets will be discarded first.

Actually, the problem is how to define the source traffic contract and how to set the leaky-bucket parameters. In plaNet the notion of *equivalent capacity* [2, 6, 7] has been used. The equivalent capacity is the bandwidth that has to be reserved for a source in order to guarantee a given QoS. This notion is based on the *fluid-flow model* [1, 14, 16, 22].

The source is modeled as a *two-state Markov chain.* In state *"off"* the source is silent. In state *"on"* the source transmits data at its peak bit rate R. In the time domain, the source traffic is characterized by a triplet (R, m, b), where m is the mean bit rate and b is the mean burst duration. This assumption leads us to consider that active periods are exponentially distributed. This model is depicted in Fig. 7. Suppose an (R, m, b) source is transmitting data to a finite length buffer of X kbytes linked to a transmission line. The equivalent capacity of the source $\widehat{C}$ (expressed in Mbits/s) is the capacity of the line that yields a probability of buffer overflow equal to ϵ. The value of $\widehat{C}$ is given by the relation [6]

$$\widehat{C} = \frac{\alpha\, b\,(1-\rho)\,R - X + \sqrt{[\alpha\, b\,(1-\rho)\,R - X]^2 + 4\,X\,\alpha\, b\,\rho\,(1-\rho)\,R}}{2\,\alpha\, b\,(1-\rho)} \quad , \tag{6}$$

[3]This number is proportional to the packet length.

where $\alpha = \log(1/\epsilon)$ and $\rho = m/R$. The buffer overflow probability ϵ is actually the QoS of the connection.

In the leaky-bucket mechanism, if the green token generation rate γ is equal to the equivalent capacity of the connection $\widehat{C}$, we find that

- if the source respects its traffic characteristics, namely (R, m, b), the QoS will be guaranteed and the behavior of the leaky bucket may be considered "transparent";
- if the source does not respect its traffic constraint then the leaky bucket will limit the mean traffic bit rate transmitted to the network to $\widehat{C}$.

Another important parameter of the leaky bucket is the green token pool size M_g. This parameter is used to establish the largest transmitted burst duration. It is computed by the relation

$$M_g = \frac{b\,(1-\rho)\,\gamma\,(R-\gamma)}{\gamma - \rho\,R}\,\mathrm{Log}\left[\frac{(\gamma - \rho\,R) + \rho\,\xi\,(R-\gamma)}{\xi\,\gamma\,(1-\rho)}\right] \quad , \tag{7}$$

where ξ is the probability that a packet will be labeled red.[4] The size of the red token pool M_r is given by the relation

$$M_r = r\,\gamma\,T_r \tag{8}$$

The coefficient r is used to bound the long-term red marking probability; ξ will never exceed r. The parameter T_r is the red token pool refilling period.[5]

We will not discuss the efficiency of the leaky-bucket mechanism here. A detailed analysis can be found in [7, 20].

3.3 Adaptive leaky bucket

The leaky-bucket mechanism has a few shortcomings:

- To obtain an efficient bandwidth management it is important to define the traffic parameters, i.e. the triplet (R, m, b), quite accurately for all the sources. However, some sources (e.g. video) are very difficult to characterize.
- The leaky bucket assumes that the traffic of a source is stationary. In practice, this is rarely true.
- In the model adopted here, burst lengths are characterized only by their mean. This assumes that their distribution is exponential. Actually, in most cases this is not true.

These considerations led T. Tedijanto and L. Gün to propose an adaptive version of the leaky-bucket mechanism [23] shown in Fig. 8. The adaptive leaky bucket is composed of three modules.

Leaky bucket: the leaky-bucket mechanism used here is identical to the system described in the previous section.

[4] Also called the *red marking probability.*

[5] The red token pool is completely refilled every T_r ms.

Measuring and estimation: the mean bit rate m and the red marking probability ξ are periodically measured, respectively, at the leaky-bucket input and output. At each time t, the raw measurements $\hat{m}_t$ and $\hat{\xi}_t$ are filtered by exponential filters to obtain the estimated values $\tilde{m}_t$ and $\tilde{\xi}_t$. If $\tilde{x}_{t-1}$ and $\hat{x}_t$ are the estimated values at time $t-1$ and the raw measurement at time t of the stochastic process[6] $\mathcal{X}$, respectively, the estimation at time t is given by

$$\tilde{x}_t = \alpha\, \tilde{x}_{t-1} + (1-\alpha)\, \hat{x}_t\ , \quad 0 < \alpha < 1 \quad . \tag{9}$$

Decision module: the decision module provides three functions:

- Decide when an adaptation of the leaky-bucket parameters should be performed.
- Send a bandwidth change request to the network management.
- Compute and update leaky-bucket parameters.

Let us focus on the decision module. At each time t, estimated values are provided by filters. Actually, the source mean burst length is estimated by the *exponential substitution* method [8]. If the current settings of the leaky bucket are known, it is possible by observing the red marking probability to estimate the mean burst length of a source equivalent to the present one such that

- it has the same impact on the leaky bucket, and
- the equivalent source burst lengths have an exponential distribution.

The source average burst length estimation, at time t, is given by the relation [8]

$$\tilde{b}_t^{\mathrm{eq}} = \frac{\eta}{\mathrm{Log}\left[\frac{R\,(\gamma - \tilde{m}_t) + \tilde{\xi}_t\, \tilde{m}_t\, (R-\gamma)}{\tilde{\xi}_t\, \gamma\, (R - \tilde{m}_t)}\right]} \quad , \tag{10}$$

where

$$\eta = \frac{M_{\mathrm{g}}\, R^2\, (\gamma - \tilde{m}_t)}{(R - \tilde{m}_t)\, (R-\gamma)\, \gamma} \quad . \tag{11}$$

Therefore, it is possible at each time t to compute an estimation of the source traffic characteristics: $(R, \tilde{m}_t, \tilde{b}_t^{\mathrm{eq}})$. Note that the traffic can be modeled by the triplet $(R, \tilde{m}_t, \tilde{\xi}_t)$ as, from the knowledge of the actual leaky-bucket parameters, $\tilde{b}_t^{\mathrm{eq}}$ is derived from $\tilde{\xi}_t$ by the exponential substitution relation.

