Effect of Input Traffic Correlation on Clock Recovery in MPEG-2 Systems Layer

Christos Tryfonas
Anujan Varma

UCSC-CRL-99-6
March 20, 1999

Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA 95064

ABSTRACT

We analyze the effect of the correlation of the delay experienced by the video traffic stream on the quality of the clock recovered by an MPEG-2 decoder. We prove analytically that, for a general input process, high correlation of the delay samples produces a large variance of the recovered clock. We verify the analytical result through actual simulation of an MPEG-2 Systems decoder PLL with correlated traffic patterns (AR processes) as the input. We also show by simulation the effects of bimodal delay distributions, such as those resulting from the PCR-unaware encapsulation scheme, on the quality of the receiver clock. The problem is also brought out in the case of practical networks through an experiment which indicates that the temporal characteristics of the delays encountered by the video stream in a packet network depend heavily on the scheduling disciplines used in the network switches.

Keywords: MPEG-2, clock recovery, ATM networks, set-top box, MPEG-2 Systems layer.

1 Introduction

Packet switching is increasingly used for real time data transfer, such as multimedia traffic (e.g., audio and video). The provision of Quality-of-Service (QoS) in newly designed packet-switched networks attempts to solve the inefficiency of mechanisms pre-existing in traditional Internet Protocol (IP) networks to support this sort of data. Although voice is the primary multimedia traffic used today in packet switching, video traffic is increasingly present.

Clearly, the ever-increasing demand for more bandwidth by multimedia traffic, in conjunction with the slower rates of its provision by public operators and the costs associated with its use, call for more efficient use of communications links. This, among others, calls for sophisticated compression algorithms.

MPEG-2 is rapidly adopted as a standard for audio and video compression in packet networks [7, 8]. The MPEG-2 Systems Layer defines two ways to multiplex elementary audio, video or private streams to form a program [15, 18]: the *Program Stream* and the *Transport Stream*. The MPEG-2 Transport Stream format is the preferred choice in environments where errors are likely to occur, as in the case of transport over a packet-switched network. It is currently used in Digital Video Broadcasting (DVB) systems for digital television and data broadcasting across a broad range of delivery media. The format makes use of explicit timestamps (called *Program Clock References* or *PCR*s in MPEG-2 terminology) embedded within the transport packets to facilitate the clock recovery at the receiver.

Several applications with stringent clock requirements need high accuracy and stability of the recovered clock to provide acceptable quality to the user. Such cases include applications in which the reconstructed system clock is used to directly synthesize a chroma sub-carrier for the composite video signal feeding the TV monitor. The degradation of the reconstructed clock at the receiver is caused primarily by the *packet delay variation (jitter)*. The jitter may be either network-induced due to the specific scheduling disciplines used in the end-to-end path, or sender-induced due to the encapsulation procedures at the originating nodes.

This paper focuses on the effects of the autocorrelation of the delays experienced by the incoming PCR samples on the clock recovery process of the MPEG-2 Systems decoder. The effect of the autocorrelation of PCR delays on the clock recovery process of the MPEG-2 Systems layer was first identified by Andreotti et al. [2], where the effect was verified through some experiments. However, we are not aware of any detailed analysis of the problem in the literature. In this paper, we provide an analysis of the effects of the correlation of PCR delays on MPEG-2 clock recovery when a standard PLL is used to derive the clock at the decoder end. We prove analytically that, for general input processes with exponentially decreasing autocorrelation functions, a high correlation produces a large variance of the recovered clock.

We verify our analytical results and provide insight for cases with more general autocorrelation functions through a series of simulation experiments. We start with simulation experiments of an MPEG-2 Systems decoder PLL with theoretical traffic patterns that have exponentially decreasing or more general autocorrelation functions as the inputs. In both cases, we verify that high correlation of the input traffic process results in poor quality of the recovered clock on the receiver side. We extend our results by showing the effects of high correlation on the resulting quality in bimodal delay distributions as in the case of PCR-unaware scheme, through simulation experiments. These experiments make use of synthetic traces in which the polarity of PCR values is determined by a random telegraph process with a tunable correlation coefficient. Finally, we demonstrate the effect

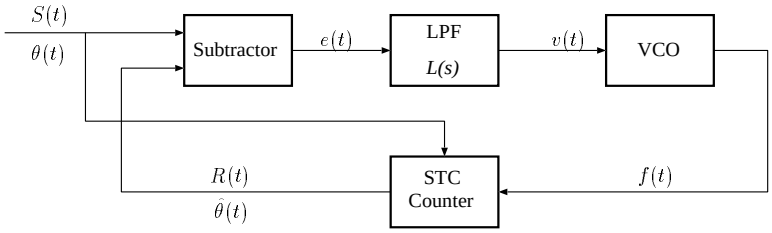


Figure 2.1: Equivalent model of the PLL used.

through a simple experiment in a ATM network in which a scheduling discipline is shown to be responsible for degrading the quality of the recovered clock by introducing higher autocorrelation than other scheduling disciplines used in the same experiment.

The rest of this paper is organized as follows: Section 2 provides analytical results for the effects of jitter correlation on the PAL/NTSC clock recovery process using a standard PLL in an MPEG-2 Systems decoder. Section 3 describes the experiments performed using an MPEG-2 Systems decoder implemented using a standard PLL with both theoretical input traffic patterns and input from MPEG-2 Transport Stream traces being sent through an ATM network. Finally, Section 4 summarizes our results and discusses the experience gained from this work.

2 Analysis

In this section, we provide an analysis of the effects of delay correlation on the MPEG-2 decoder PLL. We first derive the equations for the recovered frequency function of the decoder PLL based on the analysis given in [20]. Next, we proceed to characterize the input signal at the PLL resulting from a process with variable correlation given by an exponentially decreasing function, and show analytically that, high correlation results in high variance of the recovered clock on the MPEG-2 decoder PLL.

2.1 Derivation of the Recovered Function of the Decoder PLL

The PLL model used in the analysis is shown in Figure 2.1. The transfer function $H(s)$ of the closed-loop PLL derived in [20] is given by

$$H(s) = \frac{KL(s)}{s + KL(s)}. \tag{2.1}$$

where K is the gain of the loop, and $L(s)$ is the transfer function of the low-pass filter (LPF). Then, the Laplace transform $F(s)$ of the recovered frequency function $f(t)$ is given by

$$\begin{aligned} F(s) &= (1 - H(s))\,\Theta(s)L(s)K \\ &= K\left(1 - \frac{KL(s)}{s + KL(s)}\right) L(s)\Theta(s) \qquad (2.2) \\ &= P(s)\Theta(s), \qquad (2.3) \end{aligned}$$

where $\Theta(s)$ is the Laplace transform of the input function $\theta(t)$, and $P(s)$ is given by

$$P(s) = K\left(1 - \frac{KL(s)}{s + KL(s)}\right) L(s). \tag{2.4}$$

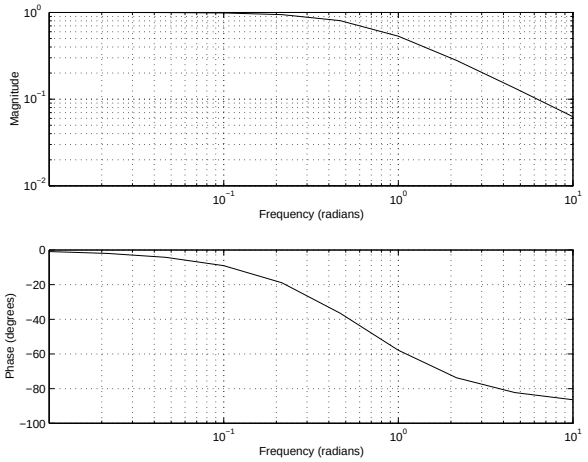


Figure 2.2: Frequency response of the first-order Butterworth LPF used in the analysis.

