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

PACKETIZED MEDIA STREAMING WITH COMPREHENSIVE EXPLOITATION OF FEEDBACK INFORMATION

Christophe De Vleeschouwer and Pascal Frossard

Swiss Federal Institute of Technology Lausanne (EPFL)

Signal Processing Institute Technical Report

TR-ITS-2005.014

June 15th, 2005

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.

Packetized Media Streaming with Comprehensive Exploitation of Feedback Information

Christophe De Vleeschouwer and Pascal Frossard
Ecole Polytechnique Fédérale de Lausanne (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)transmission using acknowledgement feedback. The different transmission scenarios associated to a group of interdependent media data units are abstracted in terms of a finite alphabet of policies, for each single data unit. A rate-distortion optimized markovian framework is proposed, which supports the use of comprehensive feedback information. Contrarily to previous works in rate-distortion optimized streaming, whose transmission policies definitions do not take into account the feedback expected for other data units, our framework considers all the acknowledgment packets in defining the streaming policy of a single data unit. More specifically, the notion of master and slave data unit is introduced, to define dependent streaming policies between media packets; the policy adopted to transmit a slave data unit becomes dependent on the acknowledgments received about its masters. One of the main contributions of our work is to propose a methodology that limits the space of dependent policies for the RD optimized streaming strategy. A number of rules are formulated to select a set of relevant master/slave relationships, defined as the dependencies that are likely to bring RD performance gain in the streaming system. These rules provide a limited complexity solution to the rate-distortion optimized streaming problem, with comprehensive use of feedback information. Based on extensive simulations, we conclude that (i) the proposed set of relevant dependent policies achieves close to optimal performance, while being computationally tractable, and (ii) the benefit of dependent policies is driven by the relative sizes and importance of interdependent data units. Our simulations demonstrate that dependent streaming policies can perform significantly better than independent streaming strategies, especially for cases where some media data units bring a relatively large gain in distortion, in comparison with other data units they depend on for correct decoding. We observe however that the benefit becomes marginal when the gain in distortion per unit of rate decreases along the media decoding dependency path. Since such a trend characterizes most conventional scalable coders, the implementation of dependent policies can reasonably be ruled out in these specific cases.

Index Terms

Media communication, streaming, rate-distortion, packet losses and retransmissions.

I. Introduction

Media streaming is getting quite a lot of attention from the research community, as multimedia applications certainly represent 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, that make their transmission quite challenging on channels with limited quality of service.

This paper addresses in particular the problem of streaming packetized media over a lossy packet network. Sender-driven (re)transmission over a single QoS network using acknowledgement feedback is considered. The media stream is composed of a series of possibly interdependent data units, with different contribution to the overall rendered media quality. Specifically, our work builds on the framework introduced by 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. Specifically, at each possible transmission opportunity, an optimized transmission policy for a data unit tells whether the data unit should be transmitted, or not, in absence of acknowledgment of the correct reception by the media client. The streaming system considers the dependencies between data units, and their relative importance to determine optimized independent streaming policies for each data unit.

We propose here to introduce dependencies between the streaming policies of different data units, and to use all the information available at the sender to define efficient streaming strategies. In contrast to [1], where the policies for sending a given data unit do no depend on the feedback received for other data units, we propose to extend the policy space so that the transmission policy of a given data unit can be made dependent on any feedback about the status of the streaming session. We introduce the notion of dependent policy to refer to a policy that tells whether the data unit should be transmitted or not, depending on the acknowledgments received about other dependent data units. As the space of all possible dependent policies grows exponentially in both the number of transmission opportunities and the number of dependent data units, the new optimization problem rapidly becomes intractable when the number of dependent policies grows, and when there are more than a few transmission opportunities [2], [3]. However, only of few of these dependencies relations are actually relevant in the rate-distortion optimization problem. One of the main contributions of our paper consists in providing a set of rules

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

to identify relevant dependencies, and thus to define a computationally tractable subset of the dependent policies so as to achieve rate-distortion performance that are close to the one expected from the intractable entire space of dependent policies. The comprehensive exploitation of the rich information provided by feedbacks from the receivers is expected to improve the performance of the adaptive streaming system. We show that the performance gain offered by considering dependent streaming policy is mostly interesting when the distortion per unit of rate does not decrease along the dependency path defined by the encoding system. In the same time, a careful definition of dependency relationships allows to benefit from performance improvement with a limited cost in computation.

Our work assumes that the network looses or corrupts packets at random, and that unlost packets are delivered after a random delay, using most of the formalism introduced in [1]. For arbitrary packetizations of encoded media content, it defines which packets to select for transmission, and when to (re)transmit them, so as to minimize the end-to-end distortion of the streaming system. The reason to follow the formalism presented in [1] is that it significantly advances the state of the art in streaming media systems [4]. It laid down the groundwork for recent studies on streaming media over multiple paths [5], from multiple servers [6], [7], or via intermediate proxy servers [8]. It has also been successful in handling different communication scenarios, including applications with severe delay constraints [9], [10], and streaming systems with rich client acknowledgments [11] or precise client requests [12], [13]. 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 [14], [15], [16] also formalize the scheduling decision as a partially observable Markov decision process. Such popularity certainly justifies our study of dependent policies, which builds on the original framework defined in [1].

The paper is organized as follows. Section II recalls the terminology and the notations introduced in [1]. It also reviews the solution of the rate-distortion optimized streaming with independent policies. Section III and IV propose to enlarge the space of independent policies studied in [1] to a computationally tractable fraction of the space of dependent policies. First, Section III considers the nature of the dependency between policies, and reveals that only policies obeying a strict master/slave dependency format have a chance to improve the streaming performance obtained based on independent policies. An algorithm is then proposed to compute the RD optimal policies respecting a given dependency pattern, defined based on the recommended master/slave format. Complementarily, Section IV identifies a computationally tractable subset of relevant dependency patterns among the set of patterns obeying the recommended master/slave format. Here, a dependency pattern is said to be relevant as long as it is likely to improve the RD streaming performance compared to the performance achieved by independent policies. Hence, computing the RD optimized streaming policies only for the relevant dependency patterns is expected to provide RD optimal streaming performance. Section V presents extensive simulation results that analyze the benefit of the extension of the policy space to dependent policies. It shows that the proposed strategy always improves the performance of a streaming system with independent strategies, and even performs close to an optimal based on a comprehensive search in the whole streaming policy space. It then demonstrates that the gain provided by considering dependent policies highly depends on the distribution of the distortion per unit of rate along the media sequence in realistic scenarios. Section VI concludes.

II. Rate-Distortion optimized streaming with independent policies

A. Framework

This section briefly reviews the framework and the terminology introduced by Chou and Miao in [1], to study streaming systems in the context of a lossy network. We strictly limit this preliminary section to the concepts needed to describe our own contribution in the rest of the paper. Interested readers are invited to refer to the original paper [1] for a detailed discussion and motivation of the assumptions that underly this framework.

In a streaming media system, 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. 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 previous literature about rate-distortion optimized streaming mainly use independent streaming policies for different data units. In that context, they solve the problem of how and when to transmit a group of interdependent data units in a rate-distortion optimal way. We mostly refer here to the methods introduced by Chou and Miao [1] and by Roder and al. [17], as they support our own contributions in improving the performance of the streaming system, as presented in the next sections. To derive an optimal transmission strategy for a group of interdependent data units, both works [1] and [17] rely on the solution provided to the problem of selecting an optimal policy when a single data unit is transmitted. It leads to a computationally tractable solution, under the assumption that transmissions of data units are independent; in this case, the transmission policy of a data unit does not depend on the feedback received for other data units. In contrast, the contribution of our work, presented in Sections III and IV, mainly consists in relaxing the assumption of independence between the transmission of distinct data units. We now summarize the part of the work in [1] and [17] that is relevant to our study.

The first problem discussed in [1] and [17] is the computation of rate-distortion (RD) optimal policies for the transmission of a single data unit. The l^{th} data unit with delivery deadline $t_{D,l}$, is assigned N_l transmission opportunities at time $t_{l,0}, t_{l,1}, ..., t_{l,N_l-1}$. Note that the proposed formulation assumes that a data unit is useless when it arrives after its delivery deadline, and we will use the same assumption in the rest of the paper. Interested readers are referred to [18] for a formal description of how retroactive recovery mechanisms are combined with the RD optimized streaming framework proposed in [1]. Based on the transmission opportunity assignments, a binary streaming policy vector $\pi_l = (\pi_l(0), \pi_l(1), ..., \pi_l(N_l-1)) \in \{0,1\}^{N_l}$ defines the transmission instants of the l^{th} data unit. Specifically, $\pi_l(i) = 1$ means that the l^{th} data unit should be sent at opportunity i if no acknowledgment has been received before time $t_{l,i}$. The notion of cost-error optimal policy is then introduced based on the following error and cost definitions. The error $\epsilon(\pi_l)$ for policy π_l is 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 is further defined as the expected number of data unit transmissions, as given by:

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

A policy π_l^* is said to be optimal if there exists no policy π_l such that $\epsilon(\pi_l) \leq \epsilon(\pi_l^*)$ and $\rho(\pi_l) < \rho(\pi_l^*)$. A branch and bound algorithm is proposed in [17], to compute both the entire set of optimal policies, and the subset of optimal policies π_l' whose cost-error points $(\rho(\pi_l'), \epsilon(\pi_l'))$ lie on the lower convex hull of the set of all achievable (ρ, ϵ) points. These policies that minimize the Lagrangian cost $J_\lambda(\pi_l) = \epsilon(\pi_l) + \lambda\rho(\pi_l)$ for $\lambda > 0$, are computed with a worst case complexity of $O(N_l 2^{N_l})$ [17].