It is possible to place the point $(\tilde{m}_t, \tilde{\xi}_t)$ in the plane $\mathcal{P}(m, \xi)$. In this plane, three regions are defined:

- $\Re_{\mathrm{c}}$ is the region where the traffic parameters are in compliance with the subscribed contract. There is no need to adapt the leaky bucket.
- $\Re_{\mathrm{u}}$ denotes the region where the traffic parameters exceed the traffic contract. In this case the leaky bucket must be adapted to increase the allocated bandwidth. The adaptation is done after a network management acknowledge. If the requested bandwidth is not available, the connection is rerouted or broken.

[6]This process can be associated with either the mean bit rate or the red marking probability.

- $\Re_{\mathrm{d}}$ is the portion of $\mathcal{P}(m,\xi)$ where the allocated bandwidth is greater than the bandwidth used by the connection. The leaky-bucket parameters should be adapted to decrease the allocated bandwidth.

In the original leaky bucket [23], these regions are defined by complicated relations. Moreover, in this algorithm the number of adaptations performed per time unit is not controlled, and this can generate intensive network management traffic. We have therefore proposed a new adaptation algorithm based on a window mechanism [5].

Let W_{f} be a time domain window. When a window is in progress the estimated values are received from the filters and stored. At the end of the window the mean estimation values are computed:

$$\tilde{m}_{W_{\mathrm{f}}} = \frac{1}{\operatorname{card}\{\tilde{m}_t\}} \sum_{t \in W_{\mathrm{f}}} \tilde{m}_t \tag{12}$$

and

$$\tilde{\xi}_{W_{\mathrm{f}}} = \frac{1}{\operatorname{card}\{\tilde{\xi}_t\}} \sum_{t \in W_{\mathrm{f}}} \tilde{\xi}_t \quad . \tag{13}$$

An adaptation is decided upon only if a window is elapsed and if a least one of the following conditions is true:

$$\mid \tilde{m}_{W_{\mathrm{f}}} - m_{\mathrm{c}} \mid > k_{\mathrm{m}}\, m_{\mathrm{c}} \tag{14}$$

or

$$\mid \tilde{\xi}_{W_{\mathrm{f}}} - \xi_{\mathrm{T}} \mid > \; k_{\xi}\, \xi_{\mathrm{T}} \tag{15}$$

where k_{m} and k_{ξ} are real positive constants, m_{c} is the mean bit rate used to compute the leaky-bucket parameters, and ξ_{T} is the desired value of ξ for the connection. The region $\Re_{\mathrm{c}}$ is a rectangular region in the plane $\mathcal{P}(m,\xi)$ centered on the point (m_{c}, ξ_t).

In the proposed algorithm, an adaptation can be performed only at the end of a window. The duration of a window is about 30 s,[7] which can be a prohibitively long time to wait before adapting. We have therefore implemented an "emergency" procedure that allows us to adapt immediately if the traffic increases significantly. An emergency adaptation is decided upon if, at a given time t, at least one of the following two conditions is true:

$$\tilde{m}_t > k_{\mathrm{m}}^{\mathrm{e}}\, m_{\mathrm{c}} \tag{16}$$

or

$$\tilde{\xi}_t > k_{\xi}^{\mathrm{e}}\, \xi_{\mathrm{T}} \quad , \tag{17}$$

where k_{m}^{e} and k_{ξ}^{e} are real positive constants.

This algorithm is depicted in Fig. 9. The performance of this adaptive leaky bucket is discussed in [20]. In the next section we study its application to a video source.

[7] This value as been determined by simulation. A smaller window would provide better tracking but would potentially increase the number of adaptations.

4 Application of adaptive leaky bucket to a compressed video source

4.1 Introduction

Transmitting compressed video over an ATM network is not an easy task. This is mainly because video has *real-time constraints* and is therefore sensitive to transfer delay jitter. The coder sends a compressed image at regular intervals (every 40 ms if the video rate is 25 images per second). The decoder must receive at the same rate the frames to decode. If T_f is the time between two frames at the coder entry and T_d is the time necessary to decode a frame, the delay jitter must respect the condition

$$D_\mathrm{j} < T_\mathrm{f} - T_\mathrm{d} \quad . \tag{18}$$

The introduction of preventive congestion-control mechanisms complicates this problem. In fact the adaptive leaky introduces a delay jitter for the following reasons.

- A queuing delay occurs at the leaky-bucket admission buffer (when the leaky bucket runs out of green and red tokens).
- During the time needed to detect and perform a bandwidth increase, the leaky bucket enters a "saturation" phase and the admission delay increases.

In the following two sections we examine the behavior of the adaptive leaky bucket loaded with traffic generated by the proposed coder. The experimental environment is presented, followed by a discussion of the results obtained.

4.2 Choice of a video source

A video sequence of 7750 images (approximately 5 min) has been chosen for this test. This sequence is built as shown in Table 3. This sequence has been compressed by the proposed coder. Hence, for each compressed image, the quantity of information to be transmitted is known. This knowledge allows us to build an (R, m, b) source equivalent to the video coder.

The peak bit rate R has been fixed to 16 Mbits/s. The transmission of each image corresponds to a burst. The active period duration is computed directly from the quantity of information to be transmitted. If Q_i is the quantity of information (expressed in bits) to be transmitted for image i, the burst duration for this image is given by

$$b_i = \frac{Q_i}{R} \quad [\mu\mathrm{s}] \quad . \tag{19}$$

Each burst is then followed by an idle period i_i such that[8]

$$i_i + b_i = 40 \times 10^3\,\mu\mathrm{s} \quad . \tag{20}$$

With this approach we obtain a source such that $m \simeq 1.51$ Mbits/s and $b \simeq 3.45 \times 10^3$ μs. However, as shown in Fig. 10, the burst lengths are not exponentially distributed.

[8]The video rate is supposed to be 25 images per second, i.e. one image arrives every 40 ms.

4.3 Simulation results

The compressed video stream has been used as input to the adaptive leaky-bucket simulation model. The parameters in Table 4 were chosen for simulation. The simulation was conducted to assess the bandwidth savings achieved with the adaptation strategy and to analyze the quality of transmission (losses and delay jitter) of the adaptive leaky bucket.

The bandwidth savings have been assessed by computing the ratio of the peak bandwidth to the average bandwidth used for transmission of a video sequence. As the duration of the adaption window (30 s) is significant compared to the sample sequence duration (300 s), the video stream has been artificially repeated three times (900 s). This allows us to better assess the adaptive leaky bucket behavior in steady state. Figure 11 shows the variation of the video stream bandwidth and of the allocation as determined by the adaptive leaky-bucket algorithm. Note that, after a transient period of about 100 s, the adaptive leaky bucket converges to the actual source parameters. Afterwards the algorithm enters the steady state. The overall simulation results are listed in Table 5.

