A Rate/Quality Controlled MPEG Video Transmission System in a TCP-Friendly Internet Scenario [1]

Francesco Licandro, Giovanni Schembra

DIIT - University of Catania - V.le A. Doria, 6 - 95125 Catania - ITALY
Email: (flicandro, schembra)@diit.unict.it

Abstract

Whereas up to a few years ago the most common applications on the Internet were only TCP-based, recently real-time video applications have also become more and more popular. This evolution is affecting the telecommunications scenario of the past, which featured traffic stability and fairness. In fact, when the same bottleneck link is shared between TCP and UDP flows, in the presence of a high network load and congestion, TCP sources reduce their output traffic, while UDP sources continue to transmit in the same way. To solve this problem some congestion control should be introduced for UDP traffic as well, in such a way that this traffic becomes "TCP-friendly". The problem dealt with in this paper arises from the fact that, although many TCP-friendly algorithms have been introduced to support real-time applications on the Internet, the only target in optimizing them was to achieve fairness with TCP flows in the network. No attention has been paid to the applications using them, and in particular, to the quality of service (QoS) perceived by their users. Starting from the consideration that using a TCP-friendly rate-control algorithm allows traffic sources to know the bandwidth available on the network, and to tune their output rate in order to match it and to minimize losses, the target of this paper is to analyze the problem of transmitting MPEG video over IP when sources use a TCP-friendly algorithm. With this aim, an MPEG video transmission system with both rate and quality control is introduced, and the characteristics of each of its elements are discussed. In order to analyze the performance of the proposed system architecture, the analysis has been carried out on a case study.

1 Introduction

The role of the Internet in daily life has changed in the last few years, and the ever-growing amount of available transmission bandwidth is changing along the Internet application scenario. Whereas, up to a few years ago, the most common applications were only TCP-based, like e-mail, FTP and WEB, real-time applications have recently become more and more popular. In this context, an increasing amount of attention has been paid to technologies for the transmission of video since, due to its high transmission rates, it is likely to constitute a large portion of the Internet traffic of the near future. Various network operators have already started to offer video streaming services and in the next few years most traditional CATV services, currently based on circuit switched (CS) networks, are likely to migrate towards packet switching (PS) technologies. The TCP transport protocol is not suitable for these applications because of its complex retransmission mechanism, and the excessive burstiness of the traffic transmitted. For this reason, all the video streaming applications at present transmit via the UDP transport protocol. This evolution is affecting the stability and fairness of the telecommunications scenario of the past [1]. In fact, when the same bottleneck link is shared between TCP and UDP flows, in the presence of a high network load and congestion, TCP sources reduce their output traffic due to their intrinsic window-based congestion control mechanism, while UDP sources continue to transmit in the same way, as if the network were underloaded. In this way, UDP sources capture most of the bandwidth, heavily penalizing TCP sources. To solve this problem some congestion control should be introduced for UDP traffic

[1] This paper was partially supported by the Project MULTICAMPUS-CT

as well, in such a way that this traffic becomes "TCP-friendly": it has to receive the same average bandwidth over the timescale of a session as a TCP flow along the same path affected by the same delay and packet loss conditions [2]. In other words, the target of TCP-friendly rate control mechanisms is to make UDP traffic sources behave like "good citizens" towards TCP sources. In this perspective, several TCP-friendly algorithms have been proposed in the literature (see the TCP-friendly WEB site [2] for a list of related works). One of the best known algorithms is the TCP-Friendly Rate Control (TFRC) [3]. It is an equation-based congestion control since it uses a control equation that explicitly gives the maximum acceptable sending rate as a function of the recent loss event rate. The sender adapts its sending rate, guided by this control equation, in response to feedback from the receiver. More specifically, in order to be fair to TCP sources which share the same bottleneck link, TFRC uses the TCP response function characterizing the steady-state sending rate of TCP as a function of the round-trip time and steady-state loss rate. The problem dealt with in this paper arises from the fact that although the above algorithms were introduced to support real-time applications on the Internet, the only target in optimizing them was to achieve fairness with TCP flows in the network. No attention has been paid to the applications using them, and in particular, to the quality of service (QoS) perceived by their users. The paper starts from the consideration that using a TCP-friendly rate-control algorithm allows traffic sources to know the available bandwidth in the network, and to tune their output rate in order to match it and to minimize losses. The target is to analyze the problem of transmitting MPEG video over IP when sources use a TCP-friendly algorithm. With this aim, an MPEG video transmission system with both rate and quality control is introduced and the characteristics of each its elements are discussed. The core element is the rate control. To this end, three rate controllers are defined and compared.

In order to analyze the performance of the proposed system architecture, the MPEG video simulation model described in [4, 5, 6] is used, and implemented for the network simulator (ns) in [7]. So, using ns, a transient analysis has been carried out to analyze the differences between the three rate controllers proposed in the paper, in terms of encoding quality stability and fidelity of the output rate to the rate suggested by the TFRC.