The second problem consists in the selection of RD optimal policies for the transmission of a group of interdependent data units. Recall that, in [1] and [17], the authors assume *independent* transmissions of data units. As a consequence, the transmission policies for the 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 single 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 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 computationally simple solution to find the policy vectors $\vec{\pi}$ minimizing $J_\lambda(\vec{\pi}) = J_\lambda(\pi_1, ..., \pi_L) = D(\vec{\pi}) + \lambda R(\vec{\pi})$, for $\lambda > 0$, is proposed in [1]. The algorithm is based on an iterative descent algorithm that minimizes $J_\lambda(\pi_1, ..., \pi_L)$ one policy at a time, keeping the other policies fixed. Let $\vec{\pi}^{(0)} = (\pi_1^{(0)}, ..., \pi_L^{(0)})$ denote the initial policy vector. A sequence of policy vectors $\vec{\pi}^{(k)}$

is then computed as follows: (i) select $l_k \in \{1, ..., L\}$, (ii) $\forall l \neq l_k$, set $\pi_l^{(k)} = \pi_l^{(k-1)}$, and (iii) let

$$\begin{aligned} \pi_{l_k}^{(k)} &= \arg\min_{\pi} J_\lambda(\pi_1^{(k)}, ..., \pi_{l_k-1}^{(k)}, \pi, \pi_{l_k+1}^{(k)}, ..., \pi_L^{(k)}) && (5) \\ &= \arg\min_{\pi} G_{l_k}^{(k)} \epsilon(\pi) + \lambda S_{l_k} \rho(\pi) && (6) \end{aligned}$$

where

$$G_{l_k}^{(k)} = \sum_{l_k \preceq l'} \Delta D_{l'} \prod_{l'' \preceq l', l'' \neq l'} (1 - \epsilon(\pi_{l''}^{(k)})). \qquad (7)$$

In practice, $\pi_l^{(0)}$ is set to $\{1\}^{N_l}$ for all $l \in \{1, ..., L\}$, and data units l_k are selected in a round robin order under the condition that a data unit can only be selected after all its ancestor have been selected for the on-going round. Because $J(\vec{\pi}^{(k)})$ is non-increasing and additionally bounded below by zero, convergence to a local optimum is guaranteed [1]. We further conjecture that the iterative descent algorithm converges to a global optimum in the particular case where the ratio $\frac{\Delta D_l}{S_l}$ decreases with the layer index in scalable coding. Alternatively, Roder and al. [17] have proposed a branch and bound algorithm to compute a global optimum to the choice of transmission policies. However its complexity is certainly to important for on-line applications.

C. Limitations of the independent streaming policies

We have seen that the most important works in rate-distortion optimized media streaming assume that the system defines independent streaming policies for different data units. In other words, the transmission strategy for one data unit is only altered by the reception of an acknowledgement for that particular data unit. This assumption is mainly justified by the aim at setting an optimization problem that is computationally tractable, and does not induce the necessity to consider all the possible dependencies between data units.

However, for media packets, the correct decoding of a data unit is often tied to the reception of another data unit. As a consequence, the information that a packet has been received should impact the strategy for sending another 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 received about 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 participate to decrease the uncertainty about the system status, and should be considered 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, so as to take into account the most recent information from feedback on any of the data packets. Such a step-wise approach handles the dependency a posteriori, i.e. after feedback reception. It results in suboptimal solutions. As an example, in the above scenario, suboptimality results from the fact that the update of the policy of j (as a function of the feedback for i) is not taken into account to define the initial transmission policy for i.

In contrast, our work considers the dependency a priori. Obviously, the system can unfortunately not consider all the possible dependency relationships in choosing the streaming policies, without rapidly facing an intractable optimization problem. Among the main contributions of our work, we show in the next sections that dependent streaming policies indeed bring a benefit to the rate-distortion optimized streaming problem. But we also show that only a few dependency relationships between media packets are relevant for defining optimized streaming policies. In the next sections, we identify these relevant dependency relationships, and we show that the hereabove optimization problem can be extended to consider dependent streaming policies.

III. Rate-Distortion optimized streaming with dependent policies

The section explores how the transmission policies of some data units, referred to as slaves, may advantageously be forced to depend on the feedback received for other data units, referred to as masters. It demonstrates that all dependent policies that are expected to provide a significant RD benefit compared to independent policies can be defined exclusively in terms of master/slave relationships (MSRs) for which the master is only transmitted once and for which a slave is only transmitted after reception of all its masters ACKs. Based on this results, we then extend the formalism presented in Section II-B to compute the optimal transmission policies corresponding to a given set of master/slaves relationships.

A. Master and Slave Policies

We introduce the notion of *master* and *slave* data units, in order to characterize the dependency relationship between the streaming policies of media packets.

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

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

Hence, a given data unit can be a slave, a master, both of them, or none of them. Dependent streaming policies based on master/slave relationships (MSR) might result in improved rate-distortion (RD) trade-offs, since the policy of a slave spares some rate at the cost of an increased distortion. Intuitively, the rate allocated to the slave is smaller in absence than in presence of master ACKs, and the increase in distortion is a consequence of the relatively fewer transmissions in absence of ACKs, and of the time elapsed before receiving the master ACKs. A master/slave relationship 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.

We now argue that all dependent transmission policies that are expected to provide a significant benefit in terms of rate-distortion trade-off, can be defined exclusively in terms of MSRs for which the master is only transmitted once and for which a slave is only transmitted after reception of all its masters ACKs. In other words, the following assumptions allow restricting the set of dependent transmission policies that are of interest in the RD sense: (i) a master data unit is only transmitted once and (ii) a slave is not transmitted as long as the master ACK has not been received.

To justify our conjecture about a unique transmission of master data unit, we reason by contradiction. If the master data unit is transmitted several times, it has a close-to-one probability to reach the client before its delivery deadline. 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 will reach the client in-time. In other words, it means that when a master candidate has a close-to-one probability to arrive in time, a slave policy that does not depend on the master feedback is expected to achieve close to optimal RD performance. We conclude that a MSR is only expected to significantly improve the RD performance obtained based on independent streaming policies, when there is a good chance for the master data unit to be lost or to arrive out of delay. 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 in a relevant MSR is only transmitted once (i.e., without retransmission).

Our second assumption proposes to restrict the analysis of dependent policies to policies for which a slave is only considered for transmission upon reception of the master acknowledgment, i.e., a slave is not transmitted if no ACK has been received for its master(s). Intuitively, this decision is motivated by the fact that the cost in rate of a streaming policy is dominated by its initial transmission, because retransmissions only happen in absence of acknowledgment. A significant gain in terms of rate is thus only expected for MSRs that cancel the initial transmission of the slave in absence of master ACKs. This result 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 remaining of the section explains how to compute the set of RD optimal dependent policies conforming to a pre-defined set of master/slave relationships.

B. RD optimal dependent policies

This section explains how to compute the set of RD optimal dependent scheduling policies following a given a set of master/slave relationships (MSRs). Based on Section III-A, we restrict our search for RD optimal policies to a subset of potentially advantageous policies that (i) transmit masters only once, and (ii) only transmit slaves upon reception of all its masters ACKs. The proposed algorithm is based on an iterative gradient descent algorithm that generalizes the approach proposed in [1] and presented in Section II-B.

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. Γ_l denotes the set of masters for the l^{th} data unit. It means that the l^{th} data unit can only be transmitted after all $m \in \Gamma_l$ have been acknowledged. The policy $\overline{\pi}_{l,\Gamma_l}$ for the l^{th} data unit is then defined by a set of N_l sub-policy vectors $\pi_{j,l,\Gamma_l} \in \{0,1\}^j, j \in \{1,...,N_l\}$. Each sub-policy vector defines the transmission policy for the l^{th} data unit when j transmission opportunities remain available after all masters of l have been acknowledged. Specifically, π_{j,l,Γ_l} becomes effective when the latest acknowledged data in Γ_l is acknowledged within $]t_{l,N_l-1-j}, t_{l,N_l-j}]$, and $\pi_{j,l,\Gamma_l}(i) = 1,\ 0 \leq i < j$, means that the l^{th} data unit has to be sent at opportunity $i + N_l - j$ if it has not yet been acknowledged. The policy vector for the group of L interdependent data units is denoted $\vec{\pi_\Gamma} = (\overline{\pi}_{1,\Gamma_1}, ..., \overline{\pi}_{L,\Gamma_L})$, where $\Gamma = \{\Gamma_1, ..., \Gamma_L\}$.

