Prioritized Distributed Video Delivery With Randomized Network Coding

Nikolaos Thomos, *Member, IEEE*, Jacob Chakareski, and Pascal Frossard, *Senior Member, IEEE*

Abstract—We address the problem of prioritized video streaming over lossy overlay networks. We propose to exploit network path diversity via a novel randomized network coding (RNC) approach that provides unequal error protection (UEP) to the packets conveying the video content. We design a distributed receiver-driven streaming solution, where a client requests packets from the different priority classes from its neighbors in the overlay. Based on the received requests, a node in turn forwards combinations of the selected packets to the requesting peers. Choosing a network coding strategy at every node can be cast as an optimization problem that determines the rate allocation between the different packet classes such that the average distortion at the requesting peer is minimized. As the optimization problem has log-concavity properties, it can be solved with low complexity by an iterative algorithm. Our simulation results demonstrate that the proposed scheme respects the relative priorities of the different packet classes and achieves a graceful quality adaptation to network resource constraints. Therefore, our scheme substantially outperforms reference schemes such as baseline network coding techniques as well as solutions that employ rateless codes with built-in UEP properties. The performance evaluation provides additional evidence of the substantial robustness of the proposed scheme in a variety of transmission scenarios.

Index Terms—Network coding, overlay networks, peer-to-peer (P2P) systems, rate allocation, scalable video delivery, unequal error protection, video streaming.

I. INTRODUCTION

S PURRED by the advances in broadband technologies and video compression, an ever-increasing amount of multimedia content is flowing over our computer networks [1], [2]. This has accelerated the development of new delivery architectures. Among them, peer-to-peer systems have experienced a fast development and emerged as one of the most popular paradigms for online media delivery [3]–[5]. In such media streaming systems, networks are often organized in overlay structures that provide better control of the delivery process.

Overlay networks offer the possibility of employing basic processing operations at intermediate network nodes, in addition to providing increased path or source diversity. These important properties can contribute to improved delivery performance. For example, the nodes can perform simple coding op-
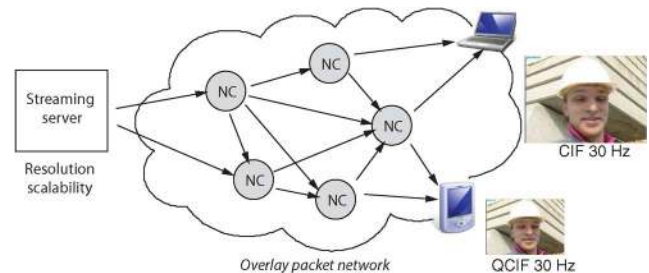
Fig. 1. Illustration of scalable video streaming in overlay networks with network coding (NC) nodes.

erations on packets before transmission in order to increase the goodput. This concept is known as network coding, and it has gone a long way from a purely analytical technique, as introduced originally in [6], to an approach applicable to data dissemination in the Internet [7] at present. Network coding has also attracted a lot of attention for multimedia communication as it enables efficient distributed delivery in lossy overlay networks. This technique also effectively deals with bandwidth variations and packet duplication that typically arise in such networking environments. While computational complexity still represents an issue in network coding, a few algorithms have been proposed recently that successfully employ network coding principles as part of multimedia streaming applications [8]. However, the growing heterogeneity of Internet access links' characteristics in terms of packet loss and bandwidth has created an important need for scalable delivery mechanisms. In particular, multimedia data is typically characterized by a variable importance of the data units in terms of their contribution to the overall reconstructed quality. Therefore, its delivery should be organized such that the peers are served the data efficiently according to their capacity, while a graceful quality degradation is experienced when resources become scarcer.

In the present paper, we address the problem of prioritized media streaming in overlay networks, where network coding operations are specifically designed to cope with media packets of different importance. Since common networks are usually characterized by a large diversity in terms of client capabilities and access speeds, we propose an efficient streaming scheme that allows for multiple levels of quality-of-service in order to accommodate for the network heterogeneity, as illustrated in Fig. 1. We build on the results of randomized network coding (RNC) [9] for the construction of a distributed streaming solution that improves the robustness to erasures without the need for centralized control, as described in [10]. However, we propose a new distributed delivery algorithm where the coding decisions are adapted to prioritized video delivery for receivers with different capabilities that extends our early work [11]. We propose a receiver-driven network coding strategy where the receiving peers request packets from classes with varying importance. Packet

classes can be constructed by considering the unequal contribution of the various video packets to the overall quality of the presentation [12] or simply from the arrangement of data in scalable video streams [13]. Prioritized transmission is then achieved by varying the number of packets from each class that are used in embedded network coding operations that are performed in the overlay nodes.

We formulate a novel optimization problem that characterizes the optimal network coding strategy undertaken at the nodes that forward data to receiving peers as a function of the available bandwidth. The optimal strategy is computed by the receiving peer, which determines the rate allocation between the packet classes that leads to the minimum expected distortion. The receiving peer subsequently requests packets from the parent nodes according to the resulting rate allocation. As the optimization problem is shown to be log-concave, we propose a new low-complexity algorithm that computes the best coding strategy in only a few iterations. It is shown that the proposed scheme has clear advantages over network coding methods that do not consider the importance of packets and leads to more graceful quality degradation when the effective bandwidth decreases. Our technique outperforms baseline network coding algorithms and other unequal error protection (UEP) approaches based on rateless codes that are specifically designed for the delivery of layered media streams.

The rest of the paper is organized as follows. In Section II, we briefly review the general characteristics of network coding as they pertain to networked media, and we describe our novel network coding algorithm. Then, we analyze in Section III the expected distortion at a decoding peer as a function of its network coding strategy. Next, we formulate in Section IV the optimization of the network coding choices that can be achieved via a low-complexity iterative solution. Subsequently, we examine, through simulation experiments, various performance aspects of the proposed scheme in Section V. Finally, we discuss related work in Section VI and conclude the paper with summarizing remarks in Section VII.

II. NETWORK CODING FOR UEP VIDEO STREAMING

A. Practical Network Coding

In this section, we first review briefly the basics of network coding (NC) and its application to practical streaming solutions before presenting our distributed prioritized NC algorithm. Network coding has been originally proposed with the goal of increasing network throughput in push-based data delivery [6]. Instead of simply forwarding packets or symbols on the outgoing network links, the nodes in the network perform linear combinations of the received packets and transmit the coded packets to the destination nodes. The receivers can then recover the original data by receiving and subsequently decoding a sufficient number of linearly combined packets.

