Content-Aware Playout and Packet Scheduling for Video Streaming Over Wireless Links

Yan Li, *Senior Member, IEEE*, Athina Markopoulou, *Member, IEEE*, John Apostolopoulos, *Fellow, IEEE*, and Nicholas Bambos

***Abstract*—Media streaming over wireless links is a challenging problem due to both the unreliable, time-varying nature of the wireless channel and the stringent delivery requirements of media traffic. In this paper, we use joint control of packet scheduling at the transmitter and content-aware playout at the receiver, so as to maximize the quality of media streaming over a wireless link. Our contributions are twofold. First, we formulate and study the problem of joint scheduling and playout control in the framework of Markov decision processes. Second, we propose a novel content-aware adaptive playout control, that takes into account the content of a video sequence, and in particular the motion characteristics of different scenes. We find that the joint scheduling and playout control can significantly improve the quality of the received video, at the expense of only a small amount of playout slowdown. Furthermore, the content-aware adaptive playout places the slowdown preferentially in the low-motion scenes, where its perceived effect is lower.**

***Index Terms*—Adaptive media playout, cross-layer optimization, multimedia delivery over wireless networks, network control, packet scheduling, video-aware adaptation and communication.**

I. Introduction

RECENT advances in video compression and streaming as well as in wireless networking technologies (next-generation cellular networks and high-throughput wireless LANs), are rapidly opening up opportunities for media streaming over wireless links. However, the erratic and time-varying nature of a wireless channel is still a serious challenge for the support of high-quality media applications. To deal with these problems, network-adaptive techniques have been proposed, [7], that try to overcome the time-variations of the wireless channel using controls at various layers at the transmitter and/or the receiver.

In this work, we consider the transmission of pre-stored media units over a wireless channel which supports a time-varying throughput. We investigate the joint control of packet scheduling at the transmitter (Tx) and playout speed at the receiver (Rx), so as to overcome the variations of the channel and maximize the perceived video quality, in terms of both picture and playout quality. We couple the action of the transmitter and the receiver, so that they coordinate to overcome the variations of the wireless channel. We jointly consider and optimize several layers, including packet scheduling at the medium access control (MAC) layer, together with playout and content-awareness at the video application layer. Video content is taken into account both in playout as well as in rate-distortion optimized packet scheduling.

We briefly note the following intuitive tradeoffs faced by the individual controls in the attempt to maximize video quality. At the Tx side, the dilemma is the following: on one hand we want to transmit all media units; on the other hand, during periods that the bandwidth is scarce, we may choose to transmit the most important units and skip some others, depending on their rate-distortion values. At the Rx side, the dilemma is the following: on one hand, we want to display the sequence at the natural frame rate; on the other hand, during bad periods of the channel, we may choose to slowdown the playout in order to extend the playout deadlines of packets in transmission, and avoid late packet arrivals (leading to buffer underflow and frame losses), but at the expense of the potentially annoying slower playout. A novel aspect of this work is that we perform content-aware playout variation; that is, we take into account the characteristics of a video scene when we adapt the playout speed. The contributions of this work are the following.

1) We formulate the problem of joint playout and scheduling within the framework of Markov decision processes and we obtain the optimal control using dynamic programming.
2) We introduce the idea of content-aware playout and demonstrate that it significantly improves the user experience. The idea is to vary the playout speed of scenes, based on the scene content; e.g., scenes with low or no motion typically may be less affected by playout variation than scenes with high motion.

The rest of the paper is structured as follows. Section II discusses related work. Section III introduces the system model and problem formulation. Section IV provides simulation results. Section V concludes the paper.

II. Related Work

Streaming media over an unreliable and/or time-varying network, whether this is the Internet or a wireless network, is a large problem space with various aspects and control parameters. Several network-adaptive techniques have been proposed [7], including rate-distortion optimized packet scheduling [4],

Manuscript received May 8, 2007; revised January 22, 2008. First published June 13, 2008; last published July 9, 2008 (projected). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pascal Frossard.

Y. Li is with Qualcomm, Campbell, CA 95008 USA (e-mail: liyan@stanfordalumni.org).

A. Markopoulou is with the Electrical Engineering and Computer Science Department, University of California, Irvine, CA 92697 USA (e-mail: athina@uci.edu).

J. Apostolopoulos is with the Streaming Media Group, Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: japos@hpl.hp.com).

N. Bambos is with the Electrical Engineering Department, Stanford University, Stanford, CA 94305 USA (e-mail: bambos@stanford.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2008.922860

[5], [13], power [17] and/or rate control at the transmitter, and playout speed adaptation at the receiver [14], [15]. Wireless video, in particular, is a very challenging problem, due to the limited, time-varying resources of the wireless channel; a survey can be found in [6]. There is a large body of work on cross-layer design for video streaming over wireless, including [1], [3], [28]–[32], [34] to mention a few representative examples. Our work also falls within the scope of cross-layer optimization. In the rest of this section, we clarify our contribution and comparison to prior work in this problem space.

A. Prior Work on Adaptive Playout

Playout control at the receiver can mitigate packet delay variation and provide smoother playout. Adaptive playout has been used in the past for media streaming over the Internet for both audio [23], [26], [33], and video [14], [15]. Our work proposes for the first time to make playout control content-aware. That is, given a certain amount of playout slowdown and variation caused by bad channel periods, we are interested in applying this slowdown to those part of the video sequence that are less sensitive from a perceptual point of view.

Within the adaptive playout literature, the closest to our work are [14] and [15]. However, there are two differences. The first difference is that we propose, for the first time, a content-aware playout; we build on and extend the metrics in [15] to include the notion of motion-intensity of a scene. A second and more subtle difference lies in the formulation. [14] models the system as a Markov chain and analyzes the performance of adaptive algorithms that slowdown or speed up the playout rate based on the buffer occupancy. However, the parameters of these algorithms, such as buffer occupancy thresholds, speedup and slowdown factors, are fixed and must be chosen offline. In contrast, we model the system as a controlled Markov chain [2], which allows for more fine-tuned control: the control policy itself is optimized for the parameters of the system, including the channel characteristics. For example, when the channel is good, the playout policy can be optimistic and use low levels of buffer occupancy, under which it starts to slow down; when the channel is bad the optimal policy should be more conservative and start slowing down even when the buffer is relatively full. Finally, another difference lies in the system design: [14] performs slowdown and speedup at the Rx, while we perform slowdown at the Rx and drop late packets at the Tx, thus saving system resources from unnecessary transmissions.

B. Prior Work on Packet Scheduling

In this paper, we use packet scheduling at the Tx to complement the playout functionality at the Rx. The main purpose of the scheduler is to discard late packets and catch up with accumulated delay caused by playout slowdown during bad channel periods; these late packets would be dropped anyway at the Rx, but dropping them at the Tx saves system resources. In addition, we enhanced the scheduler to transmit a subset of the video packets to meet the channel rate constraint with minimum distortion. This paper does not aim at improving the state of the art in radio-distortion optimized scheduling; instead, its contribution lies in the playout control. The scheduling control enhances the playout and is optimized for that purpose.

The state-of-the-art in rate-distortion optimized packet scheduling is currently the RaDiO family of techniques [4], [5], [13]: in every transmission opportunity, a decision is made as to which media units to transmit and which to discard, so as to maximize the expected quality of received video subject to a constraint in the transmission rate, and taking into consideration transmission errors, delays and decoding dependencies. Similar to RaDiO, our scheduler efficiently allocates bandwidth among packets, so as to minimize distortion and meet playout deadlines. Both works propose analytical frameworks to study video transmitted over networks. However, there are two differences. First, the two modeling approaches are different: we formulate the problem as a controlled Markov chain, thus being able to exploit the channel variations, while RaDiO formulates the problem using Lagrange multipliers,thus optimizing for the average case. Second, different simplifications are used to efficiently search for the optimal solution: RaDiO optimizes the transmission policy for one packet at a time, while we constrain our policies to in-order transmission. Our approach could also be extended to include out-of-order transmissions.

