An Analytical Design of Optimal Playout Schedulers for Packet Video Receivers*

Nikolaos Laoutaris and Ioannis Stavrakakis
{laoutaris,istavrak}@di.uoa.gr

Department of Informatics and Telecommunications,
University of Athens, 15784 Athens, Greece

Abstract

One way to reduce, or avoid, the loss of intrastream synchronization due to the delay variability introduced by best-effort networks, is by employing application layer buffering and scheduling at a Packet Video Receiver (PVR), resulting in a higher end-to-end delay. In this paper an analytical model is presented that captures the essential tradeoff between stream continuity and stream latency. Unlike past related work, stream continuity is not expressed as the average amount of synchronization loss, but as a combination of the average amount, and the variability of the duration of synchronization loss occurrences. This approach allows for a fine grained optimization of stream continuity which has the potential of providing an improved perceptual quality. It is shown that the minimization of the average amount of synchronization loss, and the minimization of the variability of the duration of synchronization loss occurrences, are two competing objectives; the minimization of variability is desirable because it can lead to the concealment of discontinuities. The aforementioned presentation quality metrics are considered by the optimal playout policy, which is derived by means of Markov decision theory.

1 Introduction

In recent years multimedia services such as internet telephony, desktop videoconference, and video on demand (VOD), have found a place next to traditional data applications like telnet, ftp or the world wide web. These new services require high transmission reliability and stringent end-to-end delay and delay jitter, to be able to maintain intrastream synchronization between successive media units. The networking community is currently focused on developing mechanisms that will enhance the Quality of Service capabilities of best-effort network [1, 2]. The IETF has standardized the Integrated Services and Differentiated Services architectures, as means of providing current best-effort networks with QoS mechanisms. Nevertheless, it is realized that the deployment of new protocols will be a slow process, so much effort is being put in coping with current network limitations by incorporating intelligent adaptive algorithms at the application layer.

*This work and its dissemination efforts have been supported in part by the IST Program of the European Union under contract IST-1999-10160 (VideoGateway).

Adaptive rate applications fall into two general categories depending on which end of the communicating parties is adapting its rate or buffering capacity. In source rate adaptation [3, 4], it is the sending system that adapts to the time-varying bandwidth availability by regulating the rate of its output video stream. On the other hand, packet video receivers (PVRs), may buffer some frames in a playout buffer or even make small adjustments in their playout rate in an effort to conceal the effects of jitter (lack of a frame to display caused by excessively delayed frames).

All PVR's buffer incoming frames as a measure against network jitter. Buffering frames in the playout buffer increases the end-to-end latency at the end-user level. Consequently, the intrastream synchronization improvement that is gained with the addition of the playout buffer is limited by the available end-to-end delay budget.

Different applications tolerate different maximum end-to-end latencies. Bidirectional applications such as desktop video conferencing place very strict latency requirements, typically of few hundreds of milliseconds. On the other hand, unidirectional applications, for example video on demand (VOD), allow for much larger latencies in the order of seconds. In a VOD application, the PVR can buffer massive numbers of frames, thus ensure an almost pauseless video presentation across the widest range of network jitter. The absence of critical latency requirements also allows for the incorporation of techniques such as data proxy-ing and client-server feedback which can help in using network resources more efficiently, especially in the case of Variable Bit Rate (VBR) encoded video [5, 6, 7].

The hard real time requirements of interactive applications [8] are met by the *absolute delay method* which delivers frames at a constant end-to-end latency and drops late frames. Applications with looser delay constraints (soft real time) may present a late frame and thus gain in stream continuity by not discarding a frame which has already harmed the continuity of the stream by causing an underflow with its late arrival. Keeping the late frame, increases the end-to-end latency of all subsequent frames resulting in a playout policy with variable overall latency for different frames. Of course, even soft real time applications have an upper bound on latency, or a limited buffer capacity, so eventually the accumulation of frames in the buffer will lead to overflows.

The family of playout schedulers that we study in this work guarantees neither a constant end-to-end latency nor a constant buffering delay. The scheduler guarantees only a statistically constant (mean value, max value) buffering delay. The gain from the relaxation of the constant latency requirement, is the ability to react better to bursty frame-arrival sequences due to network jitter. Perceptual quality can be improved by implementing *mild* latency control methods, which harm stream continuity less than the *harsh* deadline discard of late frames under the absolute delay method. At the same time, the buffering delay is not ignored – in favour of stream continuity – but is kept statistically constant, below an acceptable, user-defined level. Two such methods are proposed in Sect. 4.3. The scheduler increases the buffer occupancy as a measure against jitter, and decreases the buffer occupancy, in order to control the buffering delay and avoid overflows. The playout buffer occupancy is controlled by regulating the playout duration of frames on a per frame basis. This necessity is not present in packet audio systems, where the existence of silence periods gives the system the ability to change the size of the de-jitter buffer by modifying the duration of the silence periods, on a per talkspurt basis, without modifying the duration of media units [9, 10, 11, 12, 13, 14].

Analytical studies for PVR's that employ dynamic playout schedulers, have appeared in the literature. Yuang et al. [15] proposed a dynamic playout policy based on *slowdown* thresholds. In their work, frames are presented at a gradually decreasing rate when the playout buffer occupancy drops below TH – the slowdown threshold – and at a constant rate μ when the occupancy exceeds TH. In [16] we extended the threshold-based scheduler of [15]. The study limited the range of the threshold parameter by identifying a range of values where there is no beneficial tradeoff between continuity and reduction of mean playout rate – the two antagonistic metrics of interest. Additionally, it was shown that this range of values changes with the burstiness of the frame arrival process, revealing the danger of an initially meaningful TH appearing in the undesirable area due to the change of arrival burstiness. The work was supplemented with online algorithms for the detection and the maintenance of TH within the area of beneficial tradeoff across unknown, non-stationary, delay jitter.

With the current work we improve previous heuristic frame-playout policies, such as those in [15, 16], by formulating an optimization problem that involves the two main metrics of interest; the intrastream synchronization and the buffering delay. Furthermore, the intrastream synchronization metric used in this work is more fair and more general than previously used synchronization metrics [15, 16, 17, 18]. For the control of the buffering delay two schemes are proposed: one that favours the reduction of the maximum buffering delay and one that provides for a better stream continuity.