The specific coding strategy employed at each node could be globally optimized with a comprehensive knowledge of the network topology. However, such an assumption is not realistic in practical streaming systems. Instead, distributed algorithms are considered where each node independently chooses its coding strategy based on a local network view. RNC [9] is an efficient solution for network coding in distributed settings, since coding coefficients are selected randomly by each network coding node without any need for a central coordination. It can be adapted to practical streaming applications, as proposed in [10], where
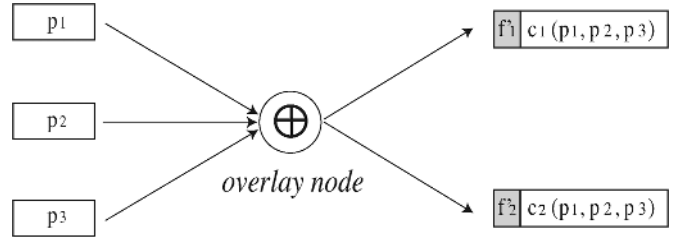

Fig. 2. Overlay node combines incoming packets p_i and generates network coding packets c_m. A header f'_m is appended to each coded packet that carries the coding coefficients.

network nodes independently perform linear combinations of packets and forward them to their neighbors. As the coding coefficients are chosen randomly, a header of constant length is appended to each packet with the coefficient information so that the decoder can decode the stream and recover the original data packets.

The network coding operations can be written as follows. If a node u generates M packets by RNC, then the mth network coded packet c_m is of the form

$$c_m = \sum_{p_i(u) \in \mathcal{S}(u)} f_{m,i} \cdot p_i(u)$$

where i is the index of the ith packet at node u, $\mathcal{S}(u)$ corresponds to the set of packets available at the node, $p_i(u)$ denotes either a network coded packet or a native (uncoded) packet, and $f_{m,i}$ is a random coefficient over the Galois field of size q, GF(q). The basis of the Galois field is typically set to $q = 256$, as it has been shown in [10] that this guarantees high symbol diversity and low probability of building duplicate packets. As the packets combined at a node are actually combinations of the original data packets, the encoded packets can be expressed as a function of the native packets

$$c_m = \sum_{i=1}^{N} f'_{m,i} \cdot n_i$$

where i denotes the index of the ith native packets and N is the total number of native packets, e.g., the number of video packets. The parameters n_i and $f'_{m,i}$ represent, respectively, the native packets and their corresponding coding coefficients after random network coding operations. It is worth noting that some of the coefficients $f'_{m,i}$ can be zero, which means that c_m does not contain information about the native packet n_i. A network coded packet is finally augmented with $N \cdot \log_2(q)$-bit header containing the vector of coding coefficients $[f'_{m,1}, \ldots, f'_{m,N}]$. Note that the header does not grow with the number of hop transmissions, but it depends on the number of native packets. The encoding procedure at a peer node is illustrated in Fig. 2.

An important aspect of streaming applications is the delay constraints that are imposed by the requirements of continuous playback at the receiver. In particular, the network nodes cannot buffer packets for long periods of time. At the same time, they should only combine packets with similar decoding time-stamps so that the decoding is not excessively delayed. In order to deal with the timing constraints, the concept of generations has been introduced in [10]. The packet stream is split into multiple generations and coding operations are restricted to packets within the same generation. Furthermore, since generations are characterized with playback deadline information,

the network nodes only transmit useful packets that correspond to generations whose decoding deadlines have not passed. All the other packets can be deleted from the nodes' buffers. The latency incurred by the streaming application waiting for the media packets to be decoded at the destination obviously becomes dependent on the length of the generation. Typically, a generation can correspond to a group of pictures (GOP) that are the images between two reference frames in encoded video sequences. In the next section, we extend practical network coding into a novel distributed solution with adaptive coding of packets of different importance.

B. Proposed Prioritized RNC

Our objective is to design a novel network coding algorithm in overlay networks that is able to deal with packets of different importance and that increases the likelihood of delivery for higher priority packets. Overlay mesh networks typically exhibit links with heterogeneous bandwidth values and dramatic variations in channel conditions due to random node departures. The traditional streaming systems utilize variations of channel coding algorithms for robust media delivery in this framework. It could be achieved with prioritized coding and UEP [14]–[16], or with variants of multiple description coding (MDC) algorithms [17]. However, most of these methods are hard to implement in distributed settings and require a full knowledge of the overall topology and channel conditions in order to determine the optimal coding strategy. Whenever this information is not available, they often overprotect the streams or conversely fail to provide the required protection.

Network coding in turn is very appealing because it can work with limited knowledge of network resources and conditions, and it can be implemented in a distributed manner. In particular, intermediate network nodes forward on their outgoing link(s) packets that represent linear combinations of the packets received on their incoming links. The destination nodes can then recover the original data by receiving and subsequently decoding a sufficient number of these linearly combined packets. Since packets have different importance, the amount of protection or equivalently the number of network coded packets in each class of importance has to be determined by taking into account the priorities of the packets. Ideally, the mixing operations should not be uniform across all packets arriving at a node, but instead packets with higher importance should be involved in more coding operations.

In order to adapt locally the network coding operations, we propose a scheme where the children peers send requests to their parents, where they specify the relative number of packets from each importance class that they would like to receive. This receiver-driven policy provides a simple way to adapt to the capabilities of the peers without large communication cost in the system. It also provides a finer control of the packet delivery compared to sender-driven error resilient coding or push-based strategies [10]. This becomes particularly important when most of the network nodes subscribe to the video content and not only forward the information, as in P2P networks. In our system, the coding operations are, therefore, driven by the children nodes that determine the optimal amount of coding allocated to each importance class of the data to which they subscribe.