When in steady state, the average video traffic bandwidth requirement is about 1.31 Mbits/s, whereas the mean allocated bandwidth is about 2.15 Mbits/s. As the peak bandwidth requested by the video source is about 4.1 Mbits/s, the allocation bandwidth gain with the adaptive leaky bucket is $\approx$1.9. Note that in steady-state operation, the allocated bandwidth correctly tracks the bandwidth variations requested by the source. Compared to a "peak rate allocation" strategy, this allows us to save bandwidth and, therefore, to improve line efficiency significantly.

The second objective of the simulation was to analyze the quality of transmission. For this purpose let us consider a more accurate analysis of image transmission. The simulation is run on the original sequence (300 s). Figure 12 shows the image transmission time and the number of lost packets versus time. As can be seen on these curves, at approximately $t = 140$ s packets are lost at the leaky bucket admission buffer and the time to transmit an image increases to 50 ms. Figure 11 shows that these losses occur when the mean bit rate increases suddenly and exceeds the allocated bandwidth. A likely conclusion is that the adaptive leaky bucket reacts too "slowly" to a sudden traffic increase, thereby temporarily degrading the connection QoS. Moreover, during this phase the delay jitter becomes significant and incompatible with video transmission requirements. A solution is proposed in Sect. 4.4.

4.4 Improvements

To avoid the problems discussed above, two solutions are considered and analyzed.

- Increase the admission buffer capacity. This solution reduces the admission losses, but it increases the delay jitter, and is thus not useful for traffic having real-time constraints.

- Increase the red token pool size. This solution enables [something] to evacuate excess traffic quickly, thus reducing both admission losses and delay jitter. On the other hand, packets are sent as red and may be discarded if there is network congestion.

Let us analyze the second solution. The red token pool size M_{r} is defined in Eq. (8). The parameter r is used to bound the amount of red traffic. In the case of an adaptive leaky bucket, red traffic is bound by the adaptation mechanism. Therefore, this parameter can be used to scale the size of the red token pool.

Figure 13 shows the admission loss probability and the maximum (and mean) image transmission time as a function of r. One can see that for $r \geq 2$ there are no more admission losses.

For $r \geq 3$, the maximum image transmission time falls below 20 ms. If the time needed to decode an image is less than 20 ms, then the source real-time constraint is satisfied.

Figure 14 shows the mean and maximum probabilities that a packet of an image will be red as a function of r. The mean probability is approximately constant and of the order of 2×10^{-2}. The maximum probability for $r \geq 3$ is $\approx$0.5. This shows that the red marking probability increases only when the leaky bucket enters a saturation phase.

A leaky bucket saturation occurs only when the source traffic increases and the bandwidth adjustment is pending. One can argue that, if the additional bandwidth is available, the traffic overhead can be transferred with a very small loss probability. In the worst case, if all the red packets are lost, the available bandwidth should be halved.[9] In this case, the coder hierarchization may be exploited to guarantee a minimum QoS. The traffic issued from the proposed coder is composed of two flows:

- A flow that has a high priority and contains "vital" data (e.g. active block positions, values used for quantization) as well as the quantized samples of active blocks belonging to sub-bands 1 to 4.
- A refinement flow containing the quantized samples of active blocks belonging to sub-bands 5 and 6 and the quantized samples of the background block.

A pre-marking mechanism is used at the leaky bucket to ensure that pre-marked packets are sent as green. If the prioritary flow is pre-marked, then in the worst case only the refinement flow may be sent as red and thus may be lost. The coder parameters are tuned such that the prioritary flow has the same amount of data as the refinement flow. Some simulations have been run to assess the image quality in the case where the refinement flow is lost. Figure 15 shows the original and reconstructed sequence in the case where only the prioritary flow is transmitted. The quality of the reconstructed image is worse than in the case where the two data flows are sent. However, the picture quality is acceptable. Moreover, this situation occurs only in the rare case that the entire refinement flow is lost. In this case only a couple of pictures of the sequence are impaired. Therefore, the impact on video QoS is negligible.

5 Conclusion

We have proposed a complete system for video coding and transmission over high-speed asynchronous networks. The system includes two key features, variable bit rate video coding and an adaptive leaky-bucket mechanism for access control, which provide significant bandwidth savings compared to peak rate allocation schemes.

The video coder combines sub-band coding for intra-image compression, and predictive coding for inter-image compression. Active areas in sub-band signals are finely quantized, whereas idle areas are coarsely quantized. Active areas are detected thanks to an original algorithm that exploits the energy correlation between sub-band signals. We have demonstrated high-quality, low-complexity video coding well suited to statistical multiplexing and robust to packet losses during transmission.

The function of the adaptive leaky bucket is to control access to the network while detecting significant changes in traffic demand. We have proposed a low-complexity algorithm that permits accurate traffic tracking with a good control of the number of adaptations per time unit. This algorithm has been shown to provide nearly optimal bandwidth savings for a real

[9] The maximum red probability is 0.5 (see Fig. 14); i.e. at least a "half image" can be transmitted.

compressed video stream, and to guarantee negligible transmission time jitter and minimal image quality impairment in cases of high cell loss in the network.

In summary, the work reported in this paper includes a comprehensive study of video compression and transmission. Therefore it can be regarded as a contribution to the introduction of video services in high-speed networking environments, which are expected to emerge in the next few years.

References

[1] D. Anick, D. Mitra, and M. Sondhi, "Stochastic Theory of a Data-Handling System with Multiple Sources," *Bell Systems Technical Journal*, vol. 61, no. 8, pp. 1871–1894, October 1982.

[2] I. Cidon, I. Gopal, and R. Guérin, "Bandwidth Management and Congestion-Control in plaNET," *IEEE Communications Magazine*, pp. 54–64, 1991.

[3] A Croisier, D. Esteban, and C. Galand, "Perfect Channel Splitting by Use of Interpolation/Decimation/Tree Decomposition Technique," presented at *Int. Symp. Information, Circuits and Systems*, Patras, Greece, August 1976.