The remainder of the paper is organized as follows. Some key concepts and definitions are presented is Section 2. A number of optimal playout adaptation policies for packet video receivers are derived is Section 3 by introducing some interesting metrics and employing Markov decision theory and a policy iteration algorithm. Numerical results are presented in Section 4 together with a comparative analysis with a non-optimal scheduler from the literature. Also in the same section two schemes for the control of the buffering delay are presented. Section 5 comments on implementation issues and future work. Section 6 concludes the paper.

2 Definitions

In the following sections we construct a mathematical model for the derivation of the optimal frame-playout policy. Among others, the need for a method to control the duration of frames will arise. Under R_T – the usual static playout policy – all frames are presented with an equal duration T, as determined by the frame production rate λ (that is, $T = 1/\lambda$); λ is equal to 25 or 30 frames/second for PAL and NTSC video. In some cases, we will need to expand the presentation duration of a frame beyond T, the normal duration. When this decision is repeated for successive frames, it leads to a temporary reduction of the playout rate (we will refer to this as a *slowdown*). On the opposite, the presentation of a sequence of frames with durations that are shorter than T, leads to a temporary increase of the playout rate (much like a *fast forward* operation on a VCR). The most general way to regulate B, the duration of a frame, is to allow it to take all non-negative values. This approach gives the utmost freedom in the search for an optimal playout policy. In this paper, however, we limit the possible values for B to a countable set of values as specified by: $B(k) \triangleq k \cdot c$, where c is an fraction of the normal frame duration T, such that $T = a \cdot c$; the value of a is the *cutting factor* of T and c is the basic unit for shortening and expanding the duration of a frame. Using the last relationship $B(k)$ becomes:

$$B(k) = \frac{k}{a} \cdot T \tag{1}$$

The reduction of freedom in the search for optimal policies is negligible for small values of c, e.g., $c = T/10$. Since k determines completely the length of a frame, $B(k)$, the selection of a specific value of k will be referred to as an *action* or a *decision*.

Expanding or shortening the duration of a frame presentation introduces a discontinuity – a loss of intrastream synchronization – quantified by the difference between the selected frame duration and the normal frame duration T. Let $d_i(k)$ denote the *discontinuity* that is incurred when the next frame is presented with a duration $B(k)$ and the current buffer occupancy is i.

$$d_i(k) \triangleq |B(k) - T| + S \cdot I_{\{i=0\}} \tag{2}$$

S is a random variable that adds to $d_i(k)$ the effect of buffer underflows. S represents the time interval between a buffer underflow instant and the next arrival instant. I is the indicator function. We define a more generalized continuity metric – called the Distortion of Playout metric (abbreviated DoP) – which includes the notion of discontinuity as in (2) but also accounts for any lack of continuity due to buffer overflows over the current presentation interval.

$$DoP_i(k, X_k) \triangleq d_i(k) + l_i(X_k) \cdot T \tag{3}$$

X_k is a random variable for the number of new arrivals over the current presentation duration. The dependence of X on the duration of the presentation interval (this duration is determined explicitly by k) is indicated by the subscript k. $l_i(X_k)$ is the number of new arrivals that are lost due to overflow from the tail of the buffer; given a current buffer occupancy i, it depends solely on the number of new arrivals X_k.

$$l_i(X_k) \triangleq \begin{cases} 0 & , X_k < N - i + 1 \\ X_k - N + i - 1 & , X_k \geq N - i + 1 \end{cases} \tag{4}$$

Note that the DoP metric measures time; it adds the duration of all time intervals during which the smooth playout of frames is disrupted. A basic idea reflected in the definitions of both $d_i(k)$ and $DoP_i(k, X_k)$ is that the perceptual cost of an idle time gap between two frames (occurring when the first frame stays on display for more than T) is equal to the perceptual cost of a loss-of-information discontinuity (an overflow or a fast-forward) of equal duration. This is based on recent perceptual studies [19] where it is shown that jitter degrades the perceptual quality of video nearly as much as packet loss does [1].

3 Design of Optimal Playout Schedulers

In this section a PVR consisting of a playout buffer and a playout scheduler is studied by means of Markov decision theory. The goal is to derive the optimal playout policy which, by controlling the

[1] For an example in support of this claim, we may think that an underflow with a duration T, degrades stream continuity, nearly as much, as does a lost frame.

duration of frames based on the current buffer occupancy, minimizes an appropriate metric that is closely related to the perceptual quality of the presentation.

3.1 MDP Problem Formulation

Let $\{I_n\}_{n>0}$ be a stochastic process describing the number of frames in the playout buffer upon the presentation completion instant[2] of the nth frame. $\{I_n\}$ is a Markov process under the assumed Poisson process for frame arrivals. The Poisson process and the associated interarrival-time exponential distribution are much more variable than actual frame arrival processes and thus lead to rather pessimistic performance results. Nevertheless, the memoryless property of the exponential distribution simplifies the model and is thus suitable for a first exposition of the basic ideas that appear in this paper (the implications of the Poisson assumption are further examined in Sect. 5).

To formulate $\{I_n\}$ as a Markov decision process (MDP), we need to define a tuple $\langle \mathcal{S}, \mathcal{A}, P, C \rangle$, where $\mathcal{S}$ is the set of possible states, $\mathcal{A}$ is the set of possible actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the state transition function specifying the probability $P\{j|i,k\} \equiv p_{ij}(k)$ of observing a transition to state $j \in \mathcal{S}$ after taking action $k \in \mathcal{A}$ in state $i \in \mathcal{S}$ and, finally, $C : \mathcal{S} \times \mathcal{A} \rightarrow \Re$ is a function specifying the cost $c_i(k)$ of taking action $k \in \mathcal{A}$ at state $i \in \mathcal{S}$.

The state space $\mathcal{S}$ of $\{I_n\}$ comprises all possible buffer occupancy levels thus takes values in $[0 \dots N]$, N being the buffer size. An *action* is defined to be the choice of an integer value k that explicitly determines $B(k)$, the presentation duration for the next frame (see (1)). The action space for the problem is $\mathcal{A} = [0 \dots K]$, where K is an integer value that results in the maximum allowable playout duration $B(K)$. For $k = 0$ the playout scheduler discards the next frame.