The paper is organized as follows. Section 2 provides some background needed for better comprehension of the following sections. Section 3 describes the MPEG video transmission system and introduces the three proposed rate controllers. Section 4 takes a case study and, by simulation, evaluates the aforementioned performance parameters. Finally, Section 5 concludes the paper.

2 Background

In order to make this paper self-consistent, in this section we will illustrate what will be necessary for comprehension of the rest of the paper. More specifically, in Section 2.1 we will describe TFRC, which is the TCP-friendly algorithm we use in our transmission system. Then, in Section 2.2 we will provide a very brief description of the aspects of MPEG encoding and MPEG rate control which will be used in the following sections.

2.1 The TFRC algorithm

In this section we briefly describe TFRC, which is the algorithm we use in our MPEG video transmission system to control the output rate in order to be friendly towards other TCP flows sharing the same congested links in the network. The TFRC algorithm was proposed in [3] as a TCP-friendly algorithm. It is not Additive Increasing Multiplicative Decreasing (AIMD) like TCP, but is based on estimation of the throughput of a TCP source experiencing the same losses and round-trip times. This estimation is made through an equation, and for this reason it belongs to the equation-based rate-control algorithm category. The reasons why a non-AIMD algorithm has been proposed to obtain behavior which has to be fair to TCP traffic are discussed in [3].

Mainly it is to prevent real-time sources using TFRC from changing their rate too abruptly for each single loss event. In any case, TFRC proposers demonstrate that their protocol is fair to TCP and provides Internet stability, as required.

The TFRC congestion control mechanism is distributed between three main protocol entities: the *TX* and *RX* entities in the sender part, and the *PR* entity in the receiver part.

The main target of the *PR* entity is to send feedback to the *RX* entity every time it receives a packet, in order to echo the sequence number from the most recent data packet, along with the time since that packet was received. This will allow the *RX* to calculate the round-trip time. Another important target of the *PR* entity is to calculate the loss event rate, p, and feed this back to the sender. The method for calculating the loss event rate has been the subject of much discussion in the past, and many solutions have been proposed in the literature (see for example [8, 3]). In our transmission system we have used the "Dynamic History Window" method, which uses a history window of packets whose length is determined by the current transmission rate. For a more detailed description of this method and its advantages, please refer to [3].

As far as the sender part is concerned, we have divided it into two different logical entities: the *RX* and *TX* entities. This division was not present in the TFRC definition. It has been introduced here in order to separate packet sending from rate calculation, and is due to the main difference between our approach and the approach followed by the TFRC authors. In fact, the TFRC authors had in mind a "greedy source", that is, a source which always has something to transmit, and whose rate can be imposed by the TFRC. In our case, on the contrary, the TFRC can only "suggest" a rate to the source, because the source considered here presents intrinsic behavior that is not dependent on the network. So the source output rate can only be modulated by the TFRC, and not imposed. In other words, in our case the output rate of the source (on the *TX* side) may not exactly coincide with the rate suggested by the TFRC (on the *RX* side), although the source tries to fit it as well as possible.

The *TX* entity receives groups of packets from the source to be sent through the UDP protocol layer and, for the generic m^{th} group, the rate to be applied to the transmission, $T_{OUT}(m)$. Consequently, the main target of the *TX* entity is to transmit the packets emitted by the source using the rate indicated by the source.

The *RX* entity, on the other hand, calculates the estimated throughput, in bits/sec, according to the equation characterizing the TFRC protocol:

$$T_{TFRC} = \frac{s}{R_S \cdot \sqrt{\frac{2}{3}\,p} + 3\,p \cdot (1 + 32\,p^2) \cdot t_{RTO} \cdot \sqrt{\frac{3}{8}\,p}} \tag{1}$$

where:

- s is the packet size, expressed in bits;
- R_S is the smoothed round-trip time (RTT); it is calculated by the *RX* entity using the information received from the *PR* entity about the times at which it has received the previous packets. The *RX* entity smoothes the RTT using an exponentially weighted moving average (EWMA):
  $$R_S = w \cdot R_S + (1 - w) \cdot SRTT \tag{2}$$
  where $SRTT$ is the last estimated RTT value, and w is the weight of the EWMA, $w \in [0, 1]$. The choice of w is very important. In fact, when its value is too close to 0, the smoothness action is weak, and the smoothed value closely follows the instantaneous process; when, on the contrary, its value is very close to 1, the smoothness action is very strong, and the smoothed process is insensitive to instantaneous variations. A typical value of w is 0.9 [3];
- p is the loss probability estimated by the *PR* entity; this entity sends the value of p to the *RX* entity at least once per round-trip time;

- t_{RTO} is the TCP retransmit timeout value, which can be estimated from the $SRTT$ process through the usual TCP algorithm, $t_{RTO} = SRTT + 4 \cdot \sigma_{RTT}$, where σ_{RTT} is the last estimated standard deviation of the RTT process.