For the sake of convenience, we need to assume a specific LPF for the PLL design. In this analysis, we use a first order Butterworth analog LPF with a cutoff frequency of 0.1 Hz. This cutoff frequency provides good stability and tracking performance to the decoder PLL as shown experimentally by Tryfonas et al. [18] and analytically by Kaiser [9]. The transfer function of a first-order Butterworth LPF is given by

$$L(s) = \frac{0.2\pi}{s - s_1}, \tag{2.5}$$

with s_1 being the single real pole of the filter that has a value of

$$\begin{aligned} s_1 &= 2\pi 0.1 \exp(j\pi) \\ &= 0.2\pi \exp(j\pi) \\ &= -0.2\pi. \end{aligned} \tag{2.6}$$

Hence, from Eq. (2.5) and (2.6) we obtain

$$L(s) = \frac{0.2\pi}{s + 0.2\pi}. \tag{2.7}$$

The frequency response of the filter is shown in Figure 2.2.

We now proceed with the characterization of the input function and the derivation of the variance of the recovered frequency after the input is applied.

2.2 General Input Process with Variable Correlation

We assume a general continuous-time input process $G(t)$ which has zero-mean and is stationary or cyclostationary. For the sake of simplicity, we also assume that the autocorrelation of this process is given by an exponentially decreasing function with a tunable decay rate given by

$$R_G(\tau) = E\{G(t)G(t+\tau)\} = e^{-\alpha|\tau|}, \alpha \geq 1, \tag{2.8}$$

in which the parameter α is used to change the decay rate of the exponentially decreasing function $R_G(\tau)$.

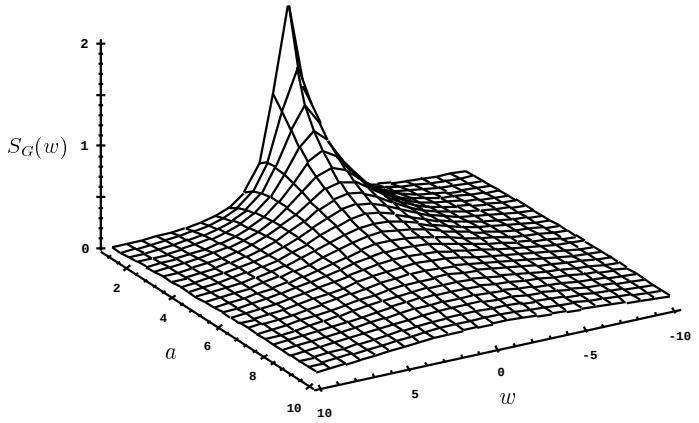


Figure 2.3: Illustration of the function $S_G(w)$ for various values of α.

Since we are interested in the tracking performance of the PLL, we also assume that the PLL is locked before the input process is applied as the input function $\theta(t)$ of the PLL. The analysis is focused on obtaining the variance or the actual function that describes the recovered clock, i.e., $f(t)$. The goal of the analysis is to show that for higher decay rates (smaller correlation), the variance of the recovered clock becomes smaller. Smaller variance of the recovered clock corresponds to a clock with better quality.

We first obtain the *power spectral density (psd)* of the input process. The psd function of the input process, denoted by $S_G(w)$, is given as the Fourier transform of the autocorrelation function $R_G(k)$. Thus,

$$\begin{aligned} S_G(w) &= \int_{-\infty}^{\infty} R_G(\tau) e^{-jw\tau} d\tau \\ &= \int_{-\infty}^{\infty} e^{-\alpha|\tau|} e^{-jw\tau} d\tau \\ &= \int_{-\infty}^{0} e^{\alpha\tau} e^{-jw\tau} d\tau + \int_{0}^{\infty} e^{-\alpha\tau} e^{-jw\tau} d\tau \\ &= \frac{1}{\alpha - jw} + \frac{1}{\alpha + jw} \\ &= \frac{2\alpha}{\alpha^2 + w^2}. \end{aligned}$$

The $S_G(w)$ function is illustrated in Figure 2.3. It is evident that $S_G(w)$ has greater low-frequency content for smaller values of α. The psd function of the recovered clock is given by

$$S_f(w) = S_G(w)|P(w)|^2,$$

where $|P(w)|$ is the magnitude of the Fourier transform of the function with Laplace transform given in Eq. (2.4). We can obtain the Fourier transform of the signal that has Laplace transform of $P(s)$ by substituting s with jw. After substitution, and using Eq. (2.7) we get

$$P(w) = \frac{Kw}{w + \left(\frac{w^2}{0.2\pi} - K\right) j}.$$

The square of the magnitude of $P(w)$ is given by

$$\begin{aligned} |P(w)|^2 &= P(w)P^*(w) \\ &= \left(\frac{Kw}{w + \left(\frac{w^2}{0.2\pi} - K\right) j}\right)\left(\frac{Kw}{w - \left(\frac{w^2}{0.2\pi} - K\right) j}\right) \\ &= \frac{(Kw)^2}{w^2 + \left(\frac{w^2}{0.2\pi} - K\right)^2}, \end{aligned} \tag{2.9}$$

where $P^*(w)$ is the conjugate of $P(w)$.