The action taken at an observation instant affects the evolution of $\{I_n\}$ by affecting the probability law for the next transition. Let P be a $(N+1) \times (N+1)$ matrix containing $(K+1)$-dimensional vector elements $\vec{p}_{ij}$. The $(k+1)$th element of vector $\vec{p}_{ij}$ is the probability of observing a transition to state j given that the previous state is i and decision $k, k \in \mathcal{A}$, is taken. Let $P\{x,t\}$ denote the probability of observing x new frame arrivals in a time interval of duration t. Since $\{I_n\}$ corresponds to the buffer occupancy, the probability of observing a transition to a state j after playing the next frame for time $B(k)$ (selecting action k), depends on the number of new frames that will arrive during $B(k)$.

$$p_{ij}(k) = \begin{cases} P\{j, B(k)\} & i = 0, 0 \leq j < N \\ 1 - \sum_{m=0}^{N-1} P\{m, B(k)\} & i = 0, j = N \\ P\{j-i+1, B(k)\} & 0 < i < N, i \leq j < N \\ 1 - \sum_{m=0}^{N-i} P\{m, B(k)\} & 0 < i \leq N, j = N \\ P\{0, B(k)\} & 0 < i \leq N, j = i-1 \\ 0 & \text{elsewhere} \end{cases} \tag{5}$$

In this paper $P\{x,t\}$ is determined by the Poisson distribution with parameter λ, the frame production rate. Using the transition matrix of (5), we can obtain $\pi(R)$, the limiting distribution of $\{I_n\}$ for a particular policy R (defined shortly below), by solving the stationary equations $\pi(R) = \pi(R) \cdot P(R)$.

The ultimate goal of this decision model is to prescribe a playout *policy* – a rule for choosing the duration of the next frame based on the observation of the current buffer occupancy. In the

[2] Hereafter called an *observation*, or a *decision* instant.

general case a policy $R \equiv \{D_{ik} : i \in \mathcal{S}, k \in \mathcal{A}\}$ is a mapping: $\mathcal{S} \times \mathcal{A} \to [0,1]$; it is completely defined for a given tuple $\langle \mathcal{S}, \mathcal{A}, P, C \rangle$ by the probabilities: $D_{ik} \triangleq P\{\text{action} = k | \text{state} = i\}$. The derived optimal policy, under the considered minimization objective, with finite action and state spaces and with the selected solution method (described in Appendix B), always prescribes the same action whenever at the same state, i.e., the optimal policy is non-randomized (see [20] for details). Under a non-randomized policy the probabilities D_{ik} are either 0 or 1. In view of this observation, the exact definition of the non-randomized policy R reduces to the definition of the function $D_i(R) = k$ which for every $i \in \mathcal{S}$ returns the selected action k.

As mentioned earlier, $c_i(k)$ denotes the cost incurred when action k is taken when the occupancy process is in state i. The optimal policy R_{opt} is defined to be the policy that minimizes $E_R\{c\}$, where $E_R\{c\}$ denotes the long-run average cost induced under some policy R. If $\pi_i(R)$ denotes the limiting probability that process $\{I_n\}$ is in state i under policy R, then

$$R_{opt} = \arg\min_R E_R\{c\} \quad \text{where} \quad E_R\{c\} = \sum_{i=0}^{N} c_i(D_i(R)) \cdot \pi_i(R) \tag{6}$$

This objective function corresponds to an infinite-horizon, undiscounted-cost planning problem [20]. A number of techniques are known for the derivation of the optimal policy of (6); these include exhaustive enumeration (only for small systems), linear programming, policy-iteration, and value iteration. The current MDP problem was solved by using a policy-iteration algorithm (described in Appendix B) which takes as input the transition probabilities of equation (5) and the state-action costs $c_i(k)$ (defined in detail in Sect. 3.2) and returns R_{opt}.

3.2 Cost Assignment

The cost considered here consists of two components: one that captures the induced lack of continuity, and one that captures the induced buffering delay. The relative weighing between the two cost components reflects the desired latency-continuity performance compromise that is to be achieved by the optimal policy.

Continuity Cost This cost component punishes for the *lack of continuity* that may arise from a certain action. This lack of continuity may be directly experienced as in the case of a frame presentation with a duration smaller or larger than T. In addition, the continuity cost also accounts for any lack of continuity due to overflowed (lost) frames occuring during that interval.

A candidate for the continuity cost is: $DoP_i(k) = E_{X_k}\{DoP_i(k, X_k)\}$, i.e., the expected value of $DoP_i(k, X_k)$ (see definition in (3)) with respect to the number of new arrivals X_k. $DoP_i(k)$ is a legitimate MDP cost as it depends only on the current state i, and the selected action k; an expression for $DoP_i(k)$ is derived in Appendix A. Such a cost assignment returns an $E\{DoP\}$-optimal policy, i.e., a policy that minimizes: $E_{\mathcal{S}}\{DoP\} = \sum_{i \in \mathcal{S}} \pi_i(R) \cdot DoP_i(D_i(R))$ over all the policies R defined in $\mathcal{S} \times \mathcal{A}$. The results of Sect. 4 indicate that R_T, the static deterministic policy with constant presentation durations equal to the frame period, is $E\{DoP\}$-optimal.

The minimization of $E_{\mathcal{S}}\{DoP\}$ calls for the minimization of the average amount of synchronization loss which is due to: underflow discontinuities, slowdown discontinuities, overflow discontinuities and fast-forward discontinuities. The minimization of $E_{\mathcal{S}}\{DoP\}$ is a rightful objective but cannot guarantee the perceptual optimality, as it only caters for the minimization of the average loss of synchronization, without paying any attention as to how this loss of synchronization spreads in time. It has been realized that the human perceptual system is more sensitive to a small frequency of long-lasting disruptions than to a higher frequency of sort-lived disruptions [15]. This is due to human perceptual inability to notice small deviations of presentation rate. As a result, a better perceptual quality can be expected by replacing large continuity disruptions (underflows and overflows) with shorter ones (slowdowns and fast-forwards), even when the later lead to a higher value for $E_{\mathcal{S}}\{DoP\}$. Thus, a playout policy should be allowed to increase $E_{\mathcal{S}}\{DoP\}$ if this increase provides for a smoother spacing between synchronization-loss occurrences, thus help in concealing them. We pursue this idea by defining the state-action cost to be:

$$c_i(k) = \beta \cdot DoP_i(k) + (1-\beta) \cdot DoP_i^{(2)}(k) \tag{7}$$

where $DoP_i^{(2)}(k) = E_{X_k}\{(DoP_i(k, X_k))^2\}$ is the expected square value of $DoP_i(k, X_k)$ with respect to X_k; an expression for $DoP_i^{(2)}(k)$ is derived in Appendix A. The weighing factor β is a user-defined input that controls the relative importance between the two minimization objectives; the minimization of the expected value of DoP over $\mathcal{S}$ and the minimization of its expected square value over $\mathcal{S}$. Setting $\beta = 1$ leads to the minimization of $E_{\mathcal{S}}\{DoP\}$ without any regard for the variability of the duration of synchronization loss occurrences. Setting $\beta = 0$ demands the minimization of $E_{\mathcal{S}}\{DoP^2\} = \sum_{i \in \mathcal{S}} \pi_i(R) \cdot DoP_i^{(2)}(D_i(R))$ – the expected square value of DoP – over all the policies R defined in $\mathcal{S} \times \mathcal{A}$. The resulting $E_{\mathcal{S}}\{DoP^2\}$-optimal policy induces more smooth synchronization losses than the $E_{\mathcal{S}}\{DoP\}$-optimal are. As will be shown later, the reduction in the expected square value of DoP comes at the cost of an increase in the expected value of DoP. Values of β that fall between the two extremes (0 and 1) provide various levels of compromise between $min\{E_{\mathcal{S}}\{DoP\}\}$ and $min\{E_{\mathcal{S}}\{DoP^2\}\}$.

Latency Cost In addition to the continuity cost a latency cost is jointly considered to allow for the control of the buffering delay.

The buffering delay for an arriving frame depends on the number of frames i found in the buffer upon arrival, the presentation duration associated with these frames, $B_n, n = 1 \ldots i$, and the remaining duration of the frame that is currently being presented, Y. The buffering delay of an arriving frame that finds i frames in the buffer is:

$$W_i = Y + \sum_{n=1}^{i} B_n \tag{8}$$

W_i cannot be expressed explicitly, as B_n's follow a distribution which is not known; in fact, this distribution is derived in the process of obtaining the optimal policy. Nevertheless, it is expected that the mean value of B_n will not differ significantly from the actual frame duration T (since the desirable mean playout rate induced by the optimal policy is close to and cannot exceed $1/T$). Thus, W_i can be approximated by some value in the interval $[i \cdot T, (i+1) \cdot T]$. $\widetilde{W_i} = i \cdot T$ is chosen as the approximation of W_i. Let $\{A_n\}_{n>0}$ denote the buffer occupancy process on frame arrival instants and let $a_i, i = 0 \ldots N$ denote the distribution of $\{A_n\}$. Then the expected buffering delay is approximated by:

$$\widetilde{W} = \sum_{i=0}^{N} a_i \cdot \widetilde{W}_i = \sum_{i=0}^{N} a_i \cdot i \cdot T \tag{9}$$

Equation (9) approximates the mean buffering delay by a function of a_i, the distribution of $\{A_n\}$. It applies that $\{A_n\}$ has the same distribution with $\{I_n\}$, that is, $a_i \equiv \pi_i$ (see Sec. 5.3 in [21] or Sec. 8.3 in [22] for details); thus $\widetilde{W}$ becomes a function of π_i, the distribution of $\{I_n\}$. By regulating the mean value of $\{I_n\}$ we also regulate the delay approximation $\widetilde{W}$ since the latter only depends on the distribution π_i. The mean value of $\{I_n\}$, and thus the delay $\widetilde{W}$, can be regulated by introducing a latency cost component to every action. We define the latency cost for taking decision k when the number of frames in buffer (the state) is i to be:

$$L_i \triangleq \frac{1}{T \cdot N} \cdot \widetilde{W}_i = \frac{i}{N} \tag{10}$$

The latency cost does not explicitly depend on the chosen playout duration (determined by k), but only implicitly, as the evolution of $\{I_n\}$ depends on the presentation duration of the current frame. The buffer size N together with T normalize the latency cost to unity.

We add the latency cost to the continuity cost introduced in the previous section by attaching to it a weighing factor γ that controls its relative importance over the continuity cost. The final expression for the transition cost is given by:

$$c_i(k) \triangleq \beta \cdot DoP_i(k) + (1-\beta) \cdot DoP_i^{(2)}(k) + \gamma \cdot L_i \tag{11}$$

4 Results and Discussion

The numerical results presented in this section are based on a reference system with the following parameters: a playout buffer with space for $N = 50$ frames; a normal frame rate $\lambda = 30$ frames/second; a basic time quantum c, equal to a tenth of the normal frame period T (i.e., a=10 in (1)). The two weighing factors, β, γ of (11), are used for the derivation of different playout policies for applications with different continuity-latency requirements.

4.1 Optimizing Stream Continuity

The following numerical results demonstrate policies that perceive the intrastream synchronization quality, as a combination of $E_{\mathcal{S}}\{DoP\}$ and $E_{\mathcal{S}}\{DoP^2\}$, following the arguments of Sect. 3.2. The latency weight γ is set to zero, thus does not affect the shape of the optimal policy. We focus on the role of continuity weight β, that regulates the relative importance between the mean value and the variability of DoP over $\mathcal{S}$.

