Optimal Rate Allocation for P2P Video Streaming

Livio Lima, Marco Dalai, Riccardo Leonardi, Pierangelo Migliorati, Riccardo Bernardini, and Roberto Rinaldo

***Abstract*—Rate control for multimedia streaming has been the subject of many recent research activities and is crucial for peer-to-peer networks, where controlling the rate at the source is ineffective. In this paper we propose a two-stage procedure for rate control. In particular, the first stage, based on Integer Linear Programming and a distortion model, optimally labels packets with a suitable priority level. This stage can be performed only once at the encoder. In a second stage, we derive an optimal strategy to choose prioritized packets for transmission, according to the available rate. This stage can be implemented at the transport level and autonomously by each peer. The techniques are specialized and used to control the rate of a standard H.264/SVC stream. We show by means of experiments that the proposed approach outperforms uncontrolled transmission, and that the proposed optimal priority selection gives substantial advantages over other simplified procedures.**

***Index Terms*—Peer-to-peer streaming, congestion control, H.264/AVC, scalable video coding.**

I. Introduction

OVER the past decades, Internet has witnessed the development of streaming services. One of the most important problems in the development of multimedia streaming services is the adaptation of the rate of the streamed video in a heterogeneous environment, where the content needs to be adapted both to the terminal capabilities and network conditions. Recently, scalable video coding (SVC), that allows decoding video at different spatial, temporal and quality resolution, emerged as a promising technique for efficient multimedia distribution in such heterogeneous scenario [1], [2].

In particular, adaptation to network conditions is actually very important since, due to the timeliness nature of streamed data, transmission is usually done over UDP that has no rate adaption mechanism. If no rate adaption is done, the link could become congested causing excessive packet losses and, in the worst case, network collapse. Indeed, it is suggested that any protocol transmitting data over UDP should control its output rate [3].

The problem of adapting the transmission rate to the network conditions is especially important in the case of video distribution over peer-to-peer networks. Indeed, while in a client-server setup the server can change the video quality in order to adapt it to the network conditions, in a peer-to-peer network this "control at the source" is not optimal. In order to make this point clearer, observe Fig. 1 where a peer-to-peer tree-structured streaming system is depicted. The multimedia source is at the root of the tree (node 1) and the link between nodes 2 and 5 has a reduced bandwidth. In this case, doing the rate control at the source would require that node 1 encodes with a quality suitable for the smallest bandwidth link, causing the whole network receiving the same quality of nodes 5, 8 and 9, although the remaining nodes could receive better content. In this case, the best solution would be having node 2 controlling the rate toward node 5 by sending to it, for example, only the base layer of a video coded with a scalable format.

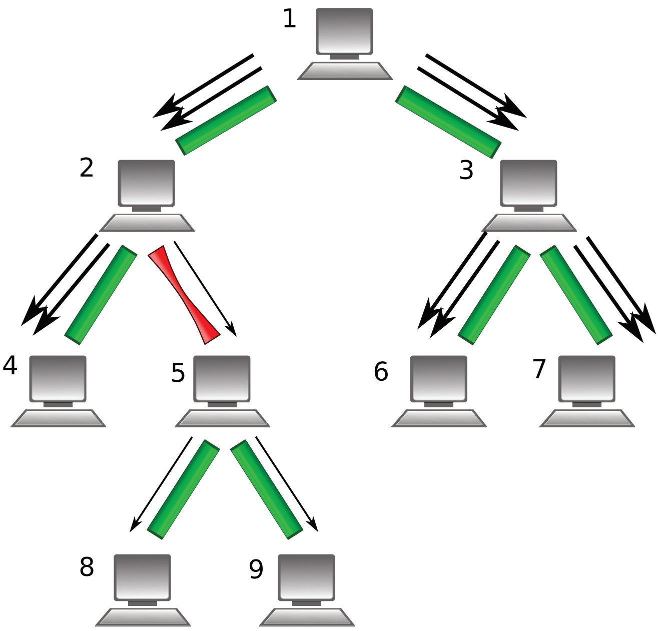


Fig. 1. Example of structure where the rate control at the source is not optimal. If the control were done at the source, the rate should be adapted to the bottleneck between nodes 2 and 5, lowering the quality in all the network.

The objective of this paper is to describe a simple technique for optimal rate adaptation in peer-to-peer systems like the one shown in Fig. 1. Several works in the literature consider the problem of resource optimization in multicast streaming of media content, see for instance [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. There are essentially two main directions of research that are relevant to the topic considered in this paper, both of which are often referred to with the common name of rate allocation, albeit with different meanings. The first one, mostly network oriented, is devoted to the problem of optimally exploiting the bandwidth available between different nodes, with the objective of maximizing some utility function. The second, mostly signal-processing oriented, focuses on the problem of allocating the bitrate during source and channel coding, with the objective of optimizing the performance in

Manuscript received February 29, 2012; revised July 14, 2012.

L. Lima, M. Dalai, R. Leonardi, and P. Migliorati are with DII–University of Brescia, Via Branze 38, 25123, Italy (e-mail: {livio.lima, marco.dalai, pierangelo.migliorati, riccardo.leonardi}@ing.unibs.it).

R. Bernardini and R. Rinaldo are with DIEGM–University of Udine, Via delle Scienze 208, 33100, Italy (e-mail: {bernardini, rinaldo}@uniud.it).

Digital Object Identifier 10.1109/JSAC.2013.SUP.0513018

terms of the quality of the reconstructed video.

In the first setting, one usually assumes that the source can be sent at different rates with different associated quality, and the problem is to optimize the achievable rate, without considering the problem of actually encoding the source at that given working point, assuming this is dealt with by some other entity. In the second setting, one usually assumes that the network conditions and the link rates are known, and the problem is to encode the source under these known constraints.

Works in these two groups use some basic techniques that are similar to the ones adopted in this paper, although the application context is different. In [4], the problem of layered multicast rate control is modelled as a dynamic programming problem where the maximization of the throughput is considered from a purely network theoretic point of view. The problem of layered source coding is not considered and it is assumed that the stream can be delivered at different rates with different values of an utility function. In [5], the problem of optimal routing for multimedia streaming is considered, but the focus is still network-oriented and no consideration is made on the coding problem. In [6], the authors propose an optimized rate control mechanism in the framework of network coding in multicast networks. The scheme requires that multicast nodes propagate cost information to the source. The optimization of the information flow in the network is then reduced to a convex optimization problem to be solved in a distributed fashion. In [7], the authors propose empirical models of the distortion due to packet loss or late arrivals as a function of rate and throughput in multiple trees connecting each peer to the source. However, no instantaneous rate control for adaptation to changing network conditions is considered in the analyzed P2P streaming scheme. They also propose a simplified classification of packet importance for optimized retransmission requests. This has to be precomputed offline for a generic video sequence, and made available to each peer. In [8], the authors consider a chunk-based P2P streaming system and adapt the number of signaling threads with neighbor peers so that the queuing delay at the transmission queue is small. In [9], a cross-layer design technique that includes rate estimation and source rate control is proposed. The work is then extended in [10] to the multicast case. The objective of the work is to optimize the quality of service in terms of delay and video quality, but no scalability and rate distortion optimization issues are considered. In [12] the problem of scalability is studied with the objective of minimizing the variance of the quality of the reproduced video under time varying network conditions. The focus of the paper is mostly source-coding oriented. The authors consider unicast transmission and the problem of packet losses in a peer-to-peer streaming system is not considered. In [13], the authors consider the problem of defining an efficient combination of scheduling, error protection and error concealment to maximize the quality of the reproduced video at the receiver in a packet loss prone network. However, the paper aims at the solution of the unicast problem and scalable approaches for the multicast case are not considered.

In this paper, we propose a joint optimization technique for streaming of scalable video in a peer-to-peer network, which is performed in two steps, one performed at the source coding stage, and the other at the packet distribution stage. The proposed scheme exploits the fact that the scalable extension of H.264/AVC standard introduces the idea of "quality layer," abstractly defined via the *priority_id* field in the Network Abstraction Layer Unit (NALU) header, where a NALU is the fundamental data unit in a SVC stream. The idea is that NALUs belonging to the same quality layer have the same importance in the adaption process. The scheme proposed in this paper operates in two loosely coupled stages: first, a *priority_id* is assigned to each NALU in an optimal way, successively the NALUs to be transmitted are optimally selected on the basis of their *priority_id* and the maximum available rate.

In order to implement the suggested scheme, the first problem to be solved is how to assign quality layers in an SVC stream in order to optimize the decoding process. The method currently adopted in the reference SVC software does not consider the possibility of multi-resolution quality layer generation and assigns priority to the NALUs according to their spatial resolution [14]. The method is extended in [15] in order to enable multi-resolution priority generation. The main drawbacks of these and other approaches as [16] [17] [18] are that they require partial decoding of the bit-stream and that they are not flexible enough to allow for further constraints to be added to the optimization problem. A more flexible, but still computationally demanding, approach is proposed in [19]. In [20], the authors proposed an adaptation method based on distortion models for Medium Grain Scalability (MGS) introducing the possibility to control the quality between different groups of pictures. However, this method requires bit-stream decoding, too. Moreover it does not support spatial scalability and does not provide any method to control the quality inside a single group of pictures. Recently, a new approach was proposed in [21] for quality layers generation based on Integer Linear Programming (ILP) and a suitable distortion model. Briefly, the distortion contribution of each NALU is estimated through the model. Then, rate and distortion contributions are used within an ILP problem in order to determine which NALUs have to be considered in an adaptation process for a particular target rate. Finally, quality layers are generated by solving the problem for a set of different rates. This approach enables real-time quality layers computation, and the flexibility to include additional constraints within the ILP model, while maintaining a performance comparable to [14].

The second problem to be solved, in order to implement the two-step scheme described above, is how to choose the NALUs to be sent, given the NALU priorities and the available bandwidth. In [13], a technique to maximize the quality of scalable video, taking into account network conditions, FEC protection and error concealment, is proposed. In [9] the authors propose a cross-layer design technique that includes a rate estimation technique (that allows for temporary violation of TCP-friendliness) and a source rate control technique. The work in [9] is extended in [22] to the wireless case, and in [10] to the multicast one. In [12], a technique for scalable video is proposed, which aims at minimizing the quality variability while maximizing the utilization of variable network bandwidth. Note that most of the papers available in the literature focus on a specific multimedia type and typically suppose to be able to control the encoder to adapt the rate to the available bandwidth. However, as already said, although the "control at the source" can be reasonable in point-to-point transmissions, it is not the optimal choice in a peer-to-peer setup.