The variance of the output process is determined by the inverse Fourier transform of $S_f(w)$ at point zero.

$$\begin{aligned} \sigma^2_{f(t)} &= \mathcal{F}\left\{S_f(w)\right\}|_0 \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_f(w)dw \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \left(\frac{2\alpha}{\alpha^2 + w^2}\right)\left(\frac{(Kw)^2}{w^2 + \left(\frac{w^2}{0.2\pi} - K\right)^2}\right) dw \\ &= \frac{\pi K^2 \alpha}{5\alpha^2 + \pi\alpha + \pi K}. \end{aligned} \tag{2.10}$$

In order to check whether the $\sigma^2_{f(t)}$ function is increasing or decreasing we need to compute the first derivative with respect to α

$$\begin{aligned} \frac{d\sigma^2_{f(t)}}{d\alpha} &= \frac{d}{d\alpha}\left(\frac{\pi K^2 \alpha}{5\alpha^2 + \pi\alpha + \pi K}\right) \\ &= \frac{\pi K^2(5\alpha^2 + \pi\alpha + \pi K) - \pi K^2\alpha(10\alpha + \pi)}{(5\alpha^2 + \pi\alpha + \pi K)^2} \\ &= \frac{\pi K^2(\pi K - 5\alpha^2)}{(5\alpha^2 + \pi\alpha + \pi K)^2}. \end{aligned}$$

The sign of the derivative is determined by the term $(\pi K - 5\alpha^2)$. In order for this to be negative, we must have

$$\pi K - 5\alpha^2 < 0, \tag{2.11}$$

which becomes true when $K < \frac{5\alpha^2}{\pi}$, with the worst case being when α is equal to 1. Therefore, K must be less than $\frac{5}{\pi}$, or 1.59. This is true in an MPEG-2 Systems decoder since K is used to scale the input signal to the appropriate levels for the MPEG-2 frequency. More specifically, the design of the VCO takes into account the maximum difference in ticks of a 27 MHz clock when the jitter is at its maximum allowed value, and the limits of the frequency of the decoder. Since, according to MPEG-2 standard [7], the maximum jitter expected is around ± 0.5 ms, the maximum allowable difference is 13500 ticks. For this maximum difference, the decoder must operate at a frequency within the limits specified in the MPEG-2 standard [7]

$$\text{27 MHz - 810 Hz} \leq \text{decoder clock frequency} \leq \text{27 MHz + 810 Hz.}$$

Therefore, the value of K must be chosen close to $\frac{810}{13500} = 0.06$, for the decoder to operate correctly. Since the inequality in Eq. (2.11) is true when $\alpha \geq 1$, the derivative of the variance in Eq. (2.10) is negative, and thus the variance is a decreasing function of α. However, smaller values of α means greater correlation in the input process. It becomes evident that the recovered clock has greater variance as the input process becomes more correlated. This is also illustrated in Figure 2.4.

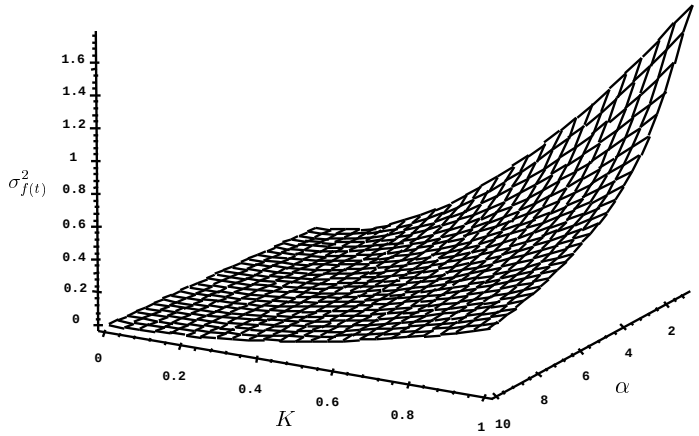


Figure 2.4: Variance of recovered clock frequency ($\sigma^2_{f(t)}$) for various values of K, α.

In this analysis, we did not make any other assumption for the input process besides the shape of its autocorrelation function. If we also assume that the input process is Gaussian, which is commonly used in communications, we can further elaborate on the resulting process of the recovered clock. Since the PLL is a linear time-invariant system, and the input process is Gaussian, the PLL's output process (the frequency of the decoder) must also be a Gaussian process. Additionally, since the recovered PAL/NTSC sub-carrier frequency is only a scaled version of the decoder frequency, it must also be Gaussian. The latter means that the probability that the recovered clock violates the predefined bound increases as its variance increases. This result is true for any process whose cumulative distribution function is monotonic with respect to the variance, since Eq. (2.10) was derived for a general process having an autocorrelation function given in Eq. (2.8). Therefore, we conclude that the quality of the recovered clock is degraded as the correlation in the input traffic process increases.

We continue with a presentation of a series of simulation experiments used not only to verify our analytical results, but also to provide us with some insight in cases in which the jitter process has a more general autocorrelation function.

3 Experimental Results

In this section, we validate the effects of jitter correlation on the clock recovery process by simulation. We present three sets of experiments: in the first, an MPEG-2 Systems decoder is fed by a theoretical input traffic pattern with variable correlation. In this case, we use autoregressive (AR) input processes with non-monotonic autocorrelation functions in addition to AR input processes with exponentially decreasing autocorrelation functions. The second set of experiments simulates the PCR-unaware encapsulation scheme [18], using input processes that take values from a bimodal distribution. These experiments are based on synthetic traces where the PCR polarity is determined by a random telegraph process with a tunable correlation coefficient. The last experiment consists of an MPEG-2 Systems decoder fed by traffic produced from a simulated multi-hop ATM network with different scheduling disciplines and varying cross-traffic load.

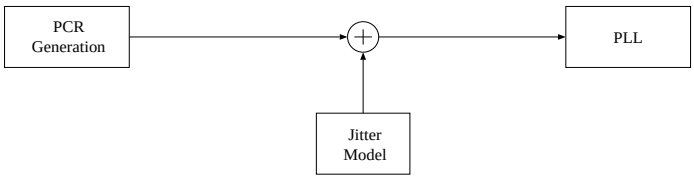


Figure 3.1: Simulation model used in the experiment to check the impact of jitter correlation on the clock recovery process.

3.1 Experiments with AR input processes

A PCR generation model is constructed that follows the MPEG-2 specifications. MPEG-2 specifications for clock insertion in the MPEG stream limit the maximum interarrival time between two consecutive PCR values to 100 ms. Thus, the PCR insertion frequency of 10 Hz can be thought as the worst case. Constant interarrival times for the generation of packets containing PCR values are considered in the following experiments.

Two different approaches can be used in order to model packet delay variation (jitter): in the first, the packets are delayed according to a delay distribution function, and therefore, get forwarded with non-regular spacings. This is the traditional way of generating jitter in which the jitter changes the actual arrival times of the packets. Another way of modeling jitter is to assume that the packets will be equally spaced and the jitter will be simulated by changing the PCR values that the packets carry according to the same delay distribution function. Although the two methods described above are not equivalent, the convergence time and the steady-state error that a PLL produces when fed by the corresponding sequence of values, are almost the same as shown in [2]. For the sake of convenience, we chose the second method to model jitter and therefore, the jitter in terms of ticks of a 27 MHz clock, is added directly to the incoming PCR values. An overview of the model used in the experiments is illustrated in Figure 3.1.