[4] D. Esteban and C. Galand, "The Application of Quadrature Mirror Filters to Split Band Voice Coding Schemes," in *Proc. Int. Conf. ASSP*, Hartford, CT, May 1977, pp. 191–195.

[5] C. Galand, P. Scotton, L. Gün, and T. Tedijanto, "Window Based Bandwidth Adjustment," *IBM Technical Disclosure Bulletin*, vol. 37, no. 10, 1994.

[6] R. Guérin, H. Ahmadi, and Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High Speed Networks," *IEEE J. Sel. Areas Commun.*, vol. JSAC–7, 1991.

[7] R. Guérin and L. Gün, "A Unified Approach to Bandwidth Allocation and Access Control in Fast Packet-Switched Networks," in *Proc. INFOCOM'92*, Florence, Italy, May 1992.

[8] L. Gün, "An Approximation Method for Capturing Complex Traffic Behavior in High Speed Networks," *Perf. Eval.*, vol. 19, no. 1, pp. 5–23, 1994.

[9] I. W. Habib and T. N. Saadawi, "Controlling Flow and Avoiding Congestion in Broadband Networks," *IEEE Communications Magazine*, pp. 46–53, 1991.

[10] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme," *Math. Anal.*, vol. 69, pp. 331–371, 1910.

[11] J. Jaffe, "Bottleneck Flow Control," *IEEE Trans. Commun.*, vol. COM–29, no. 7, pp. 954–962, 1981.

[12] G. Karlsson and M. Vetterli, "Three Dimensional Subband Coding for Video," in *Proc. Int. Conf. ASSP*, New York, NY, April 1988, pp. 1100–1103.

[13] G. Karlsson, "Sub-band Coding for Packet Video," Ph.D. Thesis, Columbia University, NY, 1989.

[14] L. Kosten, "Liquid Models for a Type of Information Storage Systems," *Delft Progress Report*, pp. 71–86, November 1986.

[15] G.G. Langdon, "An Introduction to Arithmetic Coding," *IBM J. Res. Develop.*, vol. 28, no. 2, 1984.

[16] D. Mitra, "Stochastic Theory of a Fluid Model of Multiple Failure-Susceptible Producers and Consumers Coupled by a Buffer," *Adv. Applied Probability*, vol. 20, pp. 646–676, 1988.

[17] C. Raimondo *et alii*, "Codeurs Vidéo de Faible Complexité à Base de Codage en Sous-bandes, de Compensation de Mouvement Hiérarchisée et de Quantification Vectorielle," presented at *Trezième Colloque du GRETSI*, Juan-les-Pins, France, September 1991.

[18] J. Rissanen, "A Universal Data Compression Algorithm," *IEEE Trans. Inf. Theory*, vol. IT–29, no. 5, 1983.

[19] P. Scotton and J. Menez, "A Low Complexity Video Sub-band Coder for ATM," *Signal Processing, Image Communication*, no. 6, pp. 421–433, 1994.

[20] P. Scotton, "Compression et Transmission de Signaux Vidéo sur des Réseaux Haut Débit avec Contrôle de Congestion du Flux de Données," Ph.D. Thesis, Université de Nice–Sophia Antipolis, October 1993.

[21] M. T. J. Smith and T. P. Barnwell, "Exact Reconstruction Techniques for Tree Structured Subband Coders," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP–34, no. 3, 1986.

[22] T. E. Stern and A. I. Elwalid, "Analysis of Separable Markov-Modulated Rate Models for Information-Handling Systems," *Adv. in Applied Probability*, 1991.

[23] T. Tedijanto and L. Gün, "Effectiveness of Dynamic Bandwidth Management Mechanisms in ATM Networks," in *Proc. INFOCOM'93*, San Francisco, CA, May 1993, pp. 358-367.

[24] J. S. Turner, "New Directions in Communications (or Which Way to the Information Age," *IEEE Communications Magazine*, vol. 24, no. 10, pp. 8–15, 1986.

[25] P. P. Vaidyanathan and P. Q. Hoang, "Lattice Structures for Optimal Design and Robust Implementation of Two-band Perfect Reconstruction QMF Banks," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP–36, no. 1, pp. 81–94, 1988.

[26] P. P. Vaidyanathan, *Multirate Systems and Filterbanks.* Prentice-Hall: CITY, 1993.

[27] W. Verbiest, L. Pinnoo, and B. Voeten, "The Impact of the ATM Concept on Video Coding," *IEEE J. Sel. Areas Commun.*, vol. JSAC–6, no. 9, pp. 1623–1632, 1988.

[28] M. Vetterli, "Filter Banks Allowing Perfect Reconstruction," *Signal Processing*, vol. 10, no. 3, pp. 219–244, 1986.

[29] M. Vetterli and C. Herley, "Wavelets and Filter Banks: Theory and Design," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP–40, no. 9, pp. 2207–2232, 1992.

[30] J. Woods (Ed.), *Subband Image Coding.* Kluwer: Dordrecht, 1991, Ch. 6.

Table 1: Simulation results.

Coder	*Miss America*		*Table Tennis*	
	Mean bits/pel	Mean peak SNR	Mean bits/pel	Mean peak SNR
Karlsson–Vetterli coder	1.85	29.7 dB	1.91	23.4 dB
Hybrid coder	0.94	35.0 dB	0.92	29.3 dB
Multires. coder (SBC inside)	0.83	37.3 dB	0.96	32.1 dB
Multires. coder (SBC outside)	0.83	37.4 dB	0.92	32.1 dB

Table 2: Comparison of complexity.

Coder	**Complexity (Mips)**
Karlsson–Vetterli coder	25.4
Hybrid coder	23.9
Multires. coder (SBC inside)	23.9
Multires. coder (SBC outside)	17.0

Table 3: Video sequence.

Sequence type	**Duration**
Head & Shoulders	30 s
Full motion	60 s
Head & Shoulders	40 s
Full motion	60 s
Head & Shoulders	120 s

Table 4: Simulation parameters.

Leaky-bucket parameters	$\xi_{\mathrm{T}} = 10^{-2}$, $\epsilon = 10^{-6}$, $X = 512$ kbits
Measurement time	$t = 1$ s
Window duration	$W_{\mathrm{f}} = 30$ s
Filter coefficient	$\alpha = 0.1$
Region parameters	$k_{\mathrm{m}} = 0.5$, $k_{\xi} = 0.9$, $k_{\mathrm{m}}^{\mathrm{e}} = 1.8$, $k_{\xi}^{\mathrm{e}} = 2.0$
Initialization	$m = 10$ Mbits/s, $b_{\mathrm{c}} = 5$ ms

Table 5: Simulation results.