According to the two-stage structure of the proposed scheme, the contribution of this paper is twofold and gives general solutions to the problem of designing both stages. More precisely, one contribution shows how to transform a stream of packets into a stream of *prioritized* packets, that is, a stream where each packet has a *priority label* that identifies its importance for the decoding process, in terms of Rate Distortion contribution. It is also shown how the prioritizing technique can be applied to the specific case of scalable H.264/AVC. The other contribution is a technique that, given a stream of prioritized packets and a maximum available bandwidth, shows how to select the packets to be transmitted in order to maximize the available quality. An interesting advantage of the proposed technique is that it does not require knowledge the type of the multimedia data (audio, video, 3D, ...), but only that every packet is labeled with a *priority value*. The two stages are actually separated and this allows to perform one procedure at the encoder, which is responsible of the packet priority labeling, while the other procedure, i.e., optimal packet selection, can be operated at the transport level and autonomously by each peer in a peer-to-peer system, according to the local available bandwidth. While in this paper we consider the typical scenario where the prioritization procedure is performed at the encoder, it is also possible to perform it at the peer level, in order to fully exploit the local bandwidth. This can be useful in particular cases or complex network topologies.

This paper is structured as follows. In Section II we consider the problem of optimal packet selection given a set of priorities and a limited bandwidth. In Section III, we describe a procedure to assign packet priorities using a distortion model and ILP. Section IV specializes the general results of Section III to the scalable extension of H.264/AVC standard. Experimental results are presented in Section V, while Section VI draws the conclusions.

II. Rate Control Algorithm

The problem considered in this section is the following: given a sequence of *prioritized packets* and a maximum available bandwidth of B bits/second, select which packets to send in order to satisfy the bandwidth constraint, while maximizing the quality of decoded content. Since usually multimedia packets must be received in time, otherwise they are useless, rate control cannot be done by increasing the time interval between successive packets (as done, for example, in [23]). In this section we will tackle the general problem, and suppose that we can send only a subset of the packets, while discarding the other ones. Although this approach could seem strange (we artificially introduce losses), it turns out that is more convenient to select which packet to lose rather than having the packets discarded at random by the network.

A. Definitions

a) Prioirity classes: We will suppose that the distributed content is organized in *packets* and that each packet has an associated *priority class* i, with class i "more important" (in a sense to be specified later) than class $i+1$. If i is the class of a packet, we will say that the packet *belongs to (priority) class* i. For the sake of language convenience, we will consider the whole set of packets of class i as a *virtual sub-stream* called in the following, the *i-th sub-stream*. Let L be the number of priority classes.

Remark 1

The concept of *priority class* is very similar to what H.264/AVC calls *quality layer*. However, in this section we will continue to use the term *priority class* in order to emphasize the independence of the algorithm described in this section from a specific coder.

Let r_i be the bit-rate (in bits/second) required by the i-th sub-stream and let $\mathbf{r} = [r_1, \ldots, r_L]^t$ be the vector column in $\mathbb{R}^L$ whose entries are the sub-stream rates. The total bit-rate required by the content is, clearly, $\sum_{i=1}^{L} r_i = \mathbf{u}^t\mathbf{r}$, where $\mathbf{u} = [1, \ldots, 1]^t$. Clearly, if $\mathbf{u}^t\mathbf{r} \le B$ the link between the source peer and the target peer can carry the whole content and no rate adjustment is necessary. If $\mathbf{u}^t\mathbf{r} > B$, the link cannot carry the whole contents and the source peer needs to reduce the amount of data sent to the target.

Remark 2

Although in this section we suppose that the rates r_i are known, it is clear that in an actual implementation values r_i probably will have to be estimated. A possible algorithm for estimating r_i is the following. Let $s_{i,n}$ be the size in bits of the n-th packet of class i and let $t_{i,n}$ be arrival time of the same packet. Keep the pairs $(s_{i,n}, t_{i,n})$ inside a circular buffer and estimate r_i as

$$r_i = \frac{\sum_{k=n_0}^{n_1} s_{i,k}}{t_{i,n_1} - t_{i,n_0}} \tag{1}$$

where n_0 and n_1 are, respectively, the minimum and the maximum value of n stored in the circular buffer. Note that this algorithm has a negligible computational complexity since it requires, beside the cost of updating the circular buffer, three sums (two to update the value of $\sum_{k=n_0}^{n_1} s_{i,k}$, one to compute $t_{i,n_1} - t_{i,n_0}$), and a division for every received packet.

b) Puncturing: As said above, the rate is reduced by selecting which packets to send. More precisely, the rate is reduced by "puncturing" the sub-streams, that is, by transmitting only a fraction of the packets belonging to a sub-stream and discarding the others. Let $p_i \in [0,1]$, $i = 1, \ldots, L$, be the fraction of packets of class i that are actually sent to the target and let $\mathbf{p} = [p_1, \ldots, p_L]^t \in [0,1]^L$ be the vector of p_i values.

Because of the puncturing, the bit-rate received by the target for the i-th substream will not be equal to r_i, but to $r_i p_i$. For the sake of notational convenience, we will denote with $q_i := r_i p_i$ the rate of class i *after* puncturing and with $\mathbf{q} = [q_1, \ldots, q_L]^t$ the corresponding column vector. Note that $\mathbf{q}$ belongs to the "box"

$$H := [0, r_1] \times [0, r_2] \times \cdots \times [0, r_L] \tag{2}$$

c) Quality function: We will suppose that the quality experienced by the user can be expressed by a "quality function" $Q : H \to \mathbb{R}$ that maps the vector of the received rates $\mathbf{q}$ to a real value that represents the quality perceived by the user. The only constraint about $\mathcal{Q}$ is the following set of inequalities that formalizes the intuitive idea that packets of class i are more important than packets of class $i+1$

$$\forall \mathbf{q} \in H,\ i \in \{1, \ldots, L-1\} \quad \frac{\partial \mathcal{Q}}{\partial q_i}|_{\mathbf{q}} > \frac{\partial \mathcal{Q}}{\partial q_{i+1}}|_{\mathbf{q}} > 0 \quad (3)$$

Note that condition (3) is very general and it is to be expected by any "quality function" associated with a context that allows to define priority classes. The following easy lemma will be useful.

Lemma 1. *If $\mathcal{Q}$ satisfies (3), then $\mathcal{Q}$ is monotone increasing with respect to every argument. More precisely, if $\mathbf{e}_i$ denotes the i-th canonical basis vector, then for every $i \in \{1, \ldots, L\}$, $\mathbf{q} \in H$ and $\epsilon > 0$ such that $\mathbf{q} + \epsilon \mathbf{e}_i \in H$ the following holds*

$$\mathcal{Q}(\mathbf{q} + \epsilon \mathbf{e}_i) > \mathcal{Q}(\mathbf{q}). \quad (4)$$

Proof: It suffices to integrate $f_i(x) := \partial \mathcal{Q}/\partial q_i(\mathbf{q} + x\mathbf{e}_i) > 0$ between 0 and ϵ. ■

B. Problem statement

Our objective is to find puncturing probabilities $p_1, \ldots, p_L$ (or, equivalently, actual rates $q_1, \ldots, q_L$) such that quality $\mathcal{Q}$ is maximized under the constraint that the overall transmitted rate is not larger than B, that is

$$q_1 + q_2 + \cdots + q_L = \mathbf{u}^t \mathbf{q} \leq B \quad (5)$$

In other words, our problem is to compute

$$\mathbf{q}^{(\mathrm{opt})} = \arg\max_{\mathbf{u}^t\mathbf{q} \leq B} \mathcal{Q}(q_1, q_2, \ldots, q_L) \quad (6)$$

In order to describe the solution of (6), we will need some notation.

Definition 1. *For $k \in \{0, \ldots, L\}$, let $R_k = \sum_{i=1}^{k} r_i$ denote the cumulative rate of the first k sub-streams, with, obviously, $R_0 = 0$. If $B < R_L = \sum_{i=1}^{L} r_i$ denote with $K_B \in \{1, \ldots, L-1\}$ the index such that*

$$R_{K_B} \leq B < R_{K_B+1} \quad (7)$$

(note that the constraint $B < R_L$ makes this definition well-posed); if $B \geq R_L$, define $K_B = L$.

Definition 2. *Let $C = [0, M_1] \times [0, M_2] \times \cdots \times [0, M_L]$ be an "hyperbox" of $\mathbb{R}^L$. A vector $\mathbf{x} \in C$ is said to be a* step vector with breaking point K *if*

$$\mathbf{x}_i = \begin{cases} M_i & \text{if } i < K \\ \alpha > 0 & \text{if } i = K \\ 0 & \text{if } i > K \end{cases} \quad (8)$$

In other words, the first $K-1$ components of $\mathbf{x}$ assume the maximum value, the last $L-K-1$ components of $\mathbf{x}$ assume the minimum value, and only $0 < \mathbf{x}_K < M_i$. We will denote with V_K the set of step vectors with breaking point K.

Remark 3

Note that a step vector $\mathbf{x}$ is uniquely identified by two values: the breaking point K and the corresponding value $\mathbf{x}_K$.

Now we can give the solution of (5)

Property 1. *Suppose that (3) holds. If $B \geq R_L$, then $\mathbf{p}^{(opt)} = [1, \ldots, 1]$; otherwise, if $B < R_L$, then $\mathbf{p}^{(opt)}$ is a step vector with breaking point $K_B + 1$ and*

$$\mathbf{p}_{K_B+1}^{(opt)} = \frac{B - R_{K_B}}{r_{K_B+1}} \quad (9)$$

The proof of Property 1 is given in Appendix A.

Remark 4

The solution given in Property 1 is very intuitive. Expressed in words, Property 1 can be reformulated as follows: if there is enough bandwidth to transmit everything (i.e., $R_L \leq B$), then do no puncturing at all (i.e., $\mathbf{p}^{(\mathrm{opt})} = [1, \ldots, 1]$), otherwise keep as many sub-streams as you can, choosing them in order of priority, puncture the first sub-stream that cannot be kept in order to reduce its rate and discard the remaining sub-streams. As naïve as this solution may seem, Property 1 shows that it is optimal. Also note that Property 1 does not even require a precise knowledge of $\mathcal{Q}$, as long as it is possible to say correctly when a packet is more important than another packet. This makes Property 1 a very general result and very easy to apply, since one does not need to actually derive a closed form for $\mathcal{Q}$.

Remark 5

Observe that no knowledge of the type of the sent data is necessary. This makes it possible to implement this solution directly at the transport level, as long as the API (Application Programming Interface) allows to specify the priority class of each packet. This avoids to intermingle application level details (i.e., the additional error due to a packet loss) with transport level details (i.e., congestion control).