Every time the RX entity receives feedback, it calculates a new value for the allowed sending rate through the control equation in (1). If the previously-calculated sending rate, $T_{TFRC}(t_{n-1})$ is less than the last calculated rate, $T_{TFRC}(t_n)$, the RX entity may increase the sending rate; otherwise it must decrease the sending rate, according to one of the following three choices: an exponentially decrease, a decrease towards $T_{TFRC}(t_{n-1})$, and a decrease to $T_{TFRC}(t_{n-1})$. The third choice has been demonstrated to be the best in [3], since it gives the maximum stability, and therefore we use it in our transmission system. Each time the RX entity calculates a new value for $T_{TFRC}(t_n)$, it passes this value to the source as a suggestion for the rate to be maintained in the future.

2.2 The MPEG standard and its rate control algorithm, TM-5

In this section we describe the main features of MPEG encoding and MPEG rate control. A more accurate description of the MPEG standard can be found in [9, 10]. According to the MPEG video encoding syntax, each movie to be encoded is divided into a sequence of groups of pictures (GoPs), which has a predefined structure containing three types of frames: I-frames, which are intraframe encoded (i.e. without reference to other frames) through a two-dimensional discrete cosine (DCT) transform, begin each GoP; P-frames, which are encoded with reference to previous I- and P-frames through interframe encoding, achieve a better compression ratio than I-frames; finally, B-frames (bidirectional frames), which are encoded with reference to the next and previous I- or P- frames, achieve the highest compression ratio. The number of P-frames in a GoP, N_P, and the number of B-frames in a GoP, N_B, are set by the user application. A typical GoP structure, characterized by $N_P = 1$ and $N_B = 4$, is IBBPBB. Let $N = 1 + N_P + N_B$ be the number of frames in one GoP, and F be the frame rate.

The coding algorithm is based on a division of each picture into blocks, groups of blocks and macroblocks. More specifically, in intraframe encoding, each picture is processed block by block, applying a two-dimensional DCT to each luminance and chrominance block. For the other frames the DCT transform is applied after the motion estimation process. In all cases, the resulting DCT coefficients calculated by the DCT block are then quantized through a quantization matrix for each kind of frame (I, P or B). Thanks to a parameter called *quantizer scale parameter* (*qsp*) it is possible to re-scale all the quantization levels, multiplying the above-mentioned quantization matrices by this parameter. The choice of the *qsp* to achieve a given target constitutes the aim of the so-called *rate controller*. For example, the MPEG Forum has standardized the TM-5 rate controller [11], which varies the *qsp* during encoding in order to encode each GoP with approximately the same number of bits. Its target is to encode frames in such a way that the number of bits used per GoP is equal or very close to a given target number, T. The number of bits not used in a GoP can be used in the next GoP. To this end, a variable R stores the credits or debts accumulated in the past, which are available for the future.

The TM-5 rate control algorithm works in three steps:

- *Target bit allocation*: this step estimates the number of bits available to encode the frames of the beginning GoP. It is performed at the beginning of each GoP;
- *Rate control*: by means of three virtual buffers, one for each encoding mode, this step sets the reference value of the *qsp* for each macroblock;
- *Adaptive quantization*: this step modulates the reference value of the *qsp* according to the spatial activity in the macroblock to be encoded, to derive the value of the *qsp* q to be used to quantize the macroblock.

Target bit allocation is made up of two phases: complexity estimation and frame target setting. The complexity estimation phase calculates the variables representing the 'global complexity' for each kind of frame, X_I, X_P and X_B. The second phase of Target bit allocation is frame target setting. The target number of bits for the frames of the beginning GoP is computed as follows:

$$\begin{aligned} T_I &= \max\left\{\frac{R}{\left(1+\frac{N_P X_P}{X_I K_P}+\frac{N_B X_B}{X_I K_B}\right)}, \frac{T}{8\cdot N}\right\} && \text{for the } I\text{-frames} \\ T_P &= \max\left\{\frac{R}{\left(N_P+N_B\cdot\frac{K_P X_B}{K_B X_P}\right)}, \frac{T}{8\cdot N}\right\} && \text{for the } P\text{-frames} \\ T_B &= \max\left\{\frac{R}{\left(N_B+N_P\cdot\frac{K_B X_P}{K_P X_B}\right)}, \frac{T}{8\cdot N}\right\} && \text{for the } B\text{-frames} \end{aligned} \tag{3}$$

where K_P and K_B are 'universal' constants and are set to $K_P = 1.0$ and $K_B = 1.4$ as in [11]. R is the number of bits available to encode the following frames in the GoP. At the beginning of the sequence it is set to $R = 0$. It is updated at the end of each GoP as follows:

$$R = R + T \tag{4}$$

After each frame, it is updated as $R = R - S_F$, S_F being the number of bits used to encode this frame.