Mean waiting time at admission buffer	$25.5\,\mu$s
Red traffic percentage	0.29%
Average adaptations per second	0.06

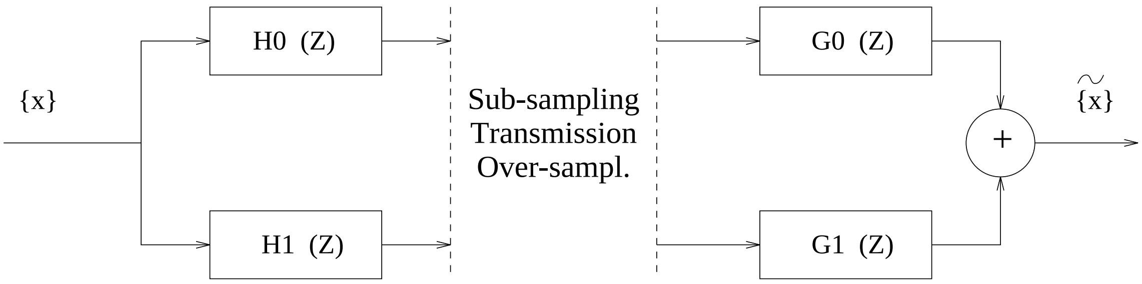

Figure 1: Two sub-band decomposition principle.

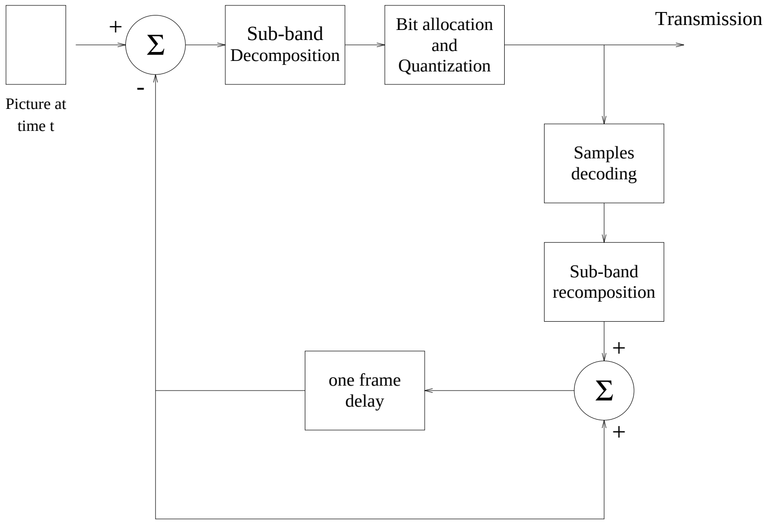

Figure 2: Hybrid coder synopsis.

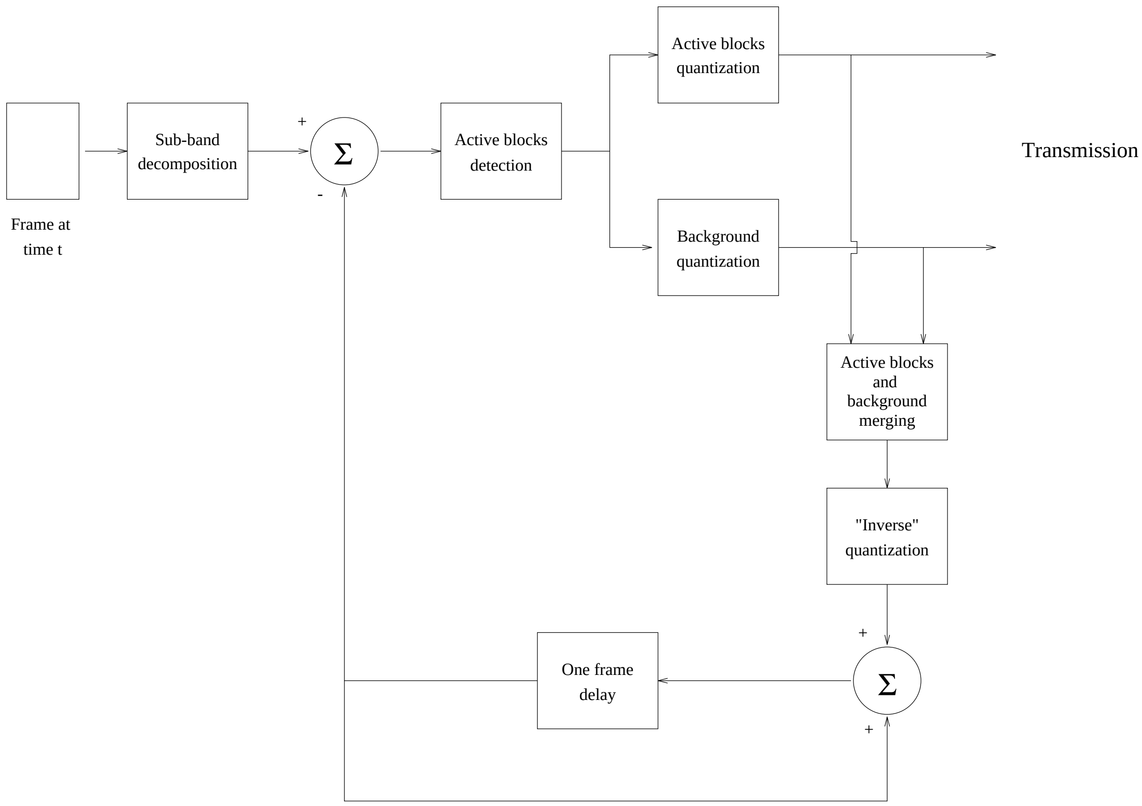

Figure 3: Proposed coder synopsis.

Figure 4: Example of (a) an original and (b) a reconstructed sequence.