Another framework for optimizing packet scheduling is CoDiO, congestion-distortion optimized streaming [27], which takes into account congestion which is detrimental to other flows but also to the stream itself; in a somewhat similar spirit, our scheme may purposely drop late packets at the transmitter in order to avoid self-inflicted increase in the stream's end-to-end delay.

Finally, we would like to note that, in this paper, we focus on nonscalable video encoding, which is the great majority of pre-encoded video today as well as in the foreseeable future. If the original video is encoded with scalable video coding then we will have more flexibility in terms of what to drop to fit the available bandwidth and delay constraints; however, some of the techniques proposed in this paper for assessing what to drop may still be applicable.

C. Relation to Our Prior Work

In the past, we have also used the general framework of controlled Markov chains to study different control problems, with emphasis on power control [17]. The closest of our past work to this paper is [19], [20], where we studied power-playout control for video streaming over a channel with time-varying error characteristics. In this paper, we use the same modeling and optimization methodology, but we study a different control problem. The first difference is that we deal with a different input: an error-free channel with time-varying rate is given to us and we try to overcome its fluctuations. We have no control over the channel characteristics, we can only adapt to their fluctuations. (In contrast, in [19], [20], we considered a channel with fixed rate and time-varying packet loss rate, which we could affect by varying the power.) From a streaming application's perspective, the setting considered in this paper is more realistic in today's wireless systems, as explained in Section III-B. Second, we introduce two novel controls. We introduce for the first time, content-aware playout: the idea is to selectively vary the playout in

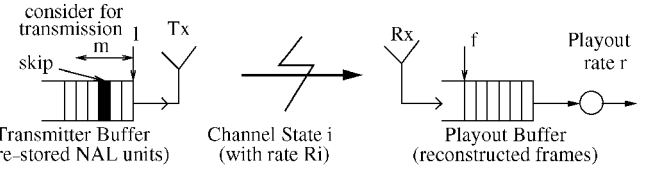


Fig. 1. Joint scheduling and playout control for streaming pre-stored NAL units over a time-varying wireless link.

scenes with less motion, where the effect will be less perceived. In addition, we use packet scheduling to discard late packets and catch up with the delay accumulated from slowing down during bad channel periods. In contrast, in [19], [20], we did not considered delay-sensitive applications and we transmitted all packets.

An early version of this work appeared in [18]. This journal paper is extended to include: the results of in-house subjective testing; a greedy algorithm for packet scheduling; more details about the video sequences used and their motion intensity; additional simulation results for a range of channels and for different rate adaptation timescales; a comparison to related work; a detailed discussion on rate adaptation techniques for video and audio;.

III. System Model and Problem Formulation

We consider the system shown in Fig. 1, which is comprised of a transmitter (Tx) and a receiver (Rx) communicating over a wireless communication link. Time is slotted. The Tx is equipped with a buffer where the content is initially stored. The Rx is equipped with a playout buffer, where received frames are queued up to be played out.

A. Video Source

We use a video sequence pre-encoded using H.264/MPEG-4 AVC [10], [11]. Let N be the total number of frames and $n = 1, \ldots, N$ be the frame index. Each frame can be further divided into a fixed number, K, of NAL (the "network abstraction layer" defibed in H.264/AVC) units, which are effectively packets for transmission. Let (n, k) be the k^{th} NAL unit in the n^{th} frame, $k = 1, \ldots, K$. This NAL unit is indexed with $l = (n-1)K+k$, has size b_l (in bits) and leads to a distortion of d_l if not received. To compute d_l, we decode the entire video sequence with this NAL unit missing. This includes the effect of error propagation between frames due to video content and coding decisions.

We consider the distortion values to be known and used as input to the control decisions. This is actually a realistic assumption. For pre-stored video, off-line exact computation of the distortion values is possible, as described above. Even when off-line computation is not an option, we can estimate the distortion by performing online analysis with a delay of a few frames. Most of the total incurred distortion occurs in the first few frames after a loss, and breaks after the next I frame; therefore, we can reduce delay at the expense of a reduced accuracy. We can even estimate the distortion using only a single frame delay, however the error bars would then be much larger; the amount of error propagation depends on the video content of subsequent frames as well as the coding decisions. Many live streaming events over the Internet support several seconds of delay. However this method is not applicable to low-delay interactive applications such as video-conferencing, which are not the target applications for the methods proposed in this paper anyway. Alternatively, a simple approach often used in the literature is to assign distortion values based solely on the group-of-pictures (GOP) structure, ignoring the actual video content and coding decisions. Another approach, referred to as ROPE [22], allows to estimate the total distortion at the decoder, given a packet loss rate and a concealment method.

Furthermore, we assume that distortions caused by loss of multiple NAL units are additive. This is a standard approximation that allows to reduce the computational complexity by separating the total distortion into a set of individual packet distortion functions and optimizing for each one of them. It is also accurate for sparse losses. In general, the actual distortion may also depend on the delivery status of prior and subsequent NALs. The distortion model can be extended to capture loss correlation between NAL units, following the recent approach in [21]. Our framework can naturally incorporate such models to capture more accurately the effect of bursty loss, which may be the case in a wireless environment.

A video sequence can consist of several scenes. Each scene s contains a group of video frames. Each scene has a different amount of motion, which we are interested in characterizing so that we can later take that into account in our content-aware playout control. Finding the appropriate metric to characterize the amount of motion in a scene is an open research problem. In this paper, we define the *motion intensity* M_s as the sum of the absolute values of all the motion vectors in the scene s averaged over the number of frames in the scene. When we computed the motion intensity defined this way for several well-known standard scenes (taken from Foreman, Mother-Daughter and other standard sequences, as described in Table I), we found that our simple heuristic captures well the different motion characteristics of these well-known scenes [as demonstrated in Fig. 3(a)]. However, our formulation can incorporate more sophisticated metrics if/as they become available.

B. Wireless Channel

The feature we are interested in capturing is the time-varying nature of the wireless channel, whether it is in 802.11, cellular or home environment. We model the wireless channel as a Markov chain, where in channel state i, the bandwidth available to the video stream is R_i; let the transition probabilities from state i to state j be q_{ij}. The rationale behind this model is the following. Fast fading, slow shadowing, path loss and interferences all affect the signal to interference/noise ratio (SIR); in turn, the SIR dictates the physical transmission rate and packet error rate, thus the channel throughput. Assuming that the physical and MAC layers use coding, retransmissions or forward error correction to combat channel variations, the wireless channel appears to the application layer as error-free but with time-varying throughput.

The throughput R_i in state i can be calculated as $R_i^p(1 - PER)$, where the R_i^p is the physical channel rate after the coding, and PER is the packet error rate with the corresponding codes; this is reasonable if the channel varies slower than a packet's transmission time, which is the case for low mobility or in-home environments. Note that the errors

TABLE I
TEST SEQUENCE

Scene Number	Frame Numbers in Test Sequence	Original Video Sequence	Frame Numbers Original Sequence	Motion Intensity	
				average	maximum
1	1-60	mother-daughter	101-160	0.19	0.37
2	61-120	carphone	171-230	4.45	6.51
3	121-180	grandma	1-60	0.14	0.63
4	181-240	foreman	271-330	12.56	25.68
5	241-300	mother-daughter	391-450	0.18	0.45
6	301-360	carphone	281-340	3.57	6.17
7	361-420	grandma	61-120	0.22	0.58
8	421-480	suite	31-90	2.69	7.14
9	481-540	mother-daughter	901-960	0.11	0.54
10	541-600	foreman	144- 203	2.56	5.31

in the wireless channel are still taken into account in this model: they are captured by the variable throughput R_i, as perceived at the application layer. In general, R_i can be thought as abstracting several lower layer details, instead of including them as additional controls in this already multidimensional problem; this allows us to emphasize the main tradeoff of the paper between video quality vs. playout quality. From the perspective of a streaming application, abstracting the wireless channel as time-varying bandwidth is realistic, especially for wireless networks with adaptive modulation; e.g., in 3G HSDPA networks, instead of using power control, the Tx uses feedback from the receiver on the channel quality to adapt modulation.