The second step is *Rate Control*, which is performed macroblock by macroblock. It is based on the state of three virtual buffers, d_I, d_P and d_B, one for each kind of frame. They are three counters which have memory of the past, in particular how many debts or credits have been accumulated for each kind of frame. Before encoding the generic macroblock j of a frame of kind ς, $\varsigma \in \{I, P, B\}$, the state of the appropriate virtual buffer, $d_j^{(\varsigma)}$, is updated as follows :

$$d_j^{(\varsigma)} = d_0^{(\varsigma)} + S_M^{(j-1)} - T_\varsigma \cdot \frac{j-1}{W} \tag{5}$$

where we have indicated:

- $d_0^{(\varsigma)}$: the state of the virtual buffer relating to the frame ς before encoding the first block in the frame;
- $S_M^{(j-1)}$: the number of bits generated by encoding the first $(j-1)$ macroblocks in the frame;
- W: the number of macroblocks in one frame;
- $d_j^{(\varsigma)}$: the current state of the virtual buffer relating to the frame ς.

The final states of each virtual buffer, $d_W^{(I)}$, $d_W^{(P)}$ and $d_W^{(B)}$, are used as $d_0^{(I)}$, $d_0^{(P)}$ and $d_0^{(B)}$ to encode the next frame of the same type.

Then, from $d_j^{(\varsigma)}$ the reference *qsp* $q_j^{(REF)}$ for the macroblock j is computed as follows:

$$q_j^{(REF)} = \frac{31 \cdot d_j^{(\varsigma)}}{r} \tag{6}$$

where r is the *reaction parameter*, given by:

$$r = 2 \cdot T/N \tag{7}$$

Finally, the last step is *Adaptive Quantization*. First the spatial activity measure for the macroblock j, A_j, is calculated from the four luminance frame-organized sub-blocks and the four luminance field-organized sub-blocks using the intra-pixel values:

$$A_j = 1 + \min\left(V_1^{(BLK)}, V_2^{(BLK)}, \dots, V_8^{(BLK)}\right) \tag{8}$$

where, if we indicate the samples in the n^{th} original 8x8 block as $P_{n,k}$, $\forall k \in \{1, \dots, 64\}$, the generic term $V_n^{(BLK)}$ in (8) is defined as follows:

$$V_n^{(BLK)} = \frac{1}{64} \cdot \sum_{k=1}^{64} \left(P_{n,k} - \overline{P}_n\right)^2 \qquad \text{and} \qquad \overline{P}_n = \frac{1}{64} \cdot \sum_{k=1}^{64} P_{n,k} \tag{9}$$

Then the activity is normalized as $A_j^{(N)} = (2 \cdot A_j + \overline{A})/(A_j + 2 \cdot \overline{A})$, where $\overline{A}$ is the average value of A_j in the last encoded frame. For the first frame it is set to $\overline{A} = 400$. Finally, the value of the *qsp* for the macroblock j, q_j, is calculated by clipping to the range $[1, \dots, 31]$ the parameter $\hat{q}_j = A_j^{(N)} \cdot q_j^{(REF)}$, where $q_j^{(REF)}$ is the reference *qsp* calculated in (6).

3 MPEG video transmission system

In this section we describe the system proposed in this paper to transmit MPEG video on the Internet using the TFRC rate control algorithm. The target of this system is twofold: on the one hand the output bit rate has to follow the bandwidth available in the network; on the other it has to respect user requirements in terms of encoding quality and, in particular, it should protect the quality of the movie being encoded from oscillations due to rapid variations in the network bandwidth. With this in mind, the system we propose is shown in Fig. 1, where the three main component blocks of the system are represented: the *TFRC*, the *Network bandwidth smoother* and the *Rate/Quality MPEG video source* (RQ-source), whose functions will be described in the following sections.

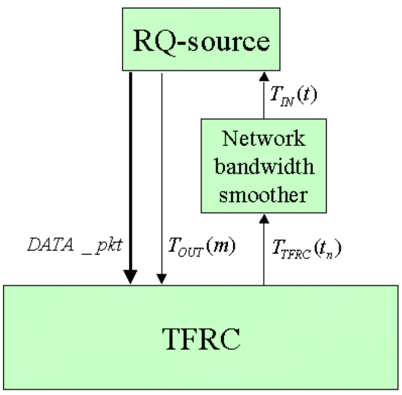


Figure 1: Rate/Quality controlled MPEG video transmission system.