Figure 1 depicts the effect of β on the shape of the optimal policy. The x,y plane illustrates the optimal policy for a specific value of β (on the z-axis). We let β take values in $[0, 0.1]$ with an increment of 0.001. For small values of β the policy tries to minimize the variability of DoP. In doing so, it presents frames slower when approaching an underflow and faster when approaching an overflow. For the given numerical example, the maximum frame duration is $B(14)$, corresponding to the buffer occupancy $i = 1$, and the minimum is $B(8)$, corresponding to a full buffer ($i = N$). It

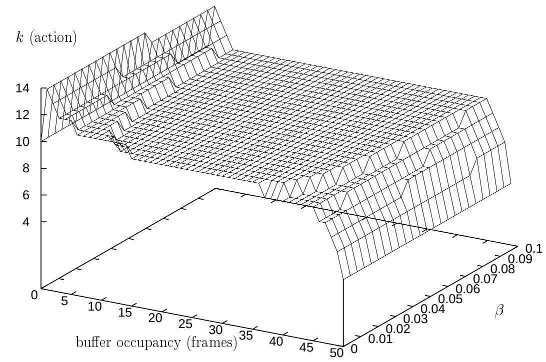

Figure 1: Optimal policies for various values of β, the weight factor that governs the tradeoff between the expected value and the expected square value of DoP. No latency cost is considered here ($\gamma = 0$). The normal frame duration T corresponds to $B(10)$ i.e., when the decision value k is ten.

may seem odd that for an occupancy equal to zero, the optimal policy plays the next frame with a normal duration $B(10) = T$ and not slower as would be expected. This decision is justified by noting that for the special case of an underflow occurrence, prior to the display of the first arriving frame, an underflow interval exists; the policy does not lengthen the duration of this frame because the associated *slowdown* discontinuity $d_0(k)$, would become very large as it also contains the underflow interval.

The amount of playout regulation decreases with increasing values of β. For $\beta = 1$ the optimal policy becomes R_T, deterministic with presentation duration T regardless of buffer occupancy. R_T is the $E_{\mathcal{S}}\{DoP\}$-optimal policy as it returns the smallest $E_{\mathcal{S}}\{DoP\}$ out of all the policies R defined in $\mathcal{S} \times \mathcal{A}$.

Figures 2,3 illustrate the effect of β on the values of $E_{\mathcal{S}}\{DoP\}$ and $E_{\mathcal{S}}\{DoP^2\}$ for different optimal policies $R_{opt}(\beta)$. It is evident that small values of β favour the reduction of $E_{\mathcal{S}}\{DoP^2\}$ by increasing $E_{\mathcal{S}}\{DoP\}$, while the opposite holds for large values of β, where the optimal policy returns a small value for $E_{\mathcal{S}}\{DoP\}$ but with a large variability, captured by $E_{\mathcal{S}}\{DoP^2\}$.

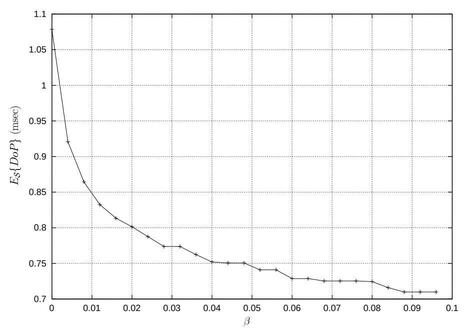

Figure 2: The effect of the continuity weight β on $E_{\mathcal{S}}\{DoP\}$. The latency weight γ is set zero.

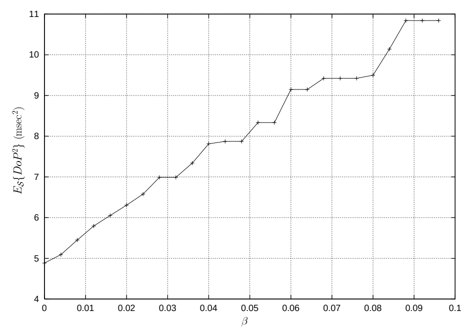

Figure 3: The effect of the continuity weight β on $E_{\mathcal{S}}\{DoP^2\}$. The latency weight γ is set zero.

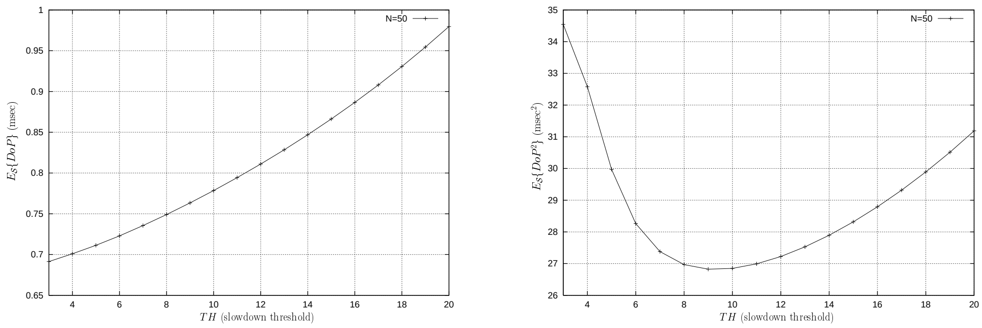

Figure 4: $E_{\mathcal{S}}\{DoP\}$ and $E_{\mathcal{S}}\{DoP^2\}$ from the threshold-based playout adaptation scheme of [16] where the value of the threshold parameter (TH) regulates the tradeoff between the two metrics.

4.2 Comparative Analysis with Non-Optimal Schedulers

This section demonstrates the performance gains of the optimal scheduler by comparing its performance with the performance of the non-optimal, empirical threshold-based playout scheduler of [16] (briefly discussed in Sec. 1). The optimality of our system stems from its ability to provide a minimal[3] $E_{\mathcal{S}}\{DoP^2\}$ ($E_{\mathcal{S}}\{DoP\}$) for a given $E_{\mathcal{S}}\{DoP\}$ ($E_{\mathcal{S}}\{DoP^2\}$). A system designer may choose an appropriate β that yields a desired value for one of $E_{\mathcal{S}}\{DoP\}$ or $E_{\mathcal{S}}\{DoP^2\}$, and know that the other continuity component will be as small as possible for this desired value.

Figure 4 illustrates the performance of the threshold scheduler, in terms of $E_{\mathcal{S}}\{DoP\}$ and $E_{\mathcal{S}}\{DoP^2\}$, both of which depend on the threshold parameter TH. These results correspond to a receiver with a buffer for 50 frames and a mean arrival rate of 30 frames/sec. The best $E_{\mathcal{S}}\{DoP^2\}$ performance, 26.8 msec^2, is achieved for the value $TH = 9$ and is approximately 5.5 times worse than the performance of the $E\{DoP^2\}$-optimal playout scheduler (26.8 msec^2 for the threshold scheduler, 4.88 msec^2 for the optimal scheduler for $\beta = 0$, see also Fig. 3). The optimal $E_{\mathcal{S}}\{DoP\}$ performance is equal for both systems and is given when they reduce to R_T (constant playout duration T for all occupancy levels); this happens for $\beta = 1$, for the proposed system, and for $TH = 1$ for the threshold scheduler (for this value it never expands the duration of frames).