In particular, the packets are organized into C classes depending on their importance, where the layers are organized along decreasing levels of priority. The class c is defined as the set of packets that are linear random combinations of packets from the c most important classes. In practice, the class of a packet is identified by a small header which is appended to the packet. Then, the packet delivery protocol proceeds in two phases. First, as shown in Fig. 3(a), children nodes u_i compute the optimal coding strategy that should be implemented at their parent nodes v_j, based on the available network bandwidth, the expected loss probability, and the importance of packets in each class. They subsequently send a request message to their parents, which specifies the distribution w_i of the relative number of packets in each class that they would like to receive. The parent nodes in turn randomly combine their packets according to the computed coding strategies and forward the corresponding coded packets to their children. This procedure is depicted in Fig. 3(b). A child node finally inspects the incoming packets to determine whether they are innovative, i.e., whether they provide any new information relative to the packets already received. Non-innovative packets are removed from the node's buffer. Based on the state of its buffer and the local network status, the child node then computes again the optimal coding strategy and sends it to its parent nodes. This procedure is repeated periodically. Finally, note that the request-based coding decisions contrasts with the work in [18] proposed for wireless transmission, where a parent node sends the same distribution of network coded packets to all of its children. Our algorithm permits to adapt to each child node on an individual basis, thereby ensuring that the resulting video quality at each of them is maximized for their given specific network conditions.

We show in the next section how a child node can compute the expected quality for each coding strategy. Then, we propose an optimization algorithm that has a simple solution for the child node to decide on the best rate allocation to be requested from its parents.

III. DISTORTION ANALYSIS

In this section, we analyze the expected distortion in a decoding peer, as a function of the coding decision and the local network state. The distortion is dependent on the number of classes that can be decoded, where the probability of decoding a class depends on the number of network coded packets that have been received. We assume that the available bandwidth between nodes is fully used by the network coding system. Its value may depend on background traffic or concurrent applications, and we assume it can be estimated locally.

A child node u requests packets of different classes from its parent nodes. It sends the same request to all its direct parents. The request typically takes the form of a rate distribution vector w, whose cth component w_c denotes the proportion of packets from class c among the requested packets. By definition, we have $\sum_{c=1}^{C} w_c = 1$ and $w_c \geq 0$, where C is the number of packet classes in the network. If we denote $N_{\mathcal{I}}(u)$ the total number of packets requested by the node u, the expected number of packets of class c sent by the node u is given as

$$r_c(u) = \lfloor N_{\mathcal{I}}(u) \cdot w_c \rfloor. \tag{1}$$

These packets are lost with probability $\pi_{\mathcal{I}}(u)$ which is the average packet erasure rate on the incoming links of node u. As the parent nodes fully use the available upload bandwidth $B_{v,u}$ between nodes v and u, the number of transmitted packets $N_{\mathcal{I}}(u)$ is determined based on the overall incoming bandwidth at node u as $N_{\mathcal{I}}(u) = \sum_{v \in \mathcal{I}(u)} B_{v,u}$, where $\mathcal{I}(u)$ denotes the set of parental nodes of node u. For the sake of clarity, we assume
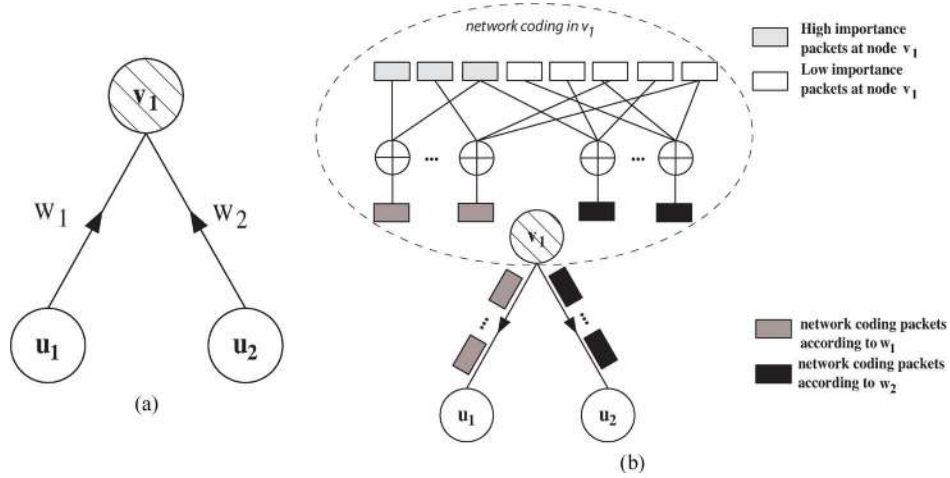
Fig. 3. Communication protocol. (a) Children nodes u_1 and u_2 calculate their optimal coding policies w_1 and w_2. These policies are forwarded to the parent node v_1. (b) Parent generates packets according to w_1 and w_2 and feeds them back to u_1 and u_2. The packets at node v are categorized into two importance groups based on their information content.

that the bandwidth is given in packets per second. The packet loss probability $\pi_{\mathcal{I}}(u)$ is equal to

$$\pi_{\mathcal{I}}(u) = \frac{\sum\limits_{v \in \mathcal{I}(u)} B_{v,u} \cdot p_{u,v}}{N_{\mathcal{I}}(u)} \quad (2)$$

where $p_{u,v}$ is the packet loss probability between nodes u and v.

The video distortion at the client peer is dependent on the number of classes that can be decoded. We denote the quality improvement or distortion reduction after successful decoding of the cth class as d_c. The values of d_c depend on the encoding parameters, e.g., the quantization values, employed by the scalable video codec. The number of native video packets in the first c classes is written as $\beta_c = \sum_{j=1}^{c} \alpha_j$, where α_j stands for the number of source video packets in class j. The total number of source or native packets is $\beta_C = N$. A client is able to decode the cth class as soon as it receives β_c innovative network coded packets. In our network coding algorithm, these packets represent linear combinations of packets from the first c classes. Recall that a packet is called innovative when it increases the rank of the equation system constructed by the set of packets received at a peer node. It should be noted that in our framework, the first class comprises the data from the base layer, the second class comprises the data from both the base and the first enhancement layer, and analogously for the subsequent coding classes.