Although the markovian model used in this paper is general enough to capture most wireless channels, the estimation of its parameters is an important issue. Several studies [12], [16], [25] have estimated these parameters from empirical data for some typical environments and could be used as input to our problem. An even better approach, in practice, would be to first spend some time to learn the channel and estimate its parameters before applying our algorithms; channel estimation is anyway an integral part of the operation of a receiver.

C. Transmission Control and Costs

We assume that all N frames of the sequence reside at the Tx. This is a realistic assumption, when the media server/proxy is co-located with the Tx or the path between the server and the Tx is not the bottleneck. Let l be the NAL unit at the head of the Tx. In this baseline model, transmission happens always in-order, the skipped units are dropped and the remaining units at the Tx have no gaps. For the rest of the paper, the term *time slot* refers to the time period over which we adjust the transmission rate (by choosing how many units to transmit). At each time slot, the Tx considers the next m units for transmission and advances the transmission index from l to $l+m$. Notice that the restriction to in-order transmission is a heuristic that simplifies the search for the optimal packet scheduling in our framework. When the Tx advances the NAL index from l to $l+m$ and skips some of the m NALs to satisfy the bandwidth constraint in the current time slot, the skipped NALs are permanently dropped from the Tx and the incurred distortion is calculated. There may be performance loss due to this in-order transmission constraint, because the skipped NALs may have less importance compared to subsequent NALs. Our model can be extended to allow for out-of-order transmission and skipped NALs to be considered for later transmissions—but we omit this extension due to space limitations.

From the considered m units, some are dropped to conform to the channel throughput R_i; which units to drop are chosen so as to minimize the total distortion $D_{tx}(m, R_i, l)$

$$\min \sum_{k\in\Theta} d_k \text{ subject to } \sum_{k=l,k\notin\Theta}^{l+m-1} b_k \le R_i \tag{1}$$

where $\Theta \subseteq \{l, l+1, \ldots, l+m-1\}$ is the set of NAL indices to be dropped.

The exact solution of this minimization can be obtained through the following dynamic programming formulation. Define $V(\Delta, R)$ to be the minimum distortion incurred of transmitting the set Δ of NAL units under rate constraint R. $V(\{l, l+1, \ldots, l+m-1\}, R_i)$ is $D_{tx}(m, R_i, l)$. If $R \ge \sum_{k\in\Delta} b_k$, $V(\Delta, R) = 0$ since all units in Δ can be transmitted. If $R < \sum_{k\in\Delta} b_k$, the system may select to drop a unit s, resulting in a distortion of d_s, and transit to a new system state of $(\Delta - s, R)$. The symbol "$-$" indicates the excluding operation. To be more precise, we formulate the recursive equation as follows:

$$V(\Delta, R) = \begin{cases} \min_{s\in\Delta} \{d_s + V(\Delta - s, R)\}, & \text{if } R < \sum_{k\in\Delta} b_k \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

However solving (2) is computationally intensive due to the large number $(2^{|\Delta|})$ of candidate subsets of Δ. We obtain a sub-optimal solution by a greedy algorithm with each NAL unit ranked by its distortion-to-size ratio, d_k/b_k. This algorithm is inspired by the greedy algorithm for a knapsack problem, if we consider d_k and b_k to be the values and weights of items in the knapsack terminology. The heuristic turned out to significantly reduce complexity at the expense of only small performance loss.

The control parameter m is chosen from a range of possible values. Large values of m advance the index further (thus helping playout) but may drop more units (thus introducing more distortion). We assume that for all practical cases, no more than 50% of units should be dropped, and we limit the range to $m \in [m_l, 2m_l]$, where m_l is the maximum number of

consecutive units that can be transmitted without exceeding the channel rate R_i

$$m_l = \arg\max_{m_l'} \sum_{k=l}^{l+m_l'-1} b_k \le R_i. \quad (3)$$

D. Content-Dependent Playout and Costs

1) Playout Speed Adaptation: Small variations in the media playout rate are subjectively less irritating than playout interruptions and long delays. To control the playout rate of media, the client scales the duration that each video frame is shown [14], and processes speech and audio [23], [35] to scale the duration without affecting its pitch. Varying the *video playout speed* is therefore straightforward. Informal tests have shown that video playout variations of up to 25%–50% can be from unnoticeable to acceptable for video [14], depending of course on the content and the frequency of the variation.

Adapting the playout rate of the audio accompanying the video stream, is less straightforward. In particular, *speech playout* time can be scaled in a way to preserve the pitch, using special processing techniques of the speech signal, such as WSOLA [35]. In [23], this technique has been used together with playout adaptation to absorb the jitter caused to VoIP when transmitted over Internet paths. This resulted to infrequent packet scaling with scaling ratios in the range of 35%–230%, which subjective experiments showed to be "inaudible" or "not annoying". *Audio playout* rate adaptation is more complicated, depending on the specific audio content, and is an active area of research.

Finally, as pointed out in [14], playout speed modification has a precedent in traditional media broadcasting although for a different purpose: motion pictures shot at a frame rate of 24 fps are shown on European PAL/SECAM broadcast television at 25 fps, i.e., a constant speedup of 4.2% even without audio time scale modification.

In the rest of this section, we discuss the variables that control rate adaptation and their associated costs.

2) Playout Control Variables: Let f be the number of fully decodable frames at the Rx, i.e., frames whose all NAL units have already been received or dropped at the Tx side; because of the in-order transmission, the units transmitted in the future will belong to subsequent frames. Note that when some NAL units are missing, the distortion has already been captured by the cost D_{tx} at the Tx. We constrain $f \le F$ to capture the physical buffer size. For small values of F, the benefit from the control is amplified; in general though, we expect memory to be cheap and thus F to be large enough and have a negligible effect on performance.

Adaptation Range and Timescale: The Rx can control the value of the playout rate $r \in \{r_1, r_2, \ldots, r_n\}$, where $r_1 < r_2 < \ldots < r_n$ and r_n is the normal rate of the video sequence (say 30 fps). New packets arrive every time slot, but we adapt r more infrequently, say every T timeslots, in order to avoid noticeable perceived effects of rapid playout variations. Similarly, to avoid large magnitude variations, we constrain r to increase or decrease only by one level, say from the previous r_k to $r \in \{r_{k+1}, r_k, r_{k-1}\}$; however, at the scene boundaries, we allow r to take any value in $\{r_1, r_2, \ldots, r_n\}$. When adapting the rate of a group of frames that span two scenes, we assign the group to the scene where most frames in the group belong to.

Removing Units From the Rx: Let t track the timeslots within a cycle of T timeslots: $t = 1, 2, \ldots T$. In the first timeslots of a cycle $(t = 1)$ the control chooses a new value for r, to use for the entire cycle; in other words, r can be adjusted only at slot $t = 1$ and not at slots $t = 2, \ldots T$. At every timeslots t, we remove and display $r^t = \lceil rt/T \rceil - \lceil r(t-1)/T \rceil$ frames from the buffer. This reduces the number of full frames in the Rx, f, by $(f - r^t)^+$, where the notation x^+ means $x^+ = x$, if $x \ge 0$ and $x^+ = 0$, otherwise.