III. Priority Class Generation with Integer Linear Programming

This section describes the proposed mechanism for priority class generation in case of scalable video content. The approach can be applied to any type of scalable video content in which data are structured as represented in Fig. 2. Video frames are processed in Group Of Pictures (GOPs) following a particular decomposition structure within a GOP. In order to supply temporal scalability, typically, scalable codecs use a hierarchical decomposition structure, with different possibilities in terms of Intra frames, P-frames, B-frames and prediction dependency between different GOPs. Within each GOP frames are encoded enabling multi-resolution (spatial scalability) and multi-quality (quality scalability) adopting a differential encoding approach.

In scalable coding architectures, prediction is typically used between different quality and resolution layers in order to reach high coding efficiency. The main drawback of this approach is that it introduces decoding constraints, since a particular data unit can not be decoded if the data unit used as a predictor is not available. In the following, we describe our method of priority class generation using this prediction model as reference, even if arbitrary prediction mechanisms between data units can be considered.

In Section III-A we describe how the ILP approach can be used in scalable video coding to determine which data units should be sent in order to optimize the reconstruction quality given a fixed value of available rate, while in Section III-B we will show how the resolution of this sub-problem is exploited to generate priority classes to be used in a peer-to-peer network. Finally, in Section III-C, the complexity of

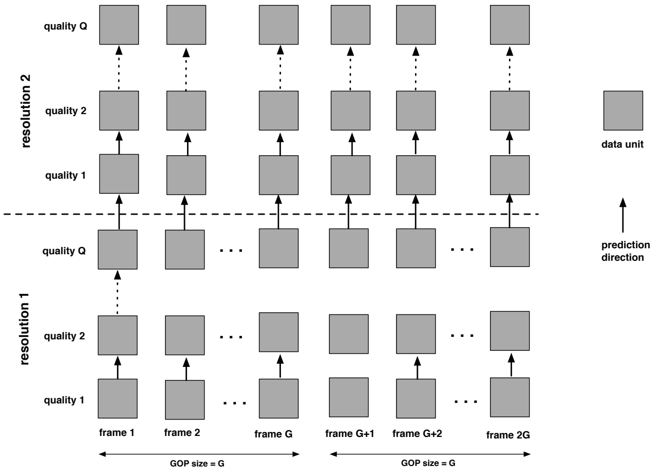


Fig. 2. Decomposition of a scalable stream in data-units. Within each resolution, a data-unit is predicted using the data unit of lower quality and the data-unit of lowest quality is predicted using the data-unit of higher quality within the lower resolution.

the algorithm is studied in order to show that the proposed approach is feasible in practical real scenarios.

A. Optimal rate allocation with Integer Linear Programming

The proposed approach for priority class generation is based on a formalization of the problem in terms of Linear Programming (LP). More precisely, since the unknown in our problem will typically be discrete valued variables, we will formalize our problem as an Integer Linear Programming (ILP) problem. ILP is a common approach used to solve optimization problems since it can offer high flexibility and computational advantages.

In this section we describe how the ILP formulation can be used to model the problem of finding the data units with higher priority given a maximum available rate. The priority here has to be measured in terms of achievable output video quality, and the optimal data unit choice could be defined by evaluating this quality independently GOP by GOP or on the entire sequence. The latter enables a solution that is globally optimal on the whole sequence, while the former is usually preferable in order to optimize the quality of service in a video streaming application.

We first need to introduce the mathematical preliminaries. The standard form of an ILP formulation is given as

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^t\mathbf{x} \\ \text{subject to} & \left\{ \begin{array}{lll} \mathbf{Ax} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & 0 \quad \mathbf{x}\in\{0,1\}^{M_v} \end{array}\right. \end{array} \tag{10}$$

where $\mathbf{c} \in \mathbb{R}^{M_v}$ and $\mathbf{b} \in \mathbb{R}^{M_c}$ are vectors of real coefficients, and $M_v \in \mathbb{Z}$ and $M_c \in \mathbb{Z}$ are, respectively, the number of variables and constraints. The product $\mathbf{c}^t\mathbf{x}$ is called objective function and $\mathbf{Ax} \leq \mathbf{b}$ is to be interpreted as an element-wise set of inequalities, which represent the set of constraints of the problem. In our setting, we will define an instantiation of this problem where the vector $\mathbf{x}$ will represent a vector of boolean flags specifying which data units have to be included in an optimal allocation of a given total available rate.

Assume now that we are encoding a sequence of N frames with L_S levels of scalability $\{S_1, S_2, \ldots, S_{L_S}\}$ and L_Q levels of quality $\{Q_1, Q_2, \ldots, Q_{L_Q}\}$, and suppose that, given the already computed decomposition of the video in data units, we want to optimally pick the units to be transmitted given a total rate budget R. Let us thus introduce a vector of variables $\mathbf{x} = \{x^i_{S,Q}\}^t$, where the generic variable $x^i_{S,Q}$ takes value 1 if the data unit of resolution S and quality Q belonging to frame i is to be encoded in order to optimally allocate the available rate R, and 0 otherwise. We now define an integer program to find the solution to this problem under the rate and prediction constraints.

As a first step, we need to define an appropriate objective function to obtain an instantiation of problem (10) that allows to find the optimal rate allocation on data units in a rate distortion sense. For notational convenience, since (10) is a maximization problem, we will look for the vector $\mathbf{x}$ that maximizes a reduction in the distortion with respect to the zero-rate case. We assume linearity in the distortion reduction, that is, we assume that the distortion reduction due to a set of data units can be written as a sum of distortion reductions due to each single data unit. This is reasonable in light of the fact that data units represent uncorrelated portions of information, since they typically contain subsets of transform coefficients and prediction residuals. This allows us to assume that a Parseval relation holds which makes it possible to compute the distortion as a sum of distortions associated to orthogonal components. Hence, we assume to have knowledge of the contribution of each data unit $x^i_{S,Q}$ in terms of distortion reduction, and express this by means of a coefficient $c^i_{S,Q}$. Consequently, the overall distortion reduction for a given vector $\mathbf{x}$ takes the form

$$\mathbf{c}^t\mathbf{x} = \sum_i \sum_S \sum_Q c^i_{S,Q} x^i_{S,Q}. \tag{11}$$

The coefficients $c^i_{S,Q}$ depend on the codec and it is usually possible to build distortion models that allow to estimate the true distortion reduction of each data unit. In our derivation, for the time being, we only assume that these coefficients are given, so that the previous expression can be evaluated. In Section IV, we will see how to model them for the particular case of the Scalable Extension of standard H.264/AVC.

Now, once defined the objective function, we need to set the constraints of the problem. The prediction structure between data units of different quality, as shown in Fig. 2, implies that if a data unit at quality k is not selected, then no higher quality data unit should be selected, since the predictor is missing. Hence, the model has to verify that if $x^i_{S,Q_k} = 0$ then $x^i_{S,Q_m} = 0$ for all $k < m \leq L_Q$; these conditions are expressed in terms of inequalities in the model (10) as:

$$-x^i_{S_h,Q_k} + x^i_{S_h,Q_{k+1}} \leq 0, \quad 1 \leq h < L_S, 1 \leq k < L_Q, \quad 1 \leq i \leq N. \tag{12}$$

The last constraint that has to be introduced in model (10) to have a complete problem description is the budget constraint (or rate constraint). If the maximum available rate is R, the budget constraint can be written as

$$\sum_i \sum_S \sum_Q x^i_{S,Q} r^i_{S,Q} \leq R \tag{13}$$

where $r^i_{S,Q}$ is the rate of each data unit, which is straightforward to obtain by data stream inspection.

B. Priority class generation algorithm

As explained above, the solution of the ILP problem (10), allows us to find the data units that give the best reconstruction performance for a given available rate R. In this section we exploit this result to find a priority map for the scalable video content by instantiating multiple problems of this type and by analyzing the maps of data units obtained as solutions to these problems.

Let us suppose that we have the optimal solution of the problem (10) for a given maximum available rate R, and let us indicate with $SP(R)$ this problem. The proposed algorithm easily generates the priority map for the scalable video content through multiple solutions of the sub-problems $SP(R)$ at different rate points R. Let us suppose that the aim is to generate a priority map for the scalable video content with L different levels of priorities. The steps of the algorithm are as follows:

1) Estimate vector $\mathbf{c}$ of distortion reduction contributions for the particular scalable video coder and video content;
2) Select a set of L rates $R_1, \dots, R_L$, starting from the rate required to include all the data units with smallest possible quality to the rate of the full data stream (R_L);
3) For every k in $1, \dots, L$, solve the sub-problem $SP(R_k)$; let $\mathbf{x}_k$ denote the corresponding solution;
4) Let $\mathcal{U}_k$ be the set of data units with related binary variable equal to 1 in $\mathbf{x}_k$. The priority value equal to k is assigned to the data units that belong to the difference set $\mathcal{U}_k \setminus \mathcal{U}_{k-1} = \mathcal{U}_k \cap \mathcal{U}^c_{k-1}$, with $\mathcal{U}_{-1} = \emptyset$.

It is worth pointing out that, in the solution of the sub-problems $SP(R_k)$, we enforce that $\mathcal{U}_{k-1} \subset \mathcal{U}_k$ for all k. This is – rarely – not verified for the independent unconstrained solution of the sub-problems due to the fact that, in the solution of problem $SP(R_{k-1})$, some small data units (usually just one) may be added, in order to achieve a rate as close as possible to R_{k-1}, which are not selected in problem $SP(R_k)$ because, with more rate available, some larger data units are preferable.

The algorithm produces, for each data unit, a priority value k to use in the rate control algorithm of Section II, starting from 1 (higher priority) to L (lower priority).

Remark 6

The computational complexity of the first stage is clearly proportional to L. This gives rise to a trade-off, i.e., by choosing L larger we gain in "priority resolution" and we can expect better performance, but a large value of L could increase excessively the computational complexity. Actually, as emphasized in Remark 7, we observe that, using the results of Section III-C, the time required for $L = 64$ (the maximum value possible allowed by the 6-bit *priority_id* field of NAL header) is sufficiently small, even on a general purpose computer, so that there is no special reason to use a smaller value for L. This is possible because, as explained in Section III-C, this specific ILP problem can be mapped into a non-integer LP problem.

However, if the algorithm is applied in a low-complexity application, a simple solution to find the best L is to solve the problem for as many rates R_k as allowed by the computational resources. In particular one can first solve the problem for the maximum value R_L, then for $R_L/2$, then for the intermediate values $R_L/4$ and $3R_L/4$ and so on ($R_L/8$, $3R_L/8$, $\dots$) in successive improvements, until the maximum time allowed for priority assignment expires.