The distortion experienced at node u is simply written as $D_{max} - D(u)$, where D_{max} represents a constant maximal distortion when no video class can be decoded. The expected reduction in video distortion $D(u)$ is a function of the number of classes that the node u can decode. It can be written as

$$D(u) = \sum_{c=0}^{C} d_c \cdot p_d(c) \quad (3)$$

where $p_d(c)$ denotes the probability that the node u is able to decode c video classes. We denote the probability that an innovative packet arrives at node u as $p_i(u)$. This probability depends on the local packet loss probability, and the probability ρ that an encoded packet at a parent node is innovative. We can provide a lower bound on this later probability that only depends

on the size of the Galois Field, q, i.e., $\rho \geq (1 - 1/q)$ [19]. The equality stands for the case when a node needs one packet to form a full rank equation system. Since we use the same GF size in each packet class, and since the number of packets in each class is small comparatively to the possible number of different packets, we consider that $p_i(u)$ is identical in each class. We further assume that the probability of two parent nodes generating the same network coded packet is negligible. Thus, the (lower bound) value of the probability for receiving an innovative packet is simply given as $p_i(u) = (1 - \pi_{\mathcal{I}}(u)) \cdot \rho$, since a packet should not be erased and it should be innovative.

We can now rewrite the expected reduction in distortion $D(u)$ from (3) as

$$D(u) = \sum_{c=1}^{C} \sum_{l_1=0}^{r_1(u)} \cdots \sum_{l_c=0}^{r_c(u)}$$
$$\sum_{l_{c+1}=0}^{\beta_{c+1}-\beta_c-1} \cdots \sum_{l_C=0}^{\beta_C-\beta_c-\sum\limits_{j=c+1}^{C} l_j - 1}$$
$$\prod_{j=1}^{C} \binom{r_j(u)}{l_j} \cdot (1 - p_i(u))^{l_j} \cdot p_i(u)^{r_j(u)-l_j} \cdot d_c \quad (4)$$

where the product term of the third line of (4) denotes simply the likelihood (probability) of decoding the first c classes exclusively. The sequence of summation terms in the first line of (4) counts the events of receiving enough packets to decode packets up to class c, but not the subsequent classes $(c+1, \ldots, C)$. This latter condition is described with the second sequence of sums in the second line of (4).

Instead of the overall incoming bandwidth at a node, we actually consider the maximum value between the incoming and outgoing bandwidths at a peer node as the capacity constraint in our network coding algorithm. This represents a crucial factor in maintaining a high packet diversity in the network. In particular, the factor $N_{\mathcal{I}}(u)$ in (1) is replaced by the rate limit $\Phi(u)$ that is defined as

$$\Phi(u) = \max\left\{ N_{\mathcal{I}}(u) \cdot (1 - \pi_{\mathcal{I}}(u)), \quad N_{\mathcal{O}}(u) \cdot (1 - \pi_{\mathcal{O}}(u)) \right\}$$

where $\pi_{\mathcal{O}}(u)$ and $N_{\mathcal{O}}(u)$ are, respectively, the average packet erasure rate on the outgoing links at node u, and the overall number of packets sent by this node. The packet loss probability $\pi_{\mathcal{O}}(u)$ is defined similarly to (2). The number of packets from class c received at node u thus becomes

$$r_c(u) = \lfloor \Phi(u) \cdot w_c \rfloor. \tag{5}$$

Specifically, the number of received packets in (1) is replaced by

$$r_c(u) = \begin{cases} \lfloor N_{\mathcal{I}}(u) \cdot w_c \rfloor, & \text{if } \Phi(u) = N_{\mathcal{I}}(u) \cdot (1 - \pi_{\mathcal{I}}(u)) \\ \lfloor N_{\mathcal{O}}(u) \cdot w_c \rfloor, & \text{if } \Phi(u) = N_{\mathcal{O}}(u) \cdot (1 - \pi_{\mathcal{O}}(u)). \end{cases} \tag{6}$$

Finally, we observe that the expected distortion depends on the local network statistics (i.e., the packet loss ratio and the incoming bandwidth at each node), but also on the number of requested packets in each class via $r_c(u)$. As the local network statistics are given, a peer node can still maximize the expected reduction in distortion by optimizing the packet class distribution that represents the relative number of network coded packets requested in each class. We study the optimization of the network coding strategy in the next section.

IV. OPTIMIZED NETWORK CODING STRATEGY

A. Optimal Rate Allocation

We can now formulate the optimization problem solved by each client node u. The node is interested in determining the number of packets it should request from its parents for each packet class. The optimal class distribution $w^\star$ is computed such that it minimizes the expected distortion, or alternatively it maximizes the expected reduction in distortion, as contributed by the received packets. Formally, the optimal rate allocation problem can be written as follows:

$$\begin{aligned} &Rate\ Allocation\ Problem(RAP): \\ &w^\star = \arg\max_{\boldsymbol{w}} D(u), \\ &\text{such that } \sum_{c=1}^{C} w_c = 1 \text{ and } w_c \geq 0, \text{ for } c = 1, \ldots, C. \end{aligned} \tag{7}$$

The peer u is interested in selecting the vector of coefficients $w^\star = [w_1^\star, \ldots, w_C^\star]$ such that the network coding strategy employed by its parent nodes maximizes the received video quality. This distribution is then sent as a request to all the parent nodes of peer u, which then perform network coding operations in order to match the requested distribution.

B. Optimization Algorithm

Every client peer has to solve the RAP problem independently based on local network information. Since the search space is huge, exhaustive algorithms are too complex and cannot be implemented even for a small number of packets. Fortunately, the objective function in (4) is a log-concave function, which leads to a simple iterative solution of the RAP problem in each peer node. We prove below that the objective function is log-concave, and we later present the optimal rate allocation algorithm.

Proposition 1: The expected distortion function $D(u)$ in (4) is log-concave.

Proof: We show that (4) is log-concave by proving that all terms $f(w_c)$ in the product are log-concave, where

$$f(w_c) = \prod_{j=1}^{C} \binom{r_j(u)}{l_j} \cdot p_i(u)^{l_j} \cdot (1 - p_i(u))^{r_j(u) - l_j} \cdot d_c.$$

We notice that the terms in $f(w_c)$ actually represent the convolution of functions of the form $g(w_c) = (1 - p_i(u))^{l_j} \cdot p_i(u)^{r_j(u) - l_j} \cdot d_c$. The factor d_c is simply a constant independent of w_c, as it represents the importance of class c. The functions $g(w_c)$ are log-concave as they represent a product of log-concave binomial coefficients and exponential terms and log-concavity is maintained under multiplication. Therefore, the function $f(w_c)$ is also log-concave, as convolution preserves log-concavity. Finally, the objective function in (4) is log-concave, as it is a cumulative distribution function [20]. ∎

The log-concavity property of the objective function permits to devise a low cost iterative optimization algorithm that is able to determine the optimum of the class rate allocation vector in a finite number of steps. We now propose a greedy algorithm that searches for the optimal distribution w^* independently at each node u.