3.1 TFRC block

The *TFRC* block has the aim of choosing the bit rate according to the result provided by the TFRC equation, and to transmit packets arriving from the MPEG video source through the UDP layer. The first function is implemented by the *RX* entity, the second by the *TX* entity. The *TFRC* block works as described in Section 2.1. The *RX* entity changes the rate according to (1), and passes the new rate to the *Network bandwidth smoother*. Let us indicate the instant when

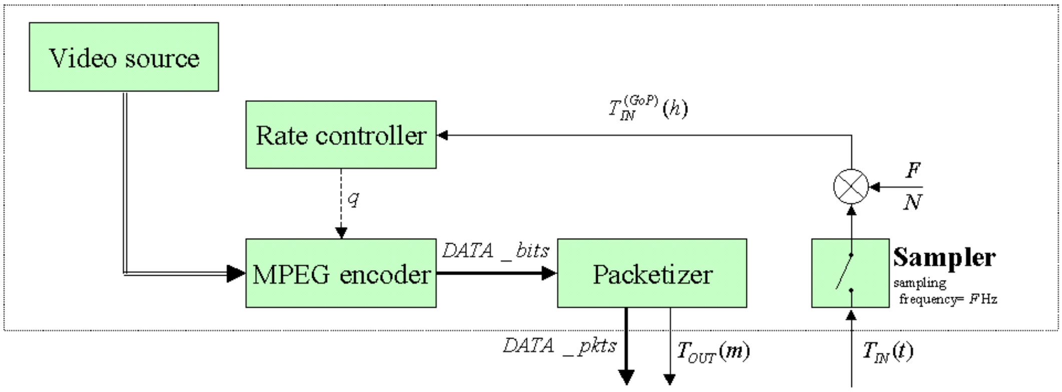


Figure 2: RQ-source.

the n^{th} rate-change event occurs as t_n, and the new rate selected at this instant as $T_{TFRC}(t_n)$. This process constitutes the input of the *Network bandwidth smoother*. The *TX* entity receives from the RQ-source the packets of the last encoded frame to be transmitted in the next frame interval, and the transmission rate, expressed in bits/sec, to be applied to transmit these packets, $T_{OUT}(m)$. So, for the whole m^{th} interval, it transmits the packets using an inter-packet gap given by:

$$ipg_m = \frac{F \cdot s}{T_{OUT}(m)} \tag{10}$$

where F is the frame rate, and s is the packet size, expressed in bits.

3.2 Network bandwidth smoother

The *Network bandwidth smoother* receives the available network bandwidth estimated by the TFRC, $T_{TFRC}(t_n)$, and has the aim of eliminating the high frequencies of the process. This is achieved by filtering this process by means of a low-pass filter with an Exponential Weighted Moving Average (EWMA):

$$\hat{T}_S(t_n) = \alpha \cdot T_{TFRC}(t_{n-1}) + (1-\alpha) \cdot \hat{T}_S(t_{n-1}) \tag{11}$$

where α is the weighting parameter, $\alpha \in [0,1]$, while t_{n-1} and t_n are the instants of the $(n-1)^{th}$ and n^{th} rate variations, respectively. Finally, the output of the *Network bandwidth smoother* is obtained as a continuous-time step function derived from $\hat{T}_S(t_n)$ as follows:

$$T_{IN}(t) = \hat{T}_S(t_{n-1}) \qquad \forall t \in [t_{n-1}, t_n[\tag{12}$$

In this way the process $T_{IN}(t)$ represents the available bandwidth calculated by the TFRC, and its variations can be regulated by the parameter α: the less α is, the more stable the process $T_{IN}(t)$, but the less the system responsiveness.

3.3 Rate/Quality MPEG video source (RQ-source)

The *Rate/Quality MPEG video source* (RQ-source) is an MPEG video source whose emission rate is controlled through a timely choice of the quantizer scale for each macroblock to be encoded, in such a way as to respect the network bandwidth calculated by the *TFRC* block, and to keep the encoding quality as stable as possible. A scheme of the RQ-source is represented in Fig. 2: the video source produces a video stream which is first encoded in the MPEG encoder, then packetized in the Packetizer, and finally passed to the *TFRC* block to be transmitted. The MPEG encoder uses the *qsp* q calculated by the *Rate controller* for each macroblock to be encoded, and encodes the movie as illustrated in Section 2.2. Adaptation of the output bit rate according to the available network bandwidth calculated by the *TFRC* is, as said previously, a task for the *Rate controller*. The *Rate controller*, at the beginning of the generic GoP h, calculates the available network bandwidth for the beginning GoP, $T_{IN}^{(GoP)}(h)$, by sampling the output of the *Network bandwidth smoother* at a frequency F, and multiplying it by F/N. This value represents the target T to be allocated to the frames in the beginning GoP according to (3).

Below we propose three different rate controllers, whose performance will be evaluated in Section 4:

1. $RC_{[TM5]}$: a rate controller using the classical TM-5 rate control algorithm defined by the MPEG forum [11], and described in Section 2.2.

2. $RC_{[TM5]}^{(H)}$: a rate controller derived from $RC_{[TM5]}$, by imposing a hysteresis mechanism to guarantee greater stability in the encoding quality. This mechanism is defined as follows. Let us use the Peak Signal-to-Noise ratio (PSNR) to represent the encoding distortion, and let $\underline{\Phi}$ be a set of L PSNR levels defined to quantize the PSNR process, $\Phi_1, \ldots, \Phi_L$. We define the following *hysteresis mechanism rules*:

   - the frame distortion must remain at the same PSNR level for at least H frames, where H is called the *hysteresis coefficient*;
   - the frame distortion can step by at most one PSNR level up or down;
   - if the PSNR associated to the quantizer scale suggested by the $RC_{[TM5]}$ by means of the rate-distortion curves [6] does not belong to one of the allowed PSNR levels, the quantizer scale providing the PSNR of the allowed level closest to the suggested one (at most one PSNR level up or down) will be chosen.