Finally, observe that in very low-complexity context, one can use the sub-optimal solution to compute the priorities not for every GOP but one every M GOPs, reusing the computed priorities for the other $M - 1$ GOPs.

C. Computational Analysis of the Proposed Method

One of the possible limitations of the proposed method for priority class generation is its potential computational complexity. In fact, in contrast to LP problems, which can be efficiently solved, ILP problems are typically NP-hard, thus requiring a computational time which increases exponentially with the problem size. This is particularly relevant if the problem (10) is solved considering the data units of the entire sequence. Furthermore, the multiple solutions of the problem needed to generate the priority map additionally increase the complexity. If we want to apply the proposed method in a real-time rate control algorithm for streaming of scalable content, this is a strong limiting factor. Even if problem (10) is in general NP-hard, efficient solutions can be found for ILP problems whose constraint matrix $\mathbf{A}$ exhibits some particular properties. In this section we show that, fortunately, our problem does fall in this category.

The constraints given by equations (12) and (13) can be expressed in matrix representation as $\mathbf{Ax} \le \mathbf{b}$, where the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are defined as:

$$\mathbf{A} = \left(\frac{\mathbf{A}_U}{\mathbf{a}} \right) = \left(\begin{array}{cccc} \mathbf{U} & \mathbf{Z} & \dots & \mathbf{Z} \\ \mathbf{Z} & \mathbf{U} & \dots & \mathbf{Z} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{Z} & \mathbf{Z} & \dots & \mathbf{U} \\ \hline \mathbf{r}^1 & \mathbf{r}^2 & \dots & \mathbf{r}^N \end{array} \right) \qquad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ R \end{pmatrix} \tag{14}$$

where $\mathbf{A}_U$ has size $NL_S(L_Q - 1) \times NL_SL_Q$ whereas $\mathbf{a}$ has dimensions $1 \times NL_SL_Q$, where NL_SL_Q is the number of data units. The matrix $\mathbf{A}_U$ is made of sub-matrices $\mathbf{U}$, each one representing the set of constraints (12) for each frame i, and $\mathbf{Z}$ (matrix of zeros), both of size $L_S(L_Q - 1) \times L_SL_Q$. The matrix $\mathbf{U}$ can be further represented as:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_p & \mathbf{Z}_p & \dots & \dots & \dots & \mathbf{Z}_p \\ \mathbf{Z}_p & \mathbf{U}_p & \dots & \dots & \dots & \mathbf{Z}_p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{Z}_p & \mathbf{Z}_p & \dots & \dots & \dots & \mathbf{U}_p \end{pmatrix}$$

$$\mathbf{U}_p = \begin{pmatrix} -1 & 1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 1 & \dots & \dots & 0 \\ 0 & 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & -1 & 1 \end{pmatrix}$$

where $\mathbf{U}_p$ has size $(L_Q - 1) \times L_Q$ and $\mathbf{Z}_p$ is a matrix of zeros of the same size. The vectors $\mathbf{r}^i$ represent the rate contributions given by each frame and can be represented as

$$(\mathbf{r}^i)^t = \big(r^i_{S_1,Q_1}, \dots, \quad r^i_{S_1,Q}, \dots, r^i_{S_1,Q_{L_Q}}, \dots, \\ r^i_{S,Q}, \dots, r^i_{S_{L_S},Q_{L_Q}} \big)$$

As previously introduced, ILP problems with a particular property of the constraints matrix $\mathbf{A}$ can be efficiently solved. Such matrices are the Totally Unimodular (TUM) matrixes.

Definition 3. *A $m \times n$ matrix $\mathbf{M}$ is* totally unimodular *if each of its square submatrices has determinant 0, +1, or -1.*

In our problem, the matrix A_U associated to the constraints (12) can be shown to be TUM, as will be proved later on. For ILP problems with TUM constraint matrix and integer right-hand side, the optimal solution of the associated LP problem (that is, the LP problem obtained by relaxing the constraint that variables $\mathbf{x}$ be integer) corresponds to the optimal integer solution, as stated by the following Theorem [24].

Theorem 1. *If M is a $m \times n$ totally unimodular matrix and $\mathbf{b}$ is an integral vector, then for each objective function $\mathbf{c}^t\mathbf{x}$ the linear programming problem:*

$$\min \left\{ \mathbf{c}^t\mathbf{x} \,|\, \mathbf{M}\mathbf{x} \geq \mathbf{b} \right\}$$

has an integral optimum solution (provided the maximum is finite).

In order to prove that our problem falls in this nice category for which the relaxed LP problem gives the same solution of the original ILP problem, we need the following theorem [24].

Theorem 2. *A $(0, +1, -1)$-valued $m \times n$ matrix $\mathbf{M}$ is totally unimodular if both of the following conditions are satisfied:*

(i) *Each column contains at most two non-zero elements.*
(ii) *The rows of $\mathbf{M}$ can be partioned into two sets $\mathcal{M}_1$ and $\mathcal{M}_2$ such that two nonzero entries in a column are in the same set of rows if they have different signs and in different sets of rows if they have the same sign.*

Theorem 2 gives a sufficient condition for the totally unimodularity of $\mathbf{M}$. The proof of the Theorems 1 and 2 can be found in [24].

It has to be noted that the constraints matrix $\mathbf{A}$ of our problem is not TUM, since we have an extra budget constraint (13). Nevertheless, this is a classical knapsack constraint [25], [26], that can be efficiently managed by common solvers for mixed integer linear programming problems. For this reason, in the following we show that the matrix $\mathbf{A}_U$, obtained removing the last row from $\mathbf{A}$ related to the budget constraint, is TUM.

Theorem 3. *The matrix $\mathbf{A}_U$, described in equation (14) is totally unimodular.*

Proof: In order to prove that $\mathbf{A}_U$ is totally unimodular we have to show that the conditions given in Theorem 2 hold. From the construction of matrices $\mathbf{U}$ and $\mathbf{U}_p$, it follows that each column of $\mathbf{A}_U$ is made by only one column of $\mathbf{U}_b$ and other zero entries (before or after the column of $\mathbf{U}_b$). By inspecting the columns of $\mathbf{U}_p$, it can be noticed that the first and the last columns contain only 1 non-zero entry, while all the other columns contain 2 non-zero entries with opposite sign. The same observation can also be used to find the sets $\mathcal{M}_1$ and $\mathcal{M}_2$ in order to verify condition (ii). Let us consider the simple partition obtained considering $\mathcal{M}_1$ as the set of rows of $\mathbf{A}_U$ and $\mathcal{M}_2 = \emptyset$. The condition (ii) is simply verified since the two elements in each column of $\mathcal{M}_1$ have different sign. ■

From the computational point of view, having an ILP problem with constraint matrix made by a TUM matrix and an additional knapsack problem gives two possible efficient solution strategies. One possible solution is to solve the problem (10) with integral variables $\mathbf{x}$ using common solvers for mixed integer linear programming as CPLEX [27]. Such kind of solvers automatically identify the TUM property of a sub-matrix of the problem constraint matrix and efficiently find an integer solution. Another possible solution is to solve problem (10) in a relaxed form where the variable vector $\mathbf{x}$ is allowed to take real values, and then solving the related LP problem. It can be shown that the LP solution of our problem (10) with constraint matrix made of a TUM matrix and an additional knapsack problem leads to all the variables $x^i_{S,Q}$ being integral ($x^i_{S,Q} \in \{0, 1\}$) except one, which assumes a real value in order to exactly fit the rate constraint R. The obtained LP problem can then be efficiently solved in polynomial time using the simplex method [28].

Remark 7

Just to give a more precise idea of the computational complexity, it is worth pointing out that even with 64 levels, which is the maximum value possible allowed by the 6-bit *priority_id*, the solution to the LP problem can be computed in real time with an non-optimized implementation on a general purpose machine. Since this operation is computed on a GOP basis, our scheme introduce at most a one-GOP-delay, but no bottlenecks in the pipeline of operations.

IV. Priority Generation for H.264/AVC Scalable Video Coding extension

As previously mentioned, the method for priority map generation described in Section III can be applied to any scalable coder with the layered structure shown in Fig. 2. In this section we specialize the results of Section III to the case of the H.264/AVC standard [1], [2], hereafter indicated as SVC. More into detail, In Section IV-A we give a brief overview of the fundamental concepts of the SVC standard and the high-level description of coded data, while in Section IV-B we describe how to model the temporal decomposition used in SVC in order to obtain the vectors used in problem (10).

A. SVC Essentials

In SVC a video sequence is essentially processed in layers. The lower layer is called Base Layer (BL), and it is independently coded using an H.264/AVC coding scheme, generating a part of the SVC bit-stream that can be decoded by H.264/AVC compatible decoders. All the other layers are called Enhancement Layers (EL). Specifically, in SVC there are three types of enhancement layers, namely: the Spatial Enhancement Layer (SEL) that provides spatial scalability, the Coarse Grain Scalability enhancement layer (CGS) and the Medium Grain Scalability enhancement layer (MGS) that provide quality scalability. Temporal scalability is achieved by hierarchical B-frames decomposition within each layer [29], followed by processing the layers in Group of Pictures (GOPs), where the GOPs are separated by the so-called key-pictures.

To increase the coding efficiency, enhancement layers are predicted from the base layer or from other enhancement layers using inter-layer prediction tools. These tools introduce additional coding modes to the classical inter- and intra- modes of H.264/AVC. Three inter-layer prediction tools have been introduced in SVC: inter-layer motion prediction, inter-layer residual prediction, and inter-layer intra prediction. Briefly, with inter-layer motion prediction the motion information (motion vectors and MB partition information) associated to other layers can be reused, after a suitable scaling. With inter-layer residual prediction, the residual signal of the other layers is used (with appropriate scaling) to predict the residual signal of the current layer. With inter-layer intra prediction it is possible to predict the intra-signal from the intra-MBs in the reference layer.

CGS and MGS use the inter-layer prediction tools in a similar way, without any scaling of reference layer information, but with some differences in the prediction of key-pictures. Moreover, they use different signaling. In fact, CGS is conceptually similar to spatial scalability with each layer having the same spatial resolution. CGS does not provide flexible SNR extraction, since the number of available rates is equal to the number of layers and, as in the case of spatial scalability, it is possible to switch between layers only at Instantaneous Decoder Refresh (IDR) pictures. MGS has been introduced to increase flexibility, with the possibility to discard quality levels at the picture level, and to distribute enhancement layer transform coefficients among different NALUs (called MGS vectors) in order to enable a finer extraction. Since MGS enables higher flexibility, in this work we consider only the combined scenario with spatial scalability and MGS.