The optimality of the proposed scheme (thus the suboptimality of the threshold scheduler) can be demonstrated by fixing the value of one of the continuity components and comparing the other. For $\beta = 0.02$ and for $TH = 11$ both systems result in approximately the same $E_{\mathcal{S}}\{DoP\}$ (0.8 msec, see Figs 2, 4). The corresponding $E_{\mathcal{S}}\{DoP^2\}$ are 6.3 msec^2 for the proposed system and 27 msec^2 for the threshold scheduler (Figs 3, 4).

4.3 Controlling the Buffering Delay and the Effect on Stream Continuity

This section proposes two methods for the regulation of the induced buffering delay and presents some results on the degrading effect of the delay control function on the overall stream continuity. The proposed delay control methods can be configured to suit the requirements of different applications, so the results presented here aim more at illustrating their differences rather than evaluating a specific delay-sensitive application.

[3]The minimum achieved value is bounded from below by the granularity of the adjustment quantum (specified by α).

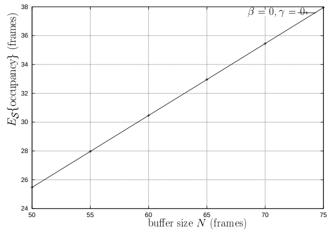

Figure 5: The effect of the buffer size on $E_{\mathcal{S}}\{\text{occupancy}\}$.

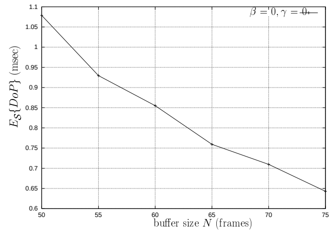

Figure 6: The effect of the buffer size on $E_{\mathcal{S}}\{DoP\}$.

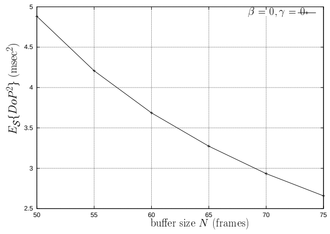

Figure 7: The effect of the buffer size on $E_{\mathcal{S}}\{DoP^2\}$.

4.3.1 Limiting the size of the buffer

A simple way to control the buffering delay is by controlling the size, N, of the playout buffer. The maximum buffering delay of a presented frame is approximately bounded by $N-1$ times the mean frame presentation duration (which in essence is equal to the normal frame duration T). The delay is passively controlled by the overflows from the tail of the buffer. The size of the buffer also affects the expected buffering delay. Figure 5 illustrates the reduction of the expected buffer occupancy (and, thus the delay as well) with small buffer sizes, for a $E\{DoP^2\}$-optimal policy ($\beta = 0$). Smaller buffer sizes, though, harm the continuity of the stream by increasing both $E_{\mathcal{S}}\{DoP\}$ and $E_{\mathcal{S}}\{DoP^2\}$ (Figs 6,7).

4.3.2 Active control using the latency cost component

The latency cost component of Sect 3.2 and the associated latency weight γ of equation (11) can be used to control the playout buffer occupancy and the buffering delay. The introduction of the latency weight leads to a playout policy that truncates some frames, not only to prevent long-lasting continuity disruptions due to overflows, but also to reduce the buffer occupancy. As with the previous approach, the active regulation of the delay has a twofold degrading effect on stream continuity; both the expected value of DoP and its variability increase when trying to reduce the delay (with non zero γ).

Figure 10 illustrates the reduction of the expected buffer occupancy with increasing values of γ. For a given latency weight γ, smaller values of β lead to smaller occupancy levels. This happens as the $DoP_i(k)$ costs (adding to $c_i(k)$ more for large βs) are higher than the corresponding $DoP_i^{(2)}(k)$ costs (adding to $c_i(k)$ more for small βs); thus larger βs balance more effectively the latency weight γ, which is pushing for greater latency reduction.

By choosing an appropriate latency weight the playout buffer occupancy can be stabilized to a level that adds an acceptable buffering delay component. Figure 8 [4] shows the increase in the expected value of DoP that is paid for the regulation of buffer occupancy. Figure 9 depicts the effect of the latency weight on the variability of DoP; not suprisingly the occupancy control function not only increases $E_{\mathcal{S}}\{DoP\}$ but also increases $E_{\mathcal{S}}\{DoP^2\}$.

[4]Note that compared to Fig. 10, we have switched the axes for γ and β in order to make the plots more comprehendible.

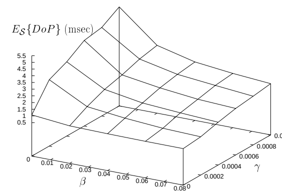

Figure 8: The effect of β and γ on $E_{\mathcal{S}}\{DoP\}$.

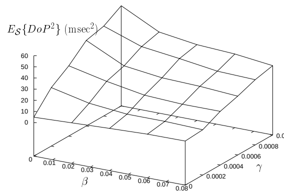

Figure 9: The effect of β and γ on $E_{\mathcal{S}}\{DoP^2\}$.

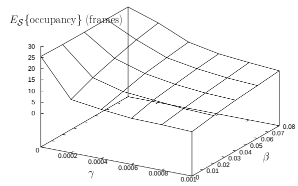

Figure 10: The effect of β and γ on $E_{\mathcal{S}}\{\text{occupancy}\}$.

4.3.3 Comparison of the two delay control methods

The buffer oriented and the cost oriented delay control methods can be configured appropriately so that they yield the same expected buffer occupancy. In that case, their differentiation lies in their performance as far as the maximum delay and the stream continuity are concerned; the buffer oriented control provides for a smaller maximum buffering delay while the cost method provides for a better continuity quality. Take for example the $E\{DoP^2\}$-optimal system with $\beta = 0, \gamma = 0, N = 50$. As shown in Fig. 5 its expected buffer occupancy is 25.5 frames. We apply the cost oriented delay control for various latency weights γ in a system with $\beta = 0$ and a larger buffer $N = 75$. The results are plotted in Figs 11, 12 and 13. Looking closely at Fig 11 one may see that for $\gamma = 1.73 \cdot 10^{-5}$ the expected buffer occupancy is 25.5 frames, i.e., it is equal to the expected occupancy of the previous system that used a smaller buffer but not latency cost. The following two figures show that for this particular value of the latency cost the cost oriented scheme provides for a better $E_{\mathcal{S}}\{DoP\}$ and a better $E_{\mathcal{S}}\{DoP^2\}$[5] compared with the buffer oriented scheme that provides the same expected buffer occupancy[6].