   Thanks to this mechanism we assure that the encoding quality remains stable for at least H frames, and that the quality variations are not too sudden.

3. $RC_{[TM5]}^{(H,ML)}$: a memory-less rate controller with hysteresis, defined as an extension of the $RC_{[TM5]}^{(H)}$ rate controller to avoid the fact that, due to hysteresis, such a great number of credits or debts are accumulated that, when it is possible to change level (after H frames), the source will not follow the behavior required by the *TFRC*. In other words, we propose the $RC_{[TM5]}^{(H,ML)}$ rate controller to avoid the unresponsiveness problems of the $RC_{[TM5]}^{(H)}$, which will be highlighted in Section 4.

   To obtain this we derive the $RC_{[TM5]}^{(H,ML)}$ rate controller from the $RC_{[TM5]}^{(H)}$ rate controller eliminating, at the beginning of each GoP, the memory of the past history contained by the variable R in its TM-5 algorithm, replacing (4) with $R = T$.

4 Numerical results

In this section we will analyze the performance of the MPEG video transmission system described in Section 3, and analyze the differences between the three rate controllers described in Section 3.3.

As a case study we have considered the network topology shown in Fig. 3, made up of one node loaded by only one MPEG video source transmitting the movie "The Silence of the Lambs", encoded with the GoP structure IBBPBB. Let us note that this constitutes the worst case for the quality stability of the video source, because it is the same source that has to face up to congestion situations, drastically reducing its output rate. Instead, if many sources load the same congested node, the required rate reduction is distributed to all the sources, and each source

Figure 3: Network topology.

experiences less rate variation. The link capacity used in our experiments is $C = 2$ Mbits/sec and the end-to-end propagation delay is 2 msec, corresponding to a link length of 360 Km.

We compare the rate processes at the input and output of the RQ-source, evaluate the PSNR process as a measure of the encoding distortion at the sender side, and show the loss process as a measure of what is received at the receiver side. Given that the input rate process $T_{IN}^{(GoP)}(h)$ is the number of bits to be used in one GoP, we define the process $T_{OUT}^{(GoP)}(h)$ by averaging the values of $T_{OUT}(m)$ over all the frames belonging to the same GoP h. As far as the measure of the encoding distortion is concerned, from a subjective analysis obtained with 300 test subjects on the movie "The Silence of the Lambs", the following $L = 5$ levels of distortion were envisaged: $\Phi_1 = [39.2, 49.2]dB$, $\Phi_2 = [36.2, 39.2]dB$, $\Phi_3 = [35.0, 36.2]dB$, $\Phi_4 = [33.7, 35.0]dB$ and $\Phi_5 = [31.5, 33.7]dB$. One frame for each of the above distortion levels is represented in Fig. 4.

(a) PSNR level Φ_1

(b) PSNR level Φ_2

(c) PSNR level Φ_3

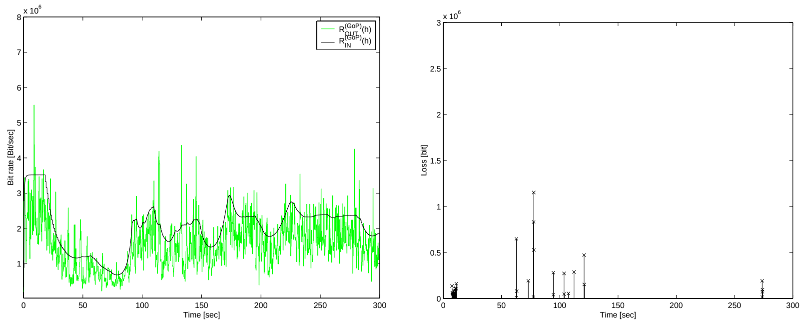
(d) PSNR level Φ_4

(e) PSNR level Φ_5

Figure 4: Image comparison for five different distortion levels calibrated through subjective testing.

Let us begin with the results obtained using the $RC_{[TM5]}$ rate controller. Fig. 5 compares the processes $T_{IN}^{(GoP)}(h)$ and $T_{OUT}^{(GoP)}(h)$ and show the loss process. In Fig. 6(a) we have shown the PSNR process. We can note that this last process strongly oscillates in a very wide range, and this represents an unacceptable oscillation of the quality perceived at the destination. To better observe quality oscillations, in Fig. 6(b) we have shown the PSNR process quantized in the five levels Φ_1, ..., Φ_5, and from this figure we can conclude that the oscillations shown in Fig. 6(a) cause unacceptable variations in the PSNR levels.

