SCHOOL OF ENGINEERING - STI
SIGNAL PROCESSING INSTITUTE
Christophe De Vleeschouwer and Pascal Frossard

CH-1015 LAUSANNE
Telephone: *+4121 6932601*
Telefax: *+4121 6937600*
e-mail: `pascal.frossard@epfl.ch`

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

ON THE DEPENDENCIES OF MEDIA PACKET SCHEDULES IN A RATE-DISTORTION OPTIMIZED FRAMEWORK

Christophe De Vleeschouwer and Pascal Frossard

Swiss Federal Institute of Technology Lausanne (EPFL)

Signal Processing Institute Technical Report

TR-ITS-2006.008

August 25th, 2006

Part of this work has been submitted to IEEE TMM.

This work has been supported by the Swiss NSF, under grant PP002-68737, and by the Belgian NSF.

On the Dependencies of Media Packet Schedules in a Rate-Distortion Optimized Framework

Christophe De Vleeschouwer and Pascal Frossard
EPFL - Signal Processing Institute - LTS4
CH-1015 Lausanne, Switzerland

Abstract

This paper addresses the problem of streaming packetized media over a lossy packet network, with sender-driven (re)transmissions and receiver acknowledgments. It proposes a rate-distortion optimized markovian framework that supports the use of dependent policies. It extends the rate-distortion optimized (RaDiO) streaming framework to render the transmission of a data unit contingent on the acknowledgements relative to other data units. The notion of master and slave data unit is introduced, in order to characterize the dependency between media packets streaming policies. The policy adopted to transmit a slave data unit potentially becomes dependent on the future acknowledgments received about its masters. One of the main contributions of our paper is to propose a methodology that limits the search space of dependent policies for the RD optimized streaming solution, such that only the dependencies that are likely to bring a rate-distortion gain are considered. A few rules are derived, which provide a computationally tractable solution to the rate-distortion optimized streaming problem, with comprehensive use of expected feedback information. Extensive simulations show that the use of relevant dependent policies achieves close-to-optimal streaming performances. We further show that the benefit of dependent streaming policies is actually quite marginal in practical scenarios where the gain in distortion per unit of rate decreases along the media decoding dependency path. But experimental results do encourage a careful investigation of dependent policies when the content is characterized by a significant 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, where sender-driven (re)transmission based on potential acknowledgement feedbacks is considered.

Our work assumes that the network loses or corrupts packets at random, and that correct packets are delivered after a random delay. The media stream is composed of an arbitrary set of possibly interdependent data units, with different contributions to the overall rendered media quality. 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. We build on the framework introduced in the seminal paper from Chou and Miao [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. This framework has been very popular in recent studies in typical streaming media scenarios [2], [3], [4], which certainly justifies a deeper analysis of its underlying assumptions. Specifically, at each possible transmission opportunity, the streaming policy defines whether a data unit should be transmitted, or re-transmitted in absence of acknowledgment. The system in [1] considers the dependencies between data units, and their relative importance, in order to determine optimized streaming policies for each data unit. It updates the policies on the fly, at regular time instants, and thus takes into account the current and expected status of other data units in defining the transmission schedules. However, Chou and Miao [1] only consider independent streaming policies, in the sense that the policy defined for a data unit only depends on its own (non-)acknowledgment at future transmission opportunities, but not on the acknowledgments potentially received for other data units. Hence, they do not envision to assign distinct schedules to a data unit, depending on the acknowledgments received at future transmission opportunities for other data units. For example, they do not offer the possibility to postpone the transmission of a data unit until another data unit has been acknowledged.

The central objective of our paper is to discuss the assumption about the independence of the streaming policies, and to determine its validity, depending on the scenario under consideration. In contrast to [1], our work has the ambition to extend the policy space so that the transmission policy of a given data unit can be made dependent on any future feedback about the status of the streaming session. However, as 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], the new optimization problem rapidly becomes intractable. As a main contribution, our paper proposes a methodology to limit the search space of the candidate dependent policies for rate-distortion optimized streaming. We first introduce the notion of master and slave data units to characterize dependent streaming policies between media packets, such that the policy adopted to transmit a slave data unit becomes dependent on the future acknowledgments received for its masters. A few rules are then derived to define a computationally

This work has been funded by the Swiss NSF, under grant PP002-68737, and by the Belgian NSF.

tractable subset of master/slaves relationships that allows to achieve close to optimal rate-distortion performances. 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 most 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 dependent streaming policies help to improve the performance. For example, we show that dependent policies can typically bring a gain of about 1 dB for conventional MPEG content, where the benefit per transmission unit is likely to be (slightly) smaller for I frames than for their depending P frames.

The paper is organized as follows. Section II briefly recalls the terminology and the methodology introduced in [1], and describes the sub-optimality that may result from the independency of streaming policies. Section III proposes to enlarge the space of independent policies studied in [1] to a computationally tractable subspace of dependent policies, based on master/slave relationships. An algorithm is then proposed to compute the RD optimal policies respecting a given dependency pattern. We show in Section IV how to further limit the dependencies to relevant master/slave relationships (MSRs) that are expected to provide RD 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

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

A. Framework

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. These data units, or possibly part of them, are eventually sent as a packet stream, to a decoder that reconstructs the media information. 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. It means 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 transmissions of data units

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 feedback received for other data units. It actually transforms the computationally intractable Markov decision process introduced in [5] into a controllable optimization problem.

[1]A 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.

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 $t_{l,0}, t_{l,1}, ..., t_{l,N_l-1}$. A binary vector $\pi_l = (\pi_l(0), \pi_l(1), ..., \pi_l(N_l-1)) \in \{0,1\}^{N_l}$ 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 $\epsilon(\pi_l)$ 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

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

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

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

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 $\vec{\pi} = (\pi_1, ..., \pi_L)$, where π_l, $l \in \{1, ..., L\}$ is the transmission policy of the l^{th} data unit. Based on the notation hereabove, the expected transmission rate and distortion for $\vec{\pi}$ are respectively

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

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

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 $J_\lambda(\vec{\pi}) = J_\lambda(\pi_1, ..., \pi_L) = D(\vec{\pi}) + \lambda R(\vec{\pi})$, for $\lambda > 0$. Because $J(\vec{\pi}^{(k)})$ is non-increasing and additionally bounded below by zero, convergence to a local optimum is guaranteed [1]. Alternatively, Roder and 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], [17], [18] also formalize the scheduling decision as a partially observable Markov decision process. Such a popularity certainly justifies a deeper analysis of the independent policy assumption underlying that framework.

C. Limitations of the independent streaming policies

Most of previous works in optimized packet media streaming implicitly assumes independency between a data unit policy and the potential feedback received for other data units at future transmission opportunities. 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 acknowledgements 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. Acknowledgements clearly participate to decrease the uncertainty about the system status, and should be considered in order to adapt the strategy of streaming inter-dependent media data units.

In [1], the authors propose to re-compute 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 by reaction to acknowledgement. It does not permit the scheduler to consider dependencies between transmission policies, a priori, before actual reception of acknowledgments. For example, sub-optimality 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 sub-optimality. 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 sub-optimality. 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

This section proposes to formalize the dependencies between the transmission strategies of interdependent data units. In particular, it investigates how the schedules defining the future transmissions of some data units (called slaves), may advantageously be forced to depend on the future feedback potentially received for other data units (called masters). As the solution of rate-distortion optimal streaming problem may rapidly become intractable due to very complex dependency relationships, we limit the dependencies to the set of relations that are likely to bring a benefit in a rate-distortion sense. We demonstrate that all 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 where the slave is only transmitted after reception of all its masters ACKs. We then extend the formalism presented in Section II-B to compute the optimal dependent transmission policies corresponding to this reduced set of master and slave data units.

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. In general, a data unit can be a slave, a master, both of them, or none of them. The objective of those definition is to be able to adjust the slave policies as a function of the arrival time of acknowledgments (ACKs) for master data units, in order to improve the rate/distortion trade-offs compared to using exclusively independent streaming policies. 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 trade-off between rate reduction and distortion increase has to be defined. Fortunately, the master/slave relationships are not all equally relevant in terms 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. In other words, it means that 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 master ACKs triggers a slave transmission, have to be considered. The study of softer dependency patterns, for which a slave progressively adapts a non-zero transmission policy as a function of the status of masters ACKs, fortunately becomes irrelevant in that case. The remainder of the section now extends the RaDiO framework surveyed in Section II-B, to compute the set of RD optimal dependent policies conforming to a pre-defined set of hard master/slave relationships.

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 $\{t_{l,0}, t_{l,1}, ..., t_{l,N_l-1}\}$ 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 sub-policy vector $\pi_{j,l} \in \{0,1\}^j$ (with $j \in \{1, ..., N_l\}$), 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 sub-policy vectors, contingent on the reception of acknowledgments for the master data units. Finally, $\vec{\pi_\Gamma} = \{\pi_1^{\Gamma_1}, ..., \pi_L^{\Gamma_L}\}$, with $\Gamma = \{\Gamma_1, ..., \Gamma_L\}$, represent the policy vector for the group of L interdependent data units, and gathers the sub-policy 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 sub-policies 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 $l \in \{1, ..., L\}$ and $j \in \{1, ..., N_l\}$) only depends on master data units, and not on the transmission policies of non-master 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 pre-computed as a function of the RTT distribution and the MSRs defined within Γ_l (interested readers are referred to Appendix D for details of that computation). Hence, using the definition of expected transmission cost $\rho(.)$ (see Section II-B), the expected rate can be written as :

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

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 pre-computed as a function of the pre-defined MSRs, and can therefore be considered as a parameter that is independent of the transmission policy assigned to non-master units. We have thus:

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

Based on these definitions, we can now extend the computation of rate-distortion optimal convex-hull policies proposed in [1], to consider a pre-defined 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 non-master sub-policy, keeping the other fixed. The sequence of policy vectors $\vec{\pi_\Gamma}^{(k)}$ is computed as follows. First select $l_k \in \{1, ..., L\}$ and $j_k \in \{1, ..., N_L\}$. Then, $\forall (l, j) \neq (l_k, j_k)$, set $\pi_{j,l}^{(k)} = \pi_{j,l}^{(k-1)}$, and let

$$\begin{aligned} \pi_{j_k,l_k}^{(k)} &= \arg\min_\pi J_\lambda(\pi_{1,1}^{(k)}, .., \pi_{j_k-1,l_k}^{(k)}, \pi, \pi_{j_k+1,l_k}^{(k)}, .., \pi_{N_L,L}^{(k)}) & (7)\\ &= \arg\min_\pi G_{j_k,l_k}^{(k)} \epsilon(\pi) + \lambda p_{l_k}(j_k) S_{l_k} \rho(\pi), \ , & (8) \end{aligned}$$

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

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

In practice, the l_k's are selected among the non-master data units. Initial policies are set to a always-send policy, and the l_k indices are selected in a round-robin order that scan ancestors first. For each l_k, the j_k indices are selected in increasing order. For the same reasons as for the independent case, convergence to the a local optimum satisfying the pre-defined 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 Equations (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 pre-defined set of master/slave relationships (MSRs). However, exploring all possible MSRs to select the one that achieves the best RD trade-off 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 $L \times O\left((L/B)^B\right)$ 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. Assignement of slaves to masters

A methodology and a set of rules are now proposed to assign slaves to master data units, under the initial assumption that the set of master data units is defined a priori.

As it is difficult to apprehend the exact impact of the degradation of the cost/benefit trade-off (ϵ, ρ) (see Section II-B) caused by the wait for master feedback, we initially propose to set a (ϵ, ρ) *non-degradation assumption* in the following development. In other words, we assume that the (ϵ, ρ) trade-offs defined for slave data units, are not significantly affected by the wait for master ACKs. Basically, everything happens as if the feedback was either lost or received instantaneously. When the time to wait for a feedback is small in comparison with the time available before expiration of the slave delivery deadline, that 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) trade-off, it is very likely that a policy where the slave is transmitted independently of its master results in optimal RD trade-off. Hence, 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) trade-offs. The non-degradation assumption stays valid for eligible slaves, which are involved in a given set of MSRs.

Based on the above arguments about the non-degradation 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 non-degradation assumption. We identify three necessary rules imposed by ancestor/descendant dependencies on the definition of relevant MSRs. First, we demonstrate that all eligible descendants of a slave are slaves themselves. Second, we observe that, in most practical cases, a slave is a descendant of its master(s). Third, we explain that, when a data unit s is a slave of one of its ancestors m, it is also a slave for all other masters m' that are ancestors of s. All these constraints are used to define the set of relevant master-slave configurations associated to a pre-defined group of masters. In a second step, the slave eligibility is checked a posteriori, by enfranchising the ineligible slaves identified in relevant MSRs, in order to further refine the set of valid dependencies. We now present and motivate each one of the 3 rules characterizing a relevant MSR.

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 $m \to s$ relation affects the descendants of s, we first neglect the delay induced by waiting for the master feedback (i.e., non-degradation assumption). In that case, we can show that $m \to s$ implies $m \to j$ 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 $m \to s$, 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 advantageously 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. Interested readers are referred to Appendix A for a detailed and formal analysis. More generally, in this appendix, we further demonstrate that the MSR $d \rightarrow a$ can only become beneficial in the RD sense (without necessarily 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 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 descendance. However, this kind of scenario is very rarely encountered in practice, because efficient media coders encode in priority 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 trade-off by sacrificing some ancestor samples, and consequently all their respective descendants, 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, our study is restricted 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 waiting for an ACK to 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 non-degradation 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 Appendix B for a formal demonstration. There, we note that for non-monotonic evolution of the $\Delta D_l/S_l$ ratio along the path of descendance, 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 either is a slave for all of them, or is transmitted independently of all of them.

The rules 1, 2 and 3, provide the toolbox for the definition of relevant MSRs for a given pre-defined set of M masters, denoted as $\{m_0, ..., m_{M-1}\}$. Relevant MSRs 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 MSRs associated to a branch is completely defined by the index of the oldest slave, s, as depicted in Figure 1. 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 MSRs for the i^{th} branch.

In practice, the branches extracted from the acyclic graph are not necessarily disjoint, so that MSRs to consider for a branch Φ_i might be constrained by the MSRs 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 MSRs to investigate. Figure 2 illustrates the computation of the number of relevant MSRs 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 MSRs have been defined based on rules 1, 2 and 3, they are then checked *a posteriori*, in order to ensure that the slave eligibility condition is not violated. This second step in the definition of relevant MSRs is made necessary by the approximations introduced by the non-degradation 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) trade-off. 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

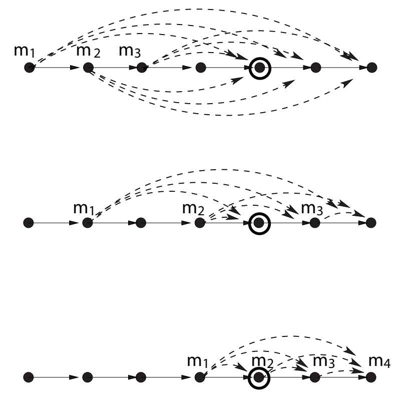


Fig. 1. Three examples of relevant MSRs definition for a branch of 7 interdependent data units. Each example considers distinct a priori selections of master candidates m. For each case, the set of relevant MSRs is defined based on the choice of the oldest slave, and follows the building properties explained 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 5^{th} data unit is chosen to be the oldest slave (= circle in the figures) and the corresponding relevant MSRs is represented by a set of dashed arrows.

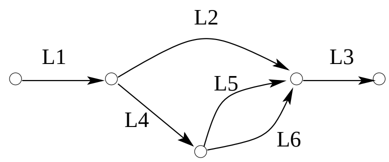


Fig. 2. Example of acyclic dependency graph, and computation of the number of slave assignment possibilities associated to that graph. Labels L_k on each link of the graph refer to the number of data units involved in the corresponding link. The graph contains 3 branches, i.e., it offers 3 different 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 $L_1 + L_2 \times (L_4 + L_5 \times L_6) + L_3$.

independently of its master, all its ancestors should also be transmitted independently of the master, which might be sub-optimal if these ancestors are not problematic and take 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 MSRs, so as to define and explore additional and potentially advantageous MSRs. The approach is described and extended to the case of multiple masters in Appendix C. 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 MSRs 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 where distortion decrement 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 Appendix C for a deeper analysis.

C. Selection of masters

This section now considers the master selection problem, in the definition of relevant MSRs. 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 trade-off. 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 labelled 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

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

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

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

and

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

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 masters data unit, 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 :

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

Note that the methodology proposed above is validated later in section V-B.2. 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 Equation (13).

D. Summary

We now summarize the previous developments for the selection of relevant MSRs. 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 MSRs 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 MSRs 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-B, a set of relevant MSRs associated to a pre-defined 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 MSRs 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 [5].
- 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, wrt the classical RaDiO approach studied in [1]. In this case, there is simply no reason to transmit a descendant data unit before making sure that its ancestors have reached the receiver. 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, in 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 to situation transmitted descendants are undecodable 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 trade-off do satisfy the features defined for relevant MSRs. However, the above arguments, supported by the developments in **??**, demonstrate that the set of relevant MSRs 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 MSRs. 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 [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 data that represent typical layered streams, and common MPEG 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 transmission delays on both paths are modeled as a shifted exponential random variable with mean μ_F (μ_B) and shift $\kappa_F = \mu_F/2$ ($\kappa_B = \mu_B/2$) [1], [19].

Our simulations basically reveal that:

- strategies based on the proposed space of relevant policies generally outperforms 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.

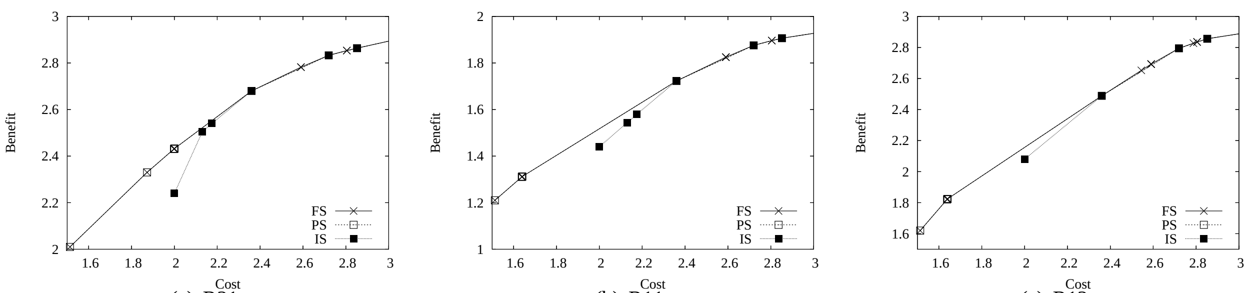
B. Synthetic (layered) media streams

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 details in Appendix E.

Three particular situations are now presented, in order to appreciate 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 non-zero 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.

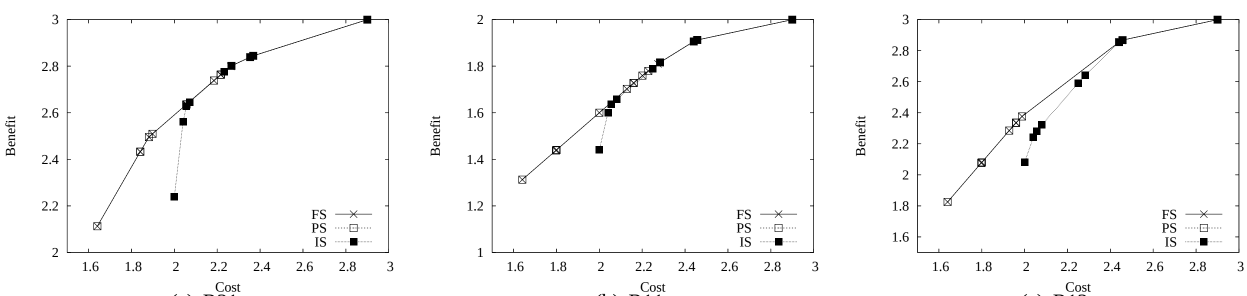
1) Two layers: In the first scenario, each frame is composed of two layers. A finite number (6) of equidistant transmission opportunities are considered for each data unit. Figures 3 and 4 show the rate-distortion convex-hull corresponding to optimized scheduling that considers the entire space of dependent 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 Figure 3 and Figure 4, 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 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 non-transmission decisions, and in turns increase the benefit obtained from the conservative behavior of dependent policies.



(a) R21 (b) R11 (c) R12

Fig. 3. 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, $\varepsilon_F = \varepsilon_B = 0.2$.



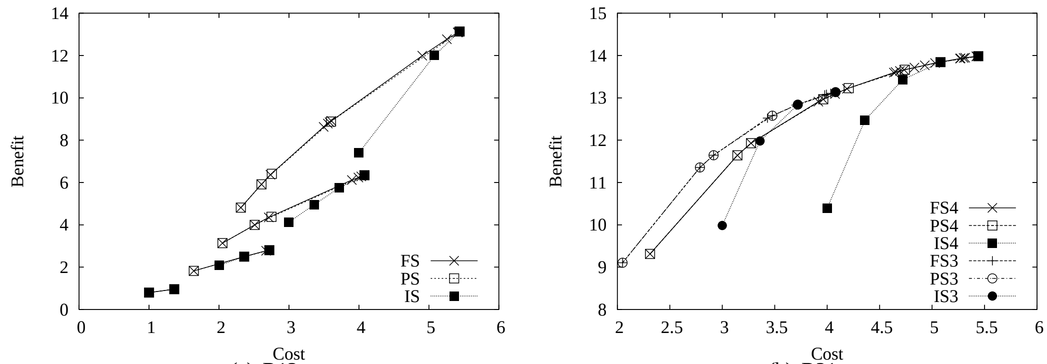
(a) R21 (b) R11 (c) R12

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 $\mu_F = \mu_B = 40ms$, $\varepsilon_F = 0.2$, $\varepsilon_B = 0$.

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.

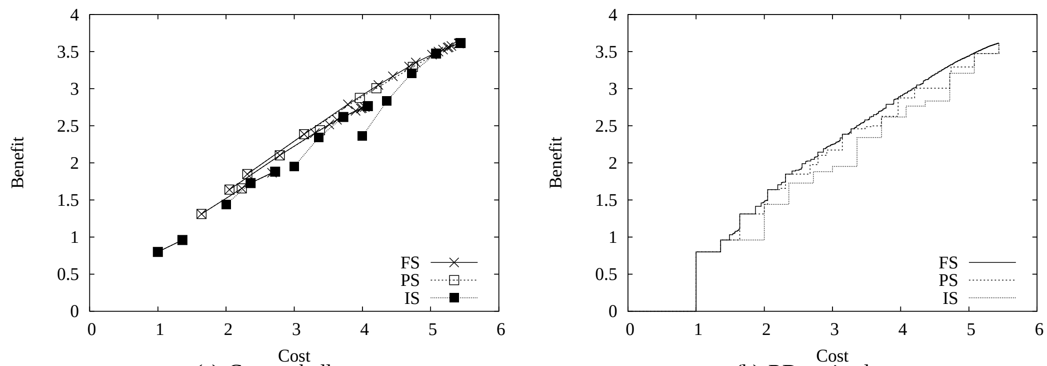
Figures 5 and 6 compare the performance of scheduling policies respectively optimized on the entire set of policies (FS), on the proposed subspace of policies defined based on relevant MSRs (PS), and on the set of independent policies (IS). Each frame is composed of at most L=4 layers, characterized by specific distortion templates (i.e., R11, R12, or R21). In Figures 5 and 6(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 Figure 5(a) and 6(a), while only 3 and 4 active layers are depicted in Figure 5(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 Equation (13).



(a) R12 (b) R21

Fig. 5. 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.



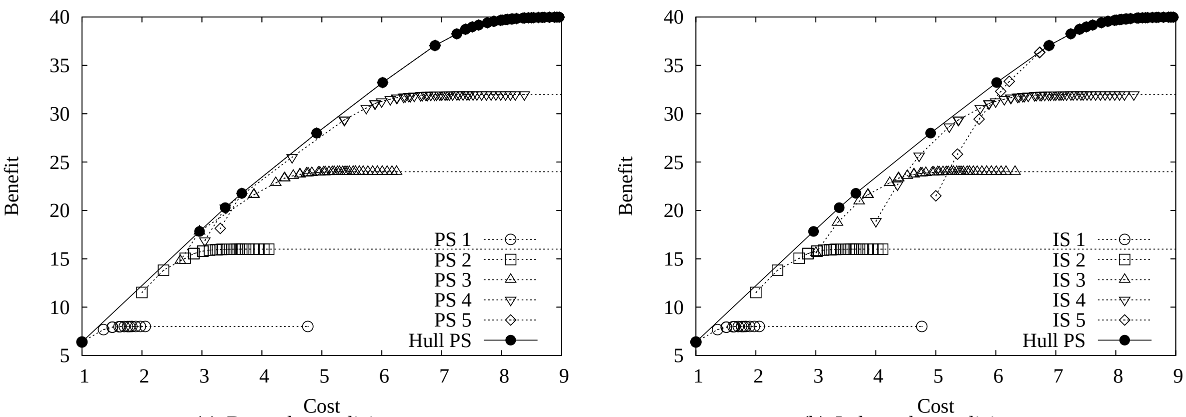
(a) Convex-hull (b) RD optimal

Fig. 6. 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.

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, Figure 6(b) extends Figure 6(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 observe 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 5 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 Eq. (8), for each possible relevant dependent policy. All relevant convex-hulls are then merged to build the PS convex-hull.



(a) Dependent policies (b) Independent policies

Fig. 7. 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 $\mu_F = \mu_B = 100ms$ and $\varepsilon_F = \varepsilon_B = 0.2$.

Figure 7 presents the results obtained for the R11 distortion template, where all data units are equally important. Figure 7 (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 Figure 7 (b), 'IS X' denotes the convex-hull computed for independent policies, with X active layers. We observe in Figure 7 (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. Figure 8 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.

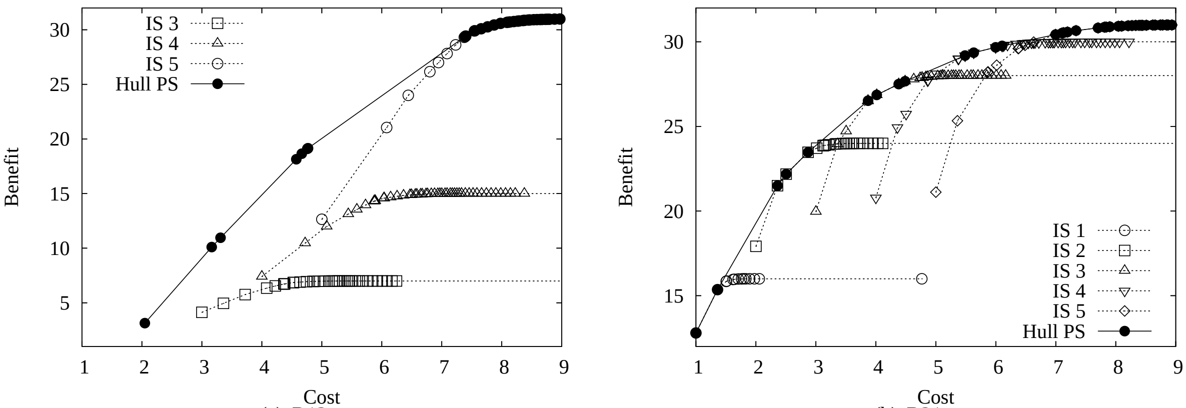
Dependent policies are mainly beneficial when the gain in distortion does not decrease along the dependency path. This observation is important because it means that there is no crucial need to implement dependent streaming policies for efficient progressive or layered coders, as they inherently try to encode the most important information first. However, when a group of interdependent data units corresponds to a group of consecutive frames, we can unfortunately not rule out that non-negligible 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. MPEG 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. To alleviate the impact of the stochastic nature of the channel on the average reconstructed Y-PSNR values, each GOP of the original sequence has been repeated a hundred of times. At the receiver, we adopt a simple concealment model, which reconstructs non-decodable frames based on the last decoded frame [20]. A frame is said to be decodable when the frame and all its ancestors have been received on time. 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 3 transmission opportunities distant by 500 ms. The pre-fetch delay is set to 2 seconds. 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

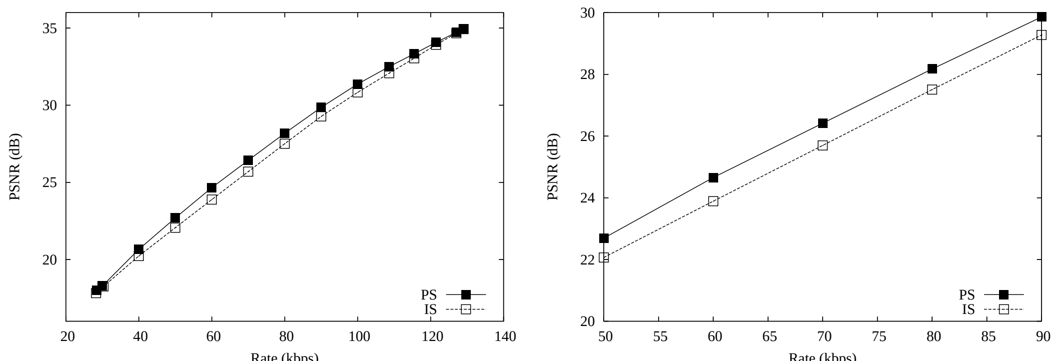
[2]To avoid λ oscillations across a GOP, the λ adjustment frequency is set to the GOP frequency.



(a) R12 (b) R21

Fig. 8. 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 $\mu_F = \mu_B = 100ms$ and $\varepsilon_F = \varepsilon_B = 0.2$.

is chosen, for all data unit in the 2 seconds transmission window.



(a) Full range (b) Zoom

Fig. 9. RD convex-hulls for Foreman QCIF encoded at 10 fps with IPPPPPPPPP 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: $\mu_F = \mu_B = 100ms$ and $\varepsilon_F = \varepsilon_B = 0.2$.

Figure 9 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, 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 correspond to N consecutive frames, with one I frame of size s_I, followed by $(N-1)$ P frames of size s_P. Each template is further characterized by its concealment profile, which basically corresponds to the evolution of the distortion among the frames that follow a loss, when concealment is activated. We consider here three concealment profiles C_k, whose PSNR values (in dB), are inspired from the actual encoding 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 $C_k(j) = 14$ dB, $\forall k, \forall j > 9$. In our notation, $C_k(0)$ represents the quality of a frame that is correctly decoded. $C_k(i)$ represents the quality of a frame that has been lost, and concealed based on its i^{th} preceeding frame. Hence, C_26 performs more efficient concealment than C_23 and C_20.

Figure 10 analyzes the impact of the concealment profile on the benefit obtained based on dependent policies ('PS'), compared to independent policies ('IS'). Figures 10 (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. Figure 10 (c) reports the maximal performance gap (i.e., PSNR difference) observed between the PS and IS strategies, as a function of $C_k(1)$. It can be seen that the benefit of dependent policies decreases as the concealment improves. A poor concealment (i.e., small $C_k(1)$) puts a lot of importance on receiving a P frame, which significantly increases

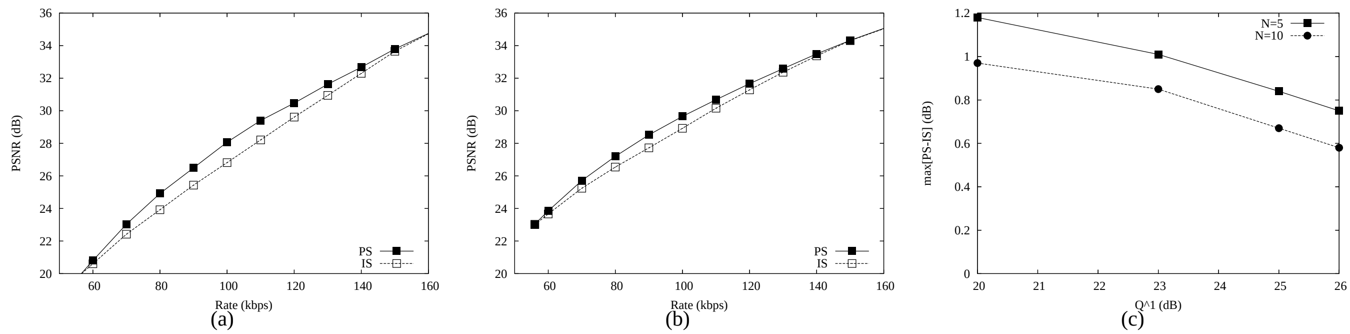


Fig. 10. Impact of the concealment profile on the benefit drawn based on dependent policies. (a) and (b) plot the IS and PS rate-distortion curves obtained with C_{20}, and C_{26} respectively. (c) depicts the maximal PSNR difference measured between PS and IS rate-distortion curves, as a function of 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 $\mu_F = \mu_B = 100ms$ and $\varepsilon_F = \varepsilon_B = 0.2$.

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).

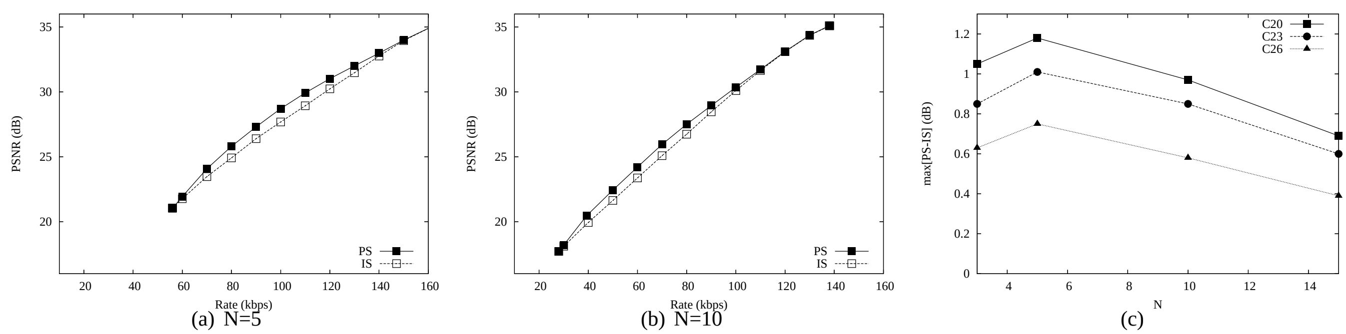


Fig. 11. Impact of the GOP size on the benefit drawn based on dependent policies. (a) and (b) plot the IS and PS rate-distortion curves obtained for $N = 5$ and $N = 10$, with the C_{23} concealment profile. (c) depicts the maximal PSNR difference measured between PS and IS rate-distortion curves, as 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 $\mu_F = \mu_B = 100ms$ and $\varepsilon_F = \varepsilon_B = 0.2$.

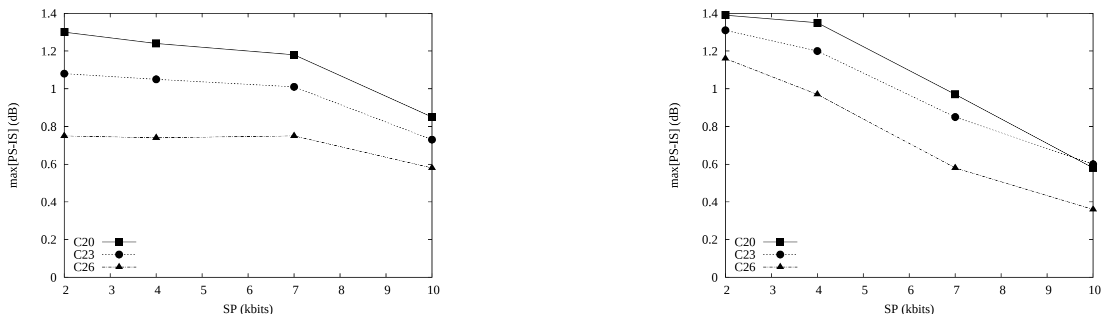
Figure 11 analyzes the impact of the GOP size on the streaming performance. Figures 11 (a) and (b) respectively depict the RD curves obtained with GOP size $N = 5$ and $N = 10$, while Figure 11 (c) plots the maximal PSNR difference measured between PS and IS as a function of the GOP size N, for three 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 the 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 while not assigning retransmission opportunities to the I frame, because, in this case, the large transmission cost of the I frame is 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.

Therefore, the results illustrated in Figure 11 (c) can be explained as follows. When N is small, both schedulers opt for policies where I frame 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, Figure 12 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 single transmission (as promoted by dependent policies), when s_P and N are large. It can also be seen in Figure 12(a) that the influence of s_P becomes negligible when both N and s_P are small. 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.



(a) $N = 5$ (b) $N = 10$

Fig. 12. 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: $\mu_F = \mu_B = 100ms$ and $\varepsilon_F = \varepsilon_B = 0.2$.

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 for 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 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 [21], [22], 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 3D scenes, as acquired based on tools like the ones presented in [23], [24], 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 3D geometry definition is generally significant [25], 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.

VI. CONCLUSIONS

A solution to select rate-distortion optimal policies for (re)transmitting packetized media has been proposed by Podolsky and al. in [5], 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 example built on streaming conventional MPEG content, dependent policies potentially provide a gain that typically lies around 1 dB.

APPENDIX A
SLAVE OF A DESCENDANT

In this Appendix, we consider dependent policies for which the slave is an ancestor of the master. For simplicity, we limit our study to the transmission of two dependent data units. Let a and d respectively denote the ancestor and descendant data units. First, we demonstrate that selecting the descendant d to be a master of its ancestor a never improves the (R,D) convex hull, as compared to the one derived based on independent transmissions. Second, we explain that the $d \rightarrow a$ MSR can only improve RD optimal solutions when the descendant brings a large gain in distortion with a relatively small cost in rate. Such allocation is rare in practical systems because most encoders try to assign the most important information first, i.e., to the ancestor data units. We conclude that the $d \rightarrow a$ MSR should not be considered as a relevant MSR because it is very unlikely to improve RD trade-offs.

As a first step, we now prove that the $d \rightarrow a$ MSR never provides a better (R,D) convex hull than the one derived based on independent transmissions. For a given λ, the optimal independent and dependent transmissions policies are computed based on (**??**) and (8). Let $\pi_a^{\perp}(\lambda)$ and $\pi_d^{\perp}(\lambda)$ denote the optimal independent policies computed based on (**??**). When the policies are constrained by the $d \rightarrow a$ MSR, the master policy can be denoted $\pi_d^{\leftarrow}(\lambda)$, and is defined by a single binary vector, just as in the independent case. In contrast, the slave policy is generally described by a set of policy vectors, each vector corresponding to the number of transmission opportunities available upon master feedback reception (see Section III). Here, we consider that all the transmissions opportunities of the slave remain available upon reception of the feedback, i.e., we neglect the impact of the wait for a feedback. Hence, the policy of the slave can be described with a single binary vector, denoted $\pi_a^{\leftarrow}(\lambda)$, and the feedback mechanism is completely defined by the probability p_d^f for the ancestor to receive a feedback from d. The RD performance that are computed based on this approximation are better than the ones obtained for the system subject to delays. As our purpose is to identify cases where the MSR brings a benefit compared to independent transmissions, it remains to demonstrate that the performance of the approximated system always remain below the ones based on independent transmissions. We now analyze the RD performance of the dependent and independent systems in more details. From Section III, we know that a master-slave relationship only significantly improves the RD performance obtained based on independent transmissions when masters are transmitted once. So, the $d \rightarrow a$ MSR can only be beneficial when d is transmitted a single time. As a consequence, no improvement can be expected when λ is so small that multiple transmissions of d are performed in the independent transmission case. We conclude that λ values that are likely to favour the $d \rightarrow a$ MSR are such that $\rho(\pi_d^{\perp}(\lambda)) = \rho(\pi_d^{\leftarrow}(\lambda)) = 1$. Regarding the ancestor, for the sake of simplicity, we omit the dependency in λ and define $\rho_a = \rho(\pi_a^{\perp}(\lambda))$ and $\rho_a^* = \rho(\pi_a^{\leftarrow}(\lambda))$. Similarly, we define $\epsilon_d = \epsilon(\pi_d^{\perp}(\lambda)) = \epsilon(\pi_d^{\leftarrow}(\lambda))$, $\epsilon_a = \epsilon(\pi_a^{\perp}(\lambda))$, and $\epsilon_a^* = \epsilon(\pi_a^{\leftarrow}(\lambda))$. Letting S_l and ΔD_l respectively denote the size and the gain in distortion of data unit l, we can now define the expected rate R and distortion D respectively associated to independent ($\perp$) and dependent ($\leftarrow$) transmissions. We have

$$\begin{aligned} R_{\perp} &= \rho_a S_a + S_d \\ D_{\perp} &= D_0 - (1-\epsilon_a)\Delta D_a - (1-\epsilon_a)(1-\epsilon_d)\Delta D_d \\ R_{\leftarrow} &= p_d^f \rho_a^* S_a + S_d \\ D_{\leftarrow} &= D_0 - p_d^f (1-\epsilon_a^*)\Delta D_a - p_d^f (1-\epsilon_a^*)\Delta D_d \end{aligned} \tag{14}$$

For a given λ, the $d \rightarrow a$ MSR improves the convex hull computed for independent transmission if and only if

$$D_{\leftarrow} + \lambda R_{\leftarrow} < D_{\perp} + \lambda R_{\perp} \tag{15}$$

Using (14), (15) becomes

$$\lambda S_a(\rho_a - p_d^f \rho_a^*) > [(1-\epsilon_a) - p_d^f(1-\epsilon_a^*)]\Delta D_a + [(1-\epsilon_a)(1-\epsilon_d) - p_d^f(1-\epsilon_a^*)]\Delta D_d \tag{16}$$

Furthermore, the transmission of data units is only beneficial when the Lagrangian resulting from the transmission is smaller than the distortion obtained in absence of transmission. For the policies corresponding to the $d \rightarrow a$ MSR, the condition becomes $D_{\leftarrow} + \lambda R_{\leftarrow} < D_0$, and is written

$$\lambda(p_d^f \rho_a^* S_a + S_d) < p_d^f (1-\epsilon_a^*)[\Delta D_a + \Delta D_d] \quad (17)$$

which implies

$$\lambda \rho_a^* S_a < \Delta D_a + \Delta D_d \quad (18)$$

Besides, we have $\rho_a^* \geq \rho_a$ because $\pi_a^{\leftarrow}(\lambda)$ is computed, knowing that d has reached the receiver in time, while $\pi_a^{\perp}(\lambda)$ only know that d has a good chance, i.e., with probability equal to $1-\epsilon_d$, to be in-time at the receiver. With respect to (18), the $\rho_a^* \geq \rho_a$ inequality implies

$$\lambda \rho_a S_a < \Delta D_a + \Delta D_d \quad (19)$$

Regarding (16), it implies

$$\lambda S_a(\rho_a - p_d^f \rho_a^*) > \lambda S_a \rho_a (1 - p_d^f) > \lambda S_a \rho_a \quad (20)$$

By introducing (20) in (16), we have

$$\lambda S_a \rho_a > [(1-\epsilon_a) - p_d^f(1-\epsilon_a^*)]\Delta D_a + [(1-\epsilon_a)(1-\epsilon_d) - p_d^f(1-\epsilon_a^*)]\Delta D_d \quad (21)$$

and, by merging (21) and (19), we have

$$\Delta D_a + \Delta D_d > [(1-\epsilon_a) - p_d^f(1-\epsilon_a^*)]\Delta D_a + [(1-\epsilon_a)(1-\epsilon_d) - p_d^f(1-\epsilon_a^*)]\Delta D_d \quad (22)$$

This inequality is never true, which proves that it is not possible to find a policy constrained by the $d \rightarrow a$ MSR that improves the convex hull computed based on independent transmissions.

As a second step in this Appendix, we now identify the cases where the $d \rightarrow a$ MSR is likely to support improved RD optimal transmission policies. We show that it is only the case when the cost S decreases and the gain in distortion ΔD increases along the path of descendance. For constrained policies, we use the tilde symbol $'$ to indicate that the data unit is a master, which in turns constrains its policy to a single transmission. In contrast, the star $*$ symbol indicates that the corresponding policy is not necessarily equal to a single transmission. one particular symbol is used for independent transmissions. We now consider the expected cost in bytes and decrease in distortion related to independent transmissions and to the $d \rightarrow a$ and $a \rightarrow d$ MSRs.

For independent transmission policies, the decrease in distortion $\Delta D_{\perp}$ and the cost in bytes $R_{\perp}$ are

$$\begin{aligned} \Delta D_{\perp} &= (1-\epsilon_a)\Delta D_a + (1-\epsilon_a)(1-\epsilon_d)\Delta D_d \\ R_{\perp} &= \rho_a S_a + \rho_d S_d \end{aligned} \quad (23)$$

For dependent policies that are constrained by $d \rightarrow a$ MSR, we have

$$\begin{aligned} \Delta D_{\leftarrow} &= p_d^f(1-\epsilon_a^*)\Delta D_a + p_d^f(1-\epsilon_a^*)\Delta D_d \\ R_{\leftarrow} &= p_d^f \rho_a^* S_a + S_d \end{aligned} \quad (24)$$

For dependent policies that are constrained by the $a \rightarrow d$ MSR, as above, we define $\rho_d^* = \epsilon(\pi_d^{\rightarrow})$ and $\epsilon_d^* = \epsilon(\pi_d^{\rightarrow})$ where $\pi_d^{\rightarrow}$ denotes the policy associated to d, and subject to the reception of a feedback from a. We can now define

$$\begin{aligned} \Delta D_{\rightarrow} &= (1-\epsilon_a')\Delta D_a + p_a^f(1-\epsilon_d^*)\Delta D_d \\ R_{\rightarrow} &= S_a + p_a^f \rho_d^* S_d \end{aligned} \quad (25)$$

To compare the above equations, we assume $p_a^f \sim p_d^f$ which makes sense as both values define the probability to receive a feedback in response to a data unit transmission. In particular, when the transmission conditions are such that the impact of delay can be neglected, both values are equal to $(1-\varepsilon_F)(1-\varepsilon_B)$, where ε_F and ε_B respectively denote the probability of loss on the forward and backward paths. By comparing, Equations (25) and (24), we observe that $\Delta D_{\leftarrow}$ is always smaller than $\Delta D_{\rightarrow}$. Here, we assume that $(1-\epsilon_a^*) \sim (1-\epsilon_d^*)$, and we note that $p_d^f(1-\epsilon_a^*) < (1-\epsilon_a')$ because $p_d^f \sim p_a^f < (1-\epsilon_a')$. So $d \rightarrow a$ can only significantly improve $a \rightarrow d$ if $R_{\rightarrow} \gg R_{\leftarrow}$. As $\rho_d > p_a^f \rho_d^*$ and $\rho_a > p_d^f \rho_a^*$, $R_{\rightarrow} \gg R_{\leftarrow}$ implies $S_a \gg S_d$.

Under the $S_a \gg S_d$ assumption, we can now compare Equations (24) and (23). We conclude that the gain in distortion per unit of rate can only be significantly higher in Equation (24) than in Equation (23) if

$$\frac{(1-\epsilon_a^*)[\Delta D_a + \Delta D_d]}{\rho_a^* S_a} \gg \frac{(1-\epsilon_a)[\Delta D_a + (1-\epsilon_d)\Delta D_d]}{\rho_a S_a} \quad (26)$$

which can only be the case when $\Delta D_d \gg \Delta D_a$.

As told above, a decreasing size and an increasing benefit of data units along the dependency path is rarely encountered in practical cases. So we have decided to ignore the descendant $\rightarrow$ ancestor MSRs when searching for RD optimal dependent transmission policies.

APPENDIX B
MULTIPLE MASTERS

In this Appendix, we explain why a data unit l should be a slave either for all or none of its master candidates. By master candidate, we refer to a data units that is only transmitted once, and that is an ancestor of s. Let m and m' denote two master candidates. We want to demonstrate that if $m \rightarrow l$ is beneficial in the RD sense then $m' \rightarrow l$ is also very likely to be beneficial. To simplify the developments, but without loss of generality, we analyze the case where there are no other master candidates than m and m'.

Before digging into our reasoning, we have to introduce the notion of *slave leader* of a master. The slave leader of m, denoted s_m, is then defined as the oldest slave of m along the path of descendance. Here, we assume here that the acyclic dependency graph defining the dependency among the descendants of m is composed of a single branch. However, the results derived based on this assumption trivially generalize to graphs that contain more than one branch by considering one branch at a time, i.e. one slave leader at a time. In addition to the definition of a slave leader, it is worth mentioning that all developments made below neglect the delay induced by the wait for the master feedback. It allows for strong simplifications of the notations introduced in Section III. We can consider that the system behaves as if the feedback about masters was either available immediately, or definitely lost.

In short, our reasoning includes two steps. First, we estimate the gain in rate and increase in distortion resulting from the MSR imposed between the master m and its slave leader s_m. Our purpose is to derive a condition under which the $m \rightarrow s_m$ is likely to produce (R,D) points that lie below the lower convex-hull of RD optimal points accessible by independent transmission policies. Second, we consider the incidence of a second master, and show that the slave leader of the oldest master should be either oldest than the youngest master, or equal to the slave leader of the youngest master. Furthermore, by assuming monotonic evolution of the $\Delta D_l / S_l$ ratio for data units l lying along the dependency path, after the oldest master, we show that when the slave leader of the oldest master is older than the youngest master, the slave leader for the youngest master is equal to its first descendant. As a consequence, a data unit l should be a slave either for all or none of its master candidates.

First, we derive the condition under which the $m \rightarrow s_m$ improves the (R,D) convex-hull computed based on independent transmissions. For a given a λ value, the RD optimal independent transmission policies $\{\pi_l\}_{l<L}$ for the L data units are computed based on Equation (**??**), and the Lagrangian $J_\lambda(\vec{\pi})$ can be written

$$J_\lambda(\vec{\pi}) = D_0 - \sum_{l=1}^{L} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) + \lambda \sum_{l=1}^{L} S_l \rho(\pi_l) \tag{27}$$

The RD optimal policies subject to $m \rightarrow s_m$ are denoted $\pi_{\vec{\{m\}}}$. For a given λ, they are computed by minimizing the Lagrangian $J_\lambda(\pi_{\vec{\{m\}}})$ defined in Equation (8). Based on Section IV-B, we know that $m \rightarrow s_m$ implies $m \rightarrow l$ for all l such that $s_m \prec l$. Furthermore, neglecting the delay induced by the wait for a feedback about m allows for major simplifications of Equation (8). Specifically, neglecting the delay means that all slave transmission opportunities remain available upon feedback reception. As a consequence, each dependent policy $\pi_{l,\{m\}}$, with $s_m \prec l$, can be abstracted by a single binary vector denoted π_l^*. In addition, we define the probability p_m^f that the feedback about master m is received by any of the slaves l, with $s_m \prec l$. Here, we assume that the probability to receive a feedback about m does not depend on l, which is acceptable if the wait for feedback is neglected. In that case, the probability to receive a feedback is directly related to the probability of losing a packet on the forward and backward paths, which do not depend on the data unit waiting for the feedback. Based on the p_m^f definition, and denoting $\pi_{\vec{\{m\}}} = (\pi_1^*, ..., \pi_L^*)$, the Lagrangian $J_\lambda(\pi_{\vec{\{m\}}})$ is written

$$\begin{aligned} J_\lambda(\pi_{\vec{\{m\}}}) \quad = \quad & D_0 - \sum_{l=1}^{s_m-1} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'}^*)) - p_m^f \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l, l' \neq m} (1 - \epsilon(\pi_{l'}^*)) \\ & + \lambda \left(\sum_{l=1}^{s_m-1} S_l \rho(\pi_l^*) + p_m^f \sum_{l=s_m}^{L} S_l \rho(\pi_l^*) \right) \end{aligned} \tag{28}$$

By definition, $m \rightarrow s$ is beneficial in the RD sense iff $J_\lambda(\pi_{\vec{\{m\}}})$ defined in Equation(28) is smaller than $J_\lambda(\vec{\pi})$ in Equation (27). To compare $J_\lambda(\pi_{\vec{\{m\}}})$ and $J_\lambda(\vec{\pi})$, we assume that the part of the Lagrangian related to ancestors of s_m is not significantly affected by the $m \rightarrow s$ relation, i.e. we assume that

$$D_0 - \sum_{l=1}^{s_m-1} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) + \lambda \sum_{l=1}^{s_m-1} S_l \rho(\pi_l) \sim D_0 - \sum_{l=1}^{s_m-1} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'}^*)) + \lambda \sum_{l=1}^{s_m-1} S_l \rho(\pi_l^*) \tag{29}$$

Based on this assumption, trivial developments show that the cost function in Equation(27) is bigger than the cost in Equation (28), i.e., $m \rightarrow s$ is beneficial, when

$$\lambda \sum_{l=s_m}^{L} S_l \left(\rho(\pi_l) - p_m^f \rho(\pi_l^*) \right) > \sum_{l=s_m}^{L} \Delta D_l \left(\prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) - p_m^f \prod_{l' \preceq l, l' \neq m} (1 - \epsilon(\pi_{l'}^*)) \right) \tag{30}$$

To interpret this condition, we consider that $\epsilon(\pi_{l'}) \sim \epsilon(\pi^*_{l'})$ for all ancestors of s_m. We also make the assumption that $\rho(\pi^*_l)$ and $\epsilon(\pi^*_l)$ can be replaced by $\rho(\pi_l)$ and $\epsilon(\pi_l)$ for all l such that $s_m \prec l$. These substitutions are acceptable because they affect the inequalities in opposite direction, and consequently partly compensate for each other. Specifically, the knowledge about correct reception of m encourage more aggressive policies, i.e. $\rho(\pi^*_l) > \rho(\pi_l)$ and $\epsilon(\pi^*_l) < \epsilon(\pi_l)$.

Based on these approximations, the $m \to s_m$ is shown to be beneficial if

$$(1-p^f_m)\lambda \sum_{l=s_m}^{L} S_l \rho(\pi_l) > \big((1-\epsilon(\pi_m)) - p^f_m\big) \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l, l' \neq m} (1-\epsilon(\pi_{l'})) \tag{31}$$

Without loss of generality, we can express p^f_m as the product of $(1-\epsilon(\pi_m)$ and p^b_m. The probability $p^b_m = p^f_m/(1-\epsilon(\pi_m))$ reflects the probability that the feedback arrives at the server before the exhaust of slaves transmission opportunities, knowing that the forward packet reached the client in-time. For eligible slaves, i.e. for slaves for which everything happens as the feedback was either lost or immediately available, p^b_m is equivalent to the probability ε_B to loose the acknowledgment on the backward channel. Based on this definition, the condition in Equation (31) becomes

$$(1-p^f_m)\lambda \sum_{l=s_m}^{L} S_l \rho(\pi_l) > (1-p^b_m) \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'})) \tag{32}$$

The inequality (32) tells whether the $m \to s_m$ is likely to be beneficial in the RD sense. Moreover, the stronger the inequality, the more benefice can be expected. As a consequence, the optimal slave leader for m, denoted s^*_m, is the descendant of m that maximizes the difference between the right and left terms of the inequality.

We now consider that a second MSR, denoted $m' \to s_{m'}$ is added to the $m \to s_m$ MSR. Without loss of generality, we assume that $s_m \preceq s'_m$, and express the condition for which the two MSRs bring a benefice in comparison with independent transmission policies. This is the case iff the Lagrangian $J_\lambda(\pi_{\vec{\{m,m'\}}})$ computed based on the two MSRs is smaller than $J_\lambda(\vec{\pi})$. Following similar developments and introducing similar definitions as above, we find a condition that is close to the one in Equation (32). Specifically, we have that $J_\lambda(\pi_{\vec{\{m\}}}) < J_\lambda(\vec{\pi}^*)$ if

$$(1-p^f_m)\lambda \sum_{l=s_m}^{s_{m'}} S_l \rho(\pi_l) + (1-p^f_m p^f_{m'})\lambda \sum_{l=s_{m'}}^{L} S_l \rho(\pi_l) > (1-p^b_m) \sum_{l=s_m}^{s_{m'}} \Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'})) + (1-p^b_m p^b_{m'}) \sum_{l=s_{m'}}^{L} \Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'})) \tag{33}$$

Based on the first order Taylor approximation, $(1-p^f_m p^f_{m'})$ and $(1-p^b_m p^b_{m'})$ are respectively written $(1-p^f_m)+(1-p^f_{m'})$ and $(1-p^b_m)+(1-p^b_{m'})$, and the condition becomes

$$(1-p^f_m)\lambda \sum_{l=s_m}^{L} S_l \rho(\pi_l) + (1-p^f_{m'})\lambda \sum_{l=s_{m'}}^{L} S_l \rho(\pi_l) > (1-p^b_m) \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'})) + (1-p^b_{m'}) \sum_{l=s_{m'}}^{L} \Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'})) \tag{34}$$

By comparing Equations (32) and (34), we observe that the benefit to draw from the $m' \to s_{m'}$ can be studied independently of other MSRs. Specifically, the optimal slave leader $s^*_{m'}$ for m' is the one that maximizes

$$(1-p^f_{m'})\lambda \sum_{l=s_{m'}}^{L} S_l \rho(\pi_l) - (1-p^b_{m'}) \sum_{l=s_{m'}}^{L} \Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'})) \tag{35}$$

among the descendants of m'. For two distinct masters m and m', we note that $p^b_m \sim p^b_{m'} \sim \varepsilon_B$. Moreover, $p^f_m = (1-\epsilon(\pi_m) p^b_m$, and we assume that $\epsilon(\pi_{m'}) \sim \epsilon(\pi_m)$. This is because, as told in Section III, a master is only transmitted once, preferably during its first transmission opportunity. As a consequence, if we assume that the time period between the first transmission opportunity of a data unit and its delivery deadline is about the same for all data units (or at least is large enough to be considered as being equivalent in terms of the probability of successful transmission), we have $\epsilon(\pi_{m'}) \sim \epsilon(\pi_m)$ for both masters m and m'. In these conditions, if $m \prec m'$ and $m' \prec s^*_m$, then $s^*_{m'} = s^*_m$. In general, if $m \prec m'$ and $s^*_m \preceq m'$, we can not say anything about $s^*_{m'}$. However, based on the condition (32), the case where $s^*_m \preceq m'$ and $s^*_{m'}$ is not equal to the first, i.e. the oldest, descendant of m' is quite unlikely. It corresponds to a case where the ratio

$$\frac{\Delta D_l \prod_{l' \preceq l} (1-\epsilon(\pi_{l'}))}{S_l \rho(\pi_l)} \tag{36}$$

encounters significant local maxima as the index l goes from a leave to the root of the acyclic dependency graph.

We conclude that when the slave leader of the oldest master is a descendant of the youngest slave leader, it is also the slave leader for the youngest master. In addition, when the slave leader of the oldest master is an ancestor of the youngest master, the first descendant of the youngest slave master is its slave leader. As all descendants of the slave leader are slaves themselves, this statement is equivalent to telling that a data unit l should be a slave either for all or none of its master candidates.

APPENDIX C
SLAVE ELIGIBILITY

This appendix explains how the eligibility issue enounced in Section II can be taken into account in practice. Given a set of relevant MSRs defined without taking slave eligibility into account, we consider the possibility for a slave s to be enfranchised *a posteriori* with respect to one or several master(s). To describe the enfranchisement process, we first consider that a single master is involved in the set of MSRs. We then generalize our developments to any master configuration.

Let m denote the index of a single master chosen among L interdependent data units, and consider the possibility to enfranchise a data unit s, chosen among the slaves of m. We first introduce the notion of useless transmission. We say that the transmission of a data unit is useless when it is triggered so late that there is (almost) no chance for the data to reach the client in-time, i.e. before its delivery deadline. An example of formal definition for a useless transmission is as follows. Let F denote the random variable corresponding to the delay experienced on the forward path by packets that are not lost. Given a parameter ν close to one, we define the time period T_{ftt} such that $P\{F > T_{ftt}\} = \nu$, and say that the transmission of data unit l is useless at time t when $t > t_{D,l} - T_{ftt}$, with $t_{D,l}$ denoting the delivery deadline of data unit l.

Based on this definition, freeing s from the mastership of m is recommended when (i) the wait for the ACK for m makes the transmission of s useless, and (ii) there exists an ancestor a of s that is also a slave for m, but for which a transmission remains useful, even after the wait for the ACK for m. Conditions (i) and (ii) respectively tell that the wait for the ACK for m penalizes s without penalizing the ancestor a. Under conditions (i) and (ii) and only under these conditions, there might be an advantage to consider a as a slave for m, while transmitting s independently of m. The corresponding MSRs are not included among relevant MSRs (because they do not respect Property 1). However, they can be derived easily based on the relevant MSRs by removing s from the set of slaves. An example is depicted in Figure 13. Figure 13(a) presents the 4 interdependent data units, and selects the first data unit to be the master. Figure 13(b) presents the 3 sets of relevant MSRs derived based on the rules defined in Section IV-B, without taking eligibility into account. Figure 13(c) defines how the last set of MSRs in Figure 13(b) is adapted when considering the enfranchisement of data unit 2. As explained above, the set of MSRs depicted in Figure(c) is likely to perform better than the last line in Figure 13(b) when the master feedback has a good chance to be received after the delivery deadline of 2, but before the delivery deadline of 1. Such a scenario is possible because the delivery deadline of data unit 2 comes earlier than the one of data unit 1.

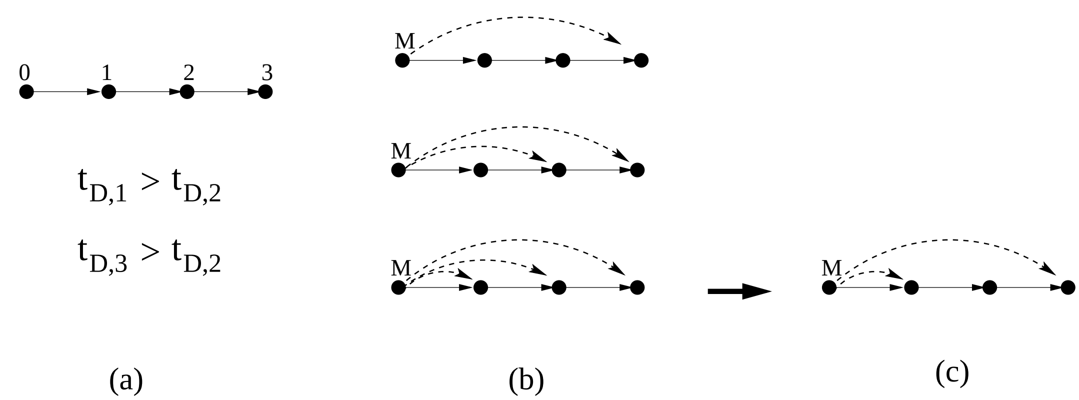


Fig. 13. Example of slave eligibility issue. (a) 4 interdependent data units. Data unit 0 is assigned to be a master M. The delivery deadline of data unit 2 comes much earlier than the deadlines for data units 1 and 3, which might cause a slave eligibility problem; (b) the 3 relevant MSRs, defined without taking slave eligibility into account. (c) The MSRs derived to circumvent the eligibility issue. Data unit 2 is enfranchised.

We now consider the case where multiple masters co-exist within a group of L interdependent data units. Similar to the single master case, the liberation of a slave s has to be considered when the transmission of s becomes useless while the transmission of one of its ancestor a, also subject to a subset of masters, remains useful. Formally, let Γ_l denote the set of masters for data unit l, and let $p_{\Gamma_l}(t)$ denote the probability that all data $m \in \Gamma_l$ are acknowledged before time t (see Appendix D). Based on these definitions, we recommend to consider the enfranchisement of a slave s as soon as the probability $p_{\Gamma_s}(t_{D,s} - T_{ftt})$ becomes smaller than a parameter ν_s, while the probability $p_{\Gamma_a}(t_{D,a} - T_{ftt})$ is larger than ν_a for at least one ancestor a of s for which $\Gamma_a \neq \{\}$. ν_s and ν_a are chosen close to one with $\nu_s < \nu_a$. Because a is an ancestor of s, we have $\Gamma_a \subset \Gamma_s$. The goal of the enfranchisement procedure is then to relax the constraint imposed on data unit s (without changing the one imposed to a) so that s has a chance to be transmitted in a useful way. In practice this is done by canceling the MSR and cascades of MSR in which s is involved. The process is illustrated by the example depicted in Figure 14. In this figure, data unit 2 is constrained by two masters. Figure 14(c) depicts two MSRs that respectively restrict or cancel the mastership constraints imposed to data unit 2. Selecting the strategy that is likely to achieve the best RD trade-offs among the multiple (two in Figure 14(c)) liberation possibilities is a complex issue. So in practice, when the eligibility of slave s is expected to be an issue, we recommend to compute the optimal dependent policies for all enfranchisement possibilities.

To conclude our discussion about eligibility issues, note that in real life streaming conditions, the playback and pre-fetch delays are generally large enough to guarantee that the initial transmission of a data unit rarely becomes useless, whatever the MSRs are. This significantly reduces the eligibility problem in practical cases. In addition, it is only worth freeing a slave s if in final the optimal policy recommend to transmit data unit s. This is because the case where s is not transmitted is already envisioned when s remains a slave. This observation reduces the eligibility problem in practical cases because most often, when the bits are so expensive that an ancestor a has an advantage to wait for a master feedback, it is better (in the RD sense) not to transmit s at all, rather than transmitting it independently of the master feedback. Cases for which this statement is not valid correspond either to cases for which descendants bring large benefit at low cost in rate, or to cases for which descendants of s are not ineligible. The first cases are often considered as pathological cases because most efficient media coders are designed to transmit most important information first. The second scenario is only possible when the descendants of s have a significantly later delivery deadline than s, which is also rare in practice.

Based on the above arguments, we conclude that cases for which freeing a slave brings a significant RD benefit are rare in practice. As a consequence, we decide not to go deeper in the study of the eligibility question. The purpose of this Appendix is to inform the reader about possible solutions to the issues raised by very heterogeneous delivery deadlines or quite unnatural allocation of rate and distortion among interdependent data units.

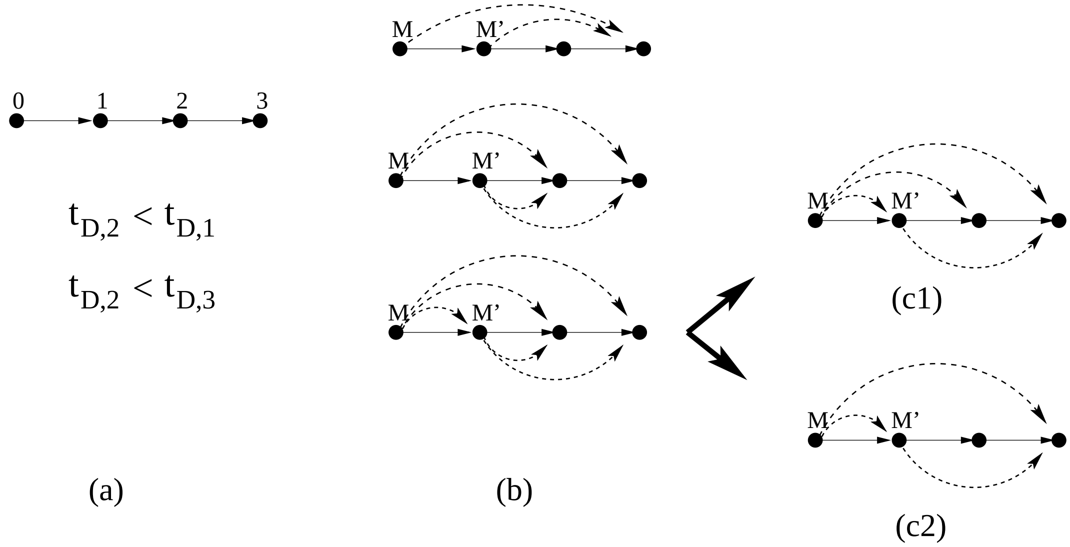


Fig. 14. Example of slave eligibility issue. (a) 4 interdependent data units. Data unit 0 and 1 are assigned to be masters M and M'. The delivery deadline comes earlier for data unit 2 than for other data units, which might cause a slave eligibility problem. This occurs when the wait for an ACK for masters M and M' penalizes data unit 2, but the wait for an ACK for M does not penalize data unit 1 (= M'); (b) The 3 relevant MSRs, defined without taking slave eligibility into account. (c) The 2 MSRs derived to circumvent the eligibility issue in the last row of (b). Data unit 2 is either partly (only M is kept as a master for 2) or completely (neither M nor M' remains masters of 2) enfranchised.

APPENDIX D
ACKNOWLEDGEMENT FEEDBACK PROBABILITIES

This Appendix considers the computation of the probability $p_{l,\Gamma_l}(j)$ that the last ACK about data units in Γ_l is received in $]t_{l,N_l-1-j}, t_{l,N_l-j}]$. The fundamental outcome of the Appendix is the fact that $p_{l,\Gamma_l}(j)$ directly depends on the data contained in Γ_l, but also on the MSRs defined among these data. For a given Γ_l, with a given set of MSRs in Γ_l, $p_{l,\Gamma_l}(j)$ is a function of the RTT random variable distribution.

Let $p_{\Gamma_l}(t)$ denote the probability that all data $m \in \Gamma_l$ are acknowledged before time t. Based on $p_{\Gamma_l}(t)$, we can write

$$p_{l,\Gamma_l}(j) = p_{\Gamma_l}(t_{l,N_l-j}) - p_{\Gamma_l}(t_{l,N_l-j-1}),\ 1 \leq j < N_l \quad (37)$$

$$= p_{\Gamma_l}(t_{l,N_l-j}), \qquad j = N_l \quad (38)$$

We now explain how to compute $p_{\Gamma_l}(t)$ as a function of the RTT variable and of the set of master/slave relationships (MSR) defined within Γ_l. A classical example of MSRs defined within Γ_l can be described by a set of disjoint master/slave dependency paths that end up in data unit l. In that case, let Υ_{Γ_l} denote the set of sources for these disjoint MSR paths, and for $m \in \Upsilon_{\Gamma_l}$, let $\ell(m,l)$ denote the length of the dependency path between m and l. Let also RTT_i denote the sum of i independent RTT random variables. Based on these definitions, we can write

$$p_{\Gamma_l}(t) = \prod_{m \in \Upsilon_{\Gamma_l}} P\{RTT_{\ell(m,l)} < t_{m,0} - t\} \quad (39)$$

where $t_{m,0}$ the first and single transmission of master data unit m.

From a practical point of view, the distribution function of a sum of RTT variable is easily estimated based on a discrete approximation of the RTT variable. Note also that a more complex MSR topology within Γ_l results in a more complex formulation of $p_{\Gamma_l}(t)$. But, whatever the topology of the MSRs in Γ_l, $p_{\Gamma_l}(t)$ can always be approximated based on the distribution function of the RTT random variable. As a consequence, for a given set of MSR defined on the group of L interdependent data units, the set of $p_{l,\Gamma_l}(j)$, $l \in \{1, ..., L\}$, $j \in \{1, ..., N_l\}$, are parameters that do not depend on the transmission policies of non-master data units.

APPENDIX E
EXAMPLES OF MASTER SELECTION

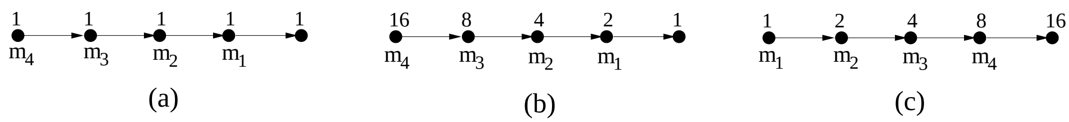


Fig. 15. Example of master assignment process for 3 different distribution of gain in distortion among 5 interdepenent data units. Bullets represent data units. Solid arrows represent dependency between data units. All data units have the same size. The gain in distortion provided by correct decoding of a data unit is defined by the number on top of its bullet. For each distortion distribution, the labels m_1, m_2, m_3, and m_4 define the order of selection of relevant masters (see text for explanations).

This Appendix illustrates the master selection procedure. Figure 15 presents an example of master assignment for 5 interdependent data units. The example corresponds to a group of data units that is extensively studied in Section V. The dependency between data units is depicted by solid arrows in Figure 15. All data units have the same size S and the same delivery deadline. For each data unit, the gain in distortion is defined by the number on top of the corresponding bullet in Figure 15. The labels m_j $(j < 5)$ below each data unit define the master selection order. As told above, masters are selected in increasing order of the ratio between the increase in distortion and the spare rate expected in return for master assignment. For data unit i, this ratio is defined by

$$\min_{s>i} \frac{\Delta D(i,s)}{\Delta R(i,s)} = \min_{s>i} \frac{\epsilon_F \Delta D_i + (1-(1-\epsilon_F)(1-\epsilon_B)) \sum_{k \geq s} \Delta D_k}{(\zeta-1)S + (5-s+1)((1-(1-\epsilon_F)(1-\epsilon_B))S} \ . \tag{40}$$

In Equation (40), we consider that data units in Figure 15 are labeled in increasing order of dependency. We now develop Equation (40) for each one of the cases depicted in Figure 15.

In Figure 15(a), the gain in distortion is equal to one for all 5 data units. As a consequence for data unit i, Equation (40) becomes

$$\min_{s>i} \frac{\Delta D(i,s)}{\Delta R(i,s)} = \min_{s>i} \frac{\epsilon_F + (5-s+1)(1-(1-\epsilon_F)(1-\epsilon_B))}{(\zeta-1)S + (5-s+1)(1-(1-\epsilon_F)(1-\epsilon_B))S} \ . \tag{41}$$

Because ζ is larger than $1/(1-\epsilon_F)$, we have

$$\frac{\epsilon_F}{(\zeta-1)S} \leq \frac{(1-\epsilon_F)}{S} \leq \frac{(1-(1-\epsilon_F)(1-\epsilon_B))}{(1-(1-\epsilon_F)(1-\epsilon_B))S} \ , \tag{42}$$

so that Equation (41) reaches a minimum for small values of $(5-s+1)$. This is the case when i is high (because then s is constrained to high values). As a consequence, for data units with constant distortion, we recommend to select masters in decreasing order of dependency.

In Figure 15(b), the gain in distortion decreases along the dependency path. As a consequence, Equation (40) is dominated by $\epsilon_F \Delta D_i/(\zeta-1)S$, and definitely decreases along the dependency path. For this reason, masters are selected in decreasing order of dependency. In contrast, in Figure 15(c), the gain in distortion increases along the dependency graph. As a consequence, Equation (40) reaches a minimum when $s = i+1$, and increases along the dependency path. For this reason, masters are selected in increasing order of dependency.

REFERENCES

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

[2] B. Girod, M. Kalman, Y. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Rochester, NY, September 2002.

[3] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Proc. ACM Multimedia*, Berkeley, CA, November 2003.

[4] A. Begen, Y. Altunbasak, and M. Begen, "Rate-distortion optimized on-demand media streaming with server diversity," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, September 2003.

[5] M. Podolsky, M. Vetterli, and S. McCanne, "Limited retransmission of real-time layered multimedia," in *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, CA, USA, December 1998, pp. 591–596.

[6] G. Matthew, M. Podolsky, M. Vetterli, and S. McCanne, "Soft ARQ for Layered Streaming Media," *Journal of VLSI Signal Processing Systems*, vol. 27, no. 1-2, pp. 81–97, February 2001.

[7] M. Kalman and B. Girod, "Rate-Distortion optimized video streaming with multiple deadlines for low latency applications," in *Packet Video Workshop*, Irvine, CA, December 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 Conference*, Snowbird, Utah, March 2004, pp. 192–201.

[9] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized media streaming," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, September 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, US Virgin Islands, December 2002.

[11] Y. Liang and B. Girod, "Prescient R-D optimized packet dependency management for low-latency video streaming," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, September 2003.

[12] E. Setton and B. Girod, "Congestion-distortion optimized Scheduling of video over a bottleneck link," in *IEEE Workshop on Multimedia Signal Processing (MMSP)*, Siena, Italy, September 2004.

[13] J. Chakareski and B. Girod, "Computing rate-distortion optimized policies for streaming media with rich acknowledgements," in *Proc. IEEE Data Compression Conference*, Snowbird, UT, April 2004.

[14] ——, "Rate-distortion optimized media streaming with rich requests," in *Proc. Packet Video Workshop*, Irvine, CA, December 2004.

[15] C.-L. Chang and B. Girod, "Receiver-based rate-distortion optimized interactive streaming for scalable bitstreams of light fields," in *Proc. IEEE Conference on Multimedia and Expo (ICME)*, Taipeh, Taiwan, June 2004.

[16] D. Li, G. Cheung, C.-N. Chuah, and S. Ben Yoo, "Joint server/peer receiver-driven rate-distortion optimized video streaming using asynchronous clocks," in *ICIP*, Singapore, October 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, March 2004.

[18] Z. Miao and A. Ortega, "Optimal scheduling for the streaming of scalable media," in *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2000.

[19] P. Ramanathan and B. Girod, "Theoretical analysis of the rate-distortion performance of a light field streaming system," in *Proc. International Picture Coding Symposium (PCS)*, San Francisco, CA, December 2004.

[20] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *IEEE Data Compression Conference (DCC)*, Snowbird, Utah, USA, March 2003.

[21] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[22] M. Rabbani and R. Joshi, "An overview of the JPEG2000 image compression standard," *Signal Processing: Image processing*, vol. 17, pp. 3–48, 2002.

[23] 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 Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 3, pp. 357–369, March 2004.

[24] M. Waschbusch, S. Wurmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3D acquisition of dynamic scenes," *The Visual Computer (Special Issues for Pacific Graphics 2005)*, , vol. 21, no. 8-10, pp. 629–638, October 2005.

[25] K. Muller, A. Smolic, M. Kautzner, P. Eisert, and T. Wiegand, "Predictive compression of dynamic 3D meshes," in *IEEE International Conference on Image Processing (ICIP)*, Genoa, Italy, October 2005.