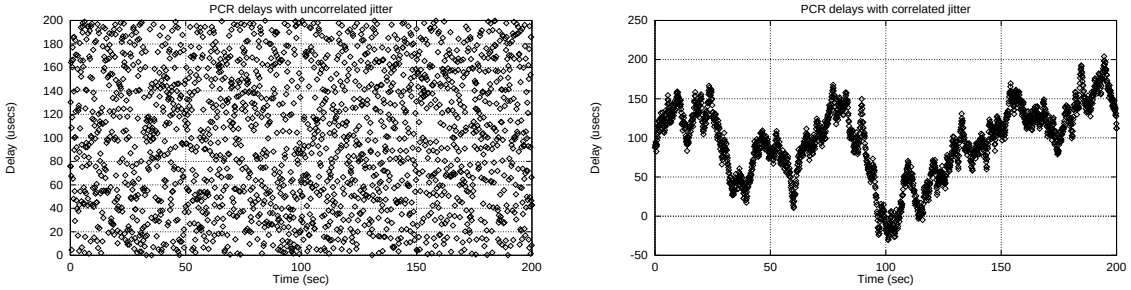
We use the model described above to check the effect of correlated jitter in the clock recovery process. A common way to generate a discrete time-correlated jitter process is through a simple autoregressive process (AR(1) process). In such a case, the jitter process can be generated using the following equation [2]:

$$J_c(nT) = \rho J_c[(n-1)T] + (1-\rho)J(nT),$$

where $J(nT)$ is an independent random variable, and $J_c(nT)$ is the correlated variable used in the experiments. The magnitude of correlation is determined by the correlation coefficient ρ. If ρ is 0, the produced jitter is not correlated whereas, if it is close to 1, high correlation can be obtained.

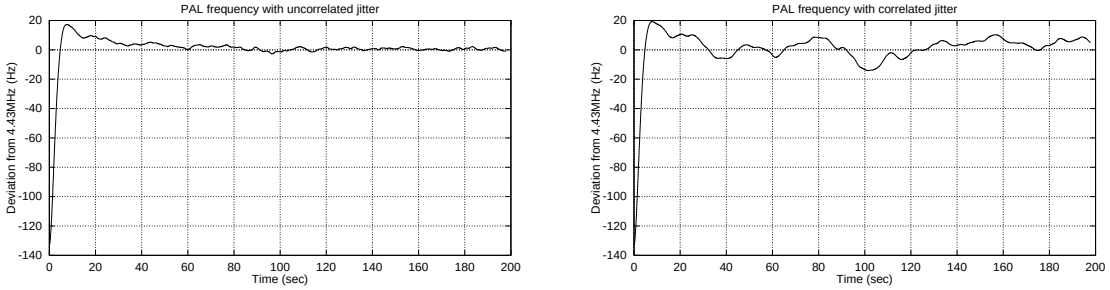
For the first experiment, an independent uniform variable is used for $J(nT)$ that generates a 200 μsec peak-to-peak jitter. We considered two cases: one with no correlation ($\rho = 0$), and one with $\rho = 0.9$ to generate correlated jitter. Since, in the latter case, the produced jitter has peak-to-peak amplitude lower than 200 μsec, we scale the correlated jitter variable to have the same amplitude as in the case of the independent uniform variable.

The jitter generated in the uncorrelated case is given in Figure 3.2(a), and the corresponding jitter for the correlated case is given in Figure 3.2(b). The frequency recovered with uncorrelated jitter is much more stable than with correlated jitter (Figures 3.3(a) and (b)). The frequency is within the PAL specifications (see Table 3.1) in Figure 3.3(a), while it violates the specifications with correlated jitter in Figure 3.3(b). This is in agreement with the observation of Andreotti et al. [2].



(a) (b)

Figure 3.2: Delays with 200 μsecs peak-to-peak uniform i.i.d. jitter. (a) Uncorrelated case (ρ=0), (b) correlated case (ρ=0.9).



(a) (b)

Figure 3.3: Recovered PAL frequency with 200 μsecs peak-to-peak uniform i.i.d. jitter. (a) Uncorrelated case (ρ=0), (b) correlated case (ρ=0.9).

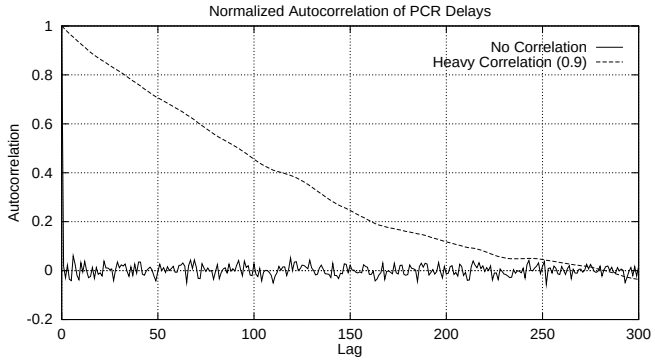


Figure 3.4: Normalized autocorrelation of PCR delays.

We can further check the validity of our results for non-monotonic autocorrelation functions. We first define *lag* as the distance of a PCR packet from another PCR packet in terms of the number of PCR packets in between. We use a higher-order AR model to generate traffic in a similar manner. The jitter in AR model used is given by:

$$J_c(nT) = a_1 J_c[(n-2)T] + a_2 J_c[(n-3)T](1 - \sum_{i=1}^{2} a_i) J(nT). \tag{3.1}$$

The selection of parameters a_1 and a_2 of the AR model is done in such a way as to (1) create non-monotonic autocorrelation functions for the jitter process, and (2) create autocorrelation functions

Standard	NTSC	PAL (M)	PAL (B, D, G, H, N)
Frequency (Hz)	35795454	3575611.49	4433618.75
Tolerance (Hz)	±10	±10	±5
Tolerance (ppm)	±3	±3	±1

Table 3.1: Specifications for the color sub-carrier of different video formats (from [2]).

Experiment	a_1	a_2
1	0.5	0.01
2	0.8	0.02
3	0.9	0.05
4	0.93	0.06

Table 3.2: Parameters used in the experiments for the AR model given in Eq. (3.1).

that have smaller values for every lag, for smaller values of the parameters a_1 and a_2. Formally, let us consider two correlated jitter processes J_{c1} and J_{c2} that are generated by:

$$J_{c1}(nT) = a_1 J_{c1}[(n-2)T] + a_2 J_{c1}[(n-3)T](1 - \sum_{i=1}^{2} a_i)J(nT),$$

$$\text{and} \quad J_{c2}(nT) = b_1 J_{c2}[(n-2)T] + b_2 J_{c2}[(n-3)T](1 - \sum_{i=1}^{2} b_i)J(nT).$$

Then, according to the second requirement, the following inequality holds for the resulting autocorrelation functions $R_{J_{c1}}(\tau)$, $R_{J_{c2}}(\tau)$.

$$R_{J_{c1}}(\tau) \leq R_{J_{c2}}(\tau), \quad \text{for any integer } \tau, a_1 \leq b_1, a_2 \leq b_2.$$

While we have experimented with a series of values for a_1 and a_2, we only present some representative cases here which magnify the problem with correlation. The values for parameters in these experiments are shown in Table 3.2.