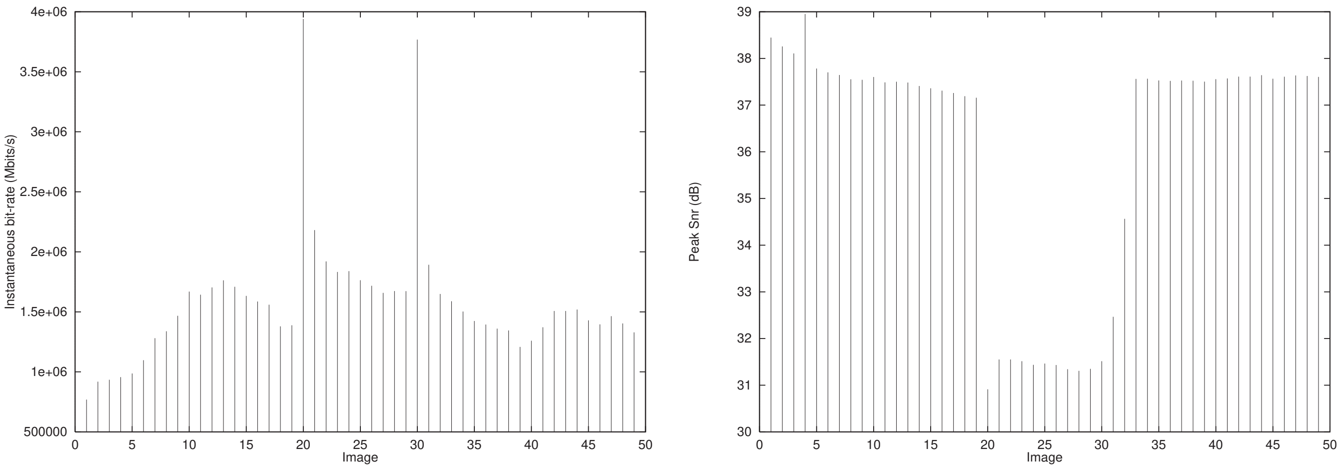

Figure 5: Example of output bit rate and SNR.

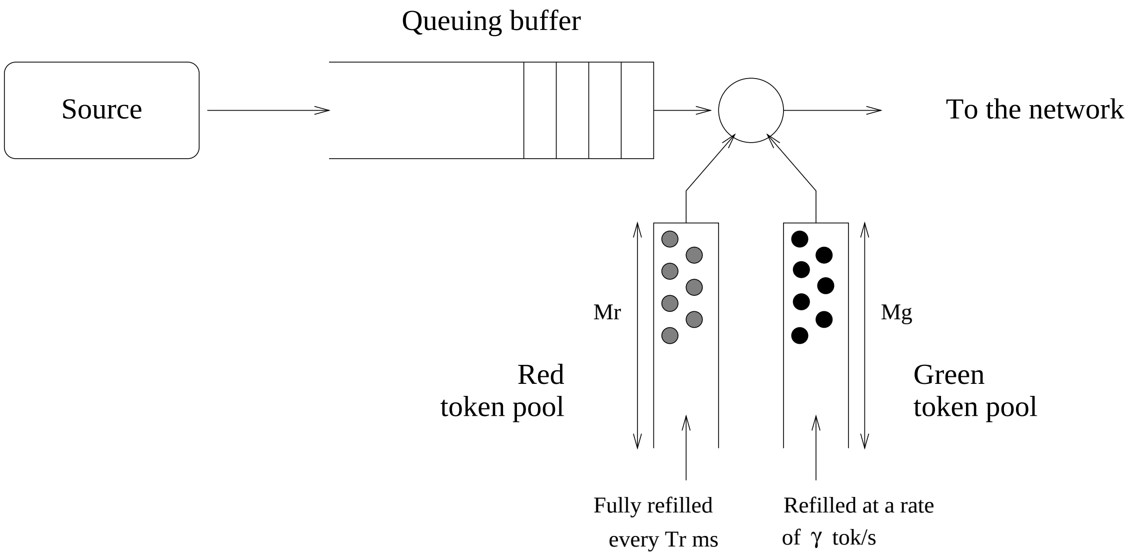

Figure 6: plaNET leaky-bucket synopsis.

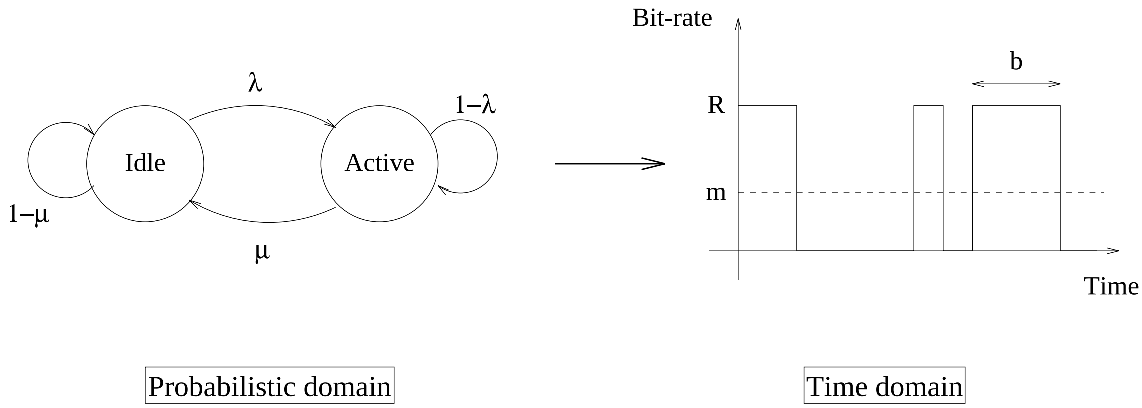

Figure 7: Source traffic model.

