

Dependent Packet Transmission Policies in Rate-Distortion Optimized Media Scheduling

Christophe De Vleeschouwer and Pascal Frossard, *Senior Member, IEEE*

Abstract—This paper addresses the problem of streaming packetized media over a lossy packet network, with sender-driven (re)transmissions and receiver acknowledgments. It extends the Markovian formulation of the rate-distortion optimized (RaDiO) streaming framework by allowing the transmission schedule of a media data unit to become contingent on the acknowledgements relative to other data units. Media decoding dependencies are generally considered in state-of-the-art rate-distortion optimized scheduling algorithms. However, the set of eligible packet schedules are restricted to independent streaming policies, where the transmission strategy envisioned for a data unit at future transmission opportunities only depends on its own acknowledgment, and not on the acknowledgments received for other data units. This paper questions the validity of this assumption in the design of rate-distortion optimal streaming solutions, and provides a first attempt in the formal derivation of the benefit offered by dependent policies. One of the main contributions of our paper is to propose a methodology that limits the search space of dependent policies to relevant dependencies that are likely to bring a rate-distortion benefit, in order to solve an optimization problem that is *a priori* computationally intractable. Extensive simulations validate the proposed approach that focuses on relevant dependencies between streaming policies. We further show that the benefit of dependent streaming policies is actually marginal in practical scenarios where the gain in distortion per unit of rate decreases along the media decoding dependency path. It represents the first demonstration that the common assumption of independent streaming policies is valid in many common streaming scenarios. However, experimental results also demonstrate significant benefits and encourage a careful investigation of dependent policies when the content is characterized by an increase of the benefit per transmission unit brought along the data unit dependency path.

I. INTRODUCTION

MEDIA streaming is getting quite a lot of attention from the research community, as it represents one of the most important components of Internet services. Media streams however present typical characteristics, like a certain tolerance to loss but quite strict timing constraints, which make their transmission challenging on channels with limited resources. This paper addresses the problem of streaming packetized media over a lossy packet network with sender-driven (re)transmission and receiver acknowledgments. We are interested in defining which

packets to select for (re)transmission, in order to minimize the end-to-end distortion of the streaming system.

Our work builds on the framework introduced in [1], which abstracts the different transmission scenarios associated to a group of interdependent media data units in terms of a finite alphabet of policies for sending a single data unit. The media units have different contributions to the decoded media quality; and the decoding dependencies between data units are represented by ancestor/descendant relationships [1], where a descendant can only be decoded upon correct reception of its ancestor(s). The framework further assumes that packets are delivered with a random delay, and that the network additionally loses packets at random. A rate-distortion optimization formulation permits to define at which transmission opportunity a data unit should be transmitted, or retransmitted in absence of acknowledgment. This framework has been very popular in recent studies in typical streaming media scenarios, e.g., [2]–[4], which certainly justifies a deeper analysis of its underlying assumptions, that are mainly twofold: the packet scheduling is restricted to independent streaming policies, and the channel is assumed to induce independent losses and delays.

This paper addresses and discusses the validity of the first assumption, and represents a first attempt in demonstrating formally the performance penalty that may result from the independent streaming policy assumption. The system in [1] considers ancestor/descendant relationships—and the relative importance of media units—in order to determine a set of streaming policies for a set of interdependent data units. It takes into account the current and expected status of other data units in defining the transmission schedules, and updates regularly the policies based on acknowledgement packets. However, the independent policy assumption considers that the (re)transmission strategy defined for a data unit is only adapted in response to the reception of its own acknowledgments, but it does not directly depend on the acknowledgments received for other data units. For example, it does not offer the possibility to postpone the transmission of a data unit until another particular data unit has been acknowledged. In this paper, we extend the policy space so that the transmission schedule defined for a given data unit at a given time becomes explicitly dependent on the acknowledgments received in the future for other data units, or equivalently on any future feedback about the status of the streaming session.

When the independent scheduling policy assumption is relaxed, the rate-distortion optimization problem rapidly becomes intractable. The space of all possible dependent policies grows exponentially in both the number of transmission opportunities and the number of dependent data units [5], [6]. One of the main contributions of our paper is to propose a methodology to limit

Manuscript received August 25, 2006; revised April 8, 2007. This work was supported by the Swiss NSF under Grant PP002-68737 and by the Belgian NSF. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Deepak S. Turaga.

C. De Vleeschouwer is with UCL-TELE, B-1348 Louvain-la-Neuve, Belgium (e-mail: devlees@tele.ucl.ac.be).

P. Frossard is with EPFL-FSTI-ITS-LTS4, CH-1015 Lausanne, Switzerland (e-mail: pascal.frossard@epfl.ch).

Digital Object Identifier 10.1109/TMM.2007.902869

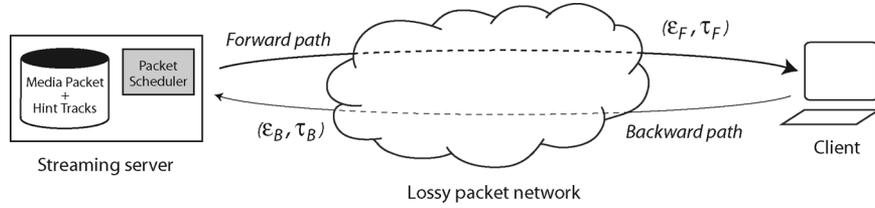


Fig. 1. Sender-driven adaptive streaming with client feedback. Media packets are sent from the streaming server to the client over a lossy packet network, whose channel paths are characterized by random delays τ_F and τ_B , and packet loss probabilities ε_F and ε_B .

the search space of the candidate dependent policies for rate-distortion optimized streaming, to policies that are likely to bring a benefit in a rate-distortion sense. To this aim, we first introduce the notion of master and slave data units to characterize dependent streaming policies between media packets. The policy adopted to transmit a slave data unit hence becomes dependent on the acknowledgments received for its masters. It is worth noting that the master/slave relationship is different from the ancestor/descendant relationship. The ancestor/descendant relationship as considered by most previous works in rate-distortion optimized streaming, refers to decoding dependencies, i.e., to the fact that a descendant can only be decoded upon correct reception of its ancestor(s). In contrast, the proposed master/slave relationship indicates that the transmission schedule assigned to the slave is a function of the instant at which the master is acknowledged, and is *a priori* independent of the ancestor/descendant relationship. A detailed analysis of the dependencies further allows to define rules for an appropriate classification of media packets into masters and slave data units. The master/slaves relationships are restricted to a computationally tractable subset that is built on the most relevant dependencies, and yet leads to near optimal rate-distortion performance.

While the comprehensive exploitation of the rich feedback information is expected to improve the performance of the adaptive streaming system, our analysis interestingly demonstrates that the gain is only marginal in most practical scenarios. In streaming systems where the importance of the data unit decreases along the dependency path, like in most of the conventional scalable image and video coding schemes, the use of dependent policies can be ruled out since the increase in computational complexity is not compensated by significantly better rate-distortion performance. However, when the hierarchical relations between dependent data units are reversed, i.e., when the distortion per unit of rate increases along the data units dependency path, we observe that the consideration of dependent streaming policies improves the performance. For example, we show that dependent policies can typically bring a gain of about 1 dB for conventional video content, where the benefit per transmission unit is likely to be (slightly) smaller for I frames than for their dependent P frames.

The paper is organized as follows. Section II briefly recalls the terminology and the methodology introduced in [1], and describes the suboptimality that may arise from the independency of streaming policies. Section III proposes to extend the space of streaming policies to a computationally tractable subspace of dependent policies, based on master/slave relationships. An algorithm is proposed to compute the RD optimal policies

respecting a given dependency pattern. We show in Section IV how the dependencies can be further restricted to relevant master/slave relationships (MSRs) that are expected to provide rate-distortion optimal streaming performance. Section V presents simulation results that demonstrate the benefit of the extension of the policy space to dependent policies.

II. RATE-DISTORTION OPTIMIZED STREAMING WITH INDEPENDENT POLICIES

A. Framework

We briefly review now the framework and the terminology introduced by Chou and Miao in [1], but we strictly limit this preliminary section to the concepts needed to describe the extension of this framework to dependent streaming policies. Interested readers are invited to refer to the original paper for a detailed discussion and motivation of the underlying assumptions. Readers that are familiar with [1] can certainly skip Sections II-A and II-B, but are invited to read Section II-C, since it motivates our work in great part.

Following the framework defined in [1], a media source is encoded and packetized into a finite set of data units that are stored on a media server. Regardless of the encoding and packetization algorithm, the interdependency between the data units can always be expressed by a direct acyclic graph. The acyclic graph induces a partial order relation among the data units. The relation is denoted \prec , and we write $l' \prec l$ when data unit l can only be correctly decoded if data unit l' has been decoded. We say that data unit $l'(l)$ is an ancestor (descendant) of data unit $l(l')$. Each data unit l is characterized by its size S_l in bytes, its decoder timestamp $t_{D,l}$, and its importance ΔD_l in units of distortion. The decoder timestamp is the delivery deadline, i.e., the time by which the data unit must be decoded to be useful.¹ The gain in distortion ΔD_l is the amount by which the distortion is decreased if data unit l is decoded, compared to the distortion if only the ancestors of l are decoded.

When the streaming server selects a data unit for transmission, the data unit is encapsulated into a packet and sent over the network. When retransmissions are possible, a data unit can be replicated in more than one packet, but we assume that a packet can contain only one single data unit. As in [1], the network forwarding path is modeled as an independent time-invariant packet erasure channel with random delays (see Fig. 1). It means

¹A data unit is thus useless when it arrives after its delivery deadline. Interested readers are referred to [7] for a formal description of how retroactive recovery mechanisms are combined with the RD optimized streaming framework proposed.

that a packet sent at time t can be either lost with probability ε_F , independent of t , or received at time t' , where the delay $\tau_F = t' - t$ is randomly drawn with probability density function p_F . Similarly, when an acknowledgment packet is sent from the client to the server through the backward channel, it is either lost with probability ε_B , or received after a delay τ_B , drawn with probability density function p_B . Each forward or backward packet is lost or delayed independently of other packets. For convenience, to combine the packet loss probability and the packet delay density into a single probability measure, we define a forward (backward) trip time random variable, denoted FTT (BTT), that is assigned to ∞ when the packet is lost, and is set to τ_F (τ_B) when the packet is not lost. The round trip time RTT is finally a random variable defined as the sum of FTT and BTT.

B. Independent Streaming Policies

Solutions proposed in the recent literature to define how and when to transmit a group of interdependent data units in a rate-distortion optimal way, formalize the problem as a finite horizon Markov decision process, and consider independent data unit policies. The independent policy assumption means that the transmission schedule of a data unit is not contingent on the future feedbacks received for other data units. It actually transforms the computationally intractable Markov decision process introduced in [6] into a controllable optimization problem.

We now give a brief overview of the methods that have been proposed to compute rate-distortion optimal policies, and interested readers are referred to [1] for more detailed information. The l^{th} data unit with delivery deadline $t_{D,l}$, is assigned N_l transmission opportunities at time $t_{l,0}, t_{l,1}, \dots, t_{l,N_l-1}$. A binary vector $\pi_l = (\pi_l(0), \pi_l(1), \dots, \pi_l(N_l - 1)) \in \{0, 1\}^{N_l}$ then defines the transmission instants of the l^{th} data unit, as the time instants $t_{l,i}$ such that $\pi_l(i) = 1$, if no acknowledgment has been received by that time. The transmission error probability $\varepsilon(\pi_l)$ for policy π_l is then defined as the probability that data unit l does not reach its destination before its delivery deadline $t_{D,l}$, as

$$\varepsilon(\pi_l) = \prod_{i:\pi_l(i)=1} P\{FTT > t_{D,l} - t_{l,i}\}. \quad (1)$$

The cost $\rho(\pi_l)$ for policy π_l further represents the expected number of transmissions for data unit l , and is given by

$$\rho(\pi_l) = \sum_{i:\pi_l(i)=1} \left(\prod_{j<i:\pi_l(j)=1} P\{RTT > t_{l,i} - t_{l,j}\} \right). \quad (2)$$

As a consequence of the assumption of independency between a policy and the potential acknowledgment of other data units, the transmission policy for a group of L interdependent data units can be described by a policy vector $\vec{\pi} = (\pi_1, \dots, \pi_L)$, where $\pi_l, l \in \{1, \dots, L\}$ is the transmission policy of the l^{th} data unit. Based on the notation above, the expected transmission rate and distortion for $\vec{\pi}$ are, respectively,

$$R(\vec{\pi}) = \sum_{l=1}^L \rho(\pi_l) S_l \quad (3)$$

$$D(\vec{\pi}) = D_0 - \sum_{l=1}^L \Delta D_l \prod_{l' \preceq l} (1 - \varepsilon(\pi_{l'})) \quad (4)$$

where D_0 denotes the distortion when no data unit has been received in time, S_l is the size of data unit l , and ΔD_l its importance. A policy vector $\vec{\pi}^*$ is said to be optimal if there exists no policy vector $\vec{\pi}$ such that $D(\vec{\pi}) \leq D(\vec{\pi}^*)$ and $R(\vec{\pi}) < R(\vec{\pi}^*)$. A simple iterative descent algorithm is proposed in [1], to find the policy vectors $\vec{\pi}$ that minimizes the Lagrangian cost function $J_\lambda(\vec{\pi}) = J_\lambda(\pi_1, \dots, \pi_L) = D(\vec{\pi}) + \lambda R(\vec{\pi})$, for $\lambda > 0$. Because $J(\vec{\pi}^{(k)})$ is nonincreasing and additionally bounded below by zero, convergence to a local optimum is guaranteed [1]. Alternatively, Roder *et al.* [8] have proposed a branch and bound algorithm to compute a global optimum to the choice of transmission policies, but with an increased computational complexity.