From the autocorrelation function plots illustrated in Figure 3.5, is evident that higher values of parameters a_1 and a_2 produce higher autocorrelation of the resulting jitter process for every lag. The quality of the reconstructed clock at the receiver end becomes better as the jitter process becomes less correlated. As shown in Figures 3.6(a) and (b), the recovered clock violates the PAL clock specifications in both cases and is worse in the case when $a_1 = 0.93$ and $a_2 = 0.06$. Although in the remaining cases (Figures 3.7(a) and (b)) the clock is within the PAL specifications for the frequency range, the quality is slightly worse in the case of $a_1 = 0.8$ and $a_2 = 0.02$, as recorded by its variance.

As shown above in this series of experiments, time-correlation of the jitter process degrades the quality and the stability of the reconstructed clock even when very low jitter is present (200 μsecs).

3.2 Experiments with Bimodal Distributions

There are cases in which the delays of the transport packets follow a bimodal distribution, as in the case of the PCR-unaware encapsulation scheme [19] in ATM networks. In that case, the encoder may perform the timestamping in such a way that results in a delay process of the transport packets

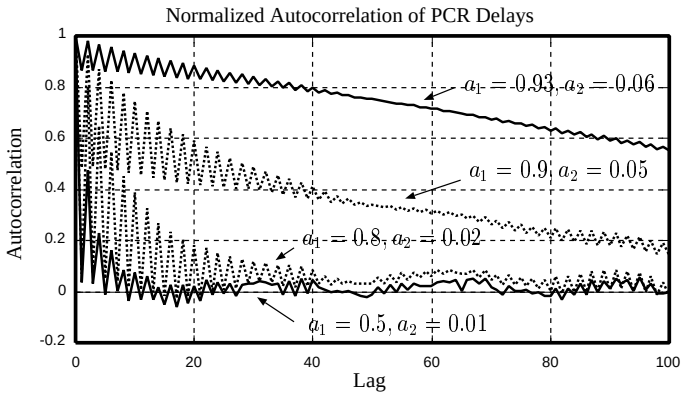
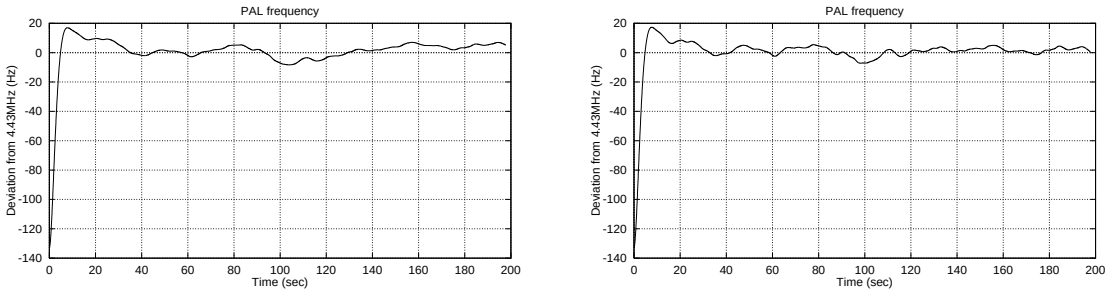
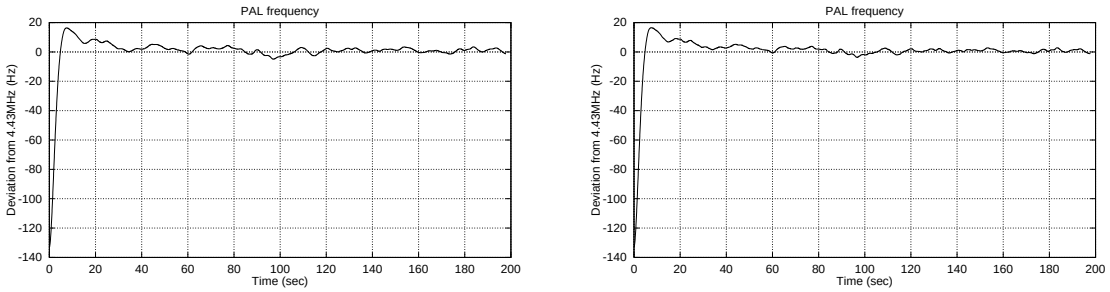


Figure 3.5: Normalized autocorrelation functions of PCR delays observed in the experiments with the AR process of Eq. (3.1).



(a) (b)

Figure 3.6: Recovered PAL frequency with 200 μsecs peak-to-peak jitter. (a) $a_1 = 0.93, a_2 = 0.06$, (b) $a_1 = 0.9, a_2 = 0.05$.

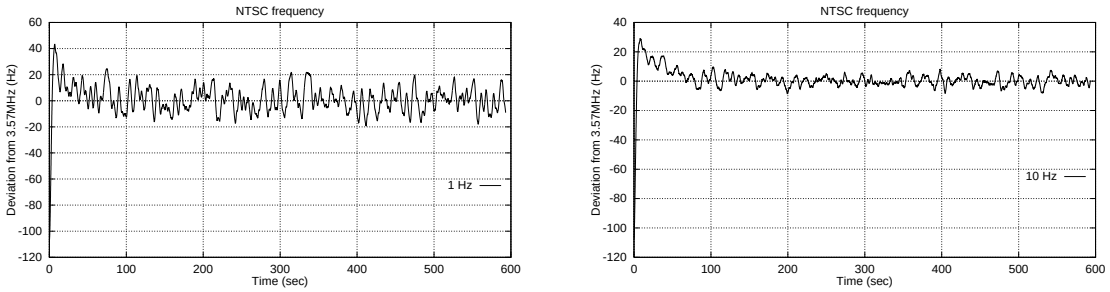


(a) (b)

Figure 3.7: Recovered PAL frequency with 200 μsecs peak-to-peak jitter. (a) $a_1 = 0.8$, $a_2 = 0.02$, (b) $a_1 = 0.5$, $a_2 = 0.01$.

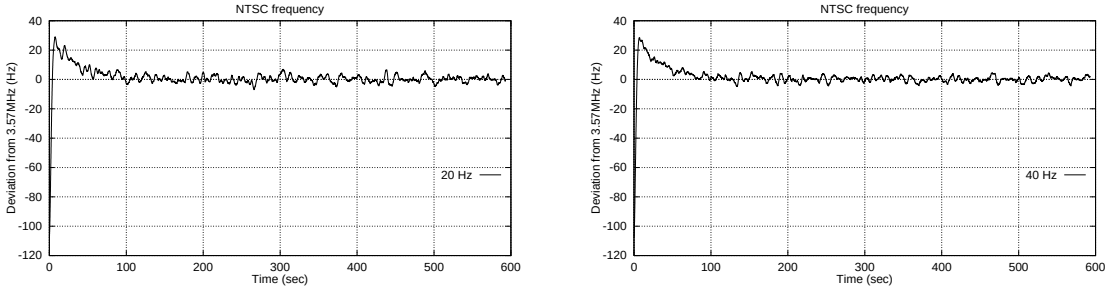
containing PCR values at the decoder with significant correlation. This, in turn, may degrade the quality of the recovered clock to unacceptable levels.