There may be timeslots when the playout control chooses to remove more frames than the currently available in the buffer, $r^t > f$. Then, the Tx is notified to drop the NAL units that miss their deadlines, and the Tx continues with subsequent units. This leads to an additional distortion cost

$$D_{rx}(r, f, l) = \sum_{k=l}^{(f_e+(r^t-f))K} d_k$$

where $f_e = \lfloor (l-1)/K \rfloor$ is the frame index of last fully decodable frame at the Rx buffer. At the Tx side, the index l is updated to $(f_e + (r^t - f))K + 1$.

Units Arriving to the Rx Buffer: At each time slot, packets arrive at the Rx. We assume a store-and-forward operation where packets that arrived in the current time slot are not available for display at the same time slot. This is a conservative assumption, as some packets may arrive before the end of the timeslots; alternatively, appropriate channel models could account for the packet arrival distributions. Taking into account both arrivals and playout, in every time slot, the new NAL index l' at the Tx and new receiver buffer level f' (which only indicates the number of fully decodable frames) are updated as follows:

$$l' = \begin{cases} (\lfloor (l-1)/K \rfloor + (r^t - f))\, K + 1 + m, & r^t > f \\ l + m, & r^t \le f \end{cases}$$
$$f' = \begin{cases} ((f - r^t)^+ + (\lfloor l'/K \rfloor - \lfloor l/K \rfloor), & r^t > f \\ \lfloor m/K \rfloor, & r^t \le f \end{cases} \quad (4)$$

Equation (4) shows how state variables l, f are updated (to l', f') after applying optimal control in one time slot, and captures the following rationale. If the buffer level is larger than the number of frames to be played out this time period, i.e., $r^t \le f$, then there are no NAL losses due to missing playout deadlines and l advances to $l' = l + m$, where m is the number of newly transmitted NAL units. If the buffer level is less than the number of frames to be played out, i.e., $r^t > f$, then the Tx drops the NAL units that will miss their deadlines, updates the NAL index to $(\lfloor (l-1)/K \rfloor / + (r^t - f))K + 1$ and transmits the next m units; therefore the next to be transmitted NAL index is $(\lfloor (l-1)/K \rfloor + (r^t - f))K + 1 + m$. $(f - r^t)^+$ guarantees that we cannot have a negative buffer.

3) Playout Variation Costs: Choosing slower playout extends the playout deadlines of the NAL units in transmission (thus reducing distortion due to dropping late units) but may also produce an annoying perceived effect. This effect is scene-dependent. For example, playout speed variations are more perceptible in scenes with significant or constant motion (e.g. a

camera pan) rather than in motionless scenes. To capture this effect we introduce the following two costs:

- Let $C_s = g_1(r, M_s)$ be the slowdown cost due to playing slower than the natural rate; M_s is the motion intensity of the scene s the current r frames belong to. If the r frames cross the scene boundary, we take a weighted average of the two costs in the two scenes. The function g_1 should be increasing with M_s and decreasing with r. In this paper, we use the simple linear function: $C_s = M_s(r_n - r)$.
- Let $C_v = g_2(r, \vec{r})$ be the playout variation cost, due to variations of r from one period to the next. The vector $\vec{r}$ records the past L playout rates and is reset at scene boundaries. The function g_2 should be decreasing in r and increasing in M_s. In this paper, we use the simple function: $C_v = |r - r_{last}|$, i.e., we ignore the effect of M_s (already accounted for in C_s) and we consider only the last chosen r_{last} instead of a longer history $\vec{r}$.

Notice, that these metrics include buffer underflow as a special case of playout variation, i.e., $r = 0$. Underflow happens when packets are late and thus are dropped at the transmitter, resulting both in a playout cost $(C_s(r = 0) + C_v(r = 0, r'))$ and in a distortion cost due to dropped packets. If desired, one could easily incorporate a separate cost for $r = 0$ in the proposed formulation, e.g. to be higher than the above C_s, C_v costs and/or a function of the freeze duration; however in this paper, for simplicity, we consider it as a special case of the general cost functions.

These costs C_s, C_v extend the ones proposed in [15], [19] by including the motion intensity of a scene. The key point captured by these cost metrics, is that the playout variation should be smooth so that a smooth movement in the video scene is still mapped, after playout rate variation, to a smooth movement. For example, if a person in the video runs from left to right across the display or moves a painting across the display, the movement should be a temporally scaled version of the original movement—without any confusing warping or abrupt stops/starts. The simple motion intensity metrics defined in this paper can be further improved. In general, defining the right metrics to capture perceptual quality is an open research problem in itself. Our framework is general enough to also incorporate any new and improved perceptual metrics.

E. System State and Optimal Control

1) Formulation: The state of the system is $(l, i, f, \vec{r}, t)$; l is the unit at the head of the Tx; i is the state of the channel (corresponding to rate R_i); f is the state at the Rx and $\vec{r}$ is the playout history; finally, $t \in \{1, \ldots T\}$ tracks whether we can adjust the playout rate in the current time slot, which is true only in the first time slot of a cycle $t = 1$. The controls exercised, (m, r), are subject to the constraints described in the previous subsection. The associated costs are: C_s, C_v for the playout slowdown and variation costs; and D_{tx}, D_{rx} for the distortion cost due to packets dropped at the Tx, to meet transmission rate constraints (D_{tx}) or because they missed their playout deadlines (D_{rx}). The overall performance cost, in a single time slot, is $C_s + C_v + w(D_{tx} + D_{rx})$, where the weighting factor w captures the trade-off between video quality and playout variation. In general, there may be additional weighting factors to stress C_s versus C_v, or D_{tx} versus D_{rx}.

Let $J(.)$ be the optimal cost to go: this is the minimum total cost from the current state until the system terminates (i.e., the last packet is transmitted and played out) assuming that optimal control is used in every time slot. The system becomes a controlled Markov chain and the optimal control can be computed from the following dynamic programming equations [2]. In the first time slot of a cycle, $t = 1$, we can control both playout r and transmission m

$$J(l, i, f; \vec{r}, t = 1) = \min_{m,r} \left\{ C_s + C_v + w(D_{tx} + D_{rx}) + \sum_{j \in \mathcal{I}} q_{ij} J(l', j, f'; \vec{r}', t + 1) \right\}. \quad (5)$$

In subsequent time slots $t = 2, \ldots T$, only the transmission control m is active, while r remains the same as in $t = 1$:

$$J(l, i, f; \vec{r}, t \neq 1) = \min_{m} \left\{ w(D_{tx} + D_{rx}) + \sum_{j} q_{ij} J(l', j, f'; \vec{r}, t + 1) \right\}. \quad (6)$$

Equations (5) (and (6) can be interpreted as follows. When we are in state $(l, i, f; \vec{r}, t)$, we exercise optimal control (m, r) that achieves the minimum cost $J(l, i, f; \vec{r}, t = 1)$ among all possible choices of control variables. J consists of two parts: 1) the cost $C_s + C_v + w(D_{tx} + D_{rx})$ we pay in the current timeslot and 2) the cost $J(l', j, f'; \vec{r}', t + 1)$ we will pay in the future, if we continue to exercise optimal control. $(l', j, f'; \vec{r}', t + 1)$ is the system state in the next time slot. There is also a summation over all possible channel states j that the wireless channel can transition from the current state i to, with probability q_{ij}, respectively.

The system starts at $l = 1$ and evolves according to (5) and (6). After all NAL units are transmitted (i.e., $l = L+1$), $\{l, i, t\}$ and m are removed from the state and control respectively; then, the Rx gradually increases the playout rate (adjusting upwards every T time slots) and plays out the remaining frames at the natural rate r_n

$$J(f; \vec{r}) = \min_{r} \{ C_s + C_v + J(f - r; \vec{r}') \}. \quad (7)$$

After all frames are played out, at the rate r chosen in (7), the system terminates.