In order to compute the rate $R(\vec{\pi_\Gamma})$ expected for $\vec{\pi_\Gamma}$, we define $p_{l,\Gamma_l}(j)$ to be 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 $p_{l,\Gamma_l}(j)$, $l \in \{1,...,L\},\ j \in \{1,...,N_l\}$ only depends on master data units, and not on the transmission policies of non-master data units. As a consequence, given the set of MSRs defined for the group of L interdependent data units, the parameters $p_{l,\Gamma_l}(j)$ can be pre-computed. Appendix D explains how to compute $p_{l,\Gamma_l}(j)$ as a function of the RTT variable distribution, and of the set of MSRs defined among the data in Γ_l. Given the values $p_{l,\Gamma_l}(j)$, and defining $\rho(.)$ and S_l as in Section II-B, we have

$$R(\vec{\pi_\Gamma}) = \sum_{l=1}^{L} \sum_{j=1}^{N_l} p_{l,\Gamma_l}(j) \rho(\pi_{j,l,\Gamma_l}) S_l \ . \tag{8}$$

Next, in order to define the distortion $D(\vec{\pi_\Gamma})$ expected for $\vec{\pi_\Gamma}$, we introduce a random vector ψ, such that $\psi(l)$ (with $0 < l \leq L$) defines the number of transmission opportunities still available for the l^{th} data unit after all data in Γ_l have been acknowledged. We define Ψ to be the set of all possible realizations of ψ. The probability of occurrence of $\psi \in \Psi$, p_ψ, clearly depends on the master units transmission policy, on the dependency between master data units, and on the RTT random variable distribution. However it does not depend on the non-master data units. Since the master transmission policies are fixed to a single transmission, and because we consider a pre-defined set of MSRs, the probability p_ψ can be considered as a fixed parameter, independent of the transmission policy assigned to non-master units. We thus have :

$$D(\vec{\pi_\Gamma}) = D_0 - \sum_{\psi\in\Psi} p_\psi \sum_{l=1}^{L} \Delta D_l \prod_{l'\preceq l} (1-\epsilon(\pi_{\psi(l'),l',\Gamma_{l'}})) \ . \tag{9}$$

Based on these definitions, the computation of rate-distortion optimal convex-hull policies is a direct extension of the algorithm proposed in [1]. The purpose is still to compute the policy vectors $\vec{\pi_\Gamma}$ minimizing $J_\lambda(\vec{\pi_\Gamma}) = J_\lambda(\pi_{1,\Gamma_1}, ..., \pi_{L,\Gamma_L}) = D(\vec{\pi_\Gamma}) + \lambda R(\vec{\pi_\Gamma})$ for $\lambda > 0$. The algorithm minimizes one policy at a time, keeping the other policies fixed. In contrast to the case of independent policies, here the algorithm has to minimize every non-master subpolicy, keeping the other fixed. The sequence of policy vectors $\vec{\pi_\Gamma}^{(k)}$ is computed as follows. Select $l_k \in \{1, ..., L\}$ and $j_k \in \{1, ..., N_L\}$. $\forall (l, j) \neq (l_k, j_k)$, set $\pi^{(k)}_{j,l,\Gamma_l} = \pi^{(k-1)}_{j,l,\Gamma_l}$, and let

$$\begin{aligned}\pi^{(k)}_{j_k,l_k,\Gamma_{l_k}} &= \arg\min_\pi J_\lambda(\pi^{(k)}_{1,1,\Gamma_1}, ..., \pi^{(k)}_{j_k-1,l_k,\Gamma_{l_k}}, \pi, \pi^{(k)}_{j_k+1,l_k,\Gamma_{l_k}}, ..., \pi^{(k)}_{N_l,L,\Gamma_L}) &(10)\\ &= \arg\min_\pi G^{(k)}_{j_k,l_k}\epsilon(\pi) + \lambda p_{l_k,\Gamma_{l_k}}(j_k) S_{l_k}\rho(\pi), \qquad \pi \in \{0,1\}^{j_k} \ , &(11)\end{aligned}$$

where

$$G^{(k)}_{j_k,l_k} = \sum_{\psi\in\Psi:\psi(l_k)=j_k} p_\psi \sum_{l_k\preceq l'} \Delta D_{l'} \prod_{l''\preceq l', l''\neq l_k} (1-\epsilon(\pi^{(k)}_{\psi(l''),l'',\Gamma_{l''}})) \ . \tag{12}$$

In practice, l_k and j_k 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 ancestor 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 set of masters associated to a data unit remains constant or enlarges as the data unit moves along the path of descendance. Such a feature reduces the cardinality of Ψ (because the slaves share common masters) and simplifies p_ψ computation.

To conclude this section, it is worth noting that Equations (8), (9) and (11) only hold because the policies associated to masters are fixed (to a single transmission). If it was not the case, $p_{l,\Gamma_l}(j)$ and p_ψ would depend on master policies, which would strongly couple the master and slave policies, resulting in a computationally intractable problem.

IV. Relevant master/slave relationships

In Section III, we have explained how to compute the RD optimal scheduling policies that respect the constraints imposed by a given 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 masters, which selects M master among the L data units, there are $2^{(L-1)M}$ possible definitions of MSRs. This is far too large to envision an exhaustive search among the entire MSRs space. Hence, we are interested in 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 becomes to define the smallest complete set of relevant MSRs. As an outcome of this section, we show that the relevant MSRs are tightly connected to the ancestor/descendant relationships defined among data units, and we propose a methodology that assigns at most $L \times O((L/B)^B)$ relevant MSRs to a group of L interdependent data units characterized by an acyclic dependency graph composed of B disjoint branches. The result is obtained in two steps. First, we assume that the set of master data units is defined a priori, and we study how slaves are assigned to these masters. Second, we consider the master selection problem, and propose a greedy algorithm to define a sequence of relevant sets of master data units.

A. 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.

The relation between a master and a slave is not easy to formalize rigorously. In particular, it is difficult to apprehend the exact impact of the degradation of (ϵ, ρ) trade-offs introduced in Section II-B, which is caused by the wait for master feedback. In our search for relevant MSRs, we propose to circumvent that problem by assuming that the slaves (ϵ, ρ) trade-offs are not significantly affected by the wait for master ACKs. That assumption is referred to as the (ϵ, ρ) *non-degradation assumption*. It simply means that, upon reception of the master feedback, a slave data unit can be transmitted with roughly the same (ϵ, ρ) performance as if it is transmitted without waiting for master feedback. In other words, everything happens as if the feedback was either lost or received instantaneously.

We now discuss the validity of the non-degradation assumption, for a slave involved in a given set of MSRs. When the time to wait for feedback is small in comparison with the time available before expiration of the slave delivery deadline, the assumption is certainly valid. Alternatively, when the time to wait for a feedback is so large that it causes significant degradation of the slave (ϵ, ρ) trade-offs, it is very likely that a policy that would transmit the slave independently of its master would result in optimal RD trade-offs. So, 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.

Based on that discussion, we propose the following approach to define relevant MSRs. As a first step, a set of relevant MSRs are defined based on the non-degradation assumption. Then, as a second step, the slave eligibility issue is handled a posteriori, by enfranchising the ineligible slaves identified in each set of relevant MSRs defined based on the non-degradation assumption.

In the rest of the section, we identify three necessary rules imposed by ancestor/descendant dependency on the definition of relevant MSRs. Our discussion relies on the non-degradation assumption. 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 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 then merged to define the set of relevant master-slave configurations associated to a pre-defined group of masters. Finally, the slave eligibility issue is considered. The enfranchisement strategy is briefly discussed, and is formally defined in Appendix C.
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 $m \rightarrow s$ relation affects the descendants of s, we first make the (ϵ, ρ) non-degradation assumption, i.e., we neglect the delay induced by waiting for the master feedback. In that case, we can show that $m \rightarrow s$ implies $m \rightarrow j$ for all $s \preceq j$, i.e., for all j that is a descendant of s. By definition, a descendant of s can only be decoded if s reaches the client before its delivery deadline. Obviously, this only happens when s is transmitted; as a consequence of $m \rightarrow s$, this is only the case when the feedback for the m^{th} data unit has been received. In final, it means that the descendants of s can not be decoded if the feedback for m has not been received. For this reason, and because of the non-degradation assumption, there is no advantage for a descendant of s to be transmitted when the feedback for m is not available. The descendant of s therefore becomes a slave for m. Note that this does not prevent s or its descendants to become 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 can therefore study a simple case in which an ancestor data unit a becomes a slave for one of its descendant d (see Appendix A). In this case, it can 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. More generally, is can further be shown that the $d \rightarrow a$ MSR 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 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 an average 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 last 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 likely to be beneficial. Based on the non-degradation assumption, the development in Appendix B estimates the gain in rate and increase in distortion respectively due to $m \rightarrow s$ only, and to both $m \rightarrow s$ and $m' \rightarrow s$. It shows that 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. Altogether it means that, if waiting for an ACK to m is beneficial, then waiting for m' to be acknowledged also brings an advantage in the RD sense. Intuitively, this is due to the fact that 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. To be complete, it has however to be noted that, for non-monotonical evolution of the $\Delta D_l/S_l$ relation along the path of descendance, this statement is only strictly valid when m is a descendant of m' (see Appendix B). For the sake of simplicity, we nevertheless omit this refinement, and admit that all RD optimal convex-hull points can be computed by considering that a descendant of multiple master candidates is either a slave for all of them, or is transmitted independently of all of them.

The rules 1, 2 and 3, provide the toolbox for the definition of relevant MSRs for a given pre-defined set of 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 is further described as follows.

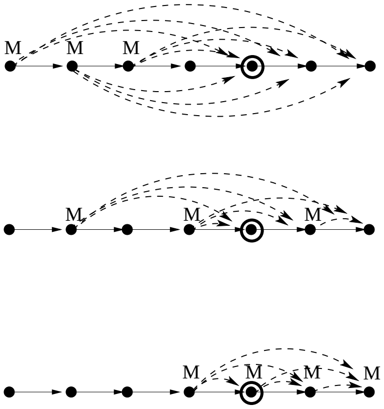


Fig. 1. Three examples of relevant MSRs definition for a branch of 7 interdependent data units. Each example considers distinct a priori selections of master candidates M. For each case, the set of relevant MSRs is defined based on the choice of the oldest slave, and follows the building properties explained in the text. Specifically, all descendants of the oldest slave are slaves themselves, and all master candidates that are ancestors of a slave are master 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.

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. Indeed, 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 $\#\Phi_i$ 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 uses an example to define the mechanism used to compute the number of relevantMSRs configurations for a complex acyclic dependency graph. It is easy to derive from that mechanism 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.

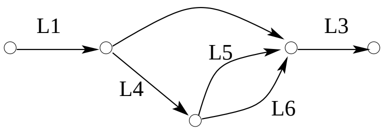


Fig. 2. Example of acyclic dependency graph, and computation of the number of slave assignment possibilities associated to that graph. Labels L_k on each link of the graph refer to the number of data units involved in the corresponding link. The graph contains 3 branches, i.e. it offers 3 different paths to connect the root to the leaf. For that example, the total number of slave assignment considered by our algorithm is equal to $L_1 + L_2 \times (L_4 + L_5 \times L_6) + L_3$.

The MSRs resulting from scanning the acyclic graph are then verified *a posteriori*, in order to ensure that the slave eligibility condition enounced earlier in this section is not violated. The eligibility question becomes relevant when the time lost in waiting for master ACKs can not be neglected. In particular, it refers to the sub-optimality arising when the wait for masters ACKs (=slaveship) strongly penalizes the transmission of some data units (referred to as problematic data units), while being accepted for some ancestors of these data units. The master/slave configuration to recommend in that situation would assign ancestors to be slaves while leaving some of their problematic descendants free. However, such configurations are not investigated by relevant MSRs (because of Property 1). To take them into account, Appendix C considers the possibility for a slave s that is a descendant of other slaves to be enfranchised *a posteriori* with regards to one or several master(s). As a main result, Appendix C reveals that the enfranchisement of a descendant slave is only expected to bring a significant RD benefit when the streamed media content is characterized by a gain in distortion (per unit of rate) that increases along the dependency path or by delivery deadlines that significantly increase beyond the enfranchised slave (along the dependency path). Intuitively, this is because in these conditions it might be worth transmitting s independently of other data units (rather than not transmitting it at all), because it brings a large gain in distortion, directly or indirectly through its non-problematic descendants. This scenario is however rare in practice. Most often, the gain in distortion decreases along the dependency path and the delivery deadlines of a group of interdependent data units come close to each others. For this reason, we do not provide here a deeper analysis of the eligibility problem, and we rather refer the reader to the Appendix for a more detailed development. In the next section, the proper selection of sets of masters that are likely to support relevant MSRs, is investigated, as the complementary problem to the slave assignment.

B. 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. We propose a greedy algorithm to define a sequence of RSMs. Let Λ denote the set of L interdependent data units. Starting from an initial empty subset $\Omega_0 = \{\}$ of relevant masters, the sequence of RSMs Ω_k is computed as follows. At each step k, the iterative algorithm selects the data unit m_k in $\Lambda \setminus \Omega_{k-1}$ that minimizes the expect ratio between the increase in distortion and the gain in rate, when m_k is selected as a master. Ω_k is then set to $\Omega_{k-1} \cup \{m_k\}$. In the rest of the section, we motivate the greedy approach and explain how to implement it in practice.

The incremental approach is motivated by the observation that, when the bits are cheap, there is little advantage to introduce master/slave relationships among data units to improve the RD trade-off. On the contrary, as bits are becoming more expensive, more data units are likely 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. When involved in one or multiple MSR(s), a slave accept to loose some gain in distortion (waiting for an ACK penalizes the slave transmission) to spare some bit budget (the slave is only transmitted upon ACK(s) reception). Such behavior only makes sense when the bits are expensive.

In practice, the process used to select the data unit m_k to add to the set of relevant masters Ω_{k-1} at each step k of our proposed greedy algorithm 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 increase in distortion and the decrease in rate expected when forcing data unit i to be master for s and its descendants, in comparison with the rate and distortion expected for independent policies. 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

$$m_1 = \arg\min_{i \in \Lambda} \left(\min_{s \succ i} \frac{\Delta D(i,s)}{\Delta R(i,s)} \right). \tag{13}$$

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. Here, ϵ_B denotes the probability of loss on the backward path. Based on the above developments, and defining ΔD_k and S_k as in Section II-A, we approximate $\Delta D(i,s)$ and $\Delta R(i,s)$ as follows

$$\Delta R(i,s) \sim (\zeta - 1) S_i + (1 - (1-\epsilon_F)(1-\epsilon_B)) \zeta \sum_{k \succeq s} S_k. \tag{14}$$

and

$$\Delta D(i,s) \sim \epsilon_F \Delta D_i + (1-(1-\epsilon_F)(1-\epsilon_B)) \sum_{k \succeq s} \Delta D_k. \tag{15}$$

Next, we consider the possibility to define RSMs with more than one masters. For that purpose, we assume that the increase in distortion and decrease in rate resulting from a MSR assignment is additive. The assumption is coarse, but is acceptable as long as it is related to the selection of promising masters and not to the computation of optimal policies. It simplifies significantly the RSMs definition. Specifically, additivity decouples the impact of multiple masters, both in terms of distortion and rate. As a consequence, masters can simply be selected in increasing order of expected increase in distortion per unit of spare rate when a single master is selected. Finally, we formally have the following master selection condition :

$$m_k = \underset{i \in \Lambda \setminus \Omega_{k-1}}{\arg\min} \left(\min_{s \succ i} \frac{\Delta D(i,s)}{\Delta R(i,s)} \right) . \tag{16}$$

To illustrate the master selection procedure, Appendix E explains how the sequence of RSMs is computed for a group of interdependent data units organized in a hierarchy of layers. That particular example is extensively studied in Section V. In particular, the results presented in Section V-B.2 validate our proposed master selection methodology in the sense that the set of masters corresponding to the optimal dependent policies derived based on a comprehensive search among all possible dependent policies are identical to the sets of masters defined based on Equation (16).

C. *Summary and discussion*

Algorithm 1 Search for optimal set of dependencies.

```
Initialization: $\Omega_0 = \{\}$, $k = 0$, $L$ is the number of interdependent data units.
Best policy ← no transmission at all.
while $k < L$ do
    for all relevant MSRs defined w.r.t. the $\Omega_k$ 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 $m_{k+1}$
    $\Omega_{k+1} \leftarrow \Omega_k \cup \{m_{k+1}\}$
    $k \leftarrow k+1$
end while.
```

This section now recapitulates the hereabove developments for the selection of relevant MSRs. The search for the optimal set of (in)dependent policies can be summarized in an iterative algorithm, as given in Algorithm 1. We can make the following observations about the iterative process:

- 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-A, 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. As explained in Section IV-A, the total number of MSRs configuration 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 [2].
- The iterative process stops when all data units belong to Ω_k, i.e., when $k = L$.

As a final remark, note that we do not claim that it is impossible to find a scheduling policy that achieves an optimal RD trade-off without satisfying the features defined in Section IV. However, the developments provided show that the set of dependent policies built on these characteristics, include most of the dependent policies susceptible to achieve a significant RD benefit in comparison to the set of independent transmission policies. As a consequence, we can reasonably assume that a scheduling policy that does not fulfill the above rules does not significantly outperform policies derived based on these rules. This assumption is confirmed below by the results presented in Section V.

V. Simulation Results

A. *Overview*

This section presents the rate-distortion performance of the streaming system proposed in the previous sections. The benefit of dependent streaming policies is evaluated, and the partial search solution, based on the selection of relevant master/slave relationships (see Section III) is compared to the independent streaming of data units [1], and to a performance upper-bound based on an exhaustive search (when possible).

The framework considered is a packet-based network with acknowledgment feedback, whose model has been described in Section II-A. Packets are lost randomly and independently on the forward (backward) path, with a probability ϵ_F (ϵ_B). The forward (F) and backward (B) transmission delays are modeled as a shifted exponential random variable with mean μ_F (μ_B) and shift $\kappa_F = \mu_F/2$ ($\kappa_B = \mu_B/2$). The rate-distortion performance are presented for several groups of interdependent data units (corresponding to typical layered or MPEG streams), and for multiple streaming scenario (different loss and delay patterns).

Our simulations reveal that:

- the proposed space of relevant policies outperforms the space of 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 possible, the proposed subspace of relevant policies results in performances similar to a full search within the entire space of policies. This validates our methodology, since it demonstrates that the proposed subspace of relevant policies includes the policies that impacts the rate-distortion performance of the streaming system.

B. Layered stream

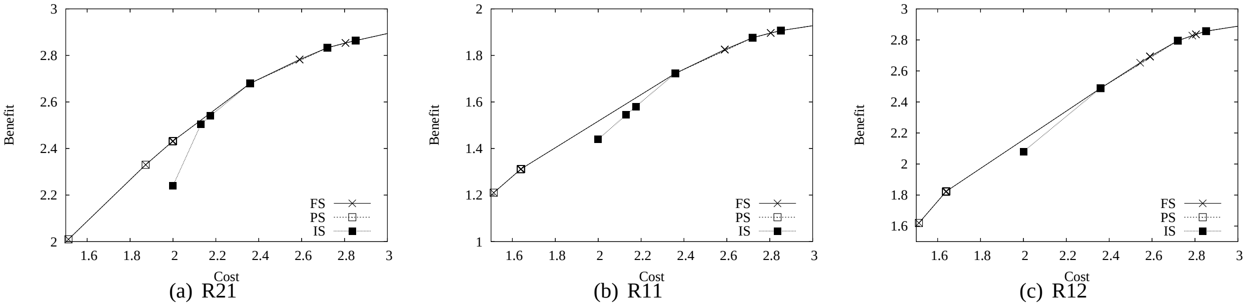
This section considers the streaming of identical and equidistant frames that are (de)coded independently of each others. The frame rate is set to 20 fps. Each frame is composed of L data units, organized in a hierarchy of layers. All data units have a unitary size. The decrease in distortion associated 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 represent most of the practical streaming scenarios, 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, 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 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 infinite delay, and (iii) a system with N layers, and finite transmission delay. The first two scenarios are quite restrictive, but allow for a full-search among all possible independent and dependent transmission policies. As a consequence, we are able to provide a comprehensive comparison of the streaming performance obtained based on 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 more realistic. It considers that the frames are composed of any number of layers, and that a finite transmission delay is available before expiration of the frame delivery deadline. We compare the RD performance obtained based on the subspace of relevant policies with the one derived based on independent policies. Our simulations reveal that the amount of RD benefit provided by the policies fitting the proposed set of relevant MSRs highly depends on the distribution of the size S_l and the distortion ΔD_l of the data units.

1) Two layers: In the first scenario, each frame is composed of two layers. A finite number (6) of equidistant transmission opportunities is considered for each data unit. Figures 3 and 4 show the RD convex-hull corresponding to 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), for three distortion templates.

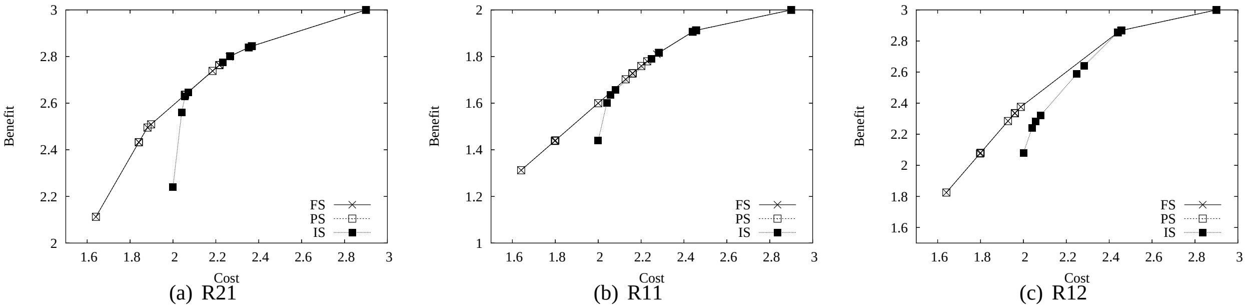
It can be observed that the proposed partial search achieves the same performance as the full search. Figure 3 considers symmetric forward and backward channels, while Figure 4 assumes that there are no losses on the backward channel. The second scenario is realistic because small acknowledgment packets can be protected efficiently against losses and errors. The comparison of these two cases shows 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 because the ancestor is not available at the client. In the 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.

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 infinite (or at least quite large in comparison with the round trip time). Whilst 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. In this case, the delay available before deadline is infinite, and the time between successive transmissions of a given data unit can also be considered as infinite, so that the transmission delay becomes negligible. As a consequence, the transmission policy does not have to specify the exact time instants at which a data unit is transmitted, but only has to tell how many times a data unit is (re)transmitted in the absence of any acknowledgment (ACK) for



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

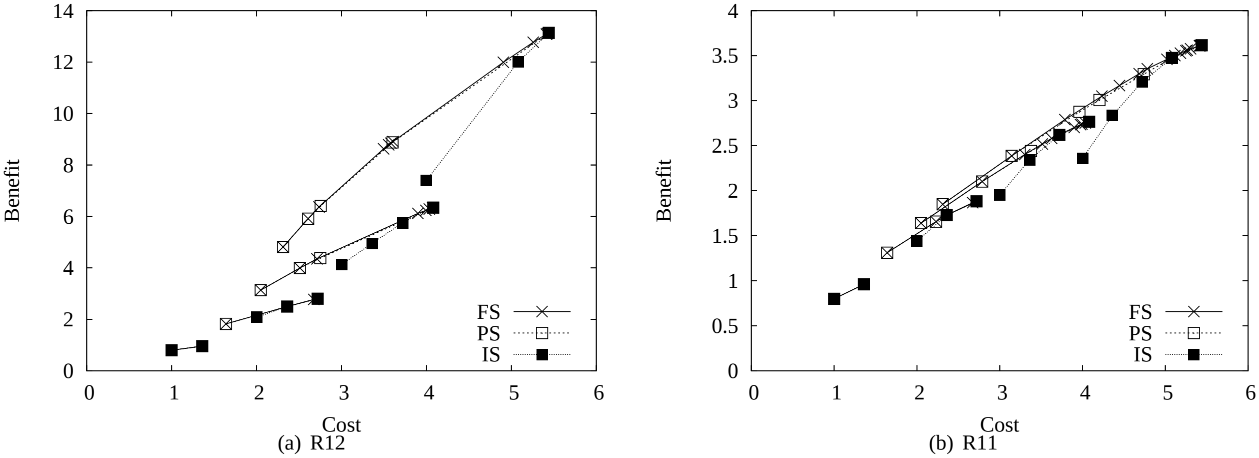


(a) R21 (b) R11 (c) R12

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

previous transmissions. The probability associated to the reception or non-reception of an ACK only depends on the probability of loss on the forward and backward paths, not on the transmission delay pdf. As a consequence, the dependent policy is much simplified as it just has to tell how many times the slave data unit has to be transmitted in presence or absence of master(s) ACK(s). The time at which ACKs are received has no importance in that particular case.

Figures 5, 6 and 7 compare the performance obtained based 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 a specific distortion template, i.e., either R11, R12, or R21. In these figures, the convex-hull corresponding to A active layers is computed based on the policies that activate the first A layers encountered along the dependency graph.



(a) R12 (b) R11

Fig. 5. RD convex-hulls 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). The time interval between two opportunities is infinite. The channel conditions are defined by $\varepsilon_F = \varepsilon_B = 0.2$.

Figures 5 and 6 compare, for different distortion templates, the convex-hulls computed based on the different streaming strategies. Between one and four active layers are considered in Figure 5, while only 3 and 4 active layers are depicted

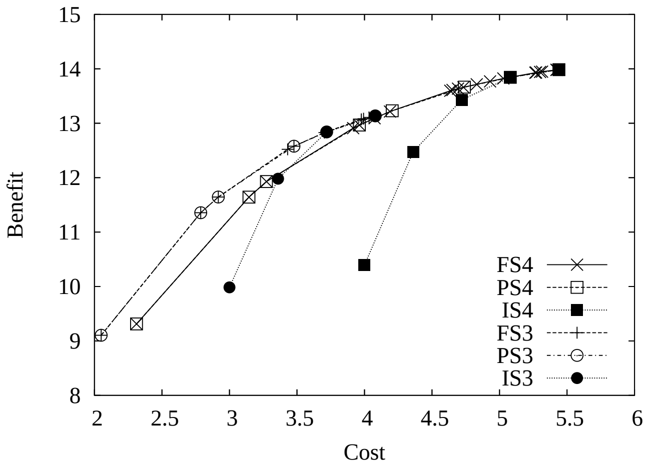


Fig. 6. RD convex-hull. R21. The convex-hulls corresponding to 3 and 4 active layers are plotted. The time interval between two opportunities is infinite. The channel conditions are defined by $\varepsilon_F = \varepsilon_B = 0.2$.

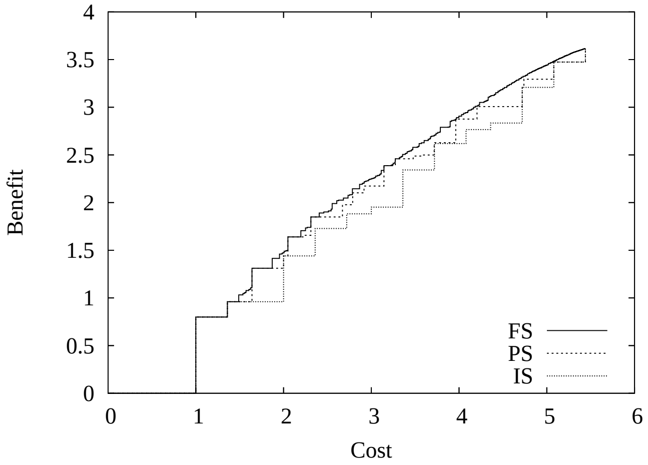


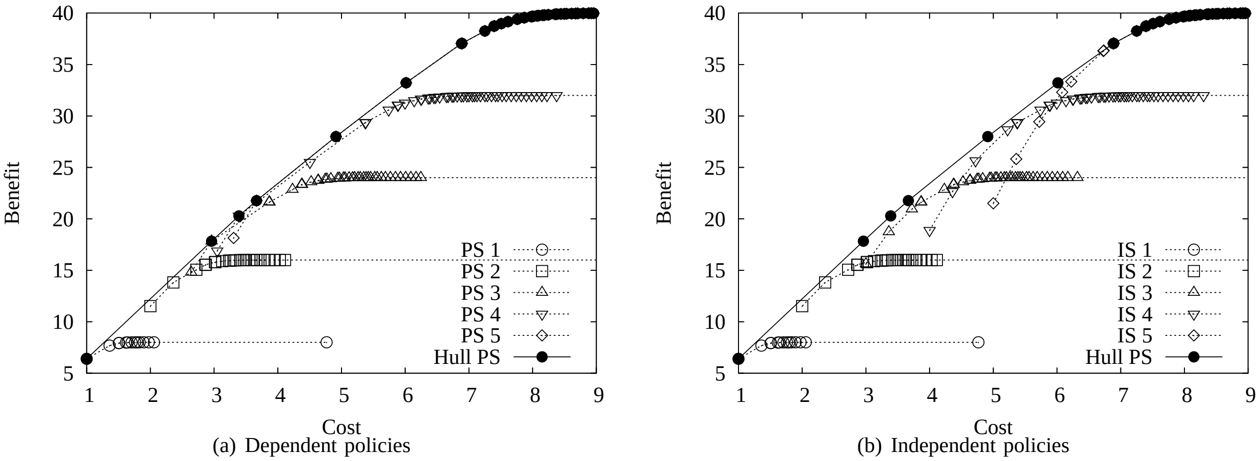
Fig. 7. Optimal RD points (not necessarily on the convex-hull). The RD points sustained by all possible numbers of active layers have been considered, for the R11 distortion template. The time interval between two opportunities is infinite. The channel conditions are defined by $\varepsilon_F = \varepsilon_B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

in Figure 6. 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 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 are identical to the sets of masters defined based on Equation (16). These simulation results also reveal that the gain provided by PS over IS is highly dependent on the relative distribution of distortion between layers. This observation becomes particularly clear when considering an imaginary *global convex-hull*, overwhelming the convex-hulls derived for all possible number of active layers.

As expected, the performance gap between the global FS and IS convex-hulls decreases when going from the R12 to R11 and R21 distortion templates. It means that there is only little benefit in dependent policies when the distortion per unit of cost decreases along the dependency path. In that case it is better (from a RD point of view) to retransmit the valuable ancestor data (when no ACK has been received), rather than to send descendants that only provide a small gain in distortion. This observation is confirmed by the next simulation results.

Finally, Figure 7 extends Figure 5 (b) and presents the RD optimal points (not necessarily on the convex hull), obtained with the three streaming strategies. We observe that PS and IS result in abrupt drops of benefit as the rate decreases, and is 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 not a surprise as 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 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, but that the delay available before a data unit delivery deadline is 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 based on Equation (6), as proposed in [1]. For each possible relevant dependent policy, a convex-hull is computed based on Equation (11). All relevant convex-hulls are then merged to build the PS convex-hull.



(a) Dependent policies

(b) Independent policies

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

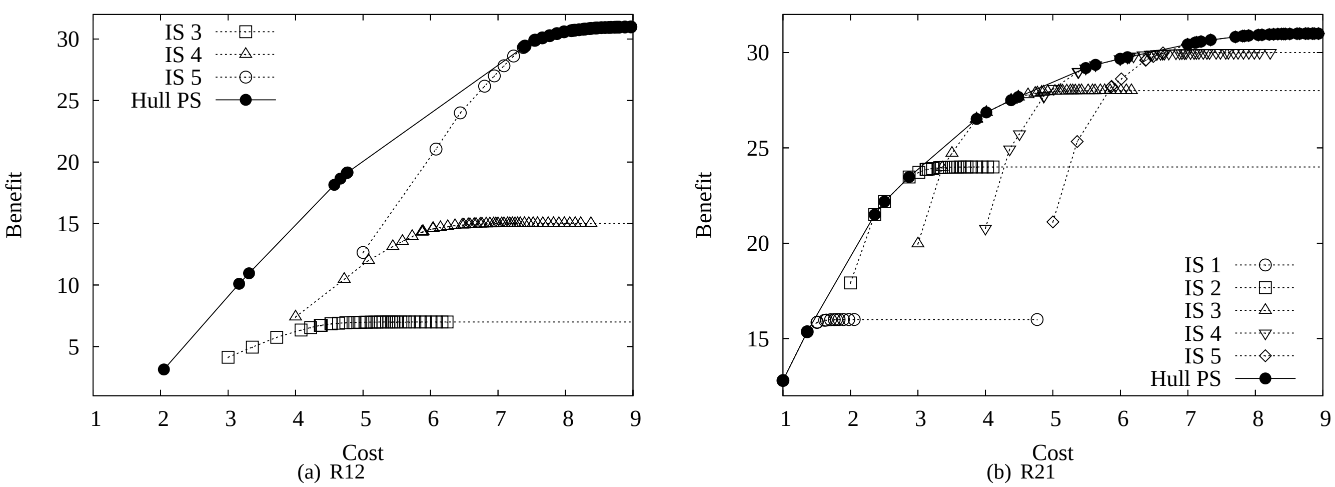
Figure 8 presents the results obtained for the R11 distortion templates, where all data units bring the same gain in distortion. Figure 8 (a) plots the convex-hulls computed for the proposed subspace of relevant policies (PS), with different number X of active layers. The global convex hull, which sustains all 'PS X' convex-hulls, is denoted 'Hull PS'. Similarly, in Figure 8 (b), 'IS X' denotes the convex-hull computed for independent policies, with X active layers. We observe in Figure 8 (b) that the proposed set of dependent policies improves the RD performance, i.e., Hull PS lies above IS X, for all X's. However, the gain appears to be quite marginal. Figure 9 provides the same analysis regarding the R12 and R21 distortion templates respectively.

To sum up, we observe that the gain provided by dependent policies is quite significant for the R12 template, but is small and even often negligible for the R21 template. As previously, we conclude that dependent policies are mainly beneficial when the gain in distortion increases (or at least does not decrease) along the dependency path. This observation is of practical importance because it means that there is no crucial need to implement dependent streaming policies when the gain in distortion per unit of rate decreases along the dependency path. Fortunately, this situation often occurs in practice, because efficient progressive or layered coders try to encode the most important information first. However, for non-scalable streams that encode a set of adjacent pictures as a group of interdependent data units, we can not rule out that non-negligible gain can be obtained, depending on the activity in the media sequences, which often drives the evolution of the distortion per unit rate along the dependency path. An example is studied in the next section.

C. MPEG stream, with temporal dependency

Finally, this section studies the relevance of the proposed dependent transmission policy, in the typical case of an MPEG streaming system. The Foreman sequence has been encoded in the MPEG-4 format at 30 fps, and groups of 10 interdependent data units (i.e., 10 video frames) have been formed, with IPPPPPPPPP dependencies. The scheduling parameters and channel conditions are defined as in the previous set of simulations: each data unit receives 20 transmission opportunities, distant by 50 ms.

Figure 10 compares the performance of dependent streaming policies with relevant master/slave relationships, with the performance of independent transmission policies 'IS X' denotes the convex-hull computed based on independent policies for X active frames in the group of interdependent data units. 'Hull PS' denotes the global convex-hull derived based on the proposed subspace of relevant policies, for any number of active layers. We observe that Hull PS lies above IS X, which means that the proposed set of relevant policies is able to improve expected RD performance in comparison with independent policies. However, the gain is marginal, and remains around half a dB.



(a) R12

(b) R21

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

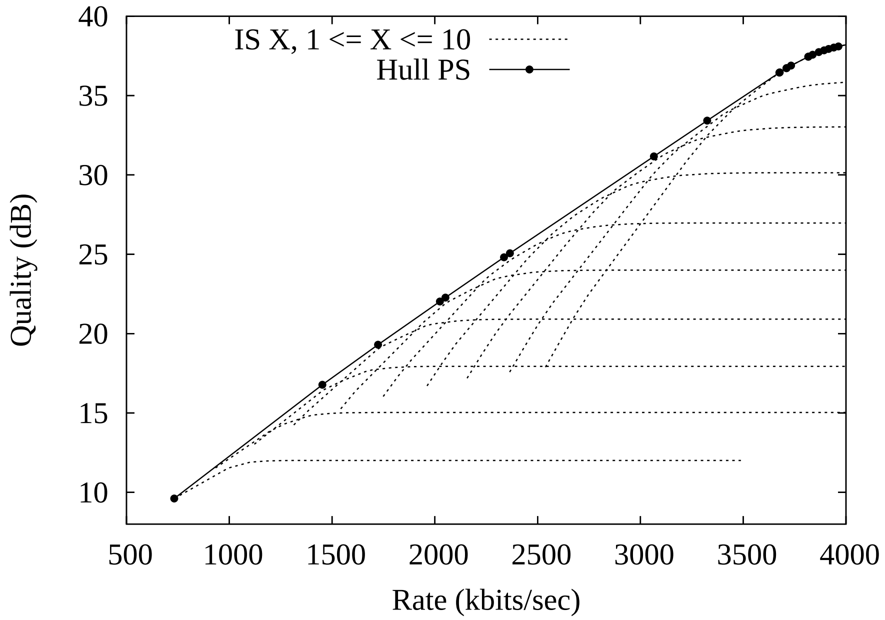


Fig. 10. RD convex-hull. 10 frames of the Foreman video sequence encoded at 30 fps with IPPPPPPPPP dependencies. Here $(S_1, ..., S_10)$ = $(244, 67, 63, 64, 73, 69, 68, 74, 69, 58)$ in kbits, and $(\Delta D_1, ..., \Delta D_{10})$ = $(12.01, 3.02, 2.91, 2.97, 3.01, 2.96, 2.86, 3.27, 2.91, 2.82)$ in dBs. Hull PS refers to the convex-hull sustained the proposed subspace of relevant dependent and independent policies, when all possible numbers of active layers are considered. IS X refers to the convex hull sustained by 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 $\varepsilon_F = \varepsilon_B = 0.2$.

VI. Conclusions

This paper has addressed the rate-distortion optimized streaming of packetized media streams, with comprehensive use of the feedback information. The notion of master and slave data unit has been introduced, to enable dependent streaming policies between media packets. Relevant master/slave relationships have been analyzed, as the dependencies that are likely to bring performance gain in the streaming system. Based on our simulation results, we conclude that (i) the proposed set of relevant dependent policies achieves close to optimal performance, while being computationally tractable, and (ii) the gain to expect from dependent policies in comparison with independent policies strongly depends on the relative sizes and distortions of interdependent data units. Even if the dependent streaming policies always perform better than independent streaming strategies, we observe that the gain becomes marginal when the gain in distortion per unit of rate decreases along the dependency path. Such a trend characterizes most conventional scalable coders, so that the implementation of dependent policies can reasonably be rules out in these practical cases. In other typical scenarios, the performance gain offered by dependent streaming policies highly depends on the evolution of the cost in rate and gain in distortion of data units encountered along the media sequence, and along the dependency graph.

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 (6) and (11). Let $\pi_a^{\perp}(\lambda)$ and $\pi_d^{\perp}(\lambda)$ denote the optimal independent policies computed based on (6). 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

$$\begin{aligned} R_{\perp} &= \rho_a S_a + S_d \\ D_{\perp} &= D_0 - (1-\epsilon_a)\Delta D_a - (1-\epsilon_a)(1-\epsilon_d)\Delta D_d \\ R_{\leftarrow} &= p_d^f \rho_a^* S_a + S_d \\ D_{\leftarrow} &= D_0 - p_d^f (1-\epsilon_a^*)\Delta D_a - p_d^f (1-\epsilon_a^*)\Delta D_d \end{aligned} \tag{17}$$

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

$$D_{\leftarrow} + \lambda R_{\leftarrow} < D_{\perp} + \lambda R_{\perp} \tag{18}$$

Using (17), (18) becomes

$$\lambda S_a(\rho_a - p_d^f \rho_a^*) > [(1-\epsilon_a) - p_d^f(1-\epsilon_a^*)]\Delta D_a + [(1-\epsilon_a)(1-\epsilon_d) - p_d^f(1-\epsilon_a^*)]\Delta D_d \tag{19}$$

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

$$\lambda(p_d^f \rho_a^* S_a + S_d) < p_d^f(1-\epsilon_a^*)[\Delta D_a + \Delta D_d] \tag{20}$$

which implies

$$\lambda \rho_a^* S_a < \Delta D_a + \Delta D_d \tag{21}$$

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 (21), the $\rho_a^* \geq \rho_a$ inequality implies

$$\lambda \rho_a S_a < \Delta D_a + \Delta D_d \tag{22}$$

Regarding (19), it implies

$$\lambda S_a(\rho_a - p_d^f\rho_a^*) > \lambda S_a\rho_a(1-p_d^f) > \lambda S_a\rho_a \tag{23}$$

By introducing (23) in (19), we have

$$\lambda S_a\rho_a > [(1-\epsilon_a) - p_d^f(1-\epsilon_a^*)]\Delta D_a + [(1-\epsilon_a)(1-\epsilon_d) - p_d^f(1-\epsilon_a^*)]\Delta D_d \tag{24}$$

and, by merging (24) and (22), we have

$$\Delta D_a + \Delta D_d > [(1-\epsilon_a) - p_d^f(1-\epsilon_a^*)]\Delta D_a + [(1-\epsilon_a)(1-\epsilon_d) - p_d^f(1-\epsilon_a^*)]\Delta D_d \tag{25}$$

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

$$\begin{aligned} \Delta D_{\perp} &= (1-\epsilon_a)\Delta D_a + (1-\epsilon_a)(1-\epsilon_d)\Delta D_d \\ R_{\perp} &= \rho_a S_a + \rho_d S_d \end{aligned} \tag{26}$$

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

$$\begin{aligned} \Delta D_{\leftarrow} &= p_d^f(1-\epsilon_a^*)\Delta D_a + p_d^f(1-\epsilon_a^*)\Delta D_d \\ R_{\leftarrow} &= p_d^f\rho_a^* S_a + S_d \end{aligned} \tag{27}$$

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

$$\begin{aligned} \Delta D_{\rightarrow} &= (1-\epsilon'_a)\Delta D_a + p_a^f(1-\epsilon_d^*)\Delta D_d \\ R_{\rightarrow} &= S_a + p_a^f\rho_d^* S_d \end{aligned} \tag{28}$$

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 (28) and (27), 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 (27) and (26). We conclude that the gain in distortion per unit of rate can only be significantly higher in Equation (27) than in Equation (26) if

$$\frac{(1-\epsilon_a^*)[\Delta D_a + \Delta D_d]}{\rho_a^* S_a} \gg \frac{(1-\epsilon_a)[\Delta D_a + (1-\epsilon_d)\Delta D_d]}{\rho_a S_a} \tag{29}$$

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 → 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, without affecting the performance attainable by the slave transmission, 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 $m < rightarrows_m$ 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 (6), and the Lagrangian $J_\lambda(\vec{\pi})$ can be written

$$J_\lambda(\vec{\pi}) = D_0 - \sum_{l=1}^{L} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) + \lambda \sum_{l=1}^{L} S_l \rho(\pi_l) \tag{30}$$

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 (11). Based on Section IV-A, 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 (11). 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

$$\begin{aligned} J_\lambda(\pi_{\vec{\{m\}}}) \quad &= \quad D_0 - \sum_{l=1}^{s_m - 1} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'}^*)) - p_m^f \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l, l' \neq m} (1 - \epsilon(\pi_{l'}^*)) \\ &\quad + \lambda \left(\sum_{l=1}^{s_m - 1} S_l \rho(\pi_l^*) + p_m^f \sum_{l=s_m}^{L} S_l \rho(\pi_l^*) \right) \end{aligned} \tag{31}$$

By definition, $m \rightarrow s$ is beneficial in the RD sense iff $J_\lambda(\pi_{\vec{\{m\}}})$ defined in Equation(31) is smaller than $J_\lambda(\vec{\pi})$ in Equation (30). 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

$$D_0 - \sum_{l=1}^{s_m - 1} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) + \lambda \sum_{l=1}^{s_m - 1} S_l \rho(\pi_l) \sim D_0 - \sum_{l=1}^{s_m - 1} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'}^*)) + \lambda \sum_{l=1}^{s_m - 1} S_l \rho(\pi_l^*) \tag{32}$$

Based on this assumption, trivial developments show that the cost function in Equation(30) is bigger than the cost in Equation (31), i.e., $m \rightarrow s$ is beneficial, when

$$\lambda \sum_{l=s_m}^{L} S_l \left(\rho(\pi_l) - p_m^f \rho(\pi_l^*) \right) > \sum_{l=s_m}^{L} \Delta D_l \left(\prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) - p_m^f \prod_{l' \preceq l, l' \neq m} (1 - \epsilon(\pi_{l'}^*)) \right) \tag{33}$$

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 $m \rightarrow s_m$ is shown to be beneficial if

$$(1 - p_m^f) \lambda \sum_{l=s_m}^{L} S_l \rho(\pi_l) > \left((1 - \epsilon(\pi_m)) - p_m^f \right) \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l, l' \neq m} (1 - \epsilon(\pi_{l'})) \tag{34}$$

Without loss of generality, we can express p_m^f as the product of $(1 - \epsilon(\pi_m)$ and p_m^b. The probability $p_m^b = p_m^f / (1 - \epsilon(\pi_m))$ 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 (34) becomes

$$(1-p^f_m)\lambda \sum_{l=s_m}^{L} S_l \rho(\pi_l) > (1-p^b_m) \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'})) \tag{35}$$

The inequality (35) tells whether the $m \rightarrow s_m$ 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 $m' \rightarrow s_{m'}$ is added to the $m \rightarrow s_m$ 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 (35). Specifically, we have that $J_\lambda(\pi_{\vec{\{m\}}}) < J_\lambda(\vec{\pi}^*)$ if

$$(1-p^f_m)\lambda \sum_{l=s_m}^{s_{m'}} S_l \rho(\pi_l) + (1-p^f_m p^f_{m'})\lambda \sum_{l=s_{m'}}^{L} S_l \rho(\pi_l) > (1-p^b_m) \sum_{l=s_m}^{s_{m'}} \Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'})) + (1-p^b_m p^b_{m'}) \sum_{l=s_{m'}}^{L} \Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'})) \tag{36}$$

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

$$(1-p^f_m)\lambda \sum_{l=s_m}^{L} S_l \rho(\pi_l) + (1-p^f_{m'})\lambda \sum_{l=s_{m'}}^{L} S_l \rho(\pi_l) > (1-p^b_m) \sum_{l=s_m}^{L} \Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'})) + (1-p^b_{m'}) \sum_{l=s_{m'}}^{L} \Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'})) \tag{37}$$

By comparing Equations (35) and (37), we observe that the benefit to draw from the $m' \rightarrow s_{m'}$ can be studied independently of other MSRs. Specifically, the optimal slave leader $s^*_{m'}$ for m' is the one that maximizes

$$(1-p^f_{m'})\lambda \sum_{l=s_{m'}}^{L} S_l \rho(\pi_l) - (1-p^b_{m'}) \sum_{l=s_{m'}}^{L} \Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'})) \tag{38}$$

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 (35), 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

$$\frac{\Delta D_l \prod_{l' \preceq l}(1-\epsilon(\pi_{l'}))}{S_l \rho(\pi_l)} \tag{39}$$

encounters significant local maxima while going up along the path of descendance, i.e. as the index l goes from the youngest data unit to the oldest one.

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 11. Figure 11(a) presents the 4 interdependent data units, and selects the first data unit to be the master. Figure 11(b) presents the 3 sets of relevant MSRs derived based on the rules defined in Section IV-A, without taking eligibility into account. Figure 11(c) defines how the last set of MSRs in Figure 11(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 11(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.

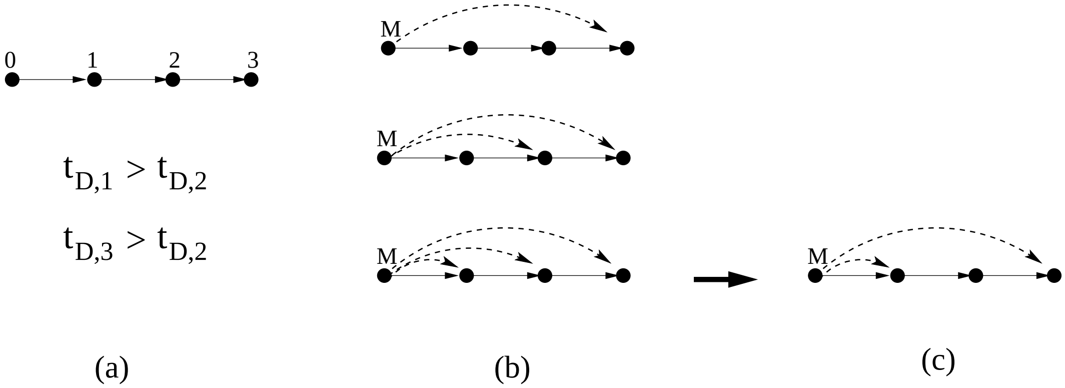


Fig. 11. Example of slave eligibility issue. (a) 4 interdependent data units. Data unit 0 is assigned to be a master M. The delivery deadline of data unit 2 comes much earlier than the deadlines for data units 1 and 3, which might cause a slave eligibility problem; (b) the 3 relevant MSRs, defined without 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 12. In this figure, data unit 2 is constrained by two masters. Figure 12(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 12(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.

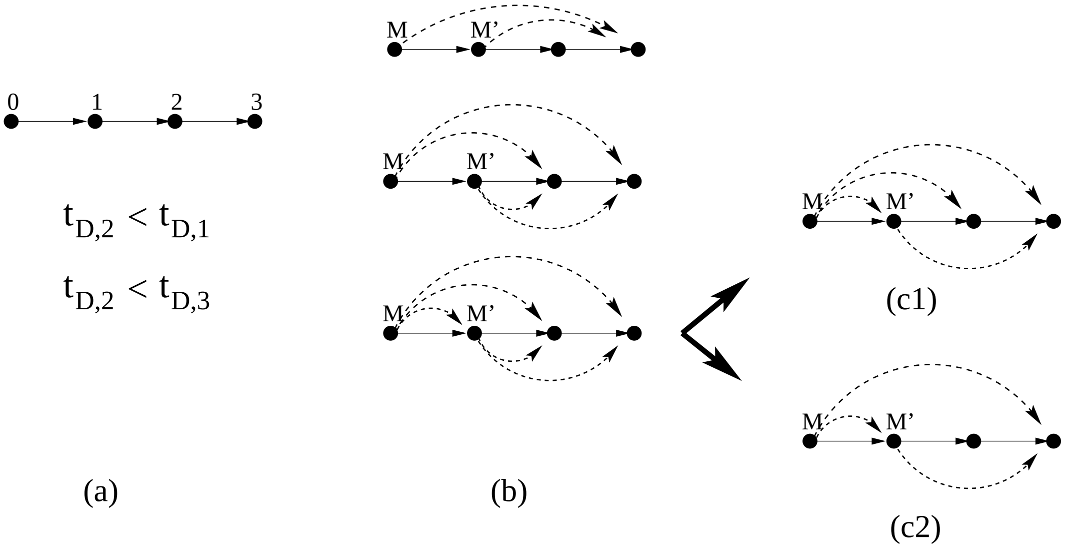


Fig. 12. Example of slave eligibility issue. (a) 4 interdependent data units. Data unit 0 and 1 are assigned to be masters M and M'. The delivery deadline comes earlier for data unit 2 than for other data units, which might cause a slave eligibility problem. This occurs when the wait for an ACK for masters 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

$$\begin{aligned} p_{l,\Gamma_l}(j) &= p_{\Gamma_l}(t_{l,N_l-j}) - p_{\Gamma_l}(t_{l,N_l-j-1}),\ 1 \leq j < N_l && (40)\\ &= p_{\Gamma_l}(t_{l,N_l-j}), \qquad j = N_l && (41)\end{aligned}$$

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

$$p_{\Gamma_l}(t) = \prod_{m \in \Upsilon_{\Gamma_l}} P\{RTT_{\ell(m,l)} < t_{m,0} - t\} \tag{42}$$

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.

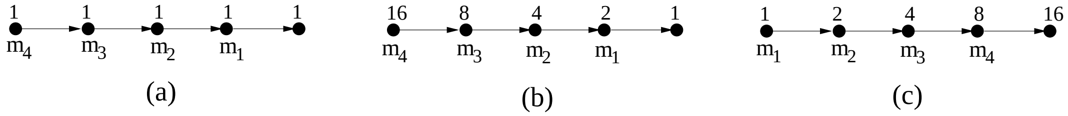


Fig. 13. Example of master assignment process for 3 different distribution of gain in distortion among 5 interdepenent data units. Bullets represent data units. Solid arrows represent dependency between data units. All data units have the same size. The gain in distortion provided by correct decoding 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).

Appendix E
Examples of master selection

This Appendix illustrates the master selection procedure. Figure 13 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 13. 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 13. 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

$$\min_{s>i} \frac{\Delta D(i,s)}{\Delta R(i,s)} = \min_{s>i} \frac{\epsilon_F \Delta D_i + (1-(1-\epsilon_F)(1-\epsilon_B)) \sum_{k\geq s} \Delta D_k}{(\zeta-1)S + (5-s+1)((1-(1-\epsilon_F)(1-\epsilon_B))S} \ . \tag{43}$$

In Equation (43), we consider that data units in Figure 13 are labeled in increasing order of dependency. We now develop Equation (43) for each one of the cases depicted in Figure 13.

In Figure 13(a), the gain in distortion is equal to one for all 5 data units. As a consequence for data unit i, Equation (43) becomes

$$\min_{s>i} \frac{\Delta D(i,s)}{\Delta R(i,s)} = \min_{s>i} \frac{\epsilon_F + (5-s+1)(1-(1-\epsilon_F)(1-\epsilon_B))}{(\zeta-1)S + (5-s+1)(1-(1-\epsilon_F)(1-\epsilon_B))S} \ . \tag{44}$$

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

$$\frac{\epsilon_F}{(\zeta-1)S} \leq \frac{(1-\epsilon_F)}{S} \leq \frac{(1-(1-\epsilon_F)(1-\epsilon_B))}{(1-(1-\epsilon_F)(1-\epsilon_B))S} \ , \tag{45}$$

so that Equation (44) 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 13(b), the gain in distortion decreases along the dependency path. As a consequence, Equation (43) 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 13(c), the gain in distortion increases along the dependency graph. As a consequence, Equation (43) 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*, Submitted 2001.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.