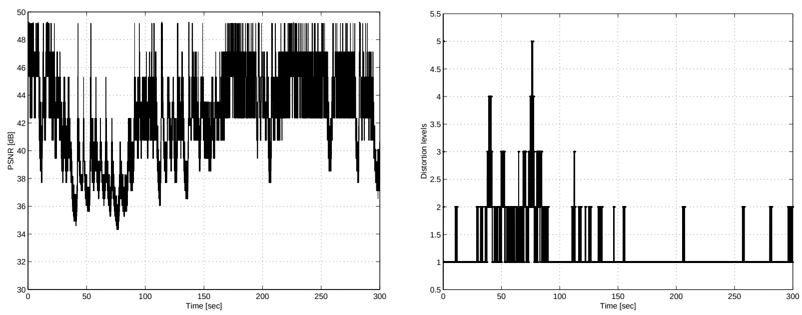
In order to improve the quality stability, we have introduced the hysteresis mechanism described in Section 3, and obtained the $RC_{[TM5]}^{(H)}$ rate controller. The performance results obtained with this rate controller are shown in Figs. 7 and 8. From these figures we can observe



(a) $T_{IN}^{(GoP)}(h)$ - $T_{OUT}^{(GoP)}(h)$ comparison.

(b) Loss process

Figure 5: Rate and loss processes for the $RC_{[TM5]}$ rate controller.



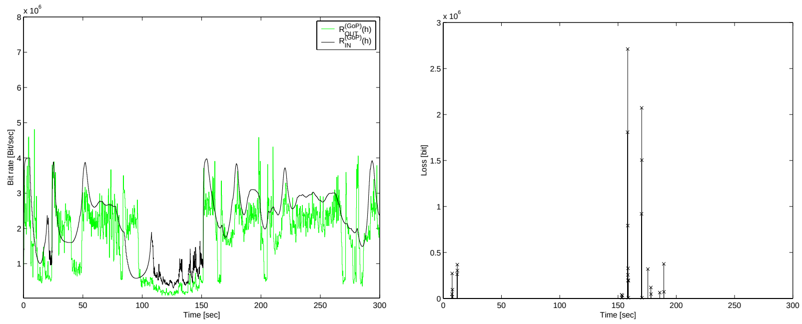
(a) PSNR process

(b) Quality level process

Figure 6: Quality processes for the $RC_{[TM5]}$ rate controller.

that, using this rate controller, although the resulting quality is more stable in terms of both the PSNR process and PSNR-level process, the source output rate does not fit the network bandwidth suggested by the TFRC very well. This is due to the fact that, during the periods in which the PSNR level cannot be changed due to the hysteresis, the RQ-source maintains its output rate almost constant, and therefore it may accumulate a very large number of credits or debts. This is the case, for example, of the interval immediately after the instant $t = 150$ sec, where the TFRC suggests an ever-increasing rate, while the source maintains a constant rate equal to that suggested at the beginning of the period. All the debts accumulated in that period cause the rate peak at the instant $t = 161$ sec, and a consequent immediate decrease in the rate suggested by the TFRC. The source rate falls, and the situation described previously recurs. This causes oscillations and a bad fitting of the rate suggested by the TFRC.

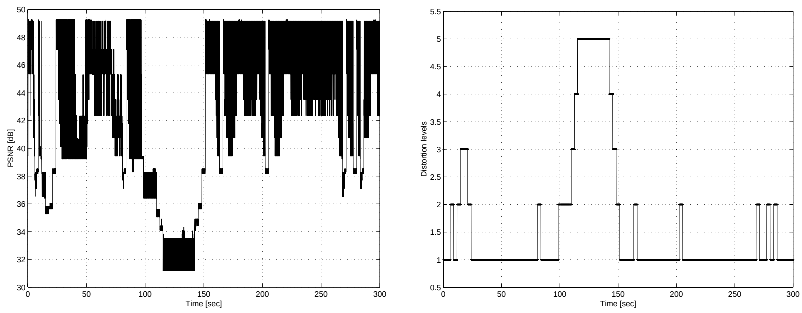
The real cause of this is the long-term memory of the $RC_{[TM5]}^{(H)}$ rate controller. Let us note that the memory maintained by the TM-5 is useless for real-time applications, and sometimes deleterious for performance, given that it is better to follow the network bandwidth in the short term, rather than trying to capture the average bandwidth in the long term only. For this reason, we have introduced the $RC_{[TM5]}^{(H,ML)}$ rate controller, which deletes its memory at the beginning of each GoP. The performance obtained is shown in Figs. 9 and 10. As we can see, thanks to this rate controller, the output rate closely fits the rate suggested by the TFRC, the losses are negligible, and the PSNR remains within the same level for a very long time.



(a) $T_{IN}^{(GoP)}(h)$ - $T_{OUT}^{(GoP)}(h)$ comparison.