2) Complexity: Solving the dynamic programming formulation involves that we recursively compute and fill up two tables, in a bottom-up way: one table stores the optimal control (m, r) and another stores the resulting cost J for every system state $(l, i, f; \vec{r}, t)$. This recursive computation depends on the size of the state space and on the possible combinations of controls. Therefore, it requires $O((L|I|FnT) \cdot (nm_l))$, where L is the total number of NAL units for transmission, $|I|$ is the number of channel states, F is the Rx buffer size (in packets), n is the number of possible playout rates, T is the number of slots in a cycle, and m_l is maximum number of NALs that can be transmitted without exceeding the channel rate. Depending on the magnitude and granularity of the above variables, the complexity can be quite high. That is why we recommend that the optimal policy is precomputed offline and stored for online use.

The optimal control policy achieves the best distortion-playout tradeoff and thus can serve as a benchmark for comparison. Furthermore, its structural properties can be used, in the future, as a guideline for the design of efficient heuristics that mimic the optimal control.

3) Using the Optimal Policy: The optimal policy should be computed offline and be stored in a table for later use: a lookup is performed in the table to obtain the optimal control (m, r) for the current state of the system $(l, i, f, \vec{r}, t)$. Some minimum communication between the transmitter (Tx) and the receiver (Rx) is required. In principle, the policy table should be precomputed at the Tx and sent to the Rx in the beginning of the session; therefore, both Tx and Rx can read the table and take the same decision given the system state. The Rx lets the Tx know about its state, namely the buffer occupancy and channel state, through a feedback channel, which is assumed low delay and error-free in this paper. The state of the wireless channel is estimated at the Rx and the methodology is robust to estimation errors and other uncertainties. In practice, it would be even better to design low-complexity heuristics that mimic the properties of the optimal control and minimize the communication between Tx and Rx.

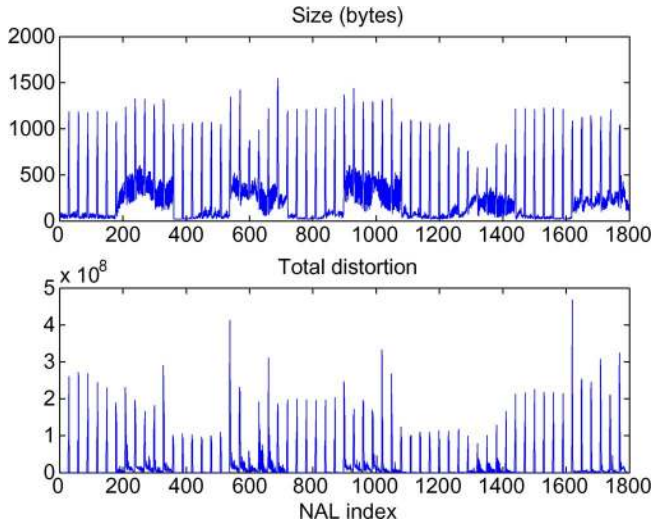


Fig. 2. Size (in bytes) and distortion (total squared error, including error propagation between frames) values for each NAL unit of our test video sequence. The test sequence consists of concatenated scenes from well-known standard sequences, with different characteristics, as described in Table I. (The standard video sequences used are all QCIF at 30 fps, encoded using only I and P frames (one I every ten frames), and packetized using three slices/frame and 33 MB/slice.)

IV. Performance Evaluation

A. Simulation Setup

We used the JM8.6 version of the H.264/MPEG-4 AVC *codec* [10], [11]. We simulated *packet loss* by erasing the corresponding NAL units from the RTP stream produced by the encoder. At the receiver side, we decoded the remaining RTP stream with error concealment enabled. In case that an entire frame is lost, we had to implement copy-concealment, which was not supported in JM 8.6. The video sequences were QCIF at 30 fps, encoded using only I and P frames (one I every ten frames), and packetized using 3 slices/frame and 33 MB/slice. The PSNR of the encoded sequence is 36.5 dB.

Our *test sequence* is shown in Table I. We concatenated ten well-known scenes from various standard sequences, which exhibit different degrees of motion. The clip for the resulting test sequence can be found at [24]. Fig. 3(a) shows the motion intensity M of the test sequence, using the metric we defined. Recall that M is defined as the weighted sum of the absolute motion vectors in each P-frame; for I-frames, $M = 0$. One can see that our heuristically defined M successfully captures the motion characteristics of these well known scenes; e.g., scene 4 corresponds to the camera pan in Foreman and has the highest M; the scenes from Grandma and Mother-Daughter have the lowest M. Fig. 2 shows the size and the distortion value of each NAL unit in the sequence. The spikes in size and distortion in the plots of Fig. 2 correspond to I-frame NAL units. The distortion value is measured as the total distortion caused by the loss of each NAL.

The parameters for the *wireless channel*, are chosen to demonstrate key features of our approach. The rate in the good and bad state was 262 kbps and 74 kbps respectively. This results in an average channel rate slightly larger than the average video rate (162 Kbps). The transition probabilities $Q = [q_{ij}]$ are set to $\begin{bmatrix} 0.67 & 0.33 \\ 0.33 & 0.67 \end{bmatrix}$ and lead to average state durations around 0.5 s, comparable to the coherence time in home or low-mobility environments. The *time slot* (for transmission and reception of a group of packets) is chosen to be five frame durations (33 ms each), i.e., 0.167 s, to allow for a reasonable number of NALs to be transmitted together. The playout rate is adjusted every $\mathrm{T} = 10$ frames.

The purpose of the simulations is to compare three policies under a range of scenarios, namely the no-control policy, the content-unaware (joint control) policy, and the optimal content-aware (joint control) policy. For the content-unaware policy, we consider M_s to be the temporal mean of the M_s value used to define the content-aware cost. Note, however, that this does not have a major impact as we vary the weighting factor w over a wide range.

B. Simulation Results

Fig. 3(b) and (c) show the playout rate (normalized relative to the natural playout rate) across frames of the test sequence without and with content-awareness, respectively. The distortion (due to dropped packets) is the same in both cases. The main observation is that the content-aware control chooses to slow down more the low motion scenes and leave the high-motion scenes intact; this reduces the perceived effect of slowdown. A secondary observation is that both controls increase the playout rate in the last 180 frames, because buffer underflow is less risky at the end of the sequence.

Fig. 4 shows the tradeoff between: 1) the increase in the playout duration due to slowdown (as a percentage of the total duration without slowdown) and 2) the increase in video quality (PSNR of the decoded sequence). The triangle on the y-axis corresponds to the no-control policy. The curve with the "X" markers corresponds to the content-unaware policy

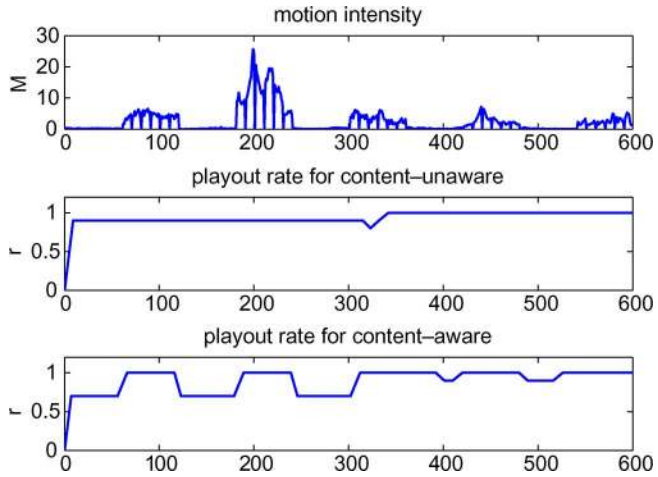


Fig. 3. (a) Motion intensity of the test sequence (described in Table I and Fig. 2). (b) Playout rate (as a fraction of the natural rate) without motion-awareness. (c) Playout rate (as a fraction of the natural rate) with motion-awareness.