With MGS coding, the process of motion-compensated prediction could introduce a drift. A drift describes the effect of an unsynchronized motion-compensated prediction loop between the encoder and the decoder, and could arise, for example, when quality refinement packets (used for the prediction at the encoder) have been discarded from the bit-stream. With MGS the drift is controlled by means of key-pictures. For each picture, a flag is transmitted which signals whether the base quality reconstruction or the enhancement layer reconstruction of the reference pictures is employed for motion-compensated prediction. All the frames of the coarsest temporal level are transmitted as key-pictures, and therefore no drift is introduced in these pictures. In contrast to that, all temporal refinement pictures typically use the reference with the highest available quality for motion-compensated prediction, thus enabling high coding efficiency but introducing a drift.

Coded video data are organized into Access Units (AUs), where each AU contains the data for a single picture. Within an AU, data are distributed into NALUs, each one identified by the following fields of the NALU header: *dependency_id* for the spatial resolution, *temporal_id* for the temporal level, and *quality_id* for the quality level. Additionally, the *priority_id* field can be used to define the "level of importance", i.e., the quality layer. If the *priority_id*s are assigned, the adaptation can be performed by discarding NALUs in decreasing order of *priority_id*. Within an AU, SVC enables each layer to be inter-predicted from any layer with lower *dependency_id* and/or *quality_id*, generating several prediction structure possibilities.

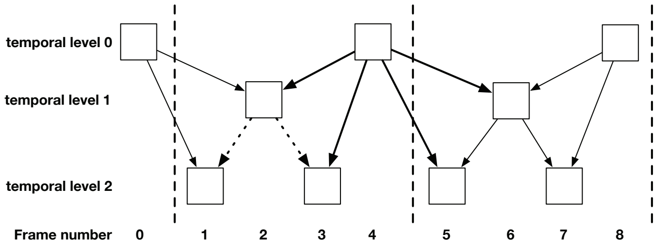


Fig. 3. Prediction path of frames belonging to different temporal levels in a Hierarchical B-frame prediction structure with GOP size equal to 4 pictures.

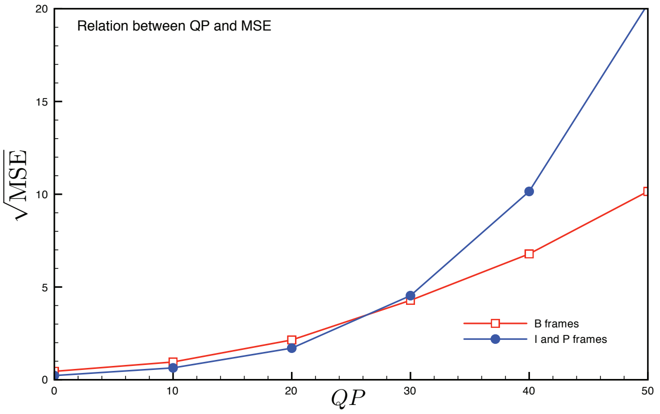


Fig. 4. Relation between Quantization parameter (QP) and Mean Square Error (MSE). This relation has been experimentally derived by averaging the results for different test sequences and encoder settings. It has to be noted that we use the same law for I and P frames, since they show similar characteristics.

The choice of the used prediction path affects both the coding performance and the robustness with respect to the drift. Nevertheless, most of the applications using SVC adopt the most efficient solution (in terms of coding efficiency), predicting each layer from the next lower layer in terms of *dependency_id* or *quality_id*.

B. Distortion model for SVC

Differently from other methods proposed in literature, in this paper the distortion contribution of each NALU is based on a distortion model. The use of a distortion model has the advantage that it does not require bit-stream decoding and, in principle, it enables the generation of a priority map at each node of a network. This feature is particularly suitable in a P2P streaming system, since each peer or group of peers could generate a different priority map that better adapts to the characteristics of the child peers.

Consider a specific GOP. Each NALU in the GOP is associated with a frame (denoted i in the following), a resolution (denoted S_h in the following) and a quality (associated with a QP and denoted Q_k in the following). For the sake of brevity we will use the notation (i, S_h, Q_k) to denote the NALU relative to frame i, resolution S_h and quality Q_k. The QP associated to the quality level Q_k will be denoted as QP_k in order to avoid the more cumbersome notation QP_{Q_k}.

We will need the concept of *prediction path*.

Definition 4. *To a GOP we associate a directed graph (G, E) where the elements of G, the set of nodes, are the frames in* *a GOP and there is an arc $(i, j) \in E \subseteq G \times G$ if frame j is predicted from frame i. A* prediction path *for a frame i is a path in (G, E) that starts from i. We will denote with $\mathcal{N}_i(\ell)$ the number of prediction paths of length ℓ that start from i.*

Example 1

In the case of Fig. 3 (that, incidentely, is a pictorial representation of the graph associated with the shown GOP) one has $\mathcal{N}_2(1) = 2$ and $\mathcal{N}_2(\ell) = 0$ for $\ell > 1$ since frames 1 and 3 are predicted from frame 2, but no frame is predicted from frames 1 and 3. As another example, $\mathcal{N}_0(2) = 2$ since frames 1 and 2 are predicted from frame 0 and frames 2 and 3 are predicted from frame 2, but no frame is predicted from frame 1.

Our objective is to obtain an estimate of the distortion increase $D^i_{S_h,Q_k}$ that one experiences when the NALU (i, S_h, Q_k) is lost. Computing $D^i_{S_h,Q_k}$ is not trivial because of the prediction employed by SVC both across frames in the same GOP and across resolution levels in the same frame. Of course, one could get the exact value of $D^i_{S_h,Q_k}$ by decoding the bit-stream, but this approach is too expensive from a computational point of view. Observe, however, that we do not need to know the exact value $D^i_{S_h,Q_k}$, but that an estimate of $D^i_{S_h,Q_k}$ good enough for assigning priorities will suffice. For example, it is acceptable that the estimated values $D^i_{S_h,Q_k}$ differ from the actual distortions by a constant offset and/or scale factor.

We will need the following definition

Definition 5. *We will denote with $\mathcal{F}(i, S_h, Q_k)$ (called Distortion on Frame) the distorsion increase* on frame i only *when the NALU (i, S_h, Q_k) is lost.*

Note that (i, S_1, Q_1) carries the most coarsely quantized versions of the coefficients, therefore $\mathcal{F}(i, S_h, Q_k)$ is expected to depend mainly on the QP value QP_1 associated to Q_1. NALU (i, S_1, Q_k), $k > 1$, carries instead the details that one must add to the coefficient quantized with QP_{k-1} in order to obtain the coefficients quantized with QP_k. It follows that for $k > 1$ it is expected that $\mathcal{F}(i, S_h, Q_k)$ will depend mainly on QP_k and QP_{k-1}.

Actually, it turns out that a better model takes into account not only the QPs, but also the frame type, since B frames have a sensibly different behaviour than frames I and P. This suggests the following model for $\mathcal{F}(i, S_h, Q_k)$

$$\mathcal{F}(i, S_h, Q_k) = \mathcal{E}(t_i, \text{QP}_k) - \mathcal{E}(t_i, \text{QP}_{k-1}) \tag{15}$$

where $\mathcal{E}(t_i, \text{QP}_k)$, shown in Fig. 4, was experimentally determined.

As said before, the effect of the loss of NALU (i, S_h, Q_k) is not limited to the frame i, so we expect $D^i_{S_h,Q_k}$ (the distorsion induced on the GOP) to be larger than $\mathcal{F}(i, S_h, Q_k)$ (the distorsion induced only on frame i). In order to get an estimate of $D^i_{S_h,Q_k}$ we correct the value of $\mathcal{F}(i, S_h, Q_k)$ by multiplying it by a factor W_i that takes into account the number of frames that are predicted from frame i, that is,

$$D^i_{S_h,Q_k} = W_i \, \mathcal{F}(i, S_h, Q_k). \tag{16}$$

Here,

$$W_i = 1 + \sum_{\ell \geq 1} \left(\frac{1}{4}\right)^{\ell} \mathcal{N}_i(\ell) \tag{17}$$

is a corrective weight that depends only on the frame number i and takes into account the fact that when NALU (i, S_h, Q_k) is lost the distortion of the frames predicted from i also increases.

Remark 8

Note that if no frame is predicted from frame i, then $\mathcal{N}_i(\ell) = 0$ for every $\ell \geq 1$ and (17) gives $W_i = 1$, as expected since the loss of (i, S_h, Q_k) influences only frame i.

In order to justify (17) observe that every frame at a temporal level $m > 0$ is predicted from the average of the two adjacent frames at the level $m - 1$. It is not difficult to show that because of the factor $1/2$ involved in the average, if frame j is directly predicted from i, the distorsion induced on frame j by the loss of NALU (i, S_h, Q_k) is the contribution $\mathcal{F}(i, S_h, Q_k)$ of (i, S_h, Q_k) multiplied by $1/4$. In general, it is not difficult to show that the effect of the loss of (i, S_h, Q_k) is multiplied by $1/4$ at each step along a prediction path, so that, if there is a prediction path of length ℓ from i to j, the distorsion induced on frame j by the loss of (i, S_h, Q_k) is $\mathcal{F}(i, S_h, Q_k)$ of (i, S_h, Q_k) multiplied by $(1/4)^{\ell}$.

Using this distortion model, we set $c^i_{S_h,Q_k} = D^i_{S_h,Q_k}$ in (11), where $D^i_{S_h,Q_k}$ is given in (16). The ILP objective function becomes

$$Z = \max \sum_{i} \sum_{S_h} \sum_{Q_k} x^i_{S_h,Q_k} c^i_{S_h,Q_k}. \tag{18}$$

Note also that the additive form of (18) is acceptable, as discussed in Section III, in view of the fact that, in our application, we assume that the base-layer is always transmitted and correctly received, so that the NALUs involved in the procedure correspond to approximately uncorrelated information.

Finally, it is worth observing that, in our model, the evaluation of the coefficients $D^i_{S_h,Q_k}$ only depends on the encoding parameters (GOP structure, QP values etc.) and not on the specific video content, so that they can be computed off-line and used for the whole sequence. More refined distortion models could be integrated that also take into account the used motion vectors, but this would then require to recompute the coefficients $D^i_{S_h,Q_k}$ for each GOP.

V. Experimental Results

In this section we present the results obtained simulating a P2P streaming system with scalable video content and traffic congestion. The tests were performed using the Oversim P2P simulation framework [30]. The framework was extended with an ad-hoc protocol implemented to support streaming of scalable content with priorities.