The above formalism has significantly advanced the state of the art in streaming media systems [2]. It has laid down the groundwork for recent studies on streaming media over multiple paths [3], from multiple servers [4], [9] or via intermediate proxy servers [10]. It has also been successful in handling different communication scenarios, including applications with severe delay constraints [11], [12] and streaming systems with rich client acknowledgments [13] or precise client requests [14], [15]. Moreover, the formalism proposed by Chou and Miao is in accordance with other works that have proposed to address the problem of scheduling media content over unreliable networks based on rate-distortion optimization techniques. Essentially, the authors in [16]–[18] also formalize the scheduling decision as a partially observable Markov decision process. Such a popularity certainly motivates a deeper analysis of the independent policy assumption, which is commonly used in rate-distortion optimized streaming.

C. Limitations of the Independent Streaming Policies

Most of previous works in optimized packet media streaming assumes independency between a data unit policy and the potential feedbacks received for other data units at future transmission opportunities. The assumption is for example implicit in [19], as the policy definition only consists in deciding whether a data unit is transmitted or not at the current transmission opportunity, without considering potential future retransmissions at all. It is explicit in [1], where the decision to transmit a data unit in the future only depends on its own acknowledgments. Yet, the correct decoding of a data unit is typically tied to the successful reception of another data unit. As a consequence, the correct reception of a packet should impact the strategy for sending a dependent data unit. To confirm this intuition, let us describe a simple example of data units i and j , with $i \prec j$. Since the decoding of data unit j is dependent on the correct reception of packet i , the transmission of a packet with data unit j should ideally be made dependent on acknowledgments relative to data unit i . Indeed, when the server receives the confirmation that a packet with data unit i has been correctly delivered, the expected benefit of sending the data unit j is modified. If this modification is significant, it can even influence the optimal streaming policy for data unit j at the server. Acknowledgments for other packets clearly participate to decrease the uncertainty about the system status, and should be considered by the scheduler in

order to adapt the strategy of streaming interdependent media data units.

In [1], the authors propose to recompute the RD optimal independent policies along the time, in order to take into account the most recent information from feedbacks on any of the data packets. Such a step-wise approach handles the dependency relationship *a posteriori* (i.e., after feedback reception) and improves the streaming performance, even if it stays suboptimal because it only modifies the streaming policies in reaction to acknowledgment. It does not permit the scheduler to consider dependencies between transmission policies, *a priori*, before actual reception of acknowledgments. For example, suboptimality may arise when a data unit i has a good chance to reach the receiver at some time (typically just after transmission), such that the scheduler decides to transmit a descendant j of i immediately. If the data unit i does not reach the client at the end, the data unit j becomes useless, and its transmission results in RD suboptimality. In this case, if the transmission deadlines allow it, a better strategy would have been to force j to wait for the acknowledgment of data unit i before transmission.

By considering dependent policies *a priori*, our work addresses these sources of suboptimality. Unfortunately, the scheduler can not consider all the possible dependency relationships in choosing the transmission policies, without rapidly facing an intractable optimization problem. As a main contribution, our paper demonstrates that only a few dependencies have to be investigated to define optimized streaming policies. In the next sections, we define these relevant dependency relationships as the ones that are likely to bring a significant rate-distortion gain compared to independent policies, and we identify the streaming scenarios where dependent policies are particularly helpful.

III. RATE-DISTORTION OPTIMIZED STREAMING WITH DEPENDENT POLICIES

We now formalize the notion of dependent transmission policies for interdependent data units. In particular, we investigate how the schedules defining the future transmissions of some data units (called slaves), may advantageously be forced to depend on the feedback potentially received for other data units (called masters) at future transmission opportunities. Section III-A first demonstrates that the dependent policies that are expected to provide a significant RD benefit compared to independent policies can be defined exclusively in terms of policies for which the master is only transmitted once, and the slave is only transmitted after reception of all its masters ACKs. Section III-B then defines how to compute RD optimized dependent policies corresponding to a predefined master/slave relationship.

A. Master and Slave Data Units

We now formally introduce the notion of *master* and *slave* data units, which are used to characterize the dependency relationships between the streaming policies of media packets.

Definition 1: A *slave* is a data unit whose transmission policy depends on the acknowledgment of other data units.

Definition 2: A *master* is a data unit for which an acknowledgment of correct reception can influence the transmission policy of other data units.

A master/slave relationship (MSR) is denoted $l \rightarrow l'$, when the reception of an acknowledgment for data unit l influences subsequent transmissions of data unit l' . In this case, we say that data unit l is a master of data unit l' , and that data unit l' is a slave for l . It is worth noting that the master/slave relationship is different from the ancestor/descendant relationship considered in most previous works about RD optimized streaming (e.g., [1]). The ancestor/descendant relationship refers to decoding dependencies, i.e., to the fact that a descendant can only be decoded upon correct reception of its ancestor(s). In contrast, the proposed master/slave relationship indicates that the transmission schedule assigned to the slave is a function of the instant at which the master is acknowledged. In general, any data unit can be a slave, a master, both of them, or none of them, independently of their classification as ancestor or descendant data unit. The purpose of adjusting the slave policy as a function of the arrival time of masters acknowledgments (ACKs) is to improve the rate/distortion tradeoffs, compared to strategies based on independent streaming policies only. Intuitively, the rate allocated to a slave data unit is smaller in absence of master ACKs, but the distortion potentially increases as a consequence of a relatively smaller number of transmissions. Hence, a proper tradeoff between rate reduction and distortion increase has to be defined.

Fortunately, the master/slave relationships are not all equally relevant in terms of improvement of the rate-distortion performance. We argue here that all the dependent policies that are likely to provide a significant benefit, can be defined exclusively in terms of MSRs for which i) the master is only transmitted once and ii) a slave is only transmitted after reception of all its masters ACKs. These conditions clearly allow to dramatically reduce the set of dependent transmission policies that are of interest in the RD sense. In order to demonstrate our conjecture about a unique transmission of master data unit, we proceed by contradiction. If the master data unit is transmitted several times, the probability that it reaches the client before its delivery deadline tends to one. In that case, there is little advantage for the slave scheduler to wait for the master feedback, as it could reasonably assume *a priori* that the master reaches the client in-time. When a master candidate is very likely to arrive in time to the client, a slave policy that does not depend on the master feedback is expected to achieve close to optimal rate-distortion performances. We conclude that a MSR is only expected to significantly improve on the performance obtained with independent streaming policies, when there is a good chance for the master data unit to be late or lost. Based on this observation, we conclude that the policy of a relevant master candidate triggers very few retransmissions. For the sake of simplicity, in the rest of the paper, we assume that a master data unit of interest is only transmitted once (i.e., without retransmission).

Our second assumption restricts the analysis of dependent policies to policies for which a slave is only considered for transmission upon reception of acknowledgments for all its master data units. This limitation is motivated by the fact that the cost

in rate of a streaming policy is dominated by its initial transmission, since retransmissions only happen in absence of acknowledgment. A significant gain in terms of rate is thus only expected for relationships that force to cancel the initial transmission of the slave data unit, in absence of master ACKs. This strongly simplifies the formalization of dependent policies. Specifically, only hard dependencies, for which the reception of all masters ACKs triggers a slave transmission, have to be considered. The study of softer dependency patterns, for which a slave progressively adapts a nonzero transmission policy as a function of the status of masters ACKs, fortunately becomes irrelevant in that case.

B. Rate-Distortion Optimal Dependent Policies

This section describes the selection of rate-distortion optimal dependent scheduling policies, conforming to a given set of master/slave relationships (MSRs). The search is limited to potentially advantageous policies that (i) transmit masters only once, and (ii) only transmit slaves upon reception of all its masters ACKs, as discussed before. The proposed solution is based on an iterative gradient descent algorithm that generalizes the approach proposed in [1], and recalled in Section II-B.

Let again $\{t_{l,0}, t_{l,1}, \dots, t_{l,N_l-1}\}$ and $t_{D,l}$, respectively, denote the N_l transmission opportunities and the delivery deadline $t_{D,l}$ assigned to the l^{th} data unit. In addition, we introduce some terminology that is specific to the dependent policy case. Let first Γ_l denote the set of masters for the l^{th} data unit. It means that the l^{th} data unit can only be transmitted after all data units m , with $m \in \Gamma_l$, have been acknowledged. As the transmission policy of data unit l is dependent on the reception of acknowledgments for its master data units, we define the subpolicy vector $\pi_{j,l} \in \{0, 1\}^j$ (with $j \in \{1, \dots, N_l\}$), as the transmission policy for the l^{th} data unit when j transmission opportunities remain available after all masters of l have been acknowledged. Specifically, the policy $\pi_{j,l}$ becomes effective if the latest acknowledgment for master data units in Γ_l is received in the time interval $[t_{l,N_l-1-j}, t_{l,N_l-j}]$. In this case, the l^{th} data unit has to be sent at opportunity $(i + N_l - j)$ if $\pi_{j,l}(i) = 1$, $0 \leq i < j$, and if it has not been acknowledged yet. The streaming policy $\pi_l^{\Gamma_l} = \{\pi_{j,l}\}$ for data unit l is then represented as a group of N_l subpolicy vectors, contingent on the reception of acknowledgments for the master data units. Finally, $\vec{\pi}_\Gamma = \{\pi_1^{\Gamma_1}, \dots, \pi_L^{\Gamma_L}\}$, with $\Gamma = \{\Gamma_1, \dots, \Gamma_L\}$, represent the policy vector for the group of L interdependent data units, and gathers the subpolicy vectors of all L data units.

In order to compute the expected rate $R(\vec{\pi}_\Gamma)$ induced by the policy $\vec{\pi}_\Gamma$, we have to determine the probability for each of the subpolicies to become effective. We therefore define $p_l(j)$ as the probability that j transmission opportunities are available for the l^{th} data unit after all data in Γ_l have been acknowledged. It is worth noting that the set of $p_l(j)$ (with $l \in \{1, \dots, L\}$ and $j \in \{1, \dots, N_l\}$) only depends on master data units, and not on the transmission policies of nonmaster data units. As a consequence, given the set of master/slave relationships defined for the group of L interdependent data units, and based on the assumption that master data units are transmitted only once, the probabilities $p_l(j)$ can be precomputed as a function of the RTT distribution and the MSRs defined within Γ_l (interested readers

are referred to in [20, App. D] for details of that computation). Hence, using the definition of expected transmission cost $\rho(\cdot)$ (see Section II-B), the expected rate can be written as

$$R(\vec{\pi}_\Gamma) = \sum_{l=1}^L \sum_{j=1}^{N_l} p_l(j) \rho(\pi_{j,l}) S_l \quad (5)$$

where S_l is the size of the data unit l .