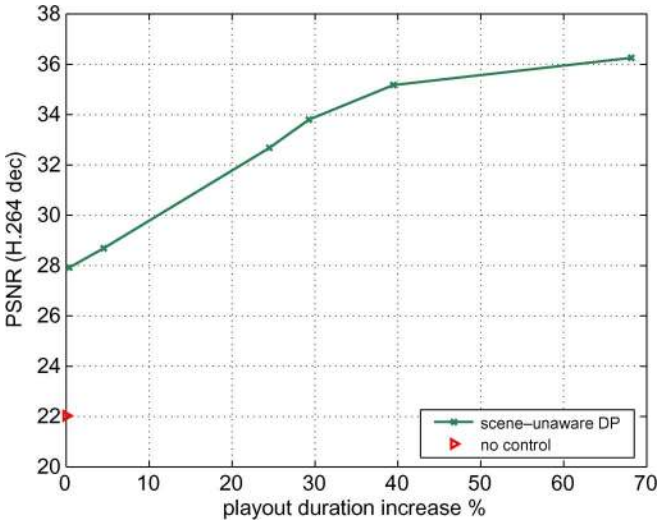


Fig. 4. Tradeoff between video quality (PSNR) and % increase in playout duration, for the no-control case and for the content-unaware control.

and is obtained by increasing the playout duration from 0% to 70% of the total playout duration without any slowdown. (The equivalent weighting factor w, between $D_{tx} + D_{rx}$ and $C_s + C_v$, needed to obtain the same result would be in the range [0.0005,0.5]). By using only the control at the transmitter (i.e., 0% increase in duration) to carefully select the right NAL units for transmission, we observe a 6-dB gain over the no-control case. Furthermore, the video quality increases approximately by 2 dB for every 10% of playout duration increase, and saturates at the PSNR of the encoded sequence. The most similar work that we are aware of is [4], where the effect of pre-roll delay on quality is studied using a different methodology. In general, the curve in Fig. 4 should depend on the wireless channel.

In Fig. 4, we characterized the effect of playout slowdown in terms of an objective metric (% increase in total duration) but did not take into account the video content. Fig. 5 shows again the tradeoff between video quality and playout variation. The lower curve corresponds to the content-unaware policy, but the effect of playout is now shown in terms of playout cost

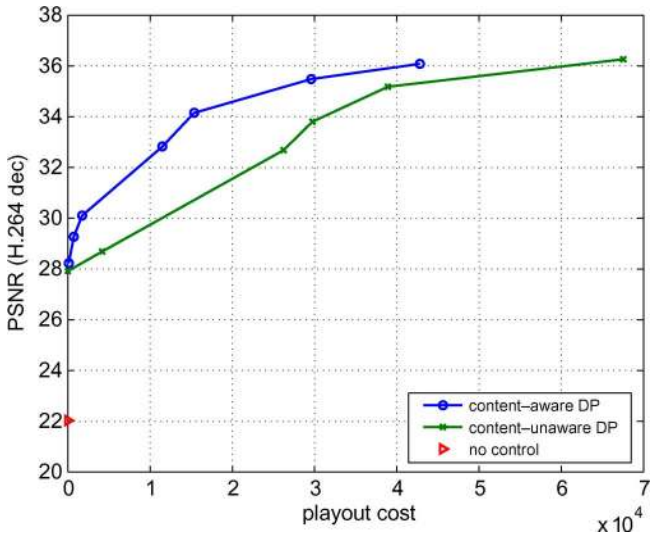


Fig. 5. Tradeoff between video quality (PSNR) and playout cost, for joint control with/without content-awareness. (DP stands for dynamic programming.)

$(C_s + C_v)$. Notice that the curves for the content-unaware policy in the two figures are based on exactly the same data, with the only difference that playout quality is expressed in terms of different metrics, namely playout duration increase in Fig. 5 as opposed to playout cost in Fig. 6. The similar shape of the same data in the two figures confirms that the playout variation cost as defined in this paper, captures the degradation in quality, as objectively captured by the increase in the total playout duration. The upper curve with the "o" markers corresponds to the content-aware policy. The second and more interesting observation from this figure is that the content-aware playout is higher and to the left, and therefore improves the distortion-playout cost tradeoff: e.g. for the same distortion, the playout variation has a smaller perceived effect, thanks to the intelligent selection of the preferred scenes for performing the slowdown. The video clips corresponding to the no-control (shown in Figs. 4 and 5 with a red triangle) and to the comparable joint control policy, can be found at [24].

Fig. 6 plots the same experimental results as Fig. 5, i.e., the tradeoff between picture quality and playout quality. However, we express picture quality in terms of distortion cost $(D = D_{tx} + D_{rx})$ as opposed to PSNR, and we express playout quality in terms of playout cost $(C_s + C_v)$ as before.

Fig. 7 examines the same tradeoff, for content-aware and content-unaware control policies, but for several channels with different characteristics. The average duration that each channel stays in the good state and bad state are set to be equal, and we consider different values for this duration. The goal is to examine performance, as a function of the channel variability or coherence time. In the previous figures, Figs. 4 and 5, this duration was set to be equal to three time-slots, i.e., 15 frame durations. We now vary this average duration from two slots to four slots, in order to examine the impact of channel variability (i.e., coherent time) on the performance. The first thing one can observe from Fig. 7 is that for larger average duration, the performance degrades, i.e., there is higher distortion for the same playout cost. This is because a longer bad channel period

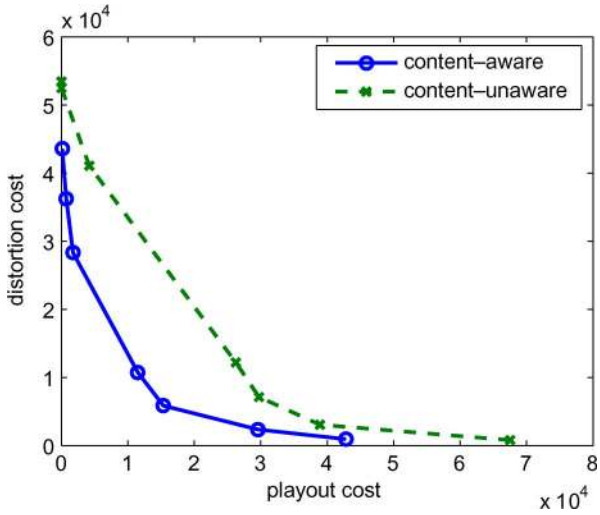


Fig. 6. Tradeoff between distortion cost and playout cost, for joint control with/without content-awareness. (Notice that this is the same tradeoff is an Fig. 5, but expressed in terms of distortion and playout cost, as opposed to PSNR and playout duration.)

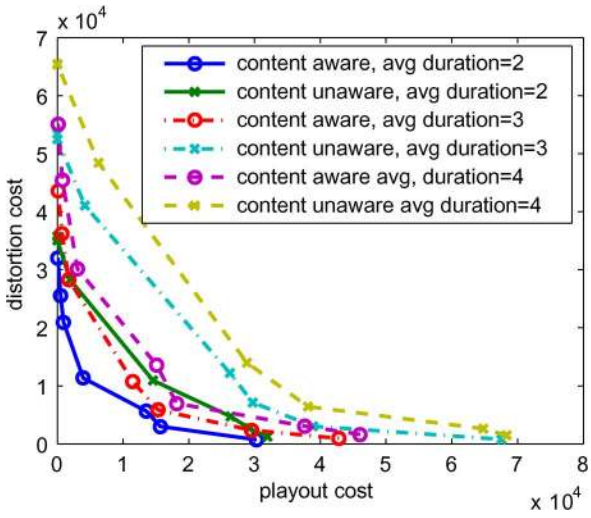


Fig. 7. Tradeoff between distortion and playout costs for three different channels. Every channel is a 2-state Markov chain with the same average duration of good an bad period (two, three, or four time slots). For each channel, we consider joint control with/without content-awareness.

is difficult to combat even using control policies; the duration of the good channel period is also longer but it does not stress the control policies. A second observation is that our content-aware policy outperforms the content-unaware policy, not only for the same channel, but also for all channels considered. Indeed, all three curves for the content-aware policies (indicated with round markers) are closer to the origin than any the content-unaware ones (indicates with "X" markers); the only exception is the content-unaware policy for the best channel (indicated in solid green with "X" marker), which is only slightly better than the content-aware policy for the worst channel (indicated in solid magenta with round markers).