(b) Loss process

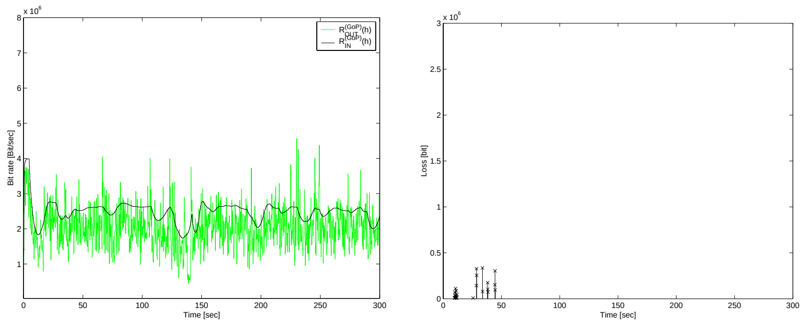
Figure 7: Rate and loss processes for the $RC_{[TM5]}^{(H)}$ rate controller.



(a) PSNR process

(b) Quality level process

Figure 8: Quality processes for the $RC_{[TM5]}^{(H)}$ rate controller.



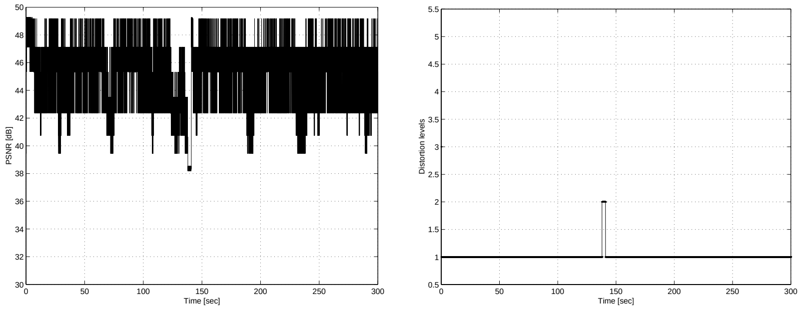
(a) $T_{IN}^{(GoP)}(h)$ - $T_{OUT}^{(GoP)}(h)$ comparison.

(b) Loss process

Figure 9: Rate and loss processes for the $RC_{[TM5]}^{(H,ML)}$ rate controller.

5 Conclusions

The paper analyzes the problem of transmitting MPEG video over IP when sources use a TCP-friendly algorithm to be friendly to TCP traffic, and to determine the bandwidth available on the network. With this aim, an MPEG video transmission system with both rate and quality control



(a) PSNR process (b) Quality level process

Figure 10: Quality processes for the $RC_{[TM5]}^{(H,ML)}$ rate controller.

have been introduced and the characteristics of each its elements discussed. In order to analyze the performance of the proposed system architecture, a transient analysis has been carried out on a case study. The core of the MPEG video transmission system is the rate controller. Three rate controllers have been introduced and compared. The results have demonstrated that the best performance in terms of quality stability are obtained with the $RC_{[TM5]}^{(H,ML)}$ rate controller, which extends the classical $RC_{[TM5]}$ based on TM5, by using a hysteresis mechanism and deletes memory at the beginning of each GoP in order to improve source responsiveness to network bandwidth variations.

References

[1] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, August 1999.

[2] J. Mahdavi, S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control", Technical note sent to the end2end-interest mailing list, January 8, 1997, available at: *http://www.psc.edu/networking/tcp_friendly.html*"

[3] S. Floyd, M. Handley, J. Padhye, J. Widmer, "Equation-based congestion control for unicast applications," *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, August 28 - September 1, 2000.

[4] A. Lombardo, G. Schembra, "TSpec Enforcement for MPEG Video Transmission over the Next Generation Internet: an Analytical Framework," *Computer Networks and ISDN Systems*, Vol. 37, No. 5, November 2001.

[5] A. Cernuto, F. Cocimano, A. Lombardo, G. Schembra, "A Queueing System Model for the Design of Feedback Laws in Rate-Controlled MPEG Video Encoders," to appear on *IEEE Transactions on Circuits and Systems for Video Technology*, 2002.

[6] A. Lombardo, G. Morabito, G. Schembra, "Traffic specifications for the Transmission of Stored MPEG Video on the Internet," *IEEE Transactions on Multimedia*, Vol. 3, No. 1, March 2001.

[7] F. Licandro, G. Schembra, "MPEG traffic generator for ns-2", available at: *www.diit.unict.it/users/schembra/Software/MPEG_over_ns2/MPEG_over_ns2_v01.zip.*

[8] J. Widmer, "Equation-based Congestion Control," Diploma Thesis, February 2000, available at: *http://www.icsi.berkeley.edu/~widmer/tfrc/thesis.ps.gz.*

[9] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Communications ACM*, vol. 4, April 1991.

[10] W. Luo, M. El Zarki, "Quality Control for VBR Video over ATM Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 6, August 1997.

[11] Test Model 5, ISO-IEC/JTC1/SC29/WG11, April 1993.