In order to check the impact of correlation on the clock recovery process in the case of bimodal delay distributions, we conducted experiments with synthetic traces. The generated synthetic traces had their PCR values switching polarity at exponentially distributed time instants. More specifically, the time intervals at which a sequence of odd-indexed PCR packets was followed by even-indexed



(a) (b)

Figure 3.8: Recovered NTSC color sub-carrier generation frequency for a random telegraph process with peak-to-peak jitter of 376 μsecs and rate of (a) 1 Hz, and (b) 10 Hz.



(a) (b)

Figure 3.9: Recovered NTSC color sub-carrier generation frequency for a random telegraph process with peak-to-peak jitter of 376 μsecs and rate of (a) 20 Hz, and (b) 40 Hz.

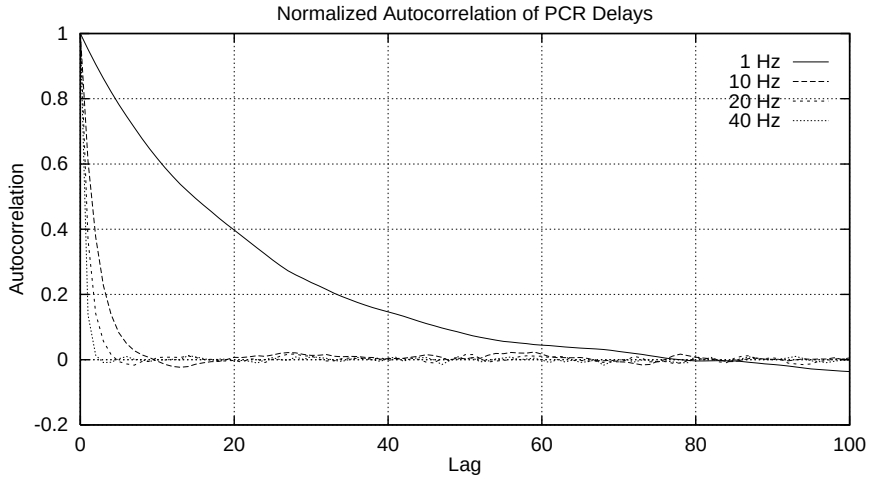


Figure 3.10: Normalized autocorrelation of PCR delays for random telegraph processes with various frequencies.

PCR packets or vice-versa were drawn from an exponential distribution. The resulting delay process follows the structure of a random telegraph process [10]. Also, the associated random telegraph process was allowed to have varying rates in order to tune its autocorrelation. The autocorrelation of the scaled random telegraph process used is derived in [19], and is given by

$$R(t_1, t_2) = \frac{f_o^2}{4r^2} e^{-2a|t_2 - t_1|}, \quad \text{for any pair of real numbers } t_1 \text{ and } t_2,$$

where f_o is MPEG-2 system frequency for timestamping purposes, r is the transport rate in pkts/sec, and a is the rate of the telegraph process. In our experiments, we started from a rate of 1 per second, and increased it to 10, 20 and 40. The transport rate of the synthetic traces was fixed at 4 Mbps so that the peak-to-peak jitter due to packetization of the PCR-unaware scheme becomes 376 μs.

The results for the recovered NTSC sub-carrier frequency from these experiments are shown in Figures 3.8 and 3.9. The worst performance is obtained when the rate of the telegraph process is 1 Hz, whereas the best is in the case of the process with the highest-frequency (40 Hz). The autocorrelation of the process is illustrated in Figure 3.10 in which we observe that high correlation is present in the 1 Hz random telegraph process even at lags more than 20 PCR packets which is not true in the other cases (10, 20 and 40 Hz).

In summary, the results from these experiments enable us to conclude that even in bimodal distributions, high correlation may result in worse performance in the quality of the reconstructed clock for the same peak-to-peak jitter.

3.3 Experiment with Delay Correlation due to Scheduling Disciplines

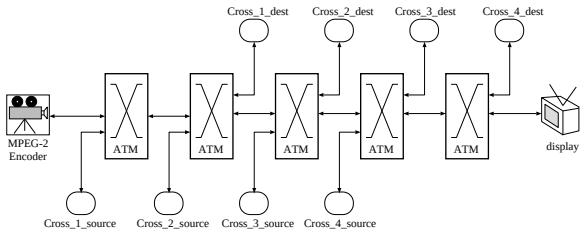


Figure 3.11: Network topology used in the experiment.

Although jitter correlation degrades the quality of the reconstructed clock in theory, it is not clear whether such correlation may occur in practice when a scheduling discipline, able to guarantee QoS, is deployed in a packet-switched network. As shown in [18], the use of a fair-queueing scheduling algorithm is essential in order to be able to provide QoS guarantees under various load conditions.

The goal of this experiment is to check whether the use of any fair-queueing algorithm can guarantee good quality of the receiver clock in applications similar to TV distribution, when the receiver does not make use of special methods for jitter compensation. Several fair-queueing algorithms have been proposed in the literature [21]. Their design often entails several trade-offs among complexity, fairness and worst-case delay bounds. In this experiment, we compare two representative fair-queueing algorithms that have different fairness properties and worst-case delay bounds: *Frame-based Fair Queueing (FFQ)* [17] and *Self-Clocked Fair Queueing (SCFQ)* [6].

A heavily loaded ATM network is used for the purpose of this experiment with cross-traffic from a large number of on-off sources with exponential on and off periods. The network topology used is shown in Figure 3.11. It consists of five cascaded ATM switches. The switch nodes are non-blocking, output-buffered crossbar switches. An actual MPEG-2 Transport Stream that contains video content in PAL format is sent through all the cascaded switches to the display device at the other end using the PCR-unaware packing scheme for the encapsulation at the adaptation layer. A set of cross traffic sources is generated at each hop of the network to stress the various scheduling disciplines

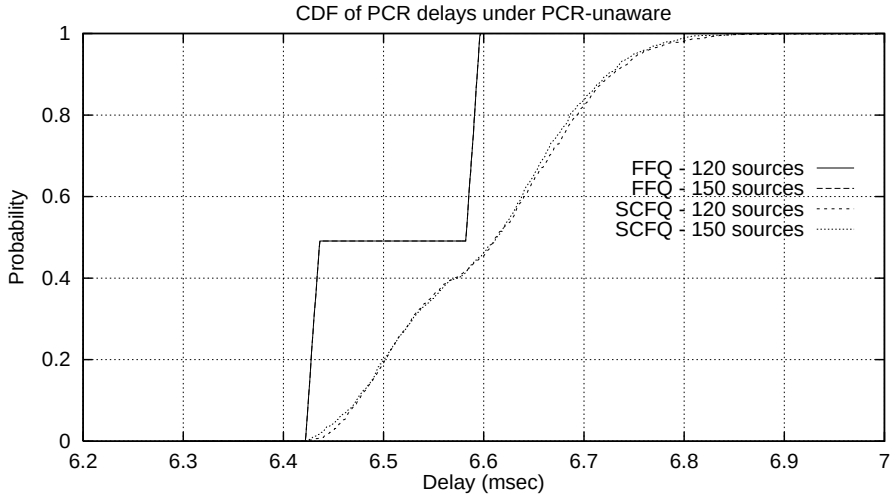


Figure 3.12: Cumulative distribution function (CDF) of PCR delays under both FFQ and SCFQ scheduling disciplines and PCR-unaware packing scheme.