The optimization algorithm starts from a pivotal packet distribution w over the priority classes that is then refined iteratively. The initial distribution depends on the number of classes that a node can optimally decode given the overall number of received packets. Specifically, when the bandwidth is adequate to decode the first c classes in an error-free case, the pivotal distribution respects the percentage fractions of the packets comprising the first c classes. No packets are requested from the other classes, i.e., the corresponding entries in w are zero. In every step of the algorithm, we examine the neighbors of the distribution vector w obtained from the previous iterations. In particular, a neighbor distribution is obtained by changing the rate allocation with the transfer of a unit rate from one class to an adjacent one, while the overall rate stays constant in order to fully utilize the available bandwidth. For example, if the video is encoded with packets in three classes arranged in order of importance, we can write the lth candidate distribution vector of the iterative search algorithm at step t as $w^{t,l} = (w_1^{t,l}, w_2^{t,l}, w_3^{t,l})$ (l is the index of the candidate distribution vector). It corresponds to a rate allocation $r^{t,l}(u) = (r_1^{t,l}(u), r_2^{t,l}(u), r_3^{t,l}(u))$. The following rate allocations are considered as neighbor vectors of $r^{t,l}(u)$, by the transfer of one rate unit between the neighbor classes:

$$\begin{cases} \left(r_1^{t,l}(u) + 1, r_2^{t,l}(u) - 1, r_3^{t,l}(u) \right), \\ \left(r_1^{t,l}(u) - 1, r_2^{t,l}(u) + 1, r_3^{t,l}(u) \right), \\ \left(r_1^{t,l}(u) + 1, r_2^{t,l}(u), r_3^{t,l}(u) - 1 \right), \\ \left(r_1^{t,l}(u) - 1, r_2^{t,l}(u), r_3^{t,l}(u) + 1 \right), \\ \left(r_1^{t,l}(u), r_2^{t,l}(u) + 1, r_3^{t,l}(u) - 1 \right), \text{ and} \\ \left(r_1^{t,l}(u), r_2^{t,l}(u) - 1, r_3^{t,l}(u) + 1 \right). \end{cases} \tag{8}$$

The algorithm checks the expected reduction in distortion for each of the neighbor rate allocations. If one of them results into a larger reduction in distortion than the starting rate allocation $r^{t,l}(u)$, then the neighbor allocation is included in the list of candidate solutions. This procedure is repeated for all new candidate solutions: the neighbor allocations are tested, as long as the unit rate transfer between the priority classes decreases the

overall distortion. When there are no further beneficial packet exchanges, the algorithm stops and the best candidate solution is retained. Note that the proposed algorithm implicitly uses the log-concavity property of the objective function, which guarantees the convergence to the optimum solution by a descent strategy, as described above. The algorithmic computation that this process embodies is summarized in Algorithm 1.

Algorithm 1: Optimal Rate Allocation Search

1: **Initialization**

- Set $t = 1$ and $l = 1$.
- Select a pivotal $w^{1,1}$ and include it in the list $\mathcal{L}^1$.
- Compute the expected reduction in distortion $D^{1,1}(u)$ corresponding to $\mathbf{w}^{1,1}$.
- Set $D_{max} = D^{1,1}(u)$.

2: **while** $\mathcal{L}^t \neq \emptyset$ **do**

3:    Pick $w^{t,l}$ in $\mathcal{L}^t$

4:    Compute $\{\mathcal{K}_i\}$, the neighbor distributions of $w^{t,l}$

5:    **for all** i **do**

6:      Compute D_i, the distortion reduction corresponding to $\mathcal{K}_i$

7:      **if** $D_i > D^{t,l}(u)$ **then**

8:        Insert $\mathcal{K}_i$ in $\mathcal{L}^{t+1}$

9:        Set $w^{t+1,l+1} = \mathcal{K}_i$

10:        Set $l = l + 1$

11:      **end if**

12:      **if** $D_i > D_{max}$ **then**

13:        Set $D_{max} = D_i$

14:        Set $w^* = \mathcal{K}_i$

15:      **end if**

16:      Remove $w^{t,l}$ from $\mathcal{L}^t$.

17:    **end for**

18: **end while**

19: $t = t + 1$

20: **if** $\mathcal{L}^t \neq \emptyset$ **then**

21:    Go to step 2

22: **end if**

23: Output w^*

Each client peer runs the above optimization problem periodically and requests the optimal distribution w^* from its parent nodes. The parents then implements network coding operations in order to match the requested distribution. Note that it might happen in practice that parents are not able to transmit the requested packets. In the case where w^* allocates nonzero weights to classes for which a parent node does not have any packets,

the parent node distributes these weight values uniformly to the classes it can transmit. This modification normally does not alter significantly the performance of the peer-to-peer system as the network can compensate for this by exploiting the existing network diversity and the fact that the nodes consider the maximum of their incoming and outgoing link capacities.

We illustrate the performance of the proposed search algorithm by comparing its solution to a full search strategy. We consider a simple scenario with three quality layers of 20 packets each and RNC in $GF(2^8)$ and one hop transmission. The loss probability between each pair of nodes is set to 5%. Fig. 4(a) and (b) illustrates, respectively, the rate allocation computed by full search and the one found by the algorithm proposed above. We can see that the proposed algorithm is always able to find the optimal coding strategy except for the rounding effects of the numerical computation of the binomial coefficients. We can also see that, whenever the bandwidth is sufficient to transmit two classes, the weights in the first two classes are identical. When the available bandwidth further increases, then the weight of the first class decreases. Though this may appear counter-intuitive, it is explained by the operation of the embedded network coding algorithm where coded packets in the second class contain information from the first two classes. Since we target a distributed solution, we consider only local statistics (singe hop). However, if the end-to-end loss rates and the min-cut values are available, we can use the same algorithm to determine the optimal rate allocation for multi-hop transmission.

V. UEP-NC PERFORMANCE EVALUATION

A. Simulation Setup

We consider the streaming of actual video content on overlay networks. We encode the Foreman sequence in CIF image size encoded with the scalable extension (SVC) of the latest video compression standard H.264 [13] into three quality layers, where the number of packets per layer and per GOP is, respectively, (38, 15, 20). The video quality achieved after the decoding of each of the three layers is, respectively, 36.48, 37.82, and 39.09 dB. The size of the GOP is set to 30 frames and the frame rate is 30 fps. We use packets of 1500 bytes that are augmented by the TCP/IP and the network coding headers. All network coding operations are performed in $GF(2^8)$, where it is expected that the size of the Galois Field does not have a large influence on the performance, as long as it is large enough [10].