Summarizing, the cost based delay control takes a rather "mild" delay control approach, harming the continuity of the stream as less as possible. A large cluster of arriving frames does not necessarily lead to frame overflows, as is the case with a small buffer and $\gamma = 0$, but is received in the buffer. This leads to a transient increase of the delay that some frames will experience but this delay is progressively reduced as frames are being presented at a higher rate. This could be beneficial for continuity-sensitive applications that are willing to trade transient increases of delay for better continuity quality.

[5]Follow the dashed horizontal and vertical lines that mark this data point in Figs 11, 12 and 13.

[6]These values are on the y axis of Figs 5,6,7 for $N = 50$ on the x axis.

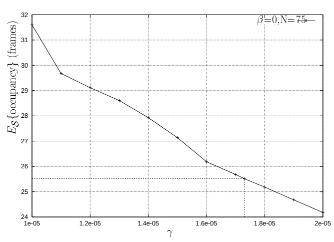

Figure 11: The effect of γ on $E_{\mathcal{S}}\{\text{occupancy}\}$.

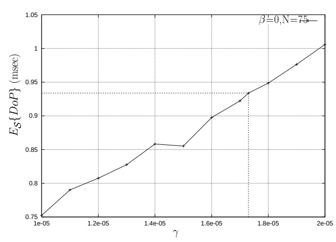

Figure 12: The effect of γ on $E_{\mathcal{S}}\{DoP\}$.

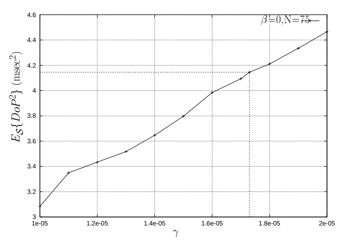

Figure 13: The effect of γ on $E_{\mathcal{S}}\{DoP^2\}$.

5 Implementation Issues and Future Work

It has already been mentioned that the exponential interarrival distribution is much more variable compared to typical interarrival distributions of periodic streams. The optimal policy derived in this paper delivers a performance approaching the optimal as long as the variability of a real interarrival distribution approaches the variability of the exponential distribution. Such high variability can be realistic in time-windows of extreme network jitter. However, in times of reduced network jitter the performance of the proposed scheme can deviate significantly from the expected optimal. To overcome this problem, the analytical model is being expanded to allow for the incorporation of arrival processes that are more regular than Poisson. This will allow for the derivation of the optimal policies that correspond to different levels of network jitter. The envisioned implementation will only need to estimate the current level of network jitter and then “load” the appropriate offline-computed optimal policy. The gain of such an architecture is twofold: the performance of the scheduler would be optimized, and the complexity of the system would be kept low as no online optimization is performed since offline computed policies would be used. These two features of the proposed scheme could make it applicable for video communications over wireless channels. In such environments jitter concealment is even more important due to the existence of an additional jitter component due to fading in the wireless link [18]. Additionally, there is a requirement for low complexity, dictated by the relative small processing capability of wireless devices, as it is the case with the proposed system that does not perform online optimization.

6 Conclusions

This paper has presented a family of playout schedulers for packet video receivers. The proposed schedulers optimize a meaningful expression of stream quality which involves the two major performance quantities: the stream continuity and the induced buffering delay. A fair and compact stream continuity metric – the Distortion of Playout – has been used as the basis for the detailed assessment of the overall intrastream synchronization quality that caters for both the average amount of synchronization loss and variability of its duration. It has been suggested that the joint consideration of the amount and the pattern of synchronization loss has the potential of improving the perceptual quality, compared to the case where only the average amount is considered.

The numerical results reveal an important tradeoff that must be considered when designing a packet video receiver. It is realized that the reduction of the average amount of synchronization loss and the attempt to evenly spread it in time, are two antagonistic objectives. Both continuity components are degraded by the introduction of a delay control function. For the control of the buffering delay two schemes are proposed: one that favours the reduction of the maximum buffering delay and one that provides a better stream continuity.

The performance gains of the optimal scheduler have been pointed out by means of comparison with an empirical scheduler from the literature. Finally, the applicability of the proposed schemes has been discussed together with directions for the future development of the work.

References

[1] Dinesh C. Verma, Hui Zhang, and Domenico Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proceedings of Tricomm '91*, Chapel Hill, North Carolina, Apr. 1991, IEEE.

[2] Domenico Ferrari, "Distributed delay jitter control in packet-switching internetworks," Technical Report TR-91-056, University of California, Berkeley, California, Oct. 1991, also in *Journal of Internetworking: Practice and Experience.*

[3] Reza Rejaie, Mark Handley, and Deborah Estrin, "An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, Mar. 1999.

[4] Sally Floyd, Mark Handley, Jitendra Padhye, and Joerg Widmer, "Equation-based congestion control for unicast applications," in *SIGCOMM Symposium on Communications Architectures and Protocols*, Stockholm, Sweden, Aug. 2000, ACM.

[5] James D. Salehi, Zhi-Li Zhang, Jim Kurose, and Don Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 397–410, Aug. 1998.

[6] Z. Jiang and L. Kleinrock, "A general optimal video smoothing algorithm," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, California, March/April 1998, p. 676.

[7] Zhi-Li Zhang, Yuewei Wang, D. H. C. Du, and Dongli Su, "Video staging: A proxy-server-based approach to end-to-end video delivery over wide-area networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, Aug. 2000.

[8] Mario Baldi and Yoram Ofek, "End-to-end delay analysis of videoconferencing over packet switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, Aug. 2000.

[9] William E. Naylor and Leonard Kleinrock, "Stream traffic communication in packet switched networks: destination buffering constraints," *IEEE Transactions on Communications*, vol. COM-30, no. 12, pp. 2527–2534, Dec. 1982.