Finally, we examined the sensitivity of the performance to the choice of T, the time period (in number of time slots), in which we adjust the playout rate. In all previous figures, T was set equal to ten frame durations or two time slots. Fig. 8 shows the

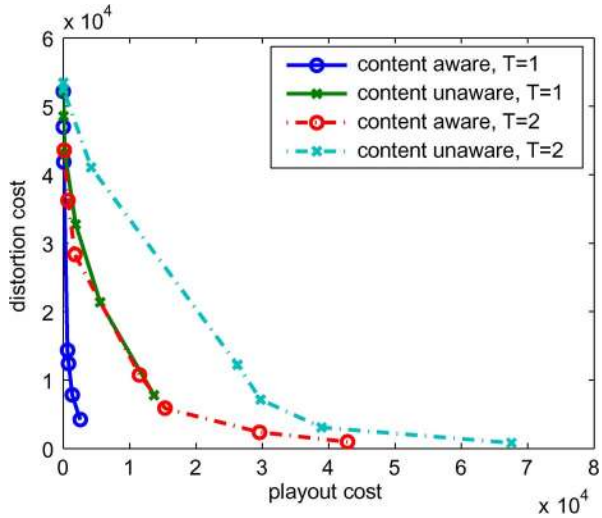


Fig. 8. Varying the period of rate adaptation, T.

performance of content-aware and content-unaware control for smaller values $T = 1, 2$. Our rationale for considering small T is to allow the playout rate to be adjusted as fast as the channel, and thus make it more responsive to channel variations. We can see that this modification improves the performance significantly.

C. Extensions of Our Framework

Defining cost functions that accurately capture the perceptual cost of adaptive playout depending on the video content is a difficult and open research problem. To the best of our knowledge, currently there are no such available perceptual metrics. This paper takes a systems approach: given appropriately defined cost metrics for content-aware playout, the goal is to determine the best control decisions, so as to have the minimum perceptual cost according to these metrics. In addition to the framework itself, we also took a first step towards defining playout cost metrics C_s and C_v that capture intuitive properties. In Section III-D3, we defined C_s as the amount of slowdown $(r - r_n)$, weighted by the amount of motion intensity in the scene M_s; we also defined C_v as the playout variation $|r_n - r|$ or the derivative of the playout rate. These metrics extend the ones proposed in [15], [19] by including the motion intensity. Intuitively, these costs increase with the amount of slowdown and the frequency of variations. The exact form of the C_s, C_v functions and the weight between the two, is a research topic in itself, orthogonal to this paper. The proposed framework can incorporate new and improved perceptual metrics as they become available.

We also defined a simple M_s function (as the sum of the absolute values of the motion vectors in a scene), which turned out to capture well the motion intensity of well known-scenes. For example, consider the scenes listed in Table I: the last two columns of this table show the (average and maximum) motion intensity of each scene. Also consider Fig. 3(a): it plots the motion intensity metric M_s for each frame in all scenes. A reader who is familiar with the these well-known scenes can verify that the M_s metric does capture the motion intensity of these scenes; e.g., scene 4 corresponds to the camera-pan in Foreman and has the highest motion intensity; scenes from Mother&Daughter and

Grandma have very low motion intensity. Therefore, at a first approximation, our heuristically defined M_s captures the motion intensity of the scenes and is thus appropriate as input to content-aware playout.

Once the metrics are defined, they can be used as input to the control problem. The purpose of the content-aware joint playout-scheduling is to distribute a certain amount of slowdown and packet drops across the sequence so as to have the least perceived effect. Fig. 3(b) and Fig. 3(c) show that our algorithms achieve this goal. Both figures correspond to the same realization of the wireless channel and the transmission of the same concatenated sequence, as described in Table I and Fig. 3(a), and for the same total distortion. Fig. 3(b) shows that the content-unware policy slows down the entire sequence, irrespectively of its content. Fig. 3(c) shows that the content-aware playout selectively slows down the parts of the sequence with less motion.

We also performed in-house, informal subjective tests to verify some intuitive assumptions made in this paper. We chose well-known scenes from Table I, with different motion intensity, and we slowed down each scene at the rates of 30 fps (original), 25 fps, 20 fps, and 15 fps. We then asked students at UC Irvine to rate the quality of these scenes in a scale from 1 to 5. The results confirmed our hypotheses that: 1) perceptual quality degrades with increasing slowdown and 2) the same amount of slowdown was found to affect less scenes with low MI. By talking to the subjects, we also realized that the motion intensity metric could be extended to also include higher layer information about the scene, such as global motion, detection of faces or objects, etc. Finally, we asked the same people to rate the longer sequence described in Table I and Fig. 3(a), with and without the proposed algorithms. We generated a wireless channel using the parameters of Section IV-A and used the same realization in both cases. This channel is stressing the system because of the long bad periods (lasting for more than one GoP) and the difference in transmission rate between the good and bad states. The three video clips for this experiment can be viewed online at [24]. Considering the original sequence as reference (5), we obtained very low ratings (average 1.5 and standard deviation 0.53) for the no-control policy and almost double ratings (average 3.13 and standard deviation 0.64) for the optimal joint control. This serves as a validation of the overall approach and not only of the content-aware part.

V. Conclusion

In this work, we formulated the problem of media streaming over a time-varying wireless channel as a stochastic control problem, and we analyzed the joint control of packet scheduling and content-aware playout. We showed that a small increase in playout duration can result in a significant increase in video quality. Furthermore, we proposed to take into account the characteristics, and in particular the motion intensity, of a video sequence in order to adapt the playout control based on the characteristics of each scene in the video sequence; this reduces the perceived effect of playout speed variation. Our proposed method can incorporate virtually any content-aware method which quantifies the perceived effect of changes to the playout speed. We jointly optimized packet scheduling at the medium access control, together with playout and content-awareness at the video application layer. Video content was taken into account both in playout (to selectively slow down scenes with the least perceived effect) and in packet scheduling (to selectively skip packets to meet the rate constraint at minimum distortion). The proposed framework can be used to characterize the underlying tradeoffs and also as a guideline for designing practical heuristics that mimic the optimal solution at lower complexity. It can also be extended to include other multimedia types, additional controls and wireless transmission systems.

References

[1] R. Berry and E. Yeh, "Cross-layer wireless resource allocation—Fundamental performance limits for wireless fading channels," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 59–68, Sep. 2004.

[2] D. Bertsekas, *Dynamic Programming and Optimal Control*. : Athena Scientific, 1995, vol. 2.

[3] J. Cabrera, A. Ortega, and J. I. Ronda, "Stochastic rate-control of video coders for wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 496–510, Jun. 2002.

[4] J. Chakareski, J. Apostolopoulos, S. Wee, W. Tan, and B. Girod, "Rate-distortion hint tracks for adaptive video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1257–1269, Oct. 2005.

[5] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[6] N. Färber and B. Girod, "Wireless Video," in *Compressed Video over Networks*, A. Reibman and M.-T. Sun, Eds. New York: Marcel Dekker, 2000.