Each evaluation point in our analysis is the average performance computed over 100 network topologies with similar statistical properties. These topologies represent irregular mesh networks that are generated by randomly modifying an initial regular network. Specifically, we start with a regular topology where the nodes are organized into stages depending on their hop-distance from the servers. In the original topology, each peer at every stage is connected to all the peers in the neighboring stages, and the nodes at the first coding stage are server nodes. We then build irregular topologies by randomly removing links from the regular topology; we denote this operation as *pruning* hereafter. We further change some links by rewiring them to different destination nodes (peers) in the network, in an operation that we denote as *shifting* [21]. The new destination nodes for these links are selected at random from the nodes at the same stage as the original destinations or
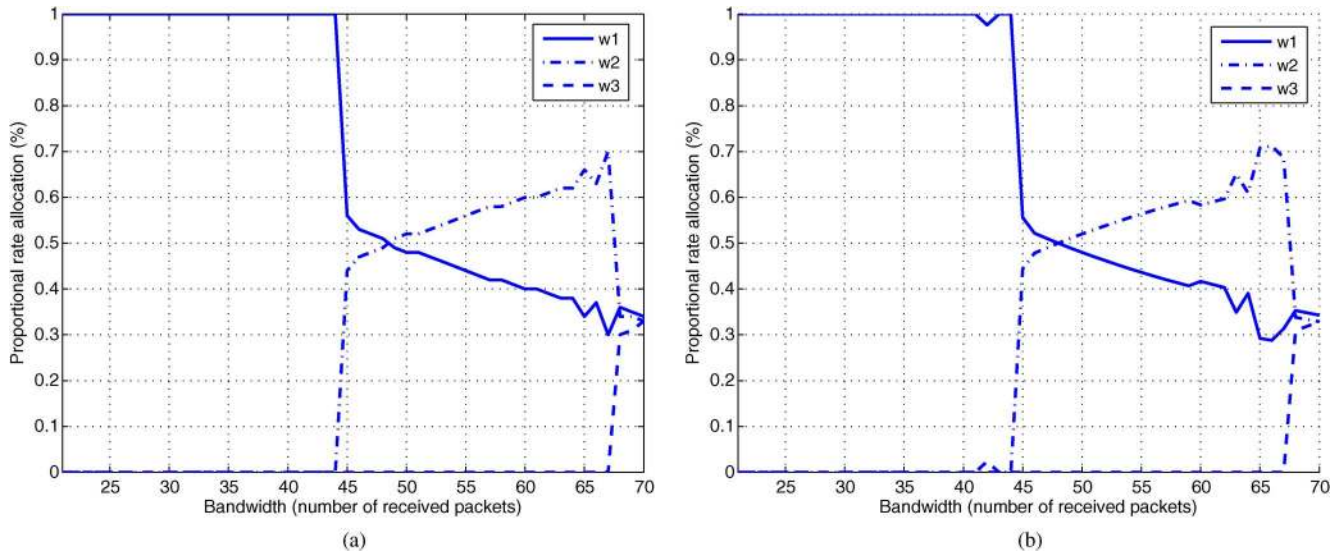
Fig. 4. Evolution of the optimal class distribution given by the coefficients w_i with respect to the available bandwidth for (a) exhaustive search and (b) the proposed search algorithm.

from nodes at subsequent stages, both in reference to the original regular topology. The pruning and shifting probabilities permit to control the "irregularity" of the resulting network. We validate the resulting topology by ensuring that it does not contain any cycles with the help of the DFS algorithm [22]. Furthermore, we make sure that every peer has at least two incoming and two outgoing links, since path diversity is critical in network coding. The capacities in each topology are finally controlled by varying the bandwidth and packet loss ratio on the network links. For the evaluation of video quality, we only consider the last stage nodes as system clients that consume the data.

We compare the performance of our UEP network coding system, denoted henceforth *UEP-NC*, against those of four competing schemes. The first scheme only considers packets from the first class. The peers perform network coding on these packets and use fully the available bandwidth. This first baseline scheme is called *Class-1*. Two other schemes, denoted *Class-2* and *Class-3*, employ only packets from the first two and three layers, respectively. The nodes perform network coding on the set of packet under consideration up to the network capacity. These three baseline schemes use the same number of source packets per class as the *UEP-NC* scheme.

The fourth competing scheme denoted as *SV-EWF* is a method based on [23] and [24] which employs expanding window fountain codes (EWF) [25] for scalable video multicasting. EWF codes consist of several LT codes [26] that are applied separately to expanding windows of data. Packet combinations are generated only with packets from the same window, which actually coincide with the importance classes of the source data. We select the robust soliton degree distribution (RSD) [26] in *SV-EWF* due to the relatively small reception overhead that it incurs. We further select the reception overhead in each window by estimating the expected error decoding probability in each window based on the local network statistics. The exact encoding procedure can be found in [23]. Finally, we apply EWF codes in a receiver-driven mode, similar to the *UEP-NC* method. The children peers compute the optimal rate allocation [23] and send requests to the parent peers,

which implement the corresponding coding strategy. In order to provide higher adaptivity to network dynamics and increase the information throughput relative to end-to-end solutions, each peer in the system decodes and then adaptively re-encodes the received information content, before forwarding it. We consider *SV-EWF* to be an excellent scheme for comparison since it shares many concepts with our UEP scheme.