Next, in order to define the distortion $D(\vec{\pi}_\Gamma)$ expected for $\vec{\pi}_\Gamma$, we define the random vector ψ , such that $\psi(l)$ (with $0 < l \leq L$) represents the number of transmission opportunities still available for the l^{th} data unit after all data in Γ_l have been acknowledged. We further denote Ψ as the set of all possible realizations of ψ . Similar to the probabilities $p_l(j)$, the probability of occurrence of $\psi \in \Psi$, p_ψ , is precomputed as a function of the predefined MSRs, and can therefore be considered as a parameter that is independent of the transmission policy assigned to nonmaster units. We have thus

$$D(\vec{\pi}_\Gamma) = D_0 - \sum_{\psi \in \Psi} p(\psi) \sum_{l=1}^L \Delta D_l \prod_{l' \leq l} (1 - \epsilon(\pi_{\psi(l'), l'})) \quad (6)$$

Based on these definitions, we can now extend the computation of rate-distortion optimal convex-hull policies proposed in [1], to consider a predefined set of MSRs. The purpose is still to compute the policy vectors $\vec{\pi}_\Gamma$ minimizing $J_\lambda(\vec{\pi}_\Gamma) = D(\vec{\pi}_\Gamma) + \lambda R(\vec{\pi}_\Gamma)$ for $\lambda > 0$. Similarly to the ISA algorithm [1] that minimizes one policy at a time, keeping the other ones fixed, we propose to proceed iteratively until a minimal cost can be reached. However, in contrast to the case of independent policies, here the algorithm has to minimize every nonmaster subpolicy, keeping the other fixed. The sequence of policy vectors $\vec{\pi}_\Gamma^{(k)}$ is computed as follows. First select $l_k \in \{1, \dots, L\}$ and $j_k \in \{1, \dots, N_{l_k}\}$. Then, $\forall (l, j) \neq (l_k, j_k)$, set $\pi_{j,l}^{(k)} = \pi_{j,l}^{(k-1)}$, and let

$$\pi_{j_k, l_k}^{(k)} = \arg \min_{\pi} J_\lambda \left(\pi_{1,1}^{(k)}, \dots, \pi_{j_k-1, l_k}^{(k)}, \pi, \right. \\ \left. \pi_{j_k+1, l_k}^{(k)}, \dots, \pi_{N_{l_k}, l_k}^{(k)} \right) \quad (7)$$

$$= \arg \min_{\pi} G_{j_k, l_k}^{(k)}(\epsilon(\pi) + \lambda p_{l_k}(j_k) S_{l_k} \rho(\pi)) \quad (8)$$

where $\pi \in \{0, 1\}^{j_k}$ and

$$G_{j_k, l_k}^{(k)} = \sum_{\psi \in \Psi: \psi(l_k) = j_k} p(\psi) \sum_{l_k \leq l'} \Delta D_{l'} \\ \times \prod_{l'' \leq l', l'' \neq l_k} (1 - \epsilon(\pi_{\psi(l''), l''}^{(k)})) \quad (9)$$

In practice, the l_k 's are selected among the nonmaster data units. Initial policies are set to a always-send policy, and the l_k indexes are selected in a round-robin order that scan ancestors first. For each l_k , the j_k indexes are selected in increasing order. For the same reasons as for the independent case, convergence to the a local optimum satisfying the predefined set of MSRs is guaranteed. From a practical point of view, we demonstrate in Section IV that, for MSRs that are worth to be studied, the masters associated to a data unit are also masters of its descendants. Such a feature reduces the cardinality of Ψ (because the slaves

share common masters) and simplifies the computation of the probability $p(\psi)$. Finally, it is worth noting that (5), (6) and (8) only hold because the policies associated to masters are fixed to a single transmission, per definition. Otherwise, $p_l(j)$ and consequently $p(\psi)$ would depend on master policies, which would strongly tie the master and slave policies, resulting in a computationally intractable problem.

IV. RELEVANT MASTER/SLAVE RELATIONSHIPS

A. Preliminaries

In Section III, we have extended the formulation of the rate-distortion optimal packet scheduling problem, and we have explained how the optimal dependent streaming policies can be computed, given a predefined set of master/slave relationships (MSRs). However, exploring all possible MSRs to select the one that achieves the best RD tradeoff remains computationally intractable because the number of MSRs grows exponentially with the number L of data units. Specifically, there are 2^L possible choices of masters among L interdependent data units, and for a given choice of M masters among the L data units, there are $2^{(L-1)M}$ possible definitions of MSRs, corresponding to the $2^{(L-1)}$ allocation of slaves to each of the M masters. This is far too large to envision an exhaustive search among the entire MSRs space. Hence, we restrict our search to those slave/master relationships that are likely to bring a benefit in the RD sense, in comparison with a scheduling strategy based on independent policies. Such MSRs are called *relevant* MSRs, and our objective in this section becomes to define the smallest complete set of relevant MSRs, as a small subset of the valid relationships defined in the previous section. We show that the relevant MSRs are tightly connected to the ancestor/descendant relationships defined among data units, and we propose a methodology that assigns at most $L \times O((L/B)^B)$ relevant MSRs to a group of L interdependent data units characterized by an acyclic dependency graph composed of B disjoint branches. The result is obtained in two steps. First, we assume that the set of master data units is defined *a priori*, and we study how slaves are assigned to these masters. Second, we consider the master selection problem, and propose a greedy algorithm to define a sequence of relevant sets of master data units.

B. Assignment of Slaves to a Predefined Set of Masters

In order to assign a collection of relevant MSRs to a predefined set of master data units, we propose to set a (ϵ, ρ) *non-degradation assumption*, which considers that the (ϵ, ρ) tradeoffs defined for slave data units are not significantly affected by the wait for master ACKs. This assumption is made because it is difficult to apprehend the exact impact of the degradation of the cost/benefit tradeoff (ϵ, ρ) caused by the wait for master feedback (see Section II-B). Basically, the nondegradation assumption considers that everything happens as if the feedback is either lost or received instantaneously. This is obviously not valid in general. However, the assumption is only made to define the relevant MSRs, not to compute the RD optimal dependent policies corresponding to these MSRs. Once the relevant

MSRs have been defined, the corresponding RD optimal dependent policies are computed based on (8) and (9), i.e., by taking the waiting time for master feedback into account.

In this context, we motivate the nondegradation assumption as follows. First, when the time to wait for feedback is small in comparison with the time available before expiration of the slave delivery deadline, the assumption is certainly valid. Alternatively, when the time to wait for a feedback is so large that it causes significant degradation of the slave (ϵ, ρ) tradeoffs, it is very likely that a policy that transmits the slave independently of its master, results in optimal RD tradeoffs. Hence, we can conclude that the relevant MSRs can be defined based on the nondegradation assumption. This stays valid as long as slavery ignores the data units for which the time wasted to wait for the master feedback penalizes too much their (ϵ, ρ) tradeoffs. In other words, given a master or a cascade of multiple masters, some data units might become *ineligible* to be a slave, because the time wasted to wait for the master feedback penalizes too much their (ϵ, ρ) tradeoffs. The nondegradation assumption stays valid for eligible slaves, which are involved in a given set of MSRs.

Based on the above arguments about the nondegradation assumption, we propose to define relevant master/slave relationships in two steps. In a first step, a set of relevant MSRs are defined based on the nondegradation assumption. In a second step, the slave eligibility is checked a posteriori, by enfranchising the ineligible slaves identified within relevant MSRs, in order to further refine the set of valid dependencies.

1) *Necessary Rules for Relevant MSRs*: We now identify three necessary rules imposed by ancestor/descendant dependencies on the definition of relevant MSRs, under the nondegradation assumption. These rules are then exploited to define the relevant master-slave configurations associated to a predefined set of masters. Eventually, enfranchisement of potential ineligible slaves is further discussed.

Rule 1: Slave descendants are slaves themselves.

The first rule for relevant MSRs definition simply states that if slaves have descendants according to the acyclic dependency graph that characterizes the encoded media streaming, then these descendants are slave data units also. Indeed, let m denote the index of a master, and s denote the index of a slave for m . To figure out how the $m \rightarrow s$ relation affects the descendants of s , we neglect the delay induced by waiting for the master feedback (i.e., nondegradation assumption). In that case, we can show that $m \rightarrow s$ implies $m \rightarrow j$ for all $s \preceq j$, i.e., for all j that are descendants of s . By definition, a descendant of s can only be decoded if s reaches the client before its delivery deadline. Obviously, this only happens when s is transmitted; as a consequence of $m \rightarrow s$, this is only the case when the feedback for the m^{th} data unit has been received. Overall, it means that the descendants of s can not be decoded if the feedback for m has not been received. For this reason, there is no advantage for a descendant of s to be transmitted when the feedback for m is not available. The descendant of s therefore becomes a slave for m . Note that this does not prevent s or its descendants to become a master data unit in other MSRs.

Rule 2: Masters are ancestors of slaves.

The second rule states that in practical settings, master data units in relevant MSRs are also ancestors of their slaves. It is easier

to support that statement by contradiction. We therefore consider the case in which an ancestor data unit a becomes a slave for one of its descendants d . In this case, it can first be proven that all rate-distortion (R,D) points respecting the $d \rightarrow a$ MSR lie above the (R,D) lower convex hull computed for independent transmission policies. In [20], a formal analysis demonstrates that the MSR $d \rightarrow a$ can only become beneficial in the RD sense (without lying on the convex-hull) when the descendant brings a large gain in distortion with a relatively small cost in rate. Intuitively, it can be explained by the fact that a significant fraction of the gain in distortion expected by the ancestor, is subject to the availability of its descendants. As a consequence, the scheduler might find a rate-distortion benefit in sending out the ancestor only when the descendant has been acknowledged. This means that a master-slave relation where a master data unit is a descendant of its slave, can only be beneficial for cases where the cost S_l decreases and the gain in distortion ΔD_l significantly increases along the path of descentance. However, this kind of scenario is very rarely encountered in practice, because efficient media coders encode first the most important information. Moreover, when streaming a sequence of groups of interdependent data units, dependent policies for which the ancestor transmission is subject to the descendant feedbacks can only achieve a beneficial RD tradeoff by sacrificing some ancestor samples and consequently all their respective descendants, in order to give other ancestor samples a chance to be transmitted. Such an allocation of transmission resources results in dramatic fluctuations of the quality at the client end, and should not be recommended. As a consequence, in realistic media streaming conditions, we restrict our study to dependent policies for which masters are also ancestors of their slaves.

Rule 3: Master candidates have the same slaves among their common descendants.

The third rule says that a data unit s should be a slave either for all, or none of its master candidates. By master candidate, we mean a data unit that is only transmitted once, and that is an ancestor of s , according to the previous discussion. To demonstrate it, let m and m' denote two master candidates. It can be shown that, if $m \rightarrow s$ is beneficial in the RD sense, then $m' \rightarrow s$ is also beneficial. In other words, if the gain in rate is worth the loss in distortion for $m \rightarrow s$, then the overall rate-distortion balance is also beneficial when forcing the $m' \rightarrow s$ relation. Equivalently, if the wait for an ACK to data unit m is beneficial, then waiting for m' to be acknowledged also brings an advantage in the RD sense. This is due to the fact that, under the nondegradation assumption, m and m' both constrain s in the same way. As they are both ancestors of s , their reception is required to decode s . As they are both master candidates, they are only transmitted once, and have about the same chance to trigger an ACK. Interested readers are referred to [20, App. B] for a formal demonstration. We note that for nonmonotonic evolution of the $\Delta D_l/S_l$ ratio along the path of descentance, the above statement is only strictly valid when m is a descendant of m' . For the sake of simplicity, we nevertheless omit this refinement, and admit that all RD optimal convex-hull points can be computed by considering that a descendant of multiple master candidates is either a slave for all of them, or is transmitted independently of all of them.

2) *Assignment of Slaves to Masters and Slave Eligibility Check:* The rules 1, 2, and 3, provide the toolbox for the defini-

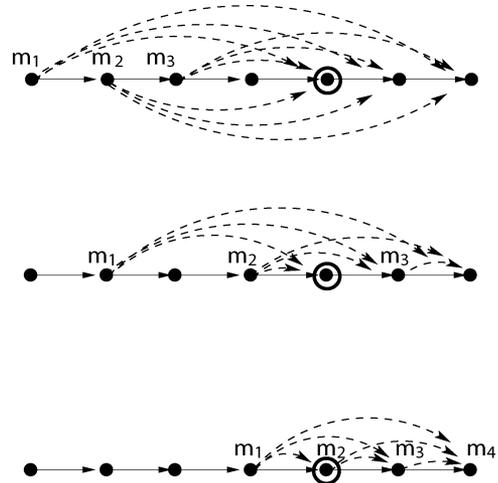


Fig. 2. Three examples of relevant MSR definitions for a branch of 7 interdependent data units. Each example considers distinct *a priori* selections of master candidates m . For each case, the set of relevant MSR is defined based on the choice of the oldest slave, and follows the building properties explained in the text. Specifically, all descendants of the oldest slave are slaves themselves, and all master candidates that are ancestors of a slave are masters of that slave. Here, the 5th data unit is chosen to be the oldest slave (= circle in the figures) and the corresponding relevant MSR is represented by a set of dashed arrows.

tion of relevant MSR for a given predefined set of M masters, denoted as $\{m_0, \dots, m_{M-1}\}$. Relevant MSR are assigned to these masters based on a sequential scan of the acyclic graph branches, which describe the data units dependencies. Each branch connects a root of the acyclic graph to one of its leaves, and is scanned in the ancestor-descendant order. The order in which branches are considered is chosen arbitrarily and does not affect the outcome of the algorithm. The MSR assignment process can be formally described as follows. Let Φ_i denote the set of data units that belong to the i^{th} branch. In each branch, data units are ordered in increasing order of dependency, i.e., data unit k in Φ_i is a descendant of all data units $j < k$ in Φ_i . A set of relevant MSR associated to a branch is completely defined by the index of the oldest slave, s , as depicted in Fig. 2. Based on rule 1, all descendants of s are slaves themselves. Moreover, rules 2 and 3 state that any given slave is a slave for all older masters, but is independent of younger masters. As a consequence, there are at most $Card(\Phi_i)$ relevant MSR for the i^{th} branch.

In practice, the branches extracted from the acyclic graph are not necessarily disjoint, so that MSR to consider for a branch Φ_i might be constrained by the MSR already defined for branches Φ_j , $j < i$. Specifically, data units that are common to Φ_i and Φ_j and that have been defined as being slaves in Φ_j should also be slaves in Φ_i . That simply reduces the number of relevant MSR to investigate. Fig. 3 illustrates the computation of the number of relevant MSR configurations for a complex acyclic dependency graph. It is easy to derive that the number of configurations to investigate for L data units characterized by an acyclic graph with B branches is upper bounded by $(L/B + 1)^B$, which remains computationally tractable for realistic media content.

Once the relevant MSR have been defined based on rules 1, 2, and 3, they are then checked *a posteriori*, in order to ensure

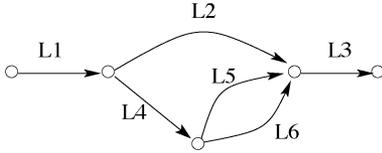


Fig. 3. Example of acyclic dependency graph, and computation of the number of slave assignment possibilities associated to that graph. Labels L_k on each link of the graph refer to the number of data units involved in the corresponding link. The graph contains 3 branches, i.e., it offers 3 different paths to connect the root to the leaf. For that example, the total number of slave assignments considered by our algorithm is equal to the number of possible sets formed with the oldest slaves, i.e., to $L_1 + L_2 \times (L_4 + L_5 \times L_6) + L_3$.

that the slave eligibility condition is not violated. This second step in the definition of relevant MSR is made necessary by the approximations introduced by the nondegradation assumption, used in rules 1 to 3. The eligibility question becomes mostly relevant when the time lost in waiting for master ACKs can not be neglected. In particular, it refers to the loss arising when the wait for masters ACKs strongly penalizes the transmission of some data units. For the sake of simplicity, we consider the case of a single master, with multiple slave candidates. A slave candidate is said to be problematic when the wait for its master ACK strongly penalizes its (ϵ, ρ) tradeoff. From a rate distortion point of view, it might certainly be advantageous to transmit the problematic slaves independently of the master ACK. However, due to Rule 1, when a problematic data unit is transmitted independently of its master, all its ancestors should also be transmitted independently of the master. This might be suboptimal if these ancestors are not problematic but rather have an advantage of being slaves of the master. To circumvent the problem, a solution consists in enfranchising, a posteriori, the problematic slaves identified in relevant MSR, so as to define and explore additional and potentially advantageous MSR. However, based on the single master example, we understand that the enfranchisement of a descendant slave s is only likely to bring a significant RD benefit (compared to relevant MSR and to independent transmission policies) when the media stream is characterized by delivery deadlines that significantly decrease along the dependency path, and by gains in quality that increase beyond the enfranchised slave. In such a case, ancestors of problematic slaves are not problematic themselves, and it is more advantageous to transmit the problematic data unit than retransmitting a master candidate. Therefore, it may be worth transmitting s independently since it brings a large gain in distortion (directly or indirectly through its descendants), rather than taking the risk of not transmitting it because of the wait for a master ACK. Such a situation is however rare in practice, where delivery deadlines most often come close to each others, and importance decreases along the dependency path. Therefore, we do not provide here a more detailed development of the eligibility problem, and we rather refer the interested reader to [20, App. C] for a deeper analysis.

C. Selection of Masters

This section now considers the master selection problem, in the definition of relevant MSR. A *relevant set of masters* (RSM) denotes a subset of data units that are expected to improve the

streaming RD performance in becoming masters. Starting from an initial empty subset $\Omega_0 = \{\}$, the sequence of RSMs Ω_k chosen from the set Λ of interdependent data units, is computed as follows. At each step k , we select the data unit m_k in $\Lambda \setminus \Omega_{k-1}$ whose role as a master minimizes the expected ratio between the increase in distortion and the gain in rate. Ω_k is then set to $\Omega_{k-1} \cup \{m_k\}$. In the rest of the section, we motivate the iterative approach and explain how to implement it in practice.

The incremental nature of the approach followed to define the RSMs is justified as follows. When the bits are cheap, there is little advantage to introduce master/slave relationships among data units to improve the rate-distortion tradeoff. On the contrary, as bits are becoming more expensive, more data units are likely to be transmitted a single time, and to bring a benefit in becoming masters. This is because their slaves are only transmitted upon reception of master ACKs, which saves some bit budget. Based on this reasoning, if a master is relevant for a given cost of bit, it remains relevant when bits become more expensive. Hence, data units can progressively be labeled as masters as the cost of bits increases, which ends up in an incremental definition of RSMs.

In practice, the iterative master selection process works as follows. We consider first the selection of the master m_1 . Let $\Delta D(i, s)$ and $\Delta R(i, s)$, respectively, denote the expected increase in distortion and decrease in rate when data unit i is forced to become a master for s and its descendants. In the Lagrangian formalism introduced in Section II-B, for a given factor λ , there is an advantage in assigning i to be a master if and only if $\Delta D(i, s) < \lambda \Delta R(i, s)$. In other words, assigning i to be a master is beneficial for all λ values larger than $\Delta D(i, s) / \Delta R(i, s)$. We are interested in the data unit for which the master assignment becomes beneficial at the smallest λ value. Formally, we have

$$m_1 = \arg \min_{i \in \Lambda} \left(\min_{s > i} \frac{\Delta D(i, s)}{\Delta R(i, s)} \right). \quad (10)$$

We now explain how $\Delta D(i, s)$ and $\Delta R(i, s)$ are estimated. Since independent policies are particularly interesting when bits are cheap, we make the coarse assumption that the RD optimal independent policies perform enough retransmissions to ensure correct delivery of all data units. To estimate the corresponding rate, we remember that the capacity of an erasure channel with probability ϵ is $(1 - \epsilon)$. As a consequence, an ideal transmission system needs an average of $\zeta = 1/(1 - \epsilon_F)$ channel packets to convey a data unit to the client, with ϵ_F denoting the probability of loss on the forward path. In contrast, when s and its descendants are slaves of master i , the master i is only transmitted once (see Section IV), while s and its descendants are only transmitted upon reception of an ACK for i . As a consequence, i has a probability lower than $(1 - \epsilon_F)$ to reach the client, whilst s and its descendants have a probability lower than $(1 - \epsilon_F)(1 - \epsilon_B)$ to be transmitted, where ϵ_B denotes the probability of loss on the backward path. Based on the above developments, we approximate $\Delta D(i, s)$ and $\Delta R(i, s)$ as follows:

$$\Delta R(i, s) \sim (\zeta - 1)S_i + (1 - (1 - \epsilon_F)(1 - \epsilon_B)) \zeta \sum_{k \geq s} S_k \quad (11)$$

and

$$\Delta D(i, s) \sim \epsilon_F \Delta D_i + (1 - (1 - \epsilon_F)(1 - \epsilon_B)) \sum_{k \geq s} \Delta D_k. \quad (12)$$

Next, we consider the possibility to define RSMs with more than one master, under the assumption that the increase in distortion and decrease in rate resulting from master assignments is additive. This assumption is coarse, yet acceptable as long as our objective is to select a set of promising master data units, rather than to compute the optimal policies. It significantly simplifies the RSMs definition, by decoupling the impact of multiple masters, both in terms of distortion and rate. As a consequence, masters can simply be selected in increasing order of the expected increase in distortion per unit of rate

$$m_k = \arg \min_{i \in \Lambda \setminus \Omega_{k-1}} \left(\min_{s > i} \frac{\Delta D(i, s)}{\Delta R(i, s)} \right). \quad (13)$$

Note that the methodology proposed above is validated later in Section V-B2. For streaming scenarios where a comprehensive search among all possible dependent policies is computationally tractable, the sets of masters that end up in optimal dependent policies, indeed correspond to the RSMs defined based on (13).

D. Summary

We now summarize the previous developments for the selection of relevant MSR. The search for the set of (in)dependent policies that offer an optimized rate-distortion scheduling strategy, is represented in Algorithm 1.

Algorithm 1 Search for optimal policies

Initialization: $\Omega_0 = \{\}$, $k = 0$, L is the number of interdependent data units.

Best policy \leftarrow no transmission at all.

while $k < L$ **do**

for all relevant MSR. defined w.r.t. the Ω_k RSMs **do**

 compute the optimal (convex-hull) policies using the algorithms proposed in Section III-B.

end for

if the optimal (convex-hull) policy computed for a set of relevant MSR. outperforms the best policy **then**

 replace the best policy by the newly computed optimal policy.

end if

 Select m_{k+1} as proposed in Section IV-C

$\Omega_{k+1} \leftarrow \Omega_k \cup \{m_{k+1}\}$

$k \leftarrow k + 1$

end while.

We can make the following observations about the iterative process defined by Algorithm 1:

- The initial set of masters is empty. It means that the policies computed for Ω_0 are the independent policies.
- As explained in Section IV-B2, a set of relevant MSR. associated to a predefined set of masters is completely defined by selecting the index of the oldest slave in every branch of the acyclic dependency graph. The total number of MSR. configurations depends on the acyclic dependency graph, but remains computationally tractable. In contrast, the full search through the entire space of streaming policy becomes computationally intractable when the number of interdependent data units becomes larger than two [6].
- The iterative process stops when all data units belong to Ω_k and become master data units, i.e., when $k = L$.

The expected benefit of dependent streaming policies is clearly dependent on the characteristics of the media data units. First, when the decrease in distortion per transmission unit decreases along the stream dependency path, we conjecture that there is not much gain to expect from dependent policies, with respect to the classical RaDiO approach studied in [1]. In this case, it is more efficient in a RD sense to define several (re)transmissions of an ancestor, rather than to give a master role to this ancestor. Indeed, when the server has a good confidence that the ancestor has been lost (assuming the backward channel is reliable), the ancestor retransmission is likely to bring a larger benefit per unit of cost than a descendant transmission (assuming respective deadlines are very close). Second, when, on the contrary, the gain in distortion per transmission units increases along the dependency path, the benefit drawn from dependent policies becomes explicit. In that case, masters are first selected among the oldest data units (based on Rule 3), and ancestors are thus likely to be transmitted less frequently than their descendants. This sounds counter-intuitive, but may be advantageous in a rate-distortion sense because the transmission resources are primarily allocated to the more beneficial data units, i.e., the descendants. In the same time, it avoids the situation where transmitted descendants cannot be decoded due to the absence of some ancestors, as it is the case with independent streaming policies.

Note that we do not claim all scheduling policies that achieve an optimal RD tradeoff do satisfy the features defined for relevant MSR. However, the above arguments, supported by the developments in [20], demonstrate that the set of relevant MSR. includes most of the dependent policies that are likely to offer a significant benefit in comparison to the set of independent transmission policies. Therefore, we can reasonably assume that a scheduling policy that does not fulfill the above rules does not significantly outperform policies based on relevant MSR. This assumption is further confirmed by the results presented in Section V.

V. SIMULATION RESULTS

A. Overview

This section evaluates the benefit of dependent streaming policies with respect to the independent streaming of data units

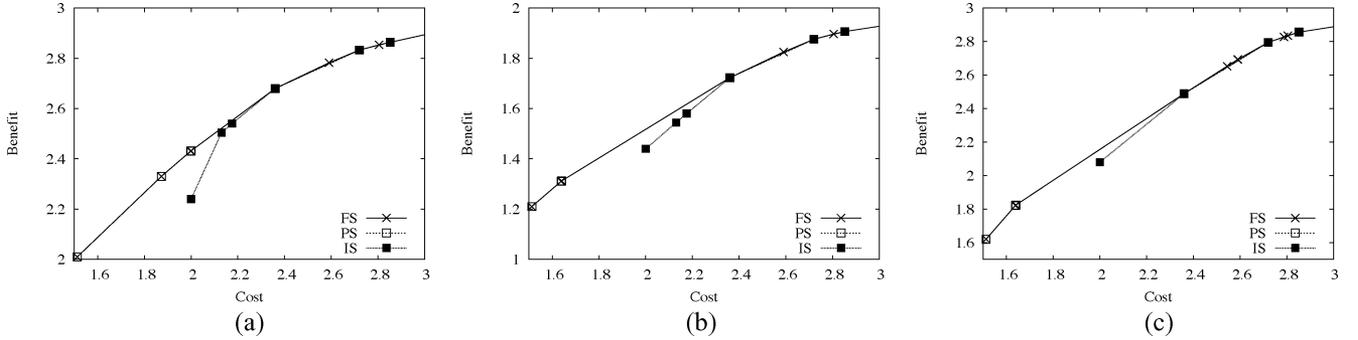


Fig. 4. RD convex-hulls computed based on a full search (FS), on the proposed partial search (PS), and based on a search among independent policies (IS). The number of transmission opportunities is $N = 6$, the time interval between two opportunities is 50 ms. The channel conditions are defined by $\mu_F = \mu_B = 40$ ms, $\epsilon_F = \epsilon_B = 0.2$. (a) R21; (b) R11; (c) R12.

[1]. It also analyzes the performance of partial search solutions based on the selection of relevant master/slave relationships, to a performance upper-bound based on an exhaustive search (when possible). The rate-distortion performances are discussed in the cases of synthetic streams that represent typical layered media data, and common video streams. Simulations are performed on several groups of interdependent data units, and for multiple streaming scenarios (different loss and delay patterns). The streaming framework under consideration is a packet-based network with acknowledgment feedbacks, as described in Section II-A. Packets are lost randomly and independently on the forward or backward path, with a probability ϵ_F or ϵ_B , respectively. The forward loss ratio is intentionally set to large values in our simulations, as the main purpose of advanced scheduling strategies is to compensate for the adverse effect of poor channel conditions. The transmission delays on both paths are modeled as a shifted exponential random variable with mean μ_F (μ_B) and shift $\kappa_F = \mu_F/2$ (respectively, $\kappa_B = \mu_B/2$) [1], [21].

Our simulations basically reveal that

- strategies based on the proposed space of relevant policies generally outperform those only based on independent policies;
- the amount of benefit obtained based on the relevant subspace of dependent policies strongly depends on the relative sizes and distortions of interdependent data units;
- for cases where a comparison is computationally tractable, the proposed subspace of relevant policies results in performances similar to a full search within the entire space of policies, which validates our methodology.

B. Synthetic (Layered) Media Streams

This section first considers the streaming of synthetic data, which represent identical and equidistant frames that are (de)coded independently of each others. The frame rate is set to 20 fps. Each frame is composed of L data units, organized in a hierarchy of layers. All data units have a unitary size. The decrease in distortion associated to a data unit only depends on its layer index in a frame, and obeys a predefined distortion template, characterized by a constant ratio between consecutive layers. Let ΔD_l denote the decrease in distortion for the l^{th} layer. We denote R11 the template for which $\Delta D_1 = 1$ and $\Delta D_{l+1} = \Delta D_l$. Similarly, we denote R21 (R12) the template

for which $\Delta D_L = 1$ ($\Delta D_1 = 1$) and $\Delta D_{l+1} = \Delta D_l/2$ (resp. $\Delta D_{l+1} = 2\Delta D_l$). For all templates the quality achieved in absence of any data unit is set to 0. This artificial data model allows to define a representative set of contents, and in the same time to carefully analyze the behavior of the streaming system. Note that the R11 and R21 templates are certainly the most realistic ones, as media coders generally encode the most important information in the first layers. The selection of relevant set of masters for these templates is illustrated in detail in [20, App. E].

Three particular situations are now presented, in order to illustrate the benefit of dependent streaming policies: i) an encoding system with two layers only, ii) a system with instantaneous transmission, and iii) a system with N layers, and nonzero transmission delay. The first two scenarios are quite restrictive, but allow for an exhaustive search among all possible independent and dependent transmission policies. Therefore, we are able in this case to provide a comprehensive comparison of the streaming performance obtained with a full search among all possible policies, a partial search limited to our proposed subspace of relevant policies, and a search restricted to independent policies. The third scenario is closer to common streaming scenarios. It considers that the frames are composed of any number of layers, and that a finite delay is available before expiration of the frame delivery deadline.

1) *Two Layers*: In the first scenario, each frame is composed of two layers. A finite number (6) of equidistant transmission opportunities are considered for each data unit. Figs. 4 and 5 show the rate-distortion convex-hull corresponding to optimized scheduling that considers the entire space of dependent and independent policies (= Full Search), the proposed subspace of policies defined by relevant MSRs (= Partial Search), and the set of independent policies (= Independent Search). Three distortion templates are considered, and different channel characteristics are simulated in Figs. 4 and 5, with, respectively, symmetric paths, and lossless backward channel.

It can be observed first that the proposed partial search achieves the same performance as the full search, which certainly validates our methodology. Then, the figures show that the gain provided by dependent policies is larger when the feedback is reliable. Dependent policies favor the wait for ancestor feedbacks, to prevent the transmission of data units that could not be decoded in absence of ancestors. In absence of ancestor

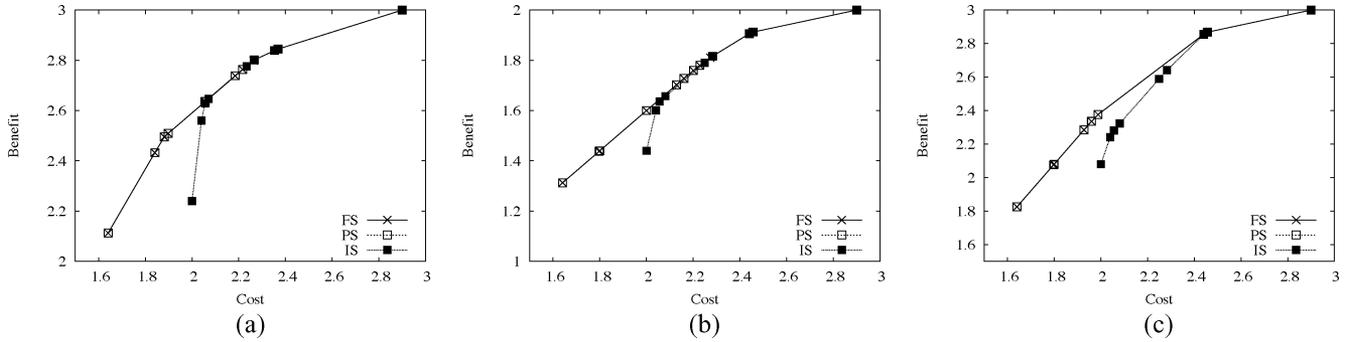


Fig. 5. RD convex-hulls computed based on a full search (FS), on the proposed partial search (PS), and based on a search among independent policies (IS). The number of transmission opportunities is $N = 6$, the time interval between two opportunities is 50 ms. The channel conditions are defined by $\mu_F = \mu_B = 40$ ms, $\varepsilon_F = 0.2$, $\varepsilon_B = 0$. (a) R21; (b) R11; (c) R12.

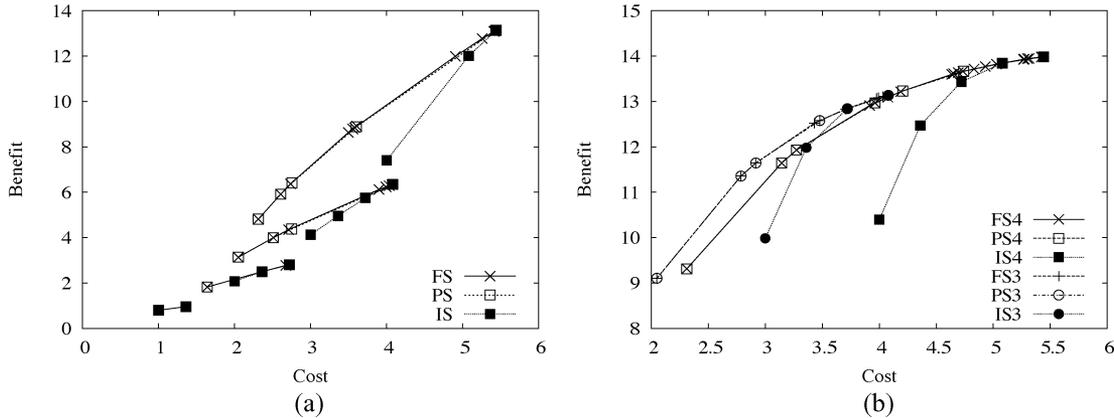


Fig. 6. RD convex-hulls. The time interval between two opportunities is infinite. The channel conditions are defined by $\varepsilon_F = \varepsilon_B = 0.2$. (a) For R12, the convex-hulls corresponding to all possible numbers of active layers are plotted (the larger the number of active layers, the higher the cost in rate). (b) For R21, for clarity purpose, only the convex-hulls corresponding to 3 and 4 active layers are plotted.

feedback, dependent policies decide not to transmit the slave data unit. A reliable feedback guarantees a better knowledge of the client state at the server, which in turns decreases the risk of inappropriate nontransmission decisions, and in turns increase the benefit obtained from the conservative behavior of dependent policies.

2) *N Layers With Infinite Delivery Deadlines:* The second scenario considers a stream with N layers. Moreover, the time available before expiration of the data delivery deadline tends to infinity. This is for example the case when the playback delay is quite large in comparison with the round trip time (instantaneous transmissions). Whilst often unrealistic, the infinite delivery deadline assumption is interesting because it simplifies the definition of data unit policy, which in turns makes a full search within the entire space of policies computationally tractable. The transmission delay, and exact transmission schedule become irrelevant in that case, where only the number of retransmissions matters.

Figs. 6 and 7 compare the performance of scheduling policies respectively optimized on the entire set of policies (FS), on the proposed subspace of policies defined based on relevant MSRs (PS), and on the set of independent policies (IS). Each frame is composed of at most $L = 4$ layers, characterized by specific distortion templates (i.e., R11, R12, or R21). In Figs. 6 and 7(a), convex-hulls are computed based on the policies that only activate the first X layers encountered along the dependency graph. By definition, a layer is said to be *active* if at least

one component of its policy vector is set to one. Between one and four active layers are considered in Fig. 6(a) and (7a), while only 3 and 4 active layers are depicted in Fig. 6(b). All figures show that the proposed partial search achieves close to optimal (= full search) performance, and that the search restricted to independent policies performs significantly worse than the proposed search. This validates our methodology, since the proposed set of relevant MSRs is able to identify RD optimal dependent policies. In particular, the set of masters corresponding to the optimal dependent policies defined based on a full search is identical to the set of masters defined based on (13).

The simulation results also reveal that the gain provided by PS over IS is highly dependent on the relative distribution of distortion between layers. Let us consider the *global convex-hull*, overwhelming the convex-hulls derived for all possible number of active layers. As expected, the gap between the global FS and the IS convex-hulls decreases when going from the R12 to R11 and R21 distortion templates. Hence, dependent policies provide negligible benefit when the distortion per unit of cost decreases along the dependency path. In that case, it is better (in a rate-distortion sense) to retransmit the ancestor data in absence of ACK, rather than to send descendants that only provide a small gain in distortion. This observation is confirmed by the simulation results presented later.

Finally, Fig. 7(b) extends Fig. 7(a) and presents the RD optimal points obtained with the three streaming strategies. Note that these points are not necessarily on the convex-hull. We ob-

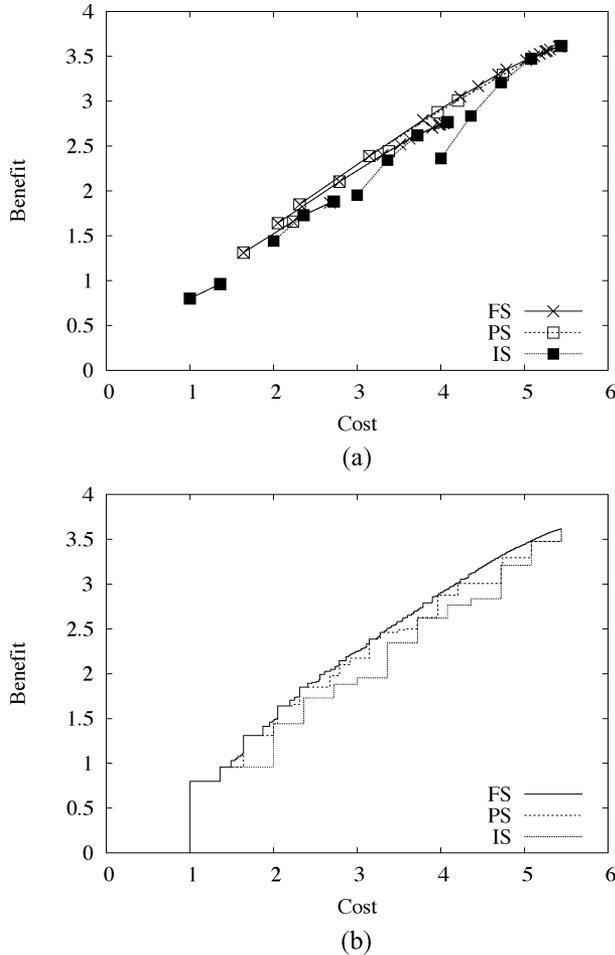


Fig. 7. Streaming performance obtained for the R11 template. The time interval between two opportunities is infinite. The channel conditions are defined by $\varepsilon_F = \varepsilon_B = 0.2$. (a) RD convex-hulls corresponding to all possible numbers of active layers. (b) Optimal RD points (not necessarily on the convex-hull). The RD points sustained by all possible numbers of active layers have been considered, for the R11 distortion template.

serve that PS and IS result in abrupt drops of benefit as the rate decreases, and that they are not able to follow the graceful evolution offered by the entire space of policies ($=$ FS). More interestingly, we also observe that PS significantly outperforms IS, but sometimes lies below FS. A careful comparison of the PS and FS curves reveals that the proposed subspace of relevant policies does not capture all optimal RD points, but rather a well-chosen subset of these points. In particular, we observe that the subset of optimal RD points selected by PS are regularly spread over the cost range, and include all optimal RD points lying on the optimal convex-hull. This is expected, since the subspace of relevant policies has been defined to include most of the policies that are expected to bring a significant benefit in the Lagrangian framework, or equivalently the policies that are expected to improve the RD convex-hull computed based on independent policies.

3) *N Layers With Finite Delivery Deadlines*: The third scenario considers that each frame is composed of five data units of unitary size, organized into a hierarchy of layers. However the delay available before a data unit delivery deadline is now limited. Specifically, we consider that a one second delay is available between the first transmission opportunity of a data unit

and its delivery deadline. During this time interval, each data unit receives 20 opportunities to be transmitted, the time interval between successive transmission opportunities being equal to 50 ms. We compare the convex-hull computed for independent policies ($=$ IS), and the convex-hull resulting from a search among the proposed subspace of relevant policies ($=$ PS). The IS convex-hull is computed as described in [1]. Alternatively, a convex-hull is computed based on (8), for each possible relevant dependent policy. All relevant convex-hulls are then merged to build the PS convex-hull.

Fig. 8 presents the results obtained for the R11 distortion template, where all data units are equally important. Fig. 8(a) plots the convex-hulls computed for the proposed subspace of relevant policies (PS), with different numbers of active layers (denoted “PS X” when X layers are active). The global convex hull, which sustains all “PS X” convex-hulls, is denoted “Hull PS”. Similarly, in Fig. 8(b), “IS X” denotes the convex-hull computed for independent policies, with X active layers. We observe in Fig. 8(b) that the proposed set of dependent policies improves the RD performance, i.e., Hull PS lies above IS X, for all X’s. However, the gain appears to be quite marginal. Fig. 9 provides the same analysis regarding the R12 and R21 distortion templates, respectively. Overall, we observe that the gain provided by dependent policies is quite significant for the R12 template, but is small for R11 and even negligible for the R21 template.

Dependent policies are mainly beneficial when the gain in distortion does not decrease along the dependency path. This observation is important because it means that there is no crucial need to implement dependent streaming policies for progressive or layered image coders, as they inherently encode the most important information first. However, when interdependent data units correspond to a group of consecutive frames with temporal dependencies, we can unfortunately not rule out that nonnegligible gain can be obtained with dependent policies. This depends on the activity in the media sequence, which often drives the evolution of the benefit per transmission unit along the dependency path. An example is studied in the next section.

C. Video Streams With Temporal Dependencies

This section now evaluates the advantages of considering dependent policies when streaming real media content. We consider a typical MPEG sequence, where the frames within a group of picture (GOP) are likely to present an increase of benefit per transmission unit along the dependency path. Reference Intra frames and predicted Inter frames bring similar benefit to the reconstruction (especially when concealment is not very efficient), but the cost of a predicted frame is much smaller.

Our proof-of-concept example is built on the following setup. The first 100 frames of the QCIF Foreman sequence are encoded at 10 fps, with IPPPPPPPPP dependencies, using JM2.1 of the JVT/H.264 compression standard. A constant quantization parameter has been used, for an average Y-PSNR of 35.8 dB. In our streaming simulations, we employ a concatenation of hundred video segments, each segment being composed of the 100 encoded frames of the Foreman sequence. The purpose of running simulations on a longer concatenated sequence is to alleviate the impact of the probabilistic channel behavior on the average reconstructed Y-PSNR values. At the receiver, previous

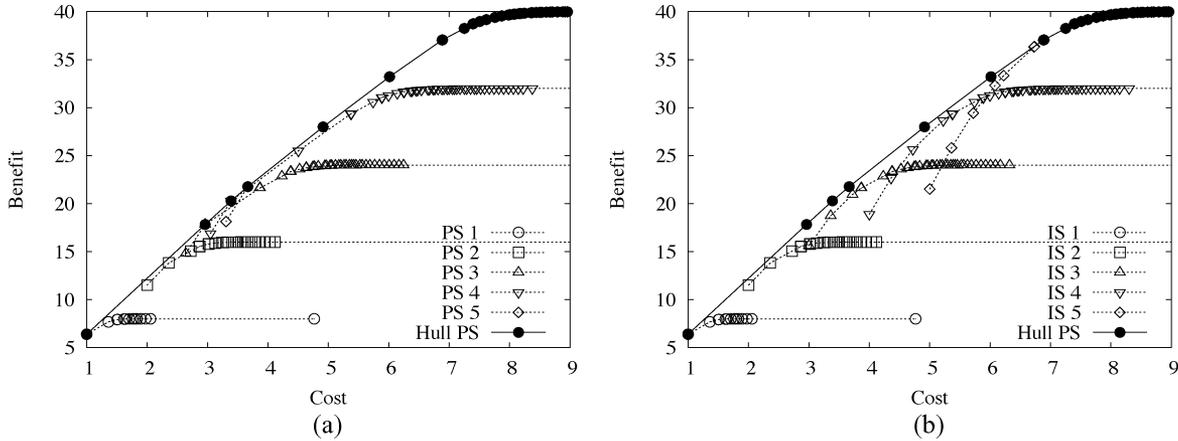


Fig. 8. RD convex-hull. R11, 5 layers, 20 fps. PSX and ISX denote, respectively, the convex hull of dependent and independent policies for X active layers. Hull PS denotes the convex hull of all PSXs. The number of transmission opportunities is $N = 20$, and the time interval between two opportunities is 50 ms. The channel conditions are defined by $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$. (a) Dependent policies and (b) independent policies.

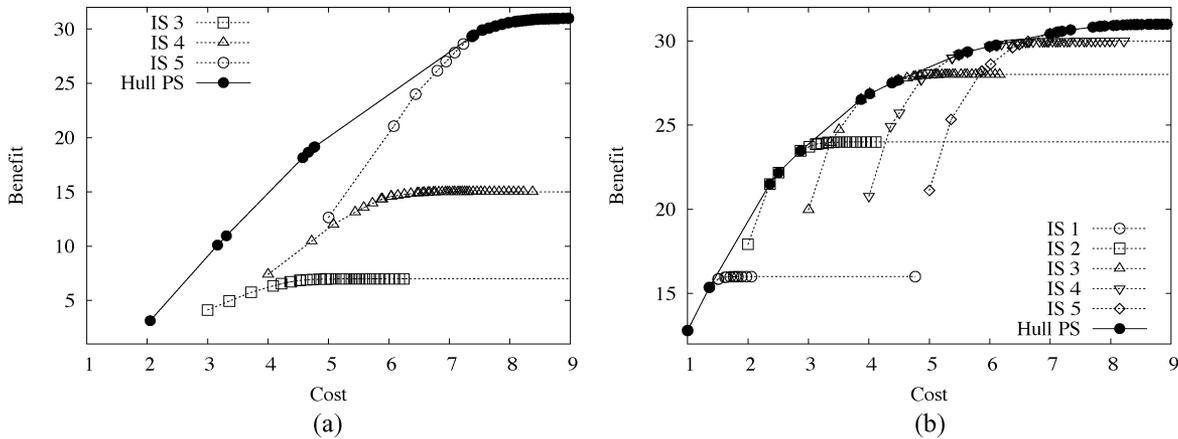


Fig. 9. RD convex-hull. 5 layers, 20 fps. Hull PS denotes the convex hull of partial search (PS) hulls computed for all possible numbers of active layers. ISX denotes the convex hull of independent policies for X active layers. The number of transmission opportunities is $N = 20$, and the time interval between two opportunities is 50 ms. The channel conditions are defined by $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$. (a) R12 and (b) R21.

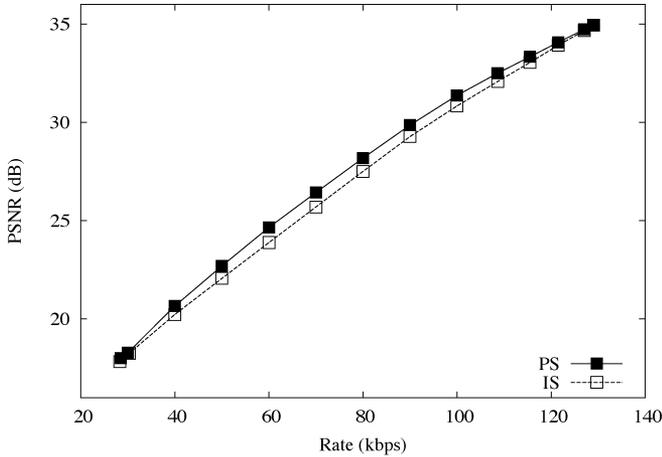
frame error concealment on missing video frames is performed by the client application. On the scheduler side this concealment strategy is approximated as proposed originally in [22], and further described in [23, Appendix]. The method assumes that a nondecodable frame is reconstructed based on the last decodable frame. A frame is said to be decodable when the frame and all of its ancestors have been received on time. The method relies on the pre-computation of the PSNR of frame i when it is approximated by the correctly decoded frame j , $\forall i, j$ such that $i - N_c < j \leq i$. The concealment horizon N_c is set to 10 in our simulations.

The channel conditions are defined by $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$. The transmission policies are adjusted every 100 ms, and each data unit receives three transmission opportunities distant by 500 ms. The pre-fetch delay is set to 2 s. An average target rate is fixed *a priori*, and a simple rate control algorithm, similar to the one described in [1], smoothly adapts the RaDiO λ parameter so as to respect the average bit-budget constraint.² At each RaDiO iteration, the convex-hull optimal policy corresponding to λ is chosen, for all data unit in the 2 seconds transmission window.

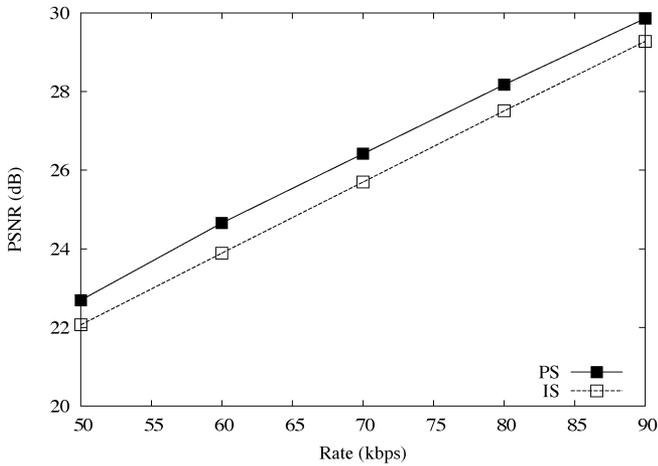
²To avoid λ oscillations across a GOP, the λ adjustment frequency is set to the GOP frequency.

Fig. 10 compares the rate-distortion performance obtained with optimized scheduling solutions based on dependent policies (“PS”), and independent policies (“IS”). We first observe that the proposed set of relevant (dependent) policies is able to improve RD performance in comparison with independent policies. While the gain is marginal (around 0.7 dB), it can be obtained with a small additional computational cost in practice. Among the relevant master/slave relationships defined based on the rules of Section IV, the most advantageous policies basically appear to be the ones where master data units are either the I frame only, or all the frames in the GOP. This is due to similar characteristics for all P frames, which are different from I frames.

We compare now the “IS” and “PS” scheduling mechanisms for different GOP structures and various error concealment strategies, in order to get a deeper understanding of the influence of the content on the streaming performance. We build different sequences simply by repetition of a constant GOP template, which corresponds to N consecutive frames, with one I frame of size s_I , followed by $(N - 1)$ P frames of size s_P . Each sequence is further characterized by a concealment profile, which basically corresponds to the PSNR observed for the (sequence of) frames that follow a loss when a particular concealment is activated. We consider



(a)

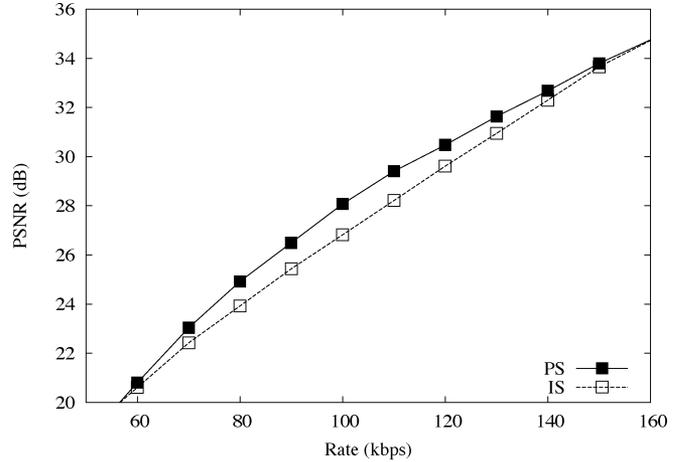


(b)

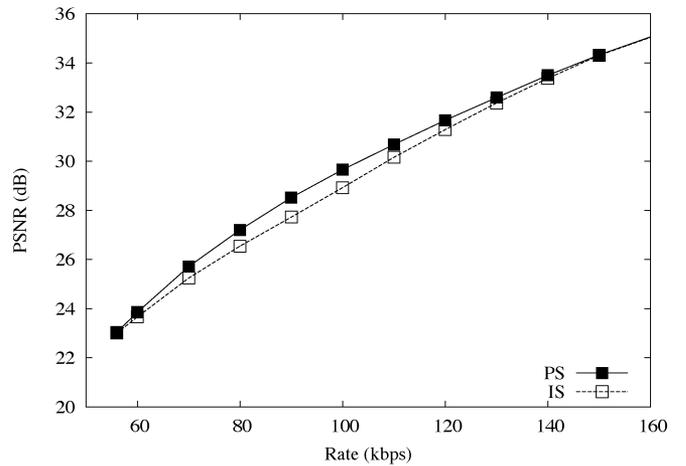
Fig. 10. RD convex-hulls for Foreman QCIF encoded at 10 fps with IPPPPPPPP dependencies. PS and IS refers, respectively, to the proposed subspace of relevant (in)dependent policies, and to the conventional set of independent policies. Channel conditions: $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$. (a) Full range and (b) zoom.

here three concealment profiles C_i , whose PSNR values (in dB) have been chosen arbitrarily, but are inspired from the actual values observed for the frames of the Foreman sequence: $C_{20} = \{36, 20, 18, 17, 16, 15, 14, 14, 14, 14\}$, $C_{23} = \{36, 23, 20, 18, 17, 16, 15, 14, 14, 14\}$, and $C_{26} = \{36, 26, 23, 21, 19, 18, 17, 16, 15, 14\}$, with $C_i(j) = 14$ dB, $\forall i, \forall j > 9$. In our notation, $C_i(0)$ represents the quality of a frame that is correctly decoded. $C_i(j)$ represents the quality of a frame that has been lost, and concealed based on its j^{th} preceding frame. Hence, C_{26} performs more efficient concealment than C_{23} and C_{20} .

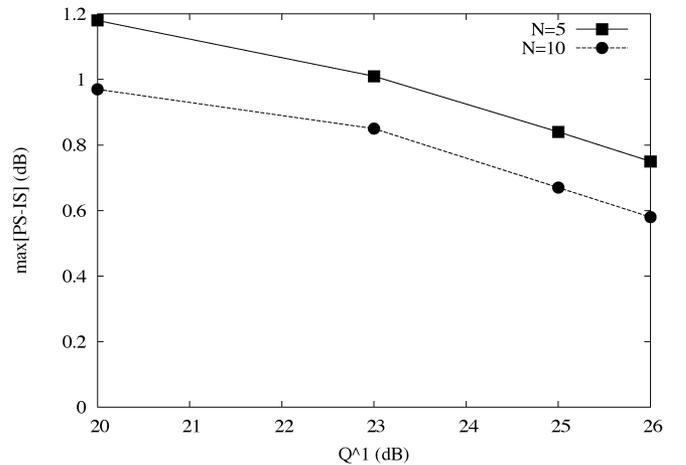
Fig. 11 analyzes the impact of the concealment profile on the benefit obtained based on dependent policies (“PS”), compared to independent policies (“IS”). Figs. 11(a) and (b), respectively, depict the rate-distortion performances obtained with the concealment profiles C_{20} and C_{26} . Unsurprisingly, we observe that the streaming quality increases when the concealment becomes more efficient. Fig. 11(c) reports the maximal performance gap (i.e., PSNR difference) observed between the PS and IS strategies, as a function of $C_i(1)$. It can be seen that the benefit of



(a)



(b)



(c)

Fig. 11. Impact of the concealment profile on the benefit drawn based on dependent policies. (a) and (b) plot the IS and PS rate-distortion curves obtained with C_{20} , and C_{26} , respectively. (c) depicts the maximal PSNR difference measured between PS and IS rate-distortion curves, as a function of the PSNR quality $C_k(1)$ of the set of concealment profiles defined in the text. For all graphs, SI and SP have been set to 28 and 7 kbits, and N has been set to 5. The channel conditions are defined by $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$.

dependent policies decreases as the concealment improves. A poor concealment (i.e., small $C_i(1)$) puts a lot of importance

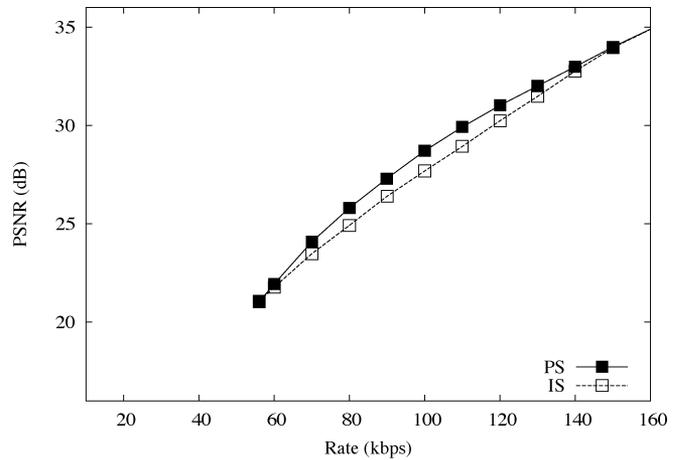
on receiving a P frame, which significantly increases the reconstructed quality compared to a decoding (and concealment) with its ancestors only. Hence, the relative benefit obtained from the transmission of a descendant (P frame) increases compared to the retransmission of an ancestor (I frame). This favors the scheduling based on dependent policies (see Section IV-D).

Fig. 12 analyzes the impact of the GOP size on the streaming performance. Figs. 12(a) and (b), respectively, depict the RD curves obtained with GOP size $N = 5$ and $N = 10$, while Fig. 12(c) plots the maximal PSNR difference measured between PS and IS as a function of the GOP size N , for three concealment profiles. They confirm that the benefit of dependent policies is larger when the concealment is not efficient. We can also observe that the benefit offered by dependent policies initially increases with small values of N (until $N = 5$), and then progressively decreases with the GOP size.

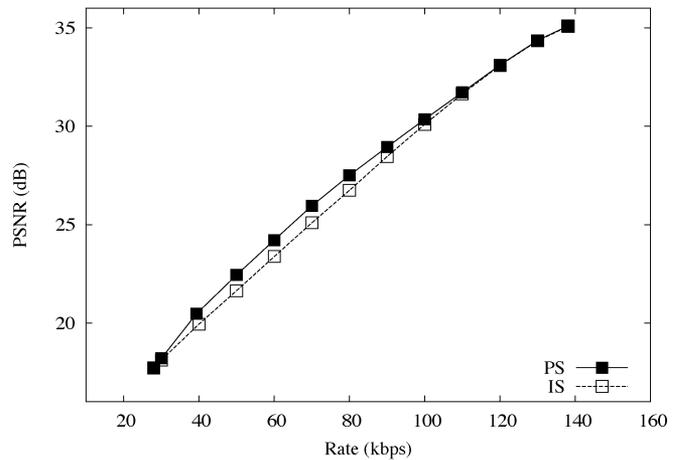
A deeper analysis of the convex hull optimal policies selected by the IS and PS schedulers is necessary to explain that observation. On one hand, we have observed that the IS scheduler ends up in promoting two different strategies to select the optimal policies, depending on the values of the size of P frames, s_P and the GOP size N . When N and s_P are large, optimal independent policies favor strategies that either transmit I a single time (and do not transmit the P frames) or transmit both I and P frames several times. This is due to the relative high transmission cost of P frames, and to the increased benefit of retransmissions of I frames when the number of descendants becomes large. In contrast, when N and s_P are small, it becomes advantageous for the IS scheduler to consider the transmission of P frames, rather than retransmitting the I frame. In this case, the large transmission cost of the I frame is indeed not compensated by the gain in quality provided by a small number of descendants. On the other hand, the PS scheduler always promotes the dependent transmission of P frames, conditionally to the reception of an ACK for the I frame, when resources are limited. In this way, it fully exploits the larger gain per unit of rate provided by P frames.

The results illustrated in Fig. 12(c) can therefore be explained as follows. When N is small, both schedulers opt for policies where I frames are only transmitted once, and P frames may be transmitted several times. The benefit of PS over IS comes from the fact that it does not waste transmission resources when no feedback is received for correct transmission of P-frames. That benefit increases with N , as it offers more opportunities to get a benefit from dependent transmission of P frames. In contrast, when N is large, it becomes more advantageous for IS to promote multiple transmissions of the I frame. In that case, the benefit of PS over IS decreases with N , as an increasing number of descendants augments the benefit obtained from I frames retransmissions. The value of N where the benefit is maximal obviously depends on the relative sizes of I and P frames, and it moves towards larger values of N when s_P decreases.

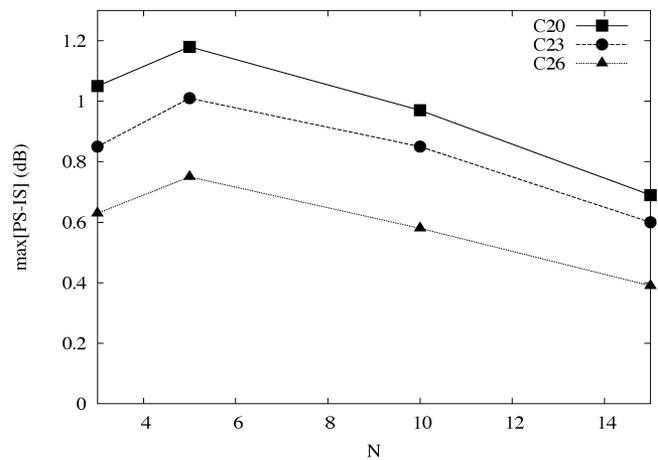
Finally, Fig. 13 analyzes the impact of the size of P frames, s_P , over the benefit obtained with dependent streaming policies. We observe that the benefit of PS over IS generally decreases when s_P increases, and that the decrease is sharper for larger N . As explained above, the retransmission of I frames by the IS scheduler, becomes increasingly advantageous compared to a



(a)



(b)



(c)

Fig. 12. Impact of the GOP size on the benefit drawn based on dependent policies. (a) and (b) plot the IS and PS rate-distortion curves obtained for $N = 5$ and $N = 10$, with the C_{23} concealment profile. (c) depicts the maximal PSNR difference measured between PS and IS rate-distortion curves, as a function of the GOP size N , for three distinct concealment profiles. For all graphs, s_I and s_P have been set to 28 and 7 kbits, respectively. The channel conditions are defined by $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$.

single transmission (as promoted by dependent policies), when s_P and N are large. It can also be seen in Fig. 13(a) that the influence of s_P becomes negligible when both N and s_P are small.

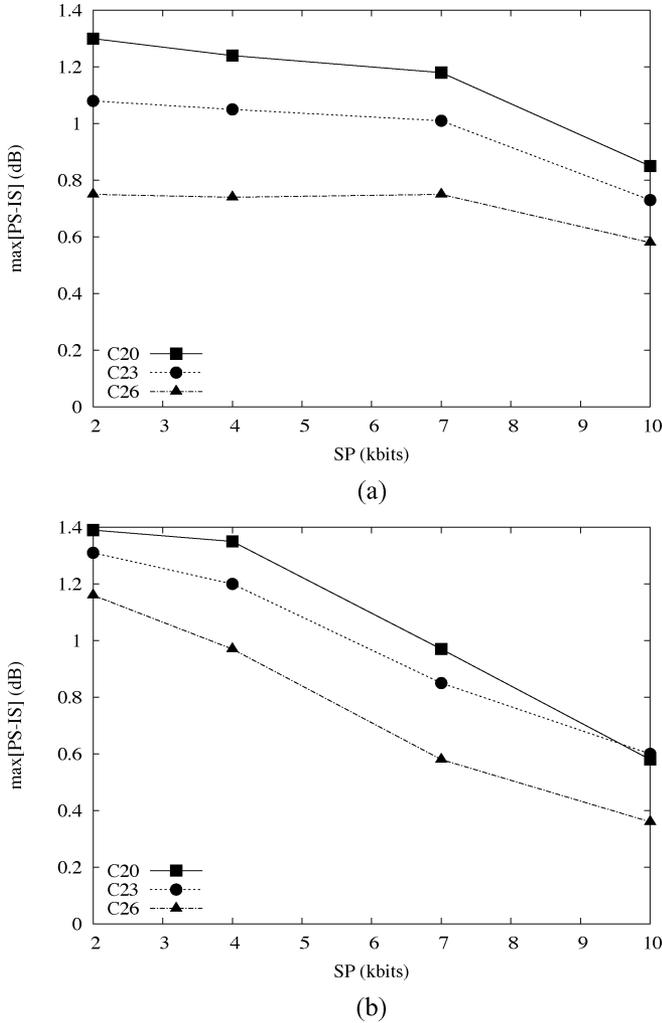


Fig. 13. Impact of the P frame size (s_P) on the benefit drawn based on dependent policies. (a) and (b) depict the maximal PSNR difference measured between PS and IS rate-distortion curves, as a function of s_P , for three distinct concealment profiles. Channel conditions: $\mu_F = \mu_B = 100$ ms and $\varepsilon_F = \varepsilon_B = 0.2$. (a) $N = 5$; (b) $N = 10$.

In this case, the benefit of PS over IS does not depend on the absolute value of s_P , since both schedulers adopt the same strategy that consists in sending I frames only once. The number of bits saved by PS in avoiding wasteful retransmissions in absence of ACKs, is proportional to s_P . In the same time, the benefit per bit that can be obtained through additional retransmissions of P frames, is inversely proportional to s_P . Hence, the benefit of PS over IS does not depend on s_P when N and s_P are small.

D. Discussion

The previous analysis demonstrates that dependent policies have the capability to outperform independent policies in video streaming applications. We have shown that the benefit provided by dependent policies generally increases when i) the concealment efficiency decreases, ii) the relative cost of P frame decreases, and iii) the GOP size is small. These characteristics correspond to sequences for which the temporal activity is quite important, but where the coding efficiency is significantly larger

for P frames than for I frames. In practice, these sound like contending requirements. However, they might be encountered in sequences for which motion compensation is efficient (high coding efficiency), but unpredictable (poor concealment). For a typical sequence, our simulations have shown that the benefit to expect from dependent policies remains smaller than 1 dB. Hence, we interestingly conclude that for most real video content, only marginal gain is expected from dependent policies, so that the independent policies assumption raised by [1] can generally be accepted.

Dependent policies can also be ruled out when the media is characterized by a decrease of the relative benefit brought along the dependency path. That is typically the case for JPEG2000 content, which encapsulates the information that provides the largest benefit per unit of rate in the first layers of an embedded stream [24], [25] so that the relative importance of corresponding data units decreases along the dependency path.

In contrast, when the relative benefit per transmission unit is likely to increase along the dependency path, dependent policies may become beneficial. For example, the transmission of dynamic 3-D scenes, as acquired based on tools like the ones presented in [26], [27] necessitates the streaming of both the scene geometry and the texture information. Obviously, the texture information brings a large incremental benefit in quality, but is only useful if the scene geometry information is known. As the cost associated to 3-D geometry definition is generally significant [28], the benefit per transmission unit is likely to be higher for texture information than for geometrical information, and thus increases along the dependency path. The application of dependent streaming policies may therefore become beneficial in such scenarios.

We finally raise two observations about the framework used in this paper. First, our study restricts the search space of dependent policies based on a number of assumptions, like the facts that a master is only transmitted once, and that a slave is only transmitted upon reception of all its master ACKs. These two assumptions are motivated in Section III-A, and do not deserve particular attention. In contrast, the nondegradation assumption introduced in Section IV-A supports most of the development and the formal analysis provided in [20]. Hence, the slave eligibility issue should be addressed carefully when the time lost in waiting for master ACKs can not be neglected for important descendant(s) of eligible slaves (see end of Section IV-B2).

Second, as in [1], our framework assumes independent losses and delays for transmitted packets. The assumption does not hold in numerous practical streaming scenarios, and the question of correlated loss and delay patterns definitely needs to be addressed in future research. However, we believe that this issue is orthogonal to the comparison of dependent and independent policies provided in this paper. In a sense, working with correlated losses means that the ACK of a data unit conveys side information about the arrival of other packets, as they are affected by correlated losses. For this reason, independent losses can be considered as minimizing the amount of side information conveyed by a specific ACK about other ACKs. Hence, it maximizes the additional information taken into account when shifting from an independent policy scheduler to a dependent policy one. The independent losses scenario is thus expected

to provide an upper bound on the benefit derived from the explicit information provided by all relevant ACKs, compared to the case where the policy is defined only based on its own ACK.

VI. CONCLUSIONS

A solution to select rate-distortion optimal policies for (re)transmitting packetized media has been proposed by Podolsky *et al.* in [6], based on a Markov chain analysis. As the space of streaming policies grows exponentially with both the number of packets and transmissions opportunities, that solution is limited to scenarios with a few media packets and transmission opportunities. In order to decrease the computational complexity, Chou *et al.* [1] proposed to factor the policy space based on the assumption of independence between streaming policies, which means that the schedule assigned to a data unit does not depend on the future potential acknowledgments of other data units. Our work in a sense bridges the gap between these two studies, and proposes a methodology to limit the space of dependent policies to the dependencies that are likely to bring a rate-distortion performance gain, compared to the performance achieved based on the independence assumption. We have introduced the notion of master and slave data units, and defined a few rules for the selection of the most relevant relationships.

Based on a careful analysis of dependency relationships, and extensive simulation results, we can draw the two following conclusions. First, the resulting set of relevant dependent policies achieves close to optimal performance, while being computationally tractable. This validates the methodology proposed in the paper. Second, the gain to expect from dependent policies in comparison with independent policies strongly depends on the relative sizes and distortions of interdependent data units. Interestingly, we have shown that the benefit of dependent streaming policies is actually quite marginal in scenarios where the gain in distortion per unit of rate decreases along the media decoding dependency path. Alternatively, in cases where some descendant data units bring a relatively large gain in distortion in comparison with other data units they depend on for correct decoding, our simulations demonstrate that dependent streaming policies can however perform significantly better than independent streaming strategies. These worthy findings partly validate the independence assumption, and allow to rule out the computation of dependency policies when streaming scalable content such as defined based on JPEG2000, for example. However, they also do encourage a careful investigation of dependent policies when the streamed content is characterized by a significant increase of the benefit per transmission unit along the data unit dependency path. In a realistic proof-of-concept video streaming, dependent policies potentially provide a gain that typically lies around 1 dB.

REFERENCES

- [1] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.
- [2] B. Girod, M. Kalman, Y. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Rochester, NY, Sep. 2002.
- [3] J. Chakareski, S. Han, and B. Girod, "Layered coding versus multiple descriptions for video streaming over multiple paths," in *Proc. ACM Multimedia*, Berkeley, CA, Nov. 2003.
- [4] A. Begen, Y. Altunbasak, and M. Begen, "Rate-distortion optimized on-demand media streaming with server diversity," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Barcelona, Spain, Sep. 2003.
- [5] G. Matthew, M. Podolsky, M. Vetterli, and S. McCanne, "Soft ARQ for layered streaming media," *J. VLSI Signal Process. Syst.*, vol. 27, no. 1–2, pp. 81–97, Feb. 2001.
- [6] M. Podolsky, M. Vetterli, and S. McCanne, "Limited retransmission of real-time layered multimedia," in *Proc. IEEE 2nd Workshop on Multimedia Signal Processing*, Redondo Beach, CA, Dec. 1998, pp. 591–596.
- [7] M. Kalman and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines for low latency applications," in *Proc. Packet Video Workshop*, Irvine, CA, Dec. 2004.
- [8] M. Roder, J. Cardinal, and R. Hamzaoui, "On the complexity of rate-distortion optimal streaming of packetized media," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2004, pp. 192–201.
- [9] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized media streaming," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Barcelona, Spain, Sep. 2003.
- [10] J. Chakareski, P. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," in *IEEE Workshop on Multimedia Signal Processing (MMSp)*, St. Thomas, U.S. Virgin Islands, Dec. 2002.
- [11] Y. Liang and B. Girod, "Prescint R-D optimized packet dependency management for low-latency video streaming," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Barcelona, Spain, Sep. 2003.
- [12] E. Setton and B. Girod, "Congestion-distortion optimized scheduling of video over a bottleneck link," in *Proc. IEEE Workshop on Multimedia Signal Processing (MMSp)*, Siena, Italy, Sep. 2004.
- [13] J. Chakareski and B. Girod, "Computing rate-distortion optimized policies for streaming media with rich acknowledgements," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Apr. 2004.
- [14] J. Chakareski and B. Girod, "Rate-distortion optimized media streaming with rich requests," in *Proc. Packet Video Workshop*, Irvine, CA, Dec. 2004.
- [15] C.-L. Chang and B. Girod, "Receiver-based rate-distortion optimized interactive streaming for scalable bitstreams of light fields," in *Proc. IEEE Conf. Multimedia and Expo (ICME)*, Taipei, Taiwan, R.O.C., Jun. 2004.
- [16] D. Li, G. Cheung, C.-N. Chuah, and S. B. Yoo, "Joint server/peer receiver-driven rate-distortion optimized video streaming using asynchronous clocks," in *Proc. IEEE ICIP*, Singapore, Oct. 2004.
- [17] D. Tian, X. Li, G. Al-Regib, Y. Altunbasak, and J. Jakson, "Optimal packet scheduling for wireless video streaming with error-prone feedback," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, GA, Mar. 2004.
- [18] Z. Miao and A. Ortega, "Optimal scheduling for the streaming of scalable media," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2000.
- [19] J. Chakareski, J. Apostolopoulos, S. Wee, W. T. 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.
- [20] C. De Vleeschouwer and P. Frossard, "On the Dependencies of Media Packet Schedules in a Rate-Distortion Optimized Framework EPFL-Signal Processing Institute, Tech. Rep. TR-ITS-2006.008, Aug. 2006 [Online]. Available: http://its4www.epfl.ch/~frossard/publications/pdfs/tr_dradio.pdf
- [21] P. Ramanathan and B. Girod, "Theoretical analysis of the rate-distortion performance of a light field streaming system," in *Proc. Int. Picture Coding Symp. (PCS)*, San Francisco, CA, Dec. 2004.
- [22] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. IEEE Data Compression Conf. (DCC)*, Snowbird, UT, Mar. 2003.
- [23] C. De Vleeschouwer, J. Chakareski, and P. Frossard, "The virtue of patience in low-complexity scheduling of packetized media with feedback," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 348–365, Feb. 2007.
- [24] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.
- [25] M. Rabbani and R. Joshi, "An overview of the JPEG2000 image compression standard," *Signal Process.: Image Process.*, vol. 17, pp. 3–48, 2002.
- [26] T. Matsuyama, X. Wu, T. Takai, and T. Wada, "Real-time dynamic 3-D Object shape reconstruction and high-fidelity texture mapping for 3-D video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 3, pp. 357–369, Mar. 2004.

- [27] M. Waschbusch, S. Wurmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3-D acquisition of dynamic scenes," in *The Visual Computer (Special Issues for Pacific Graphics 2005)*, Oct. 2005, vol. 21, no. 8–10, pp. 629–638.
- [28] K. Muller, A. Smolic, M. Kautzner, P. Eisert, and T. Wiegand, "Predictive compression of dynamic 3-D meshes," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Genoa, Italy, Oct. 2005.



Christophe De Vleeschouwer was born in Namur, Belgium, in 1972. He received the Electrical Engineering and Ph.D. degrees from the Université catholique de Louvain (UCL) Louvain-la-Neuve, Belgium, in 1995 and 1999, respectively.

He was a Research Engineer with the IMEC Multimedia Information Compression Systems Group in 1999–2000. He was a Visiting Research Fellow at the University of California, Berkeley (2001–2002) and a Postdoctoral Researcher at the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

(2004). He is now a Belgian NSF Research Associate at UCL in the Communication and Remote Sensing Laboratory (TELE). His main interests concern video and image processing for communication and networking applications, including content management and security issues. He is also interested in non-linear signal expansion techniques and their use for signal analysis and signal interpretation.



Pascal Frossard (S'96–M'01–SM'04) received the M.S. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively.

From 2001 to 2003, he was a Member of Research Staff at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on media compression and streaming technologies. He is now a Tenure-Track Assistant Professor at EPFL, where he heads the LTS4 Laboratory in the Signal

Processing Institute. His research interests include image representation and coding, nonlinear representations, visual information analysis, joint source and channel coding, multimedia communications, and multimedia content distribution.

Dr. Frossard has been the General Chair of IEEE ICME 2002, Lausanne, and member of the organizing or technical program committees of numerous conferences. He has served as lead Guest Editor of a special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Cross-Layer Optimized Wireless Multimedia Communications, May 2007), and as Guest Editor of a special issue of the IEEE TRANSACTIONS ON MULTIMEDIA (Streaming Media, April 2004). He is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2004–present) and of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2006–present). He serves or has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee, as a member of the IEEE Image and Multidimensional Signal Processing Technical Committee, of the IEEE Multimedia Signal Processing Technical Committee, of the IEEE Visual Signal Processing and Communications Technical Committee, and of the IEEE Multimedia Systems and Applications Technical Committee. He received the Swiss NSF Professorship Award in 2003 and the IBM Faculty Award in 2005.