[7] B. Girod, J. Chakareski, M. Kalman, Y. J. Liang, E. Setton, and R. Zhang, "Advances in network-adaptive video streaming," in *Proc. IWDC 2002*, Capri, Italy, Sept. 2002.

[8] *IEEE Standard for Information Technology—LAN/MAN—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) spec.*, IEEE Std 802.11, 1999 / 8802-11 (ISO/IEC 8802-11:1999), 1999.

[9] *Draft Amendment to Standard for Information Technology—Telecom. & Information Exchange Between Systems-LAN/MAN Specific Requirements—Part 11 Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements*, IEEE P802.11E (D13), 2005.

[10] H.264/MPEG-4 AVC Recommendation, ITU-T Video Coding and ISO/IEC Moving Picture Experts Groups.

[11] H.264/AVC Reference Software Version: JM 8.6 [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm

[12] J. Hartwell and A. Fapojuwo, "Modeling and characterization of frame loss process in IEEE 802.11 wireless local area networks," in *Proc. IEEE Vehicular Technology Conf. (VTC-Fall 2004)*, Los Angeles, CA, Sep. 26–29, 2004.

[13] M. Kalman and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines for low latency applications," in *Packet Video Workshop*, Irvine, CA, Dec. 2004.

[14] M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low delay video streaming over error-prone channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 841–851, Jun. 2004.

[15] M. Kalman, E. Steinbach, and B. Girod, "Rate-distortion optimized video streaming with adaptive playout," in *Proc. IEEE ICIP 2002*, Sep. 2002, vol. 3, pp. 189–192.

[16] S. Karande, S. Khayam, M. Krappel, and H. Radha, "Analysis and modeling of errors at the 802.11 b link layer," in *Proc. IEEE ICME 2003*, Baltimore, MD, Jun. 2003.

[17] Y. Li and N. Bambos, "Power-controlled streaming in interference-limited wireless networks," in *Proc. IEEE Broadband Networks*, San Jose, CA, Oct. 2004, pp. 560–568.

[18] Y. Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos, "Packet transmission and content-dependent playout variation for video streaming over wireless networks," in *Proc. IEEE MMSP 2005 (Special Session on Content Aware Video Coding and Transmission)*, Shanghai, China, Oct. 2005.

[19] Y. Li, A. Markopoulou, N. Bambos, and J. Apostolopoulos, "Joint power-playout control schemes for media streaming over wireless links," in *Proc. IEEE Packet Video*, Irvine, CA, Dec. 2004.

[20] Y. Li, A. Markopoulou, N. Bambos, and J. Apostolopoulos, "Joint power-playout control for media streaming over wireless links," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 830–843, Aug. 2006.
[21] Y. Liang, J. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: Does burst-length matter?," in *Proc. IEEE ICASSP-2003*, Hong Kong, Apr. 2003, vol. 5, pp. 684–687.
[22] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.
[23] Y. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling and loss concealment for voice communication over IP networks," *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 532–543, Dec. 2003.
[24] A. Markopoulou, Informal subjective tests at [Online]. Available: http://newport.eecs.uci.edu/~athina/publications.html
[25] J. McDougall and S. Miller, "Sensitivity of wireless network simulations to a two-state Markov model channel approximation," in *Proc. GLOBECOM 2003*, San Francisco, CA, Dec. 2003.
[26] S. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: Performance bounds and algorithms," *ACM/Springer Multimedia Syst.*, vol. 6, pp. 17–28, Jan. 1998.
[27] E. Setton and B. Girod, "Congestion-distortion optimized scheduling of video over a bottleneck link," in *Proc. IEEE Int. Workshop on Multimedia Signal Processing, MMSP 2004*, Siena, Italy, Sep. 2004.
[28] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming," *Wireless Commun. Mag.*, vol. 12, no. 4, pp. 59–65, Aug. 2005.
[29] S. Shankar, Z. Hu, and M. Van der Schaar, "Cross layer optimized transmission of wavelet video over IEEE 802.11 a/e WLANs," in *Proc. IEEE Packet Video 2004*, Irvine, CA, Dec. 2004.
[30] "Special issue on Advances in Wireless Video," *IEEE Wireless Commun. Mag.*, Aug. 2005.
[31] "Special issue on Multimedia over Broadband Wireless Networks," *IEEE Network Mag.*, Mar. 2006.
[32] G.-M. Su, Z. Han, M. Wu, and R. Liu, "Multiuser cross-layer resource allocation for video transmission over wireless networks," *IEEE Network Mag.*, Mar. 2006.
[33] D. Towsley, H. Schulzrinne, R. Ramjee, and J. Kurose, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *Proc. IEEE Infocom 1994*, Ontario, Canada, Jun. 1994, vol. 2, pp. 680–688.
[34] M. Van der Schaar and S. Shankar, "Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, pp. 50–58, Aug. 2005.
[35] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in *Proc. ICASSP '93*, Apr. 1993, vol. II, pp. 554–557.

Yan Li (SM'02) received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, in 2001 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford CA, in 2003 and 2006, respectively.

Since then, he has been with Texas Instruments, San Jose CA, and with Qualcomm, Campbell, CA. His research interests include supporting multimedia streaming over wireless networks and performance engineering in *ad hoc* networks.

Athina Markopoulou (SM'98–M'02) received the diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1996 and the M.S. and Ph.D. degrees, both in electrical engineering, from Stanford University, Stanford, CA, in 1998 and 2002, respectively.

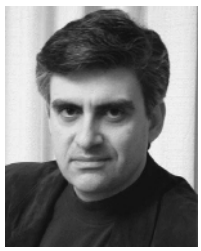
She has been a Postdoctoral Fellow at Sprint Advanced Technologies Labs (2003) and at Stanford University (2004–2005), and a Member of the Technical Staff at Arastra Inc. (2005). In 2006, she joined the Electrical Engineering and Computer Science Department at the University of California, Irvine, as an Assistant Professor. Her research interests include voice and video over IP networks, Internet denial-of-service, network measurement and control, and applications of network coding.

Dr. Markopoulou received the National Science Foundation CAREER Award in 2008.

John Apostolopoulos (S'91–M'97–SM'06–F'08) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA.

He joined Hewlett-Packard Laboratories in 1997, where he is currently a Distinguished Technologist and Lab Director for the Multimedia Communications and Networking Lab. He also teaches and conducts joint research at Stanford University, where he is a Consulting Associate Professor of electrical engineering. In graduate school, he worked on the U.S. Digital TV standard and received an Emmy Award Certificate for his contributions. His work on media transcoding in the middle of a network while preserving end-to-end security (secure transcoding) has recently been adopted by the JPEG-2000 Security (JPSEC) standard. His research interests include improving the reliability, fidelity, scalability, and security of media communication over wired and wireless packet networks.

Dr. Apostolopoulos received a Best Student Paper Award for part of his Ph.D. dissertation, the Young Investigator Award (Best Paper Award) at VCIP 2001 for his work on multiple description video coding and path diversity, was co-author for the Best Paper Award at ICME 2006 on authentication for streaming media, and was named "one of the world's top 100 young (under 35) innovators in science and technology" (TR100) by *Technology Review* in 2003. He currently serves as chair of the IEEE IMDSP and member of MMSP technical committees, and recently was General Co-Chair of VCIP'06 and Technical Co-Chair for ICIP'07.



Nick Bambos received the diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1984 and the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley in 1989.

He is a Professor of Electrical Engineering and Management Science at Stanford University, Stanford, CA, where he heads the Network Architecture and Performance Engineering Group and the Networking Research Lab. His current research interests are in wireless network architectures, high-speed switching, robust networking, queueing, and scheduling processes.

Dr. Bambos has been the Cisco Systems Faculty Scholar and the Morgenthaler Scholar at Stanford, has won the IBM Faculty Award, the National Science Foundation National Young Investigator Award, and the Research Initiation Award, among others.