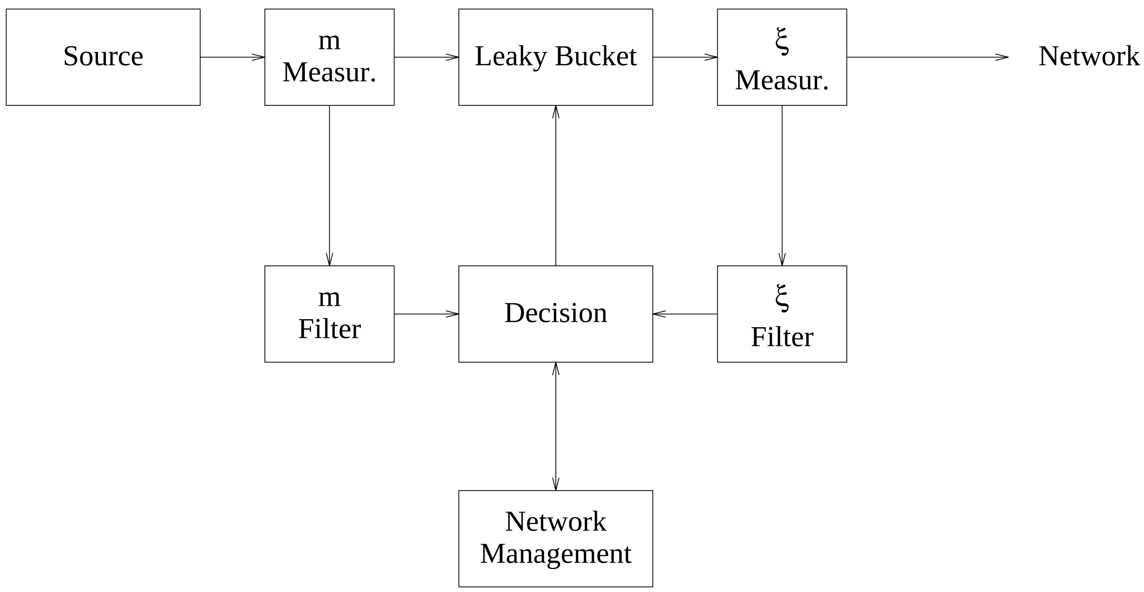

Figure 8: Adaptive leaky-bucket synopsis.

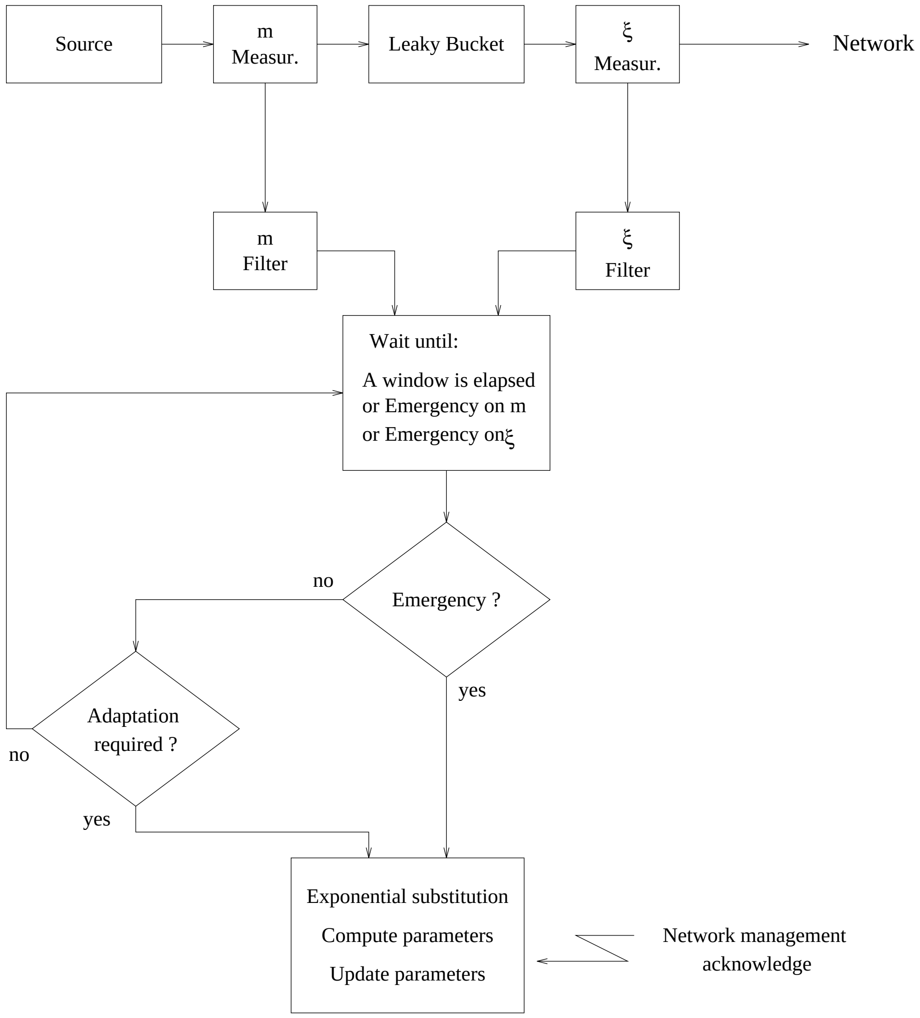

Figure 9: Adaptive leaky-bucket algorithm.

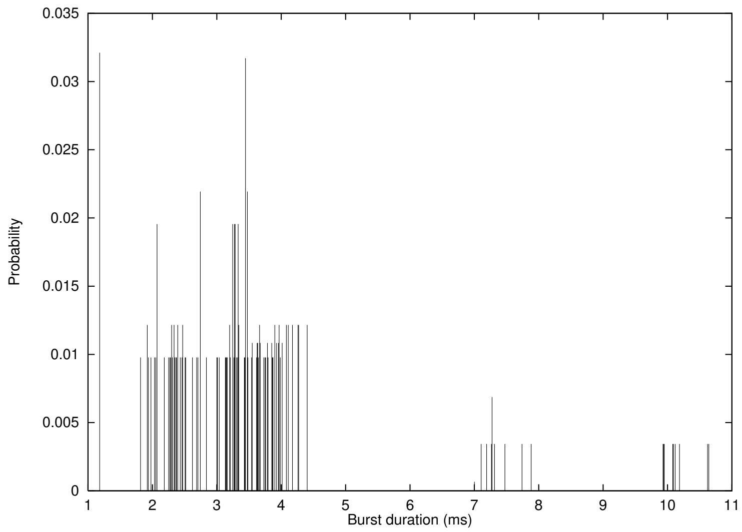

Figure 10: Burst length histogram.