The simulation environment is built using a tree-based approach for data dissemination. In our experiments, we consider a full ternary tree with one source and four levels, in order to simulate 120 peers. Note that the purpose of the experiments is to evidence the effect of different congestion control procedures in a tractable setting, and not to provide a fully realistic simulation of a large P2P network. Each network link has a different randomly generated value of congestion. In particular, for each link ℓ we randomly generate $\eta_\ell \in [0, 1]$ indicating the bit rate reduction in each link with respect to the rate required for full transmission.

Example 2

For instance, if the bit-rate required is 1 Mbit/s and $\eta_\ell = 0.3 = 30\%$, then the maximum available rate over link ℓ is 700 kbit/s and we will say that the congestion of link ℓ is equal to 30%. A link with $\eta_\ell = 0$ is able to transmit the whole content, while a link with $\eta_\ell = 1$ is fully congestioned and it cannot transmit anything.

Clearly nodes that are farther from the root of the tree are expected to experience an higher congestion and receive less data. In order to avoid pathological cases, congestion configuration was forced to guarantee that each node receives at least 50% of the full scalable bit-stream, and the same configuration has been used in all the simulations. We observed that the average experienced congestion was about 15% for the nodes closer to the source, approximately $40-45\%$ for the leaf nodes and approximately $25-35\%$ for the intermediate nodes.

Simulations have been performed using test sequences[1] with different features. The video content has been generated by encoding each sequence using the scalable extension of H.264/AVC. In our tests, we do not consider spatial scalability, but only quality scalability generated with a Medium Grain Scalability (MGS) approach and MGS vector mode, in order to generate a base layer plus seven enhancement layers for each frame. In order to obtain a good tradeoff between coding performance and robustness to drift (see Section IV), we code each sequence with a GOP size of 16 frames using an IDR period equal to the GOP size, in order to guarantee that eventual packet drops within a GOP do not affect successive GOPs. The quantization parameter (QP) is set in order to cover a wide range of qualities. In our experiments we consider a QP range from 37 to 24. The metric used for the evaluation was the average decoded streaming quality in terms of PSNR. Before streaming, the SVC stream is processed by the network source in order to generate the priority map, described using the *priority_id* field of NAL unit header. In all the experiments we use 64 values of priority, that correspond to the maximum allowed by the 6 bits of *priority_id* field. An high number of priority values enables to better adapt to fluctuations of the available rate. We additionally assume that packet drop does not incur in base layer data. This is a reasonable assumption, since the base layer represents approximately 10% of the bit-stream and could be protected by adding appropriate redundancy.

The goal of this evaluation process was to determine the behaviour of the P2P system adopting different approaches for congestion and rate control:

- Approach 1 (PROPOSED): The first approach is the proposed method for rate control with priority map. The server uses the procedure of Section III to generate priorities for NALUs in the ELs. Each peer adopts a bandwidth estimation algorithm to determine the level of congestion on the transmission link and applies the algorithm described in sections II to transmit the data to the other peers.
- Approach 2 (SVC QUALITY): The second approach still adopts a bandwidth estimation algorithm to determine the level of congestion but transmits the data units in order of quality levels associated with the SVC layers. Referring to Fig. 2, this approach transmits data by horizontal planes.
- Approach 3 (UNCONTROLLED): The last approach does not use any bandwidth estimation algorithms. It tries to transmit as much data as possible incurring to potential packet drop by the network depending on the congestion level.

  Losses are modeled by using a Gilbert-Elliot model, with loss probabilities chosen in order to get a transmitted bit-rate equal to the available one. More precisely, we used the Gilbert-Elliot model associated with the finite state machine shown in Fig. 5 with the convention that if the machine is in state *Good*, then the packet arrives, otherwise it is lost. The probabilities used in the model of Fig. 5 are computed by solving

$$\eta_\ell = \frac{p_{GB}}{1 + p_{GB} - p_{BG}} \tag{19a}$$

$$\mathfrak{L}_{\text{burst}} = \frac{1}{p_{BG}} \tag{19b}$$

  where $\mathfrak{L}_{\text{burst}}$ is the average length of a burst of losses. In all the experiments we used $\mathfrak{L}_{\text{burst}} = 10$.

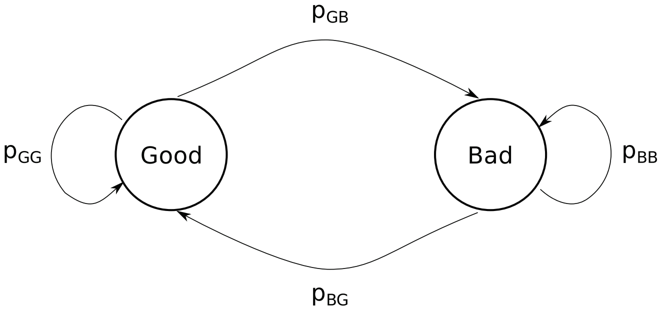


Fig. 5. Gilbert-Elliot model used for the simulation of packet losses.

In all approaches, if an enhancement layer packet is lost, no concealment is done and the video is decoded using the received packets only.

Remark 9

The aim of the experiment is the validation of the proposed method for rate control with priority map. Consequently other aspects of a P2P system that are more related to network issues, as the churning problem, bit error rate, delay, etc..., are not addressed in our experiments.

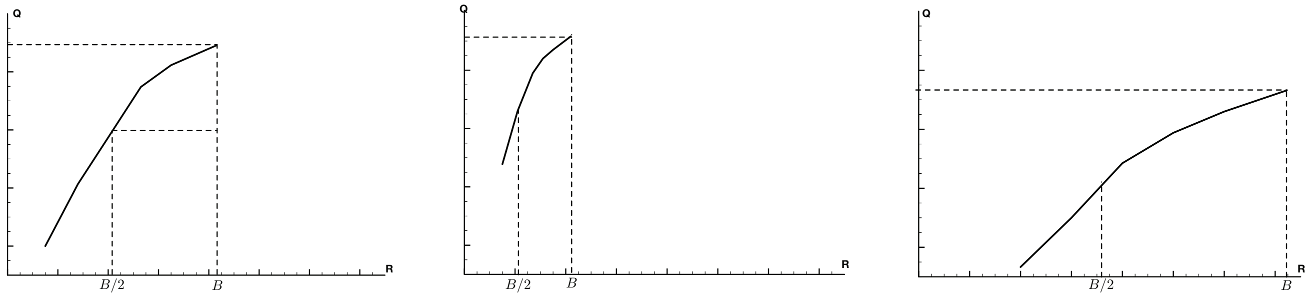
An overview of the results of the experiments is reported in Table I. For each test sequence we present the rate required to transmit the full stream, the maximum decoded quality (if the full stream is received) and the average performance for each approach in case of network congestion. The PSNR values are averages of the quality experienced by all the peers in the network. Note that the proposed Approach 1 enables the best performance for all the test sequences. In comparison to Approach 2, the gain is at least 1 dB. This demonstrates how the proposed method for priority generation enables better Rate Distortion performance. Results obtained with Approach 3 suggest that the performance quickly degrades without appropriate mechanisms for rate control.

In order to investigate in greater detail the behavior of the

[1] The test sequences were downloaded from ftp://ftp.tnt.uni-hannover.de

TABLE I
OVERVIEW OF EXPERIMENTAL RESULTS.

Sequence	$R^{TOT}(Kb/s)$	Max Quality (dB)	**Approach 1**	**Approach 2**	**Approach 3**
BasketballDrill	8300	38.9	37.9	36.8	32.8
BQmall	4200	39.2	38.65	37.53	33.01
PartyScene	14500	37.3	36.1	34.5	29.6
RaceHorse	8900	38.8	37.44	36.4	31.06
Keiba	5500	40.1	38.97	38.45	34.00



(a) BasketballDrill

(b) BQmall

(c) PartyScene

Fig. 6. Rate-Quality characteristics for test sequence BasketballDrill (6a), BQmall (6b) and PartyScene (6c).

three considered approaches, in the following we presents detailed results for a subset of the test sequences, namely: BasketballDrill, BQmall and PartyScene. Rate-Quality plots for these sequences are shown in Fig. 6. For each test sequence, in Fig. 7 we present the frame-by-frame quality detail for a peer closer to the source (Fig. 7a, 7c, 7e) and for a peer belonging to the tree leaves (Fig. 7b, 7d, 7f). As it can be noted in Fig. 7e and 7f, for the PartyScene sequence we observe a rapid performance decrease even for the nodes that experience a small congestion (i.e., those directly attached to the source). This also affects data propagation along the tree, since for this sequence we observe the larger difference of mean quality between nodes closer to the source and leaf nodes, especially for Approaches 2 and 3 where this difference reaches 2 dB. For Approach 1 the difference is limited to 1.2 dB. Furthermore, it has to be noted that such sequence is particularly vulnerable to uncontrolled packet losses, as shown by the poor performance provided by Approach 3. It is interesting to note from Fig. 7e that for intra frames Approaches 1 and 2 give the same quality. For this particular sequence, priority levels tend to coincide with SVC layers, so that Approaches 1 and 2 happen to discard the same information in intra frames.

Differently from the PartyScene sequence, BQmall is more robust to rate reductions due to congestion. It can be noted in Fig. 7c and 7d that the difference of mean quality between nodes closer to the source and leaf nodes is limited to 0.5 dB. Furthermore, it is interesting to note that, although Approach 2 offers performance comparable to Approach 1 for nodes close to the source, the difference between the two approaches increases as one moves from the source to the leaves. Sequence BasketballDrill presents performance between BQmall and PartyScene. It has to be noted that the results in the presence of data congestion follow the Rate-Quality characteristics shown in Fig. 6.

VI. CONCLUSION

In this paper a two-stage procedure for rate control has been proposed. The first stage, based on ILP, optimally labels packets with priority levels and it can be performed at the encoder. The second stage prioritizes the packets for transmission, according to the available rate and requires only to know the priorities associated to the packets, so it can be implemented at the transport level, autonomously by each peer. Examples of use with H.264/SVC have been presented and the performance of the scheme verified by means of experiments. It has been found that the proposed approach outperforms uncontrolled transmission, and that the proposed priority selection gives substantial advantages over other simplified procedures.