[10] Giulio Barberis and Daniele Pazzaglia, "Analysis and optimal design of a packet-voice receiver," *IEEE Transactions on Communications*, vol. COM-28, no. 2, pp. 217–227, Feb. 1980.

[11] Warren A. Montgomery, "Techniques for packet voice synchronization," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, pp. 1022–1028, Dec. 1983.

[12] Felipe Alvarez-Cuevas, Miquel Bertran, Francesc Oller, and Josep M. Selga, "Voice synchronization in packet switching networks," *IEEE Network*, vol. 7, no. 5, pp. 20–25, Sept. 1993.

[13] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *Proceedings of the*

Conference on Computer Communications (IEEE Infocom), Toronto, Canada, June 1994, pp. 680–688, IEEE Computer Society Press, Los Alamitos, California.

[14] Sue B. Moon, Jim Kurose, and Don Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms," *ACM/Springer Multimedia Systems*, vol. 5, no. 1, pp. 17–28, Jan. 1998.

[15] Maria C. Yuang, Shih T. Liang, Yu G. Chen, and Chi L. Shen, "Dynamic video playout smoothing method for multimedia applications," in *Proceedings of the IEEE International Conference on Communications (IEEE ICC)*, Dallas, Texas, June 1996, p. S44.

[16] Nikolaos Laoutaris and Ioannis Stavrakakis, "Adaptive playout strategies for packet video receivers with finite buffer capacity," in *Proceedings of the IEEE International Conference on Communications (IEEE ICC)*, Helsinki, Finland, June 2001.

[17] Maria C. Yuang, Po L. Tien, and Shih T. Liang, "Intelligent video smoother for multimedia communications," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, pp. 136–146, Feb. 1997.

[18] Hang Liu and Magda El Zarki, "Delay and synchronization control middleware to support real-time multimedia services over wireless PCS networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 9, pp. 1660–1671, Sept. 1999.

[19] Mark Claypool and Jonathan Tanner, "The effects of jitter on the perceptual quality of video," in *ACM Multimedia '99*, Orlando, FL, USA, 1999.

[20] Sheldon M. Ross, *Applied Probability Models with Optimization Applications*, Dover Publications, New York, 1992.

[21] Leonard Kleinrock, *Queueing Systems Volume I: Theory*, Wiley-Interscience, New York, 1975.

[22] Ronald W. Wolff, *Stochastic Modeling and the Theory of Queues*, Prentice Hall, New Jersey, 1988.

[23] Frederick S. Hillier and Gerald J. Lieberman, *Introduction to Operations Research*, McGraw-Hill, 2000.

[24] Henk C. Tijms, *Stochastic Modelling and Analysis: A Computational Approach*, John Wiley & Sons, 1986.

A Derivation of $DoP_i(k)$ and $DoP_i^{(2)}(k)$

$DoP_i(k)$ removes the dependence on the number of new arrivals X_k. Taking the expected value of (3) we have:

$$DoP_i(k) = E_{X_k}\{DoP_i(k, X_k)\} = d_i(k) + T \cdot E_{X_k}\{l_i(X_k)\} \tag{12}$$

The expected number of overflowed frames over an interval with duration $B(k)$ and with an initial occupancy i is:

$$E_{X_k}\{l_i(X_k)\} = \sum_{x=0}^{\infty} P\{X_k = x\} l_i(x) = \sum_{x=N-i+1}^{\infty} P\{X_k = x\}(x - N + i - 1) \tag{13}$$

For Poisson distributed X_k we have: $P\{X_k = x\} = e^{-\lambda B(k)}(\lambda B(k))^x / x!$, thus (13) becomes:

$$E_{X_k}\{l_i(X_k)\} \sum_{x=N-i+1}^{\infty} e^{-\lambda B(k)} \frac{(\lambda B(k))^x}{x!}(x - N + i - 1) \tag{14}$$

Similarly we obtain the expected square value of $DoP_i(k, X_k)$ with respect to X_k.

$$\begin{aligned} DoP_i^{(2)}(k) &= E_{X_k}\{(DoP_i(k, X_k))^2\} \\ &= d_i^2(k) + 2d_i(k)TE_{X_k}\{l_i(X_k)\} + T^2 E_{X_k}\{(l_i(X_k))^2\} \end{aligned} \tag{15}$$

Finally, the expected square value of loss is given by:

$$\begin{aligned} E_{X_k}\{(l_i(X_k))^2\} &= \sum_{x=0}^{\infty} P\{X_k = x\}(l_i(x))^2 \\ &= \sum_{x=N-i+1}^{\infty} P\{X_k = x\}(x - N + i - 1)^2 \end{aligned} \tag{16}$$

B Policy Iteration Algorithm

In the following we describe Howard's policy-improvement method which is used for the derivation of the optimal policy of the MDP of Sect. 3.1. The algorithm starts with an arbitrary initial policy R_0, which in this case is the deterministic policy R_T, and enters an iteration that produces a gradually improving policy, until the optimal policy is reached. In the following we outline the operation of the algorithm. For details and for the interpretation of values $v_i(R)$ see [23, 24].

Initialization Set $R_0 \equiv R_T$, i.e., $D_i(R_0) = a, \forall i \in \mathcal{S}$.

Iteration n Perform the following steps:

Value determination For policy R_n solve the system of $N + 1$ equations

$$g(R_n) = c_i(k) + \sum_{j=0}^{N} p_{ij}(k) v_j(R_n) - v_i(R_n), \quad \text{for } 0 \le i \le N$$

for the $N + 1$ unknown values $g(R_n), v_0(R_n), v_1(R_n), \ldots, v_{N-1}(R_n)$.

Policy improvement Using the values of $v_i(R_n)$ computed for policy R_n, find an improved policy R_{n+1} such that for each state i, $D_i(R_{n+1}) = k$ is the decision that minimizes:

$$c_i(k) + \sum_{j=0}^{N} p_{ij}(k) v_j(R_n) - v_i(R_n), \quad \text{for } 0 \le i \le N \tag{17}$$

i.e, for each state i minimize (17) and set $d_i(R_{n+1})$ equal to the minimizing value of k. This procedure defines a new policy R_{n+1}.

Optimality test If the current policy R_{n+1} is identical to the previous R_n, then it is the optimal policy. Otherwise set $n = n + 1$ and perform another iteration.