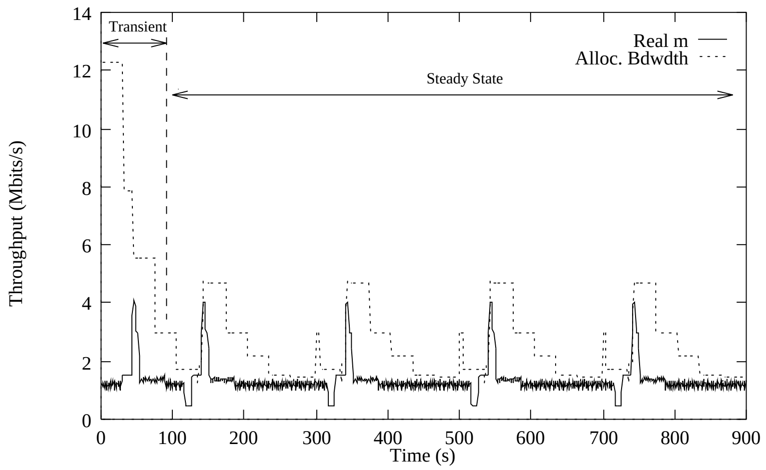

Figure 11: Real mean bit rate and allocated bandwidth.

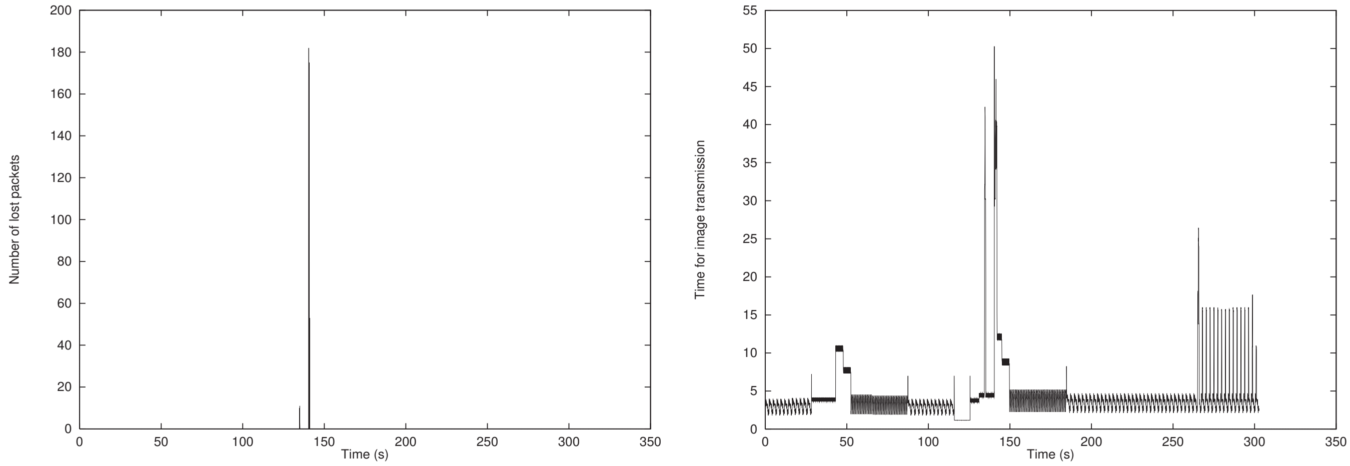

Figure 12: Packet losses and image transmission time.

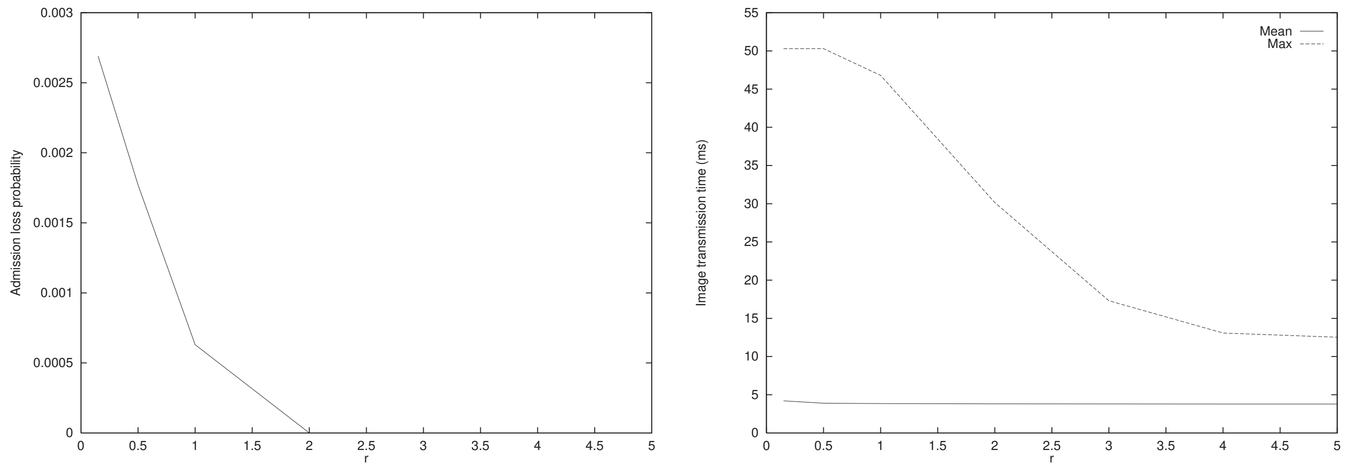

Figure 13: Packet losses and image transmission time.

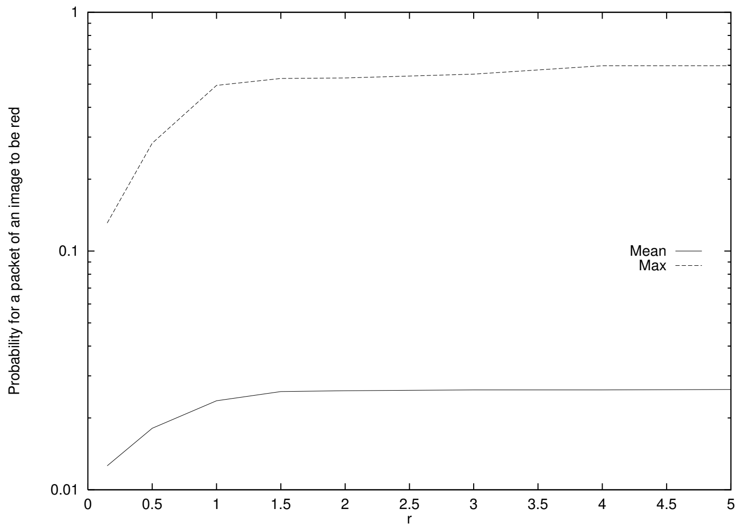

Figure 14: Probability that a packet of an image will be red.

Figure 15: Example of (a) an original and (b) a reconstructed sequence.