B. Decoding Performance

We first analyze the performances of the competing schemes as a function of the link bandwidth in the network. We consider irregular network topologies, generated by randomly pruning and shifting links from a regular topology with seven coding stages and three peers per stage. The shifting and pruning probabilities are set to 5%, i.e., a link is either shifted or removed from the network, each, respectively, with a probability of 5%. Furthermore, the packet loss ratio on each link is set to 5%. We show in Fig. 5(a) the average performance of each scheme given as the average video quality as a function of the link bandwidth in the network, which varies from 160 to 360 kbps. The proposed scheme performs better than the other schemes over the whole range of bandwidth values. The *Class-1* solution exhibits a comparable performance. However, it tends to overprotect the data as it only considers packets from the first layer to guarantee the decoding of a minimum video quality. The performance gap increases as the link capacity increases since our *UEP-NC* scheme is able to take advantage of the increased bandwidth in order to transmit data from other layers. The other two baseline schemes perform poorly for low link capacities as they do not transmit enough packets for the successful decoding of layers two and three. These schemes become competitive only at high bandwidth values. However, even in this case, they still have an inferior performance as they suffer from the on-off performance characteristic, (i.e., they are either able to decode a full class or do not decode anything) while on the other hand, our *UEP-NC* scheme can provide more adaptivity. The performance of *SV-EWF* lies in between the performances of the *Class-1* and *Class-2* schemes. As the bandwidth increases *SV-EWF* performance converges to that of *Class-2* scheme, while it performs
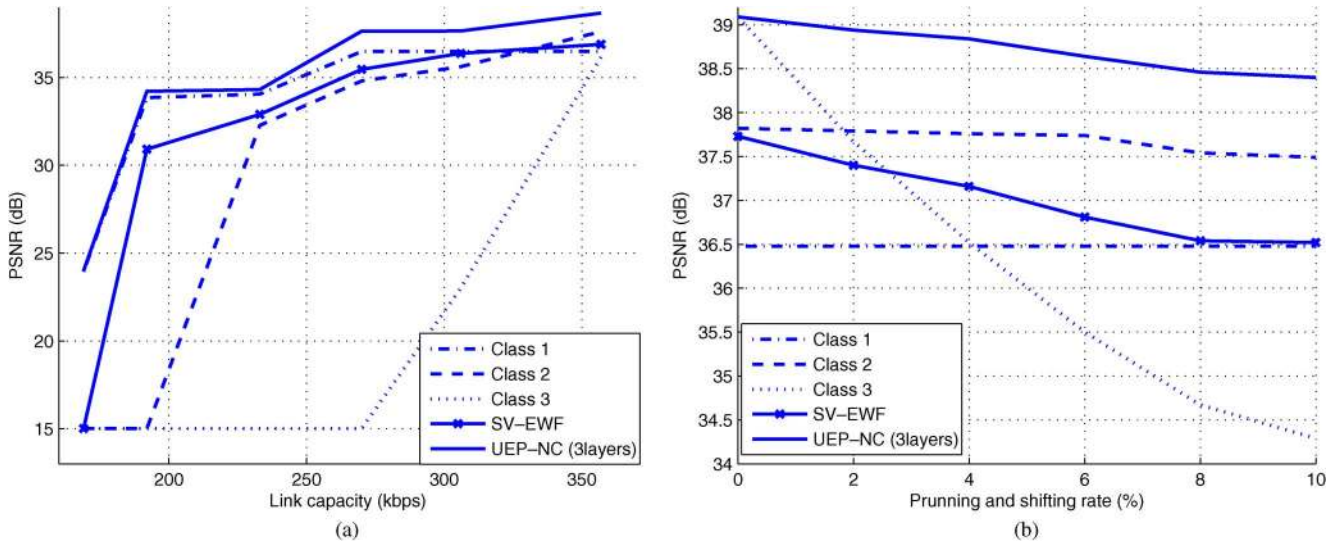
Fig. 5. PSNR comparison of the proposed UEP scheme with baseline NC solutions and the *SV-EWF* scheme: (a) influence of the available bandwidth and (b) influence of the network regularity (pruning and shifting probabilities).

considerably worse than the *UEP-NC* scheme. This can be attributed to the rather conservative allocation of symbols to the three expanding windows (classes) given the available network bandwidth, which in turn does not allow *SV-EWF* to produce symbols from the third class. The conservative allocation is due to the higher reception overhead of *SV-EWF* codes for short codeblocks (they perform close to MDS codes only for large codeblocks).

Next, we examine the influence of the irregularity of the network topology. In particular, we fix the packet loss rate to 5% and the link bandwidth to 360 kbps. We then vary the characteristics of the network by changing the pruning and shifting probabilities in the construction of the irregular topologies. Both of these probabilities are always selected to be identical. The corresponding performance results are shown in Fig. 5(b). It can be seen that when the network is quite regular (i.e., when the shifting and pruning probabilities are low), the *Class-3* scheme performs equally well with *UEP-NC* as both schemes are able to exploit efficiently the network resources. However, when the irregularity of the network increases, the performance of *UEP-NC* degrades gracefully while the *Class-3* scheme exhibits a significantly lower performance since its adaptivity is reduced by the predefined encoding strategy. The other two baseline schemes also seem to be robust to network variations, but their performance is limited by the smaller number of video layers that they consider. The performance of *SV-EWF* in topologies with low irregularity is comparable to the performance of the *Class-2* scheme. However, its performance rapidly approaches the one of the *Class-1* scheme when the network topology becomes more irregular. Furthermore, it can be seen in Fig. 5(b) that *UEP-NC* outperforms *SV-EWF* at all topology irregularity levels. This is because bandwidth variations force *SV-EWF* to generate symbols from the first class only as the available bandwidth is insufficient to transmit data from other classes. Furthermore, the second class cannot often be retrieved and the source symbols diversity in the network degrades quickly. This is the case even if next hop nodes have enough bandwidth for serving transmission of data from higher classes.

We further study the influence of the size of the network on the performance of the competing algorithms in order to under-

stand how successive bottlenecks affect the robustness of the algorithms. We consider irregular topologies with three nodes per stage, but we vary the number of stages. The link pruning and shifting probability is equal to 5% in the construction of the network topologies. The packet loss rate on each link is set to 5%. The bandwidth of all the links is set to 270 kbps and 338 kbps, in Fig. 6(a) and (b), respectively. Fig. 6(a) shows that the *UEP-NC* scheme is extremely robust and that its performance remains unaltered by the size of the network. The *SV-EWF* scheme is also robust since it generates mainly packets from the first window (the first video layer) when the link capacities are limited. The *Class-2* solution is, however, affected more significantly in small size networks due to the on-off performance of the decoding algorithm (i.e., either all packets are decoded, or none). For larger networks, the penalty is smaller since the network diversity assists this scheme to cope with the bandwidth variations. The bandwidth variations are due to the removal and shifting of some links. When the link bandwidth is larger, we see in Fig. 6(b) that the *UEP-NC* still shows a sustained performance for all network sizes. In the *Class-2* solution, the symbol diversity remains high as network size increases and the performance does not vary a lot. At the same time, the performance of *SV-EWF* degrades smoothly with the network size. This performance drop can be explained by the fact that few initial nodes are unable to retrieve the second video layer. Subsequently, the next nodes also lose the second layer as they cannot collect a sufficient number of packets to decode this layer.