APPENDIX A
PROOF OF PROPERTY 1

Our first step will be to simplify the problem by replacing constraint $\mathbf{q}^t\mathbf{u} \leq B$ with $\mathbf{q}^t\mathbf{u} = B$. We will need the following notation: if $\mathbf{a}, \mathbf{b} \in \mathbb{R}^L$, we will write $\mathbf{a} \gneq \mathbf{b}$ if $\mathbf{a}_i \geq \mathbf{b}_i$ for every i and $\mathbf{a} \neq \mathbf{b}$ (so that there is at least a j such that $\mathbf{a}_j > \mathbf{b}_j$). The following lemma is a direct consequence of Lemma 1.

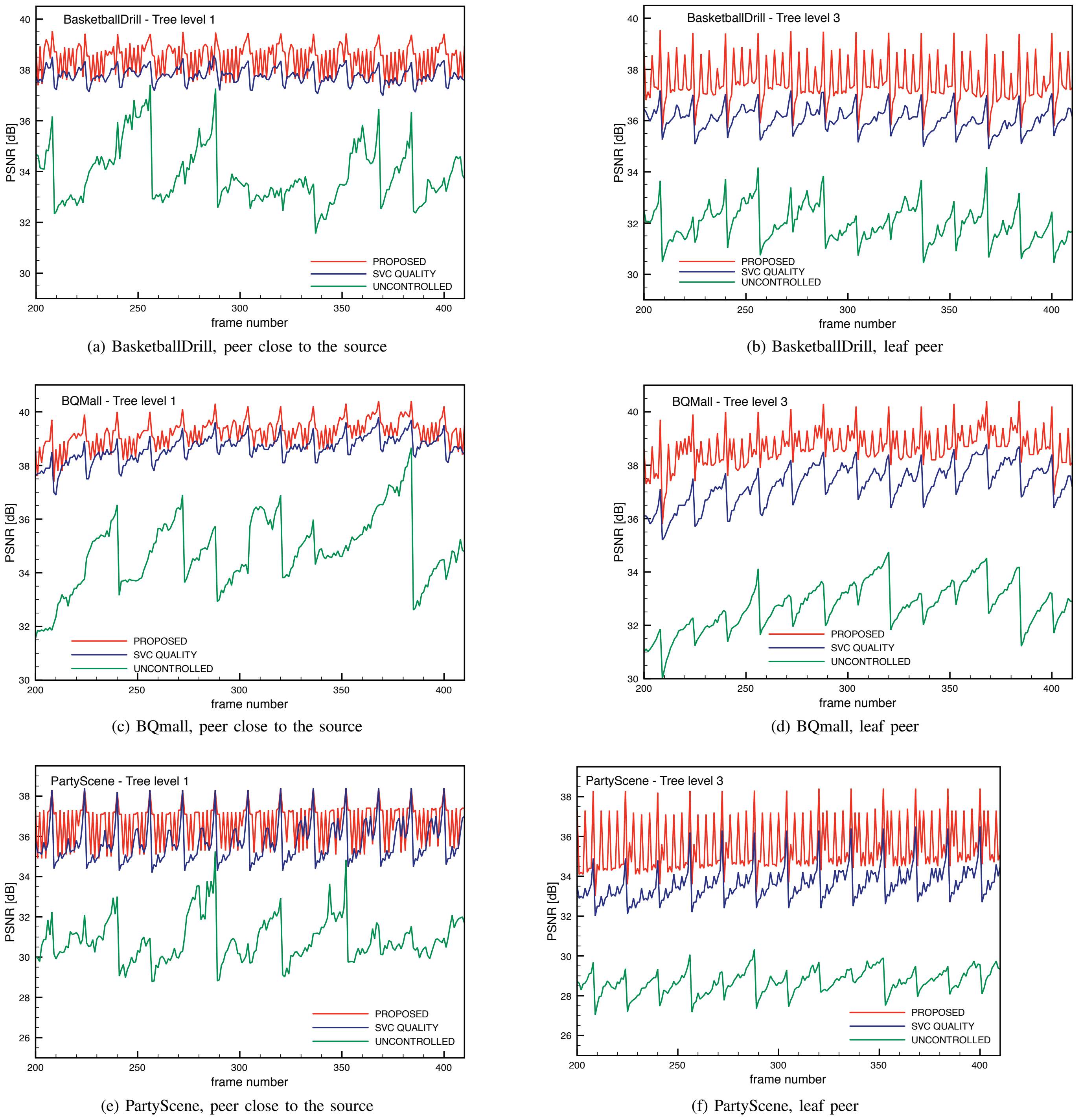
Lemma 2. *Let H be as defined in (2). If $\mathbf{a}, \mathbf{b} \in H$ and $\mathbf{a} \gneq \mathbf{b}$, then $\mathcal{Q}(\mathbf{a}) > \mathcal{Q}(\mathbf{b})$.*

Lemma 3. *If $R_L > B$, then the optimal $\mathbf{q}$ satisfies the constraint in (6) with the equality sign, that is, $\mathbf{u}^t\mathbf{q}^{(opt)} = B$.*

Proof: We will prove the contrappositive, that is, if $\mathbf{q}$ is such that $\mathbf{q}^t\mathbf{u} < B$, then $\mathbf{q}$ cannot be optimal. Suppose $\mathbf{u}^t\mathbf{q} < B$. Let $\mathbf{Q} = [r_1, r_2, \ldots, r_L]$. Note that $\mathbf{q} \lneq \mathbf{Q}$ since $\mathbf{u}^t\mathbf{Q} = R_L > B$. Consider the convex combination

$$\mathbf{x}_\lambda := \lambda\mathbf{Q} + (1-\lambda)\mathbf{q} = \mathbf{q} + \lambda(\mathbf{Q} - \mathbf{q}); \quad \lambda \in [0, 1] \quad (20)$$

Since H is convex, $x_\lambda \in H$ for every $\lambda \in [0, 1]$. Note that if $\lambda > 0$, then $\mathbf{x}_\lambda \gneq \mathbf{q}$. Since $\mathbf{u}^t\mathbf{q} = \mathbf{u}^t\mathbf{x}_0 < B$ one can find a



(a) BasketballDrill, peer close to the source

(b) BasketballDrill, leaf peer

(c) BQmall, peer close to the source

(d) BQmall, leaf peer

(e) PartyScene, peer close to the source

(f) PartyScene, leaf peer

Fig. 7. Frame-by-frame detail for test sequences BasketballDrill, BQmall and PartyScene.

$\lambda_0 > 0$ such that $\mathbf{u}^t \mathbf{x}_{\lambda_0} < B$. Since $\mathbf{Q} - \mathbf{q} \gneq 0$ one deduces $\mathbf{x}_{\lambda_0} \gneq \mathbf{q}$, therefore $\mathbf{q}$ is not optimal. ■

The second step is to show that $\mathbf{p}^{(\mathrm{opt})}$ can have at most one component that is not 0 nor 1.

Lemma 4. *Suppose* $\mathbf{q} \in \mathcal{Q}$ *is such that* $\mathbf{u}^t \mathbf{q} = B$*. Vector* $\mathbf{q}$ *is not optimal if there exist* $a > b$ *such that*

$$q_a > 0 \qquad ; \qquad q_b < r_b \tag{21}$$

Proof: The idea is that we can lower q_a while increasing q_b in order to mantain the constraint $\mathbf{u}^t \mathbf{q} = B$ and this will increase the objective function $\mathcal{Q}(\mathbf{q})$. To such an end choose ϵ such that $0 < \epsilon < \min(q_a, r_b - q_b)$. Such an ϵ exists because of (21). Let $\mathbf{d}^{(a,b)}$ be the "dipole" defined as $\mathbf{d}_a^{(a,b)} = -1$, $\mathbf{d}_b^{(a,b)} = 1$ and $\mathbf{d}_i^{(a,b)} = 0$ if $i \neq a, b$. Let $\hat{\mathbf{q}} = \mathbf{q} + \epsilon \mathbf{d}^{(a,b)}$. Note that $\mathbf{u}^t \hat{\mathbf{q}} = B$ and $\hat{\mathbf{q}} \in \mathcal{Q}$ (since $q_a - \epsilon > 0$ and $q_b + \epsilon < r_b$). In order to compute the difference of objective functions $\mathcal{Q}(\hat{\mathbf{q}}) - \mathcal{Q}(\mathbf{q})$ define $f : [0, \epsilon] \to \mathbb{R}$ as $f(x) := \mathcal{Q}(\mathbf{q} + x\mathbf{d}^{(a,b)})$ and observe that

$$\mathcal{Q}(\hat{\mathbf{q}}) - \mathcal{Q}(\mathbf{q}) = f(\epsilon) - f(0) = \int_0^\epsilon f'(x) dx. \tag{22}$$

Since

$$\begin{aligned} f'(x) &= \sum_{i=1}^{L} \mathbf{d}_i^{(a,b)} \frac{\partial \mathcal{Q}}{\partial q_i}\Big|_{\mathbf{q}+x\mathbf{d}^{(a,b)}} \\ &= \frac{\partial \mathcal{Q}}{\partial q_b}\Big|_{\mathbf{q}+x\mathbf{d}^{(a,b)}} - \frac{\partial \mathcal{Q}}{\partial q_a}\Big|_{\mathbf{q}+x\mathbf{d}^{(a,b)}} \end{aligned} \qquad (23)$$

and (23) is always positive because of condition (3), it follows that the integral in (22) is positive and that $\mathcal{Q}(\hat{\mathbf{q}}) > \mathcal{Q}(\mathbf{q})$. Therefore, $\mathbf{q}$ is not optimal. ■

From Lemma 4 an easy corollary follows.

Corollary 1. *If (3) holds, then* $\mathbf{q}^{(opt)}$ *is a step vector.*

Proof: According to Lemma 4, if $\mathbf{q}^{(\text{opt})}$ is optimal and $0 < \mathbf{q}_K^{(\text{opt})} < r_K/B$, then it must be $\mathbf{q}_\ell^{(\text{opt})} = 0$ if $\ell > K$ and $\mathbf{q}_\ell^{(\text{opt})} = r_\ell/B$ if $\ell < K$, that is, $\mathbf{q}^{(\text{opt})}$ is a step vector. ■

Lemma 5. *There is a unique step vector* $\mathbf{q}$ *such that* $\mathbf{u}^t\mathbf{q} = 1$.