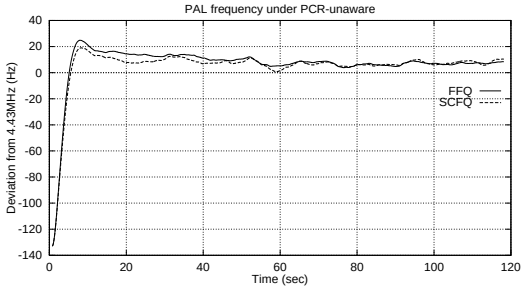
at each network switch and test their effects on the quality of the end-to-end video stream. All the cross-connections are between nodes that are connected to adjacent ATM switches. As before, we set the propagation delay for each network link to 1 msec, and the frame-size parameter in the FFQ algorithm to 3 ms.

Since, SCFQ has a worst-case delay bound that is of the order of the number of the multiplexed connections [6], we multiplex a large number of cross-traffic sources to check the effect on the PCR delay jitter and its autocorrelation. The number of cross-traffic sources starting at the second ATM switch is varied from 120 to 150 depending on the experiment. The speed of each link in the network is set to 30 Mbps.

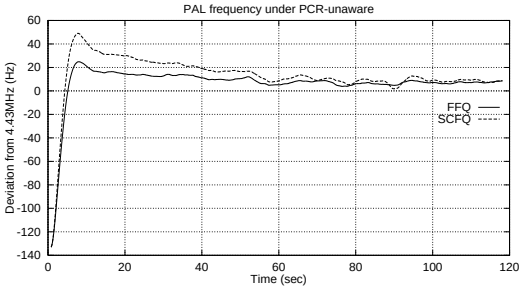
From the cumulative distribution functions (CDF) of the PCR delays under SCFQ (Figure 3.12), we observe that the delays with 120 sources are larger than the ones with 150 sources. This can be explained since in the former case, each of the 120 sources has higher rate than in the case of 150 sources, resulting in higher burstiness. Thus, the 120 cross-traffic sources tend to get synchronized more often than the case of 150 sources yielding higher delays. However, the CDFs are very close. The maximum jitter observed is 0.43 msecs. In the FFQ case, the CDFs of the PCR delays under both experiments (120 and 150 cross-traffic sources) are identical as shown in Figure 3.12. It is evident that there is a clear distinction between transport packets that suffer from packetization delay and the rest that do not. However, under the SCFQ scheduling discipline, the delays of the PCR packets are spread almost uniformly between their minimum and maximum values (also shown in Figure 3.12).

When testing the quality of the PAL frequency generated, we observe that SCFQ gives acceptable quality for 120 sources (Figures 3.13(a) and 3.14(a)). However, in the case of 150 sources, both the PAL frequency and the drift of the PAL frequency could not meet their specifications (Figures 3.13(b) and 3.14(b)). As can be seen in Figure 3.14(b), there is a clock drift of 0.22 Hz/sec at time 111 seconds. On the other hand, FFQ provides almost perfect isolation of the MPEG-2 flow that results in very good quality of the recovered clock in all the cases (Figures 3.13 and 3.14).

The inferior behavior of SCFQ with 150 sources suggests that the temporal characteristics of the jitter process are different in the case of 150 sources than that of 120. In fact, even though the

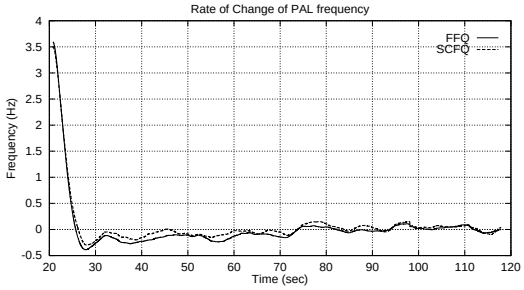


(a)

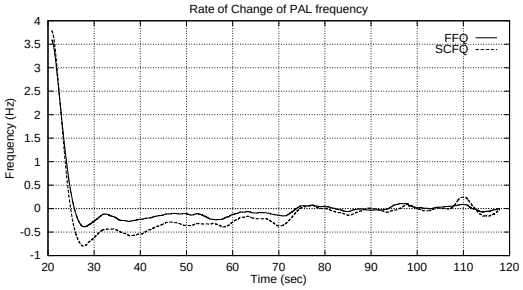


(b)

Figure 3.13: PAL color sub-carrier generation frequency under PCR-unaware scheme using (a) 120 sources, and (b) 150 sources.



(a)



(b)

Figure 3.14: Rate of change of PAL color sub-carrier generation frequency (averaged over 40 secs) under PCR-unaware scheme using (a) 120 sources, and (b) 150 sources.

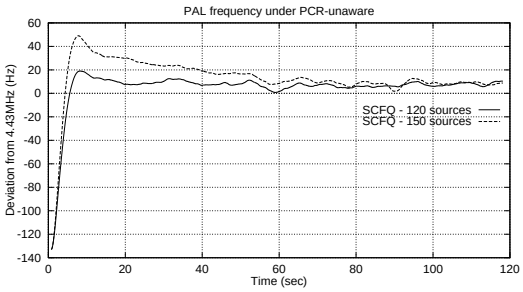


Figure 3.15: PAL color sub-carrier generation frequency under SCFQ using PCR-unaware scheme with both 120 and 150 cross-traffic sources.

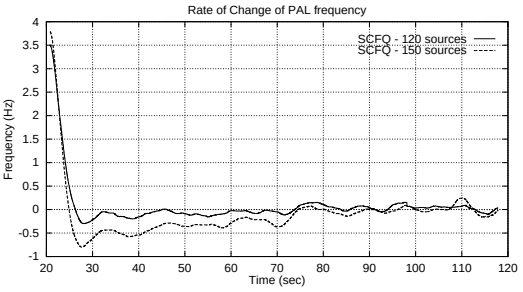


Figure 3.16: Rate of change of PAL color sub-carrier generation frequency under SCFQ using PCR-unaware scheme with both 120 and 150 cross-traffic sources (averaged over 40 secs).

CDFs for the delays are approximately the same the correlation of the jitter is not, with the case of 150 sources generating slightly more correlated delays, which, in turn, degrades the quality of the recovered clock compared to the case of 120 sources (Figure 3.17).