C. Performance Under Timing Constraints

As we target streaming applications, we propose now to analyze the performance of the peer-to-peer streaming solutions under timing constraints. We build irregular topologies with seven stages and three nodes per stage, where the topology generation uses pruning and shifting probabilities of 5%. The packet loss rate is set to 5% and the link bandwidth varies in the range [160, 360] kbps. All the network links have the same bandwidth. We implement the different schemes in the NS-3 network simulator [27], which permits to study the timing
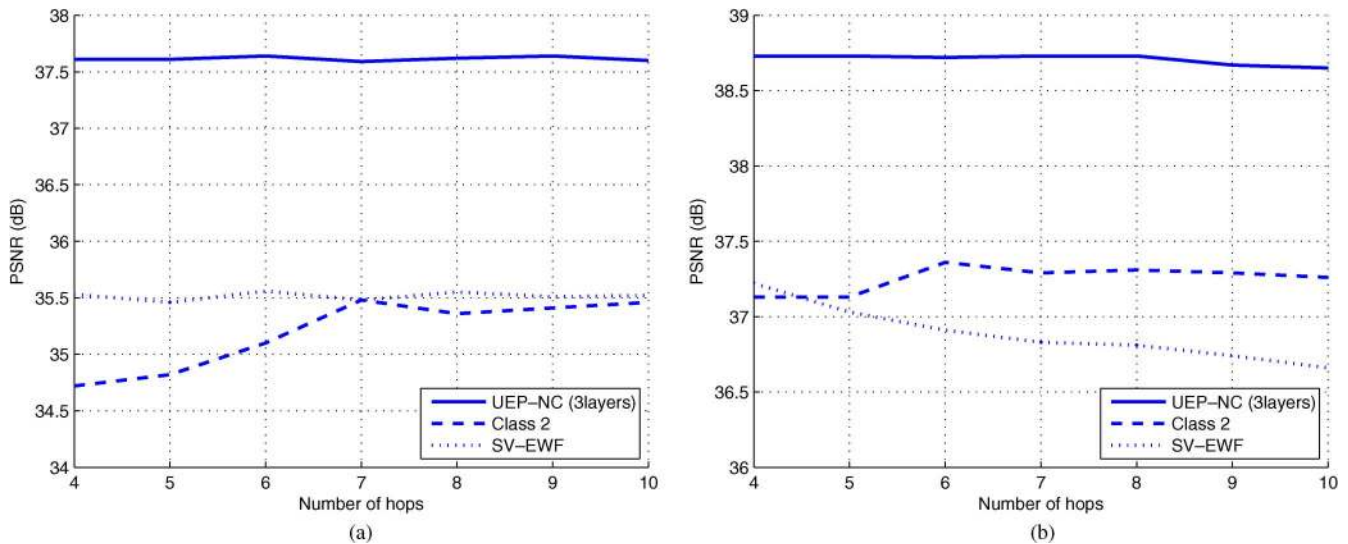
Fig. 6. Network scalability. *Class-2* scheme is compared with the *UEP-NC* scheme with three video layers and *SV-EWF* for network of various scales where all links have capacity: (a) 270 kbps and (b) 338 kbps.
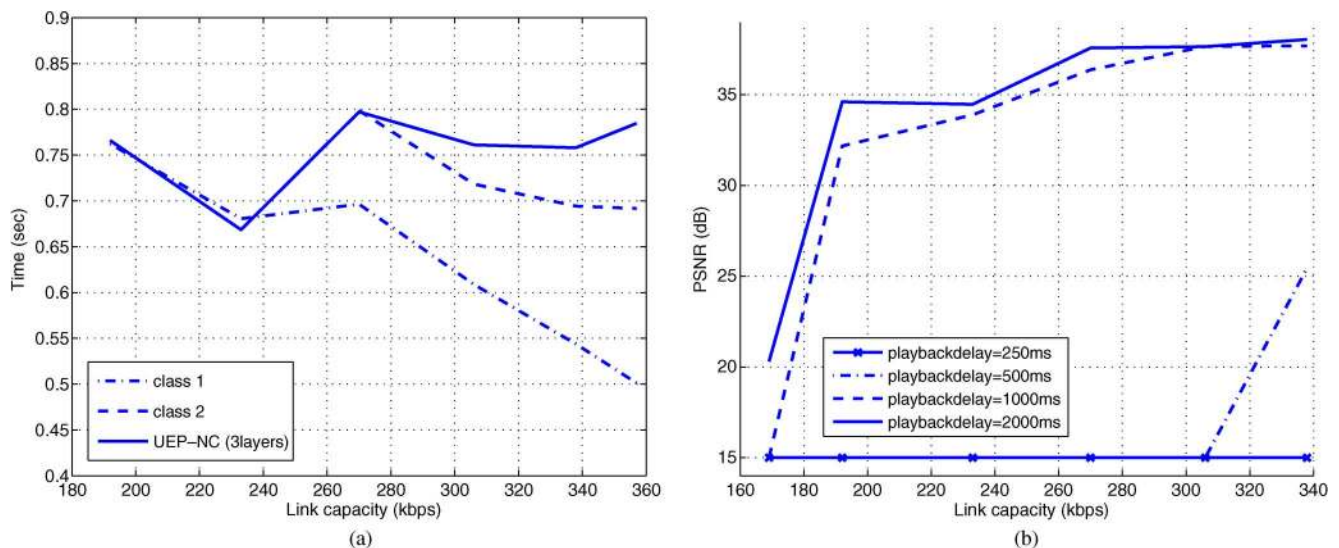

Fig. 7. (a) Decoding time comparison of the proposed *UEP-NC* scheme with the baseline network coding solutions. (b) PSNR comparison of the *UEP-NC* scheme for various link capacities and various playback delays.

information appropriately. As in the previous evaluations, all presented results represent averages over 100 simulations.

First, we analyze the average decoding time in Fig. 7(a). We observe that the decoding times for the baseline network coding schemes decrease with the increasing link bandwidth. When the links are faster