Proof: Observe that if $\mathbf{q} \in V_K$, then

$$\begin{aligned} \mathbf{u}^t\mathbf{q} &= \sum_{i=1}^{L} q_i = \underbrace{\sum_{i=1}^{K-1} r_i/B}_{R_{K-1}/B} + q_K + \underbrace{\sum_{i=K+1}^{L} q_i}_{0} \\ &= R_{K-1}/B + p_K r_K/B \end{aligned} \qquad (24)$$

From (24) it follows that if $\mathbf{q} \in V_K$, then $\mathbf{u}^t\mathbf{q} \in M_K := [R_{K-1}/B, R_K/B)$. Since intervals M_K are mutually disjoint, there is only one K such that $1 \in M_K$. The uniqueness of $\mathbf{q}$ easily follows. ■

Now Property 1 follows easily. Indeed, by Corollary 1, $\mathbf{q}^{(\text{opt})}$ must be a step vector and by Lemma 5 K is uniquely determined by condition $1 \in [R_{K-1}/B, R_K/B)$, that is equivalent to (7). Value $\mathbf{q}_K^{(\text{opt})}$ is uniquely determined by condition $\mathbf{u}^t\mathbf{q}^{(\text{opt})} = 1$ that gives $\mathbf{q}_K^{(\text{opt})} = 1 - R_{K-1}/B$.

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. CSVT*, vol. 17, no. 9, pp. 1103–1120, 2007.

[2] "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 8 : Consented in July 2007.

[3] L. Eggert and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers," RFC 5405 (Best Current Practice), Nov. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5405.txt

[4] K. Kar and L. Tassiulas, "Layered multicast rate control based on lagrangian relaxation and dynamic programming," *Sel. Areas Commun., IEEE J.*, vol. 24, no. 8, pp. 1464 –1474, Aug. 2006.

[5] P. Troubil and H. Rudov, "Integer linear programming models for media streams planning," no. 3, pp. 509–522, 2011.

[6] L. Chen, T. Ho, S. Low, M. Chiang, and J. Doyle, "Optimization based rate control for multicast with network coding," in *INFOCOM 2007. 26th IEEE International Conf. Comput. Commun.. IEEE*, May 2007, pp. 1163–1171.

[7] E. Setton, J. Noh, and B. Girod, "Rate-distortion optimized video peer-to-peer multicast streaming," in *Proc. ACM Workshop Advances Peer-To-Peer Multimedia Streaming*, ser. P2PMMS'05. New York, NY, USA: ACM, 2005, pp. 39–48. [Online]. Available: http://doi.acm.org/10.1145/1099384.1099390

[8] R. Birke, C. Kiraly, E. Leonardi, M. Mellia, M. Meo, and S. Traverso, "Hose rate control for p2p-tv streaming systems," in *Peer-to-Peer Comput. (P2P), 2011 IEEE International Conf.*, 31 2011-Sept. 2 2011, pp. 202–205.

[9] P. Zhu, W. Zeng, and C. Li, "Joint design of source rate control and QoS-aware congestion control for video streaming over the internet," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 366–376, 2007.

[10] P. Zhu, H. Yoshiuchi, and S. Yoshizawa, "QoS-aware multicast for internet video applications," in *Packet Video 2007*, Nov. 2007, pp. 46–51.

[11] N. M. Tiglao, J. M. Monteiro, A. M. Grilo, M. S. Nunes, and J. M. Xavier, "Dynamic programming optimization of multi-rate multicast video-streaming services," *Science Diliman*, vol. 22, no. 1, 2011. [Online]. Available: http://journals.upd.edu.ph/index.php/sciencediliman/article/view/1619/2004

[12] T. Kim and M. Ammar, "Optimal quality adaptation for mpeg-4 fine-grained scalable video," in *INFOCOM 2003. Twenty-Second Annual Joint Conf. IEEE Comput. Commun., IEEE Societies*, Mar. 2003, pp. 641–651.

[13] P. D. De Cuetos and K. W. Ross, "Unified Framework for Optimal Video Streaming." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.4772

[14] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized rate-distortion extraction with quality layers in the Scalable extension of H.264/AVC," *IEEE Trans. CSVT*, vol. 17, no. 9, pp. 1186–1193, 2007.

[15] "Multi layer quality layers," ITU-T VCEQ JVT-S043, Geneva, Apr. 2006.

[16] E. Maani and K. Katsaggelos, "Optimized bit extraction using distortion modeling in the scalable extension of H.264/AVC," *IEEE Trans. IP*, vol. 18, no. 9, pp. 2022–2029, 2009.

[17] C. Gu, D. Zhao, and X. Ji, "Fast rate allocation based on distortion estimation modeling in scalable video coding," in *Proc. SPIE*, 2008.

[18] T. Rusert and J. Ohm, "Application to quality layer assignment in h.264/avc based scalable video coding," in *Proc. International Conf. Acoustics, Speech, Signal Process. (ICASSP)*, 2007.

[19] T. Cong Thang, J. Kang, J.-J. Yoo, and Y. Ro, "Optimal multi-layer adaptation of svc video over heterogeneous environments," *Advances Multimedia*, 2008.

[20] R. Li, J. Sun, and W. Gao, "Fast weighted algorithms for bitstream extraction of svc medium grain scalable video coding," in *Proc. International Conf. Multimedia Expo (ICME)*, 2010.

[21] L. Lima, M. Mauro, and R. Leonardi, "Optimal rate adaptation in the scalable extension of H.264/AVC with combined scalability," in *Proc. European Signal Process. Conf. (EUSIPCO)*, 2011.

[22] P. Zhu, W. Zeng, and C. Li, "Cross-layer design of source rate control and QoS-aware congestion control for wireless video streaming." Los Alamitos, CA, USA: IEEE Comput. Society, 2006, pp. 1133–1136.

[23] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," RFC 5348 (Proposed Standard), Sept. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5348.txt

[24] E. Lawler, *Combinatorial Optimization: Networks and Matroids*. Dover Publications, inc., Mineola, NY, 2001.

[25] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer Verlag, 2004.

[26] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementation*. John Wiley and Sons, 1990.

[27] I. INC., "ILOG CPLEX 12.1 reference manual," Incline Village: ILOG Inc., CPLEX Div., 2009.

[28] G. Dantzig, *Linear Programming and Extensions*. Princeton University Press and the RAND Corporation, 1963.

[29] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical b pictures," JVT input document P014, July 2005.

[30] I. Baumgart, B. Heep, and S. Krause, "Oversim: A flexible overlay network simulation framework," in *Proc. IEEE Global Internet Symp. (GI)*, 2007.

Livio Lima was born in Gavardo (Brescia, Italy) in 1980. He received the degree in Telecommunication Engineering (cum laude) from University of Brescia in 2005, and the PhD in Information Engineering from University of Brescia, in 2009, respectively. From 2005 he was active in the area of digital image-video processing, coding and multimedia applications. In 2007 he was visiting researcher at University of New South Wales (Sydney, Australia). From 2009 he has been postdoc at the University of Brescia. His main research interests cover the field of Digital Signal Processing applications, with a specific expertise on video and image coding, and transmission of visual information. Recent activity on P2P video streaming systems is founded by the Italian Ministry of Education and Research (MIUR).

Marco Dalai was born in Manerbio (Brescia, Italy) in 1979. He received the degree in Electronic Engineering (cum laude) and the PhD in Information Engineering, in 2003 and 2007 respectively, from the University of Brescia. Since 2008, he has been an assistant professor with the Department of Information Engineering in this same university. He is a member of the IEEE Information Theory Society. His main research interests cover the field of classical and quantum information theory, signal processing, and statistical inference for system identification.

Riccardo Leonardi received the Diploma and Ph.D.degrees in electrical engineering from the Swiss FederalInstitute of Technology, Lausanne, Switzerland,in 1984 and 1987, respectively. After spending one year withthe Information Research Laboratory atUniversity of California, Santa Barbara (USA), he joined Bell Laboratories for a period of 3 years as a Memberof Technical Staff.In 1991, he returned briefly to the Swiss FederalInstitute of Technology before being appointed in 1992 at the University of Brescia to leadresearch and teaching in the field of telecommunications. His main researchinterests cover the field of multimedia signal processing applications, with a specificexpertise on visual communications, and content-based analysis of audio-visualinformation. He has published more than 200 papers on these topics. Prof. Riccardo Leonardi has been coordinating several national and international initiatives to conduct research and doctoral training at both National and International levels (US, Europe, South America, ...).

Pierangelo Migliorati was born in Seniga (Brescia, Italy) in 1962. He received the degree in Electronic Engineering (cum laude) from Politecnico di Milano in 1988, and the Master in Information Technology from CEFRIEL Research Centre, Milan, in 1989, respectively. He joined CEFRIEL Research Center in 1990. From 1990 to 1992 he was active in the area of digital image-video processing and coding. From 1993 to 1995 he leaded the Area of Digital Communication, focusing on nonlinear channel modeling and equalization. From 1995 he has been Assistant and then Associate Professor at the University of Brescia, Italy. His teaching activities focus mainly on Digital Signal Processing and Digital Communications. His main research interests cover the field of Digital Signal Processing applications, with a specific expertise on audio-visual communications, and content-based analysis of audio-visual information. He published more than 90 papers on these topics. He has been an active reviewer of several journal and conferences in this research field. He participated to several international research programs in audio-visual communications, funded by the European Commission, the Italian National Council for Research (CNR), and the Italian Ministry of Education and Research (MIUR).

Riccardo Bernardini (M02) was born in Genova, Italy, in 1964. He received the Laurea in Ingegneria Elettronica degree and the Ph.D. degre from the University of Padova, Padova, Italy, in 1990 and 1995, respectively. He spent the last year of his PhD as a visiting scientist in formerly AT&T Bell Laboratories, Murray Hill, NJ. From April 1996 to April 1997, he was a Postdoctoral Fellow with EPFL, Lausanne, Switzerland. He is currently working as a Aggregate Professor in the Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, University of Udine, (Italy) teaching digital signal processing. His main interests are in the area of multidimensional signal processing, wavelets & filter banks, robust transmission. Recently he started working also in networking-related matters, especially on peer-to-peer systems for streaming. He is author of several patents and approximately one hundred scientific publications, mainly on IEEE Transactions and IEEE conferences.

Roberto Rinaldo obtained the “Laurea in Ingegneria Elettronica” degree in 1987 from the University of Padova, Padova, Italy. From 1990 to 1992, he was with the University of California at Berkeley, where he received the MS degree in 1992. He received the Doctorate degree in “Ingegneria Elettronica e dell’Informazione” from the University of Padova in 1992. In 1992 he joined the Dipartimento di Elettronica e Informatica of the University of Padova as a “ricercatore”. Starting from November 1st 1998, he was associate professor in Communications and Signal Processing in the same Department. Since November 2001, he was an associate professor in the “Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica” of the University of Udine. Starting December 2003, he is now a professor in the same department. His interests are in the field of multidimensional signal processing, video signal coding, fractal theory and image coding. Prof. Rinaldo is part of the Telecommunication and Signal Processing group (TSP) of the University of Udine.