The above results suggest that the quality of the recovered clock is affected not only by the magnitude of the delay variation, but also by the correlation between delay samples.

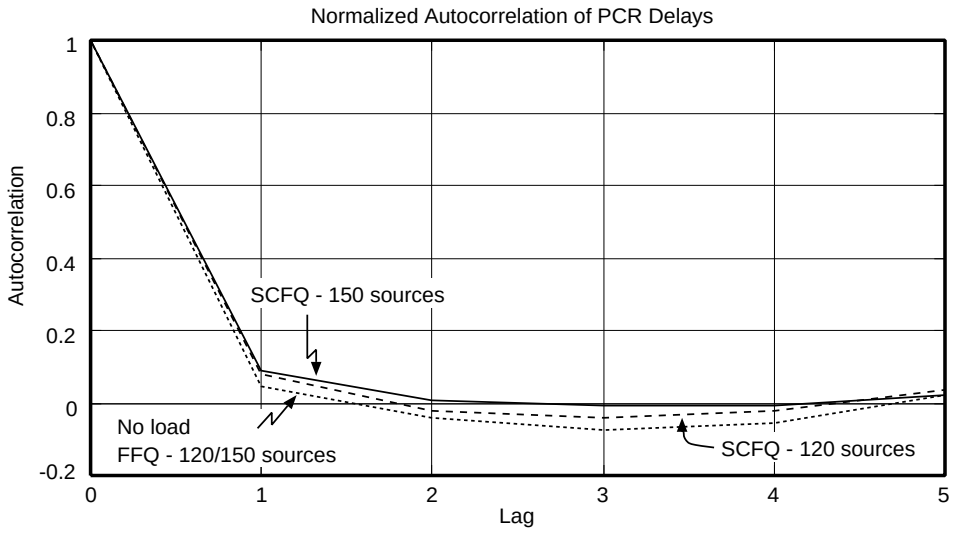


Figure 3.17: Normalized autocorrelation of PCR delays for SCFQ and FFQ scheduling disciplines under PCR-unaware scheme.

4 Conclusions

In this paper, we addressed the effect of the correlation of the packet delays on the clock recovery process in an MPEG-2 decoder. We first presented an analytical model for a standard PLL used in an MPEG-2 Systems decoder. We used a correlated general input process with a variable correlation coefficient as an input for the PLL. We showed analytically that high correlation produces a large variance of the recovered clock compared to the case with low correlation of the input traffic pattern.

We verified that high correlation of the input traffic process yields to poor quality of the recovered clock through an actual simulation of an MPEG-2 Systems decoder PLL with theoretical correlated traffic patterns (AR processes) as the input. We also showed how high correlation can result in poor quality even in bimodal delay distributions as in the case of PCR-unaware scheme, through simulation experiments. The experiments were performed with synthetic traces in which the PCR polarities were determined by a random telegraph process with a tunable correlation coefficient. We demonstrated the differences in the autocorrelation of packet delays introduced by two different packet scheduling algorithms and the resulting variation in receiver clock quality.

5 Acknowledgments

The authors would like to thank Prof. Yiannis Kontoyiannis at Purdue University for his valuable input on the analysis.

References

[1] I. F. Akyildiz, S. Hrastr, H. Uzunalioglu, and W. Yen. Comparison and evaluation of packing schemes for MPEG-2 over ATM using AAL5. In *Proceedings of ICC '96*, volume 3, pages 1411–1415, June 1996.

[2] G. F. Andreotti, G. Michieletto, L. Mori, and A. Profumo. Clock recovery and reconstruction of PAL pictures for MPEG coded streams transported over ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):508–514, December 1995.

[3] M. De Prycker. *Asynchronous Transfer Mode : Solution for Broadband ISDN.* Ellis Horwood, second edition, 1993.

[4] S. Dixit and P. Skelly. MPEG-2 over ATM for video dial tone networks: issues and strategies. *IEEE Network*, 9(5):30–40, September–October 1995.

[5] D. Fibush. *Subcarrier Frequency, Drift and Jitter in NTSC Systems.* ATM Forum, July 1994. ATM94-0722.

[6] S. J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *Proceedings of IEEE INFOCOM '94*, volume 2, pages 643–646, June 1994.

[7] International Organization for Standardization. *Information Technology — Generic Coding of Moving Pictures and Associated Audio: Systems, Recommendation H.222.0, ISO/IEC 13818-1*, draft international standard edition, November 1994.

[8] International Organization for Standardization. *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video, Recommendation ITU-T H.262, ISO/IEC 13818-2*, draft international standard edition, November 1994.

[9] Y. Kaiser. Synchronization and dejittering of a TV decoder in ATM networks. In *Proceedings of Packet Video Workshop '93*, volume 1, 1993.

[10] A. Leon Garcia. *Probability and Random Processes for Electrical Engineering.* Addison-Wesley Publishing Company, second edition, May 1994.

[11] H. Meyr and G. Ascheid. *Synchronization in Digital Communications*, volume 1 of *Wiley Series in Telecommunications.* John Wiley & Sons, 1990.

[12] M. Nilsson. MPEG-2 over ATM. In *IEE Colloqium on "MPEG-2 – what it is and what it isn't"*, number 1995/012. The Institute of Electrical Engineers, January 1995.

[13] M. Perkins and P. Skelly. *A Hardware MPEG Clock Recovery Experiment in the Presence of ATM Jitter.* ATM Forum, May 1994. ATM94-0434.

[14] J. Proakis and D. G. Manolakis. *Introduction to Digital Signal Processing.* Macmillan, 1988.

[15] P. A. Sarginson. MPEG-2 – a tutorial introduction to the systems layer. In *IEE Colloqium on "MPEG-2 – what it is and what it isn't"*, number 1995/012. The Institute of Electrical Engineers, January 1995.

[16] M. Schwartz and D. Beaumont. *Quality of Service Requirements for Audio-Visual Multimedia Services.* ATM Forum, July 1994. ATM94-0640.

[17] D. Stiliadis and A. Varma. Efficient fair-queueing algorithms for packet-switched networks. *IEEE/ACM Transactions on Networking*, April 1998.

[18] C. Tryfonas. MPEG-2 transport over ATM networks. Master's thesis, University of California at Santa Cruz, September 1996. Available at `http://www.cse.ucsc.edu/research/hsnlab`.

[19] C. Tryfonas and A. Varma. Timestamping schemes for MPEG-2 systems layer and their effect on receiver clock recovery. Technical Report UCSC-CRL-98-2, University of California at Santa Cruz, Dept. of Computer Engineering, May 1998.

[20] C. Tryfonas and A. Varma. A restamping approach to clock recovery in MPEG-2 systems layer. to appear in Proceedings of ICC '99, June 1999.

[21] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. In *Proceedings of the IEEE*, pages 1374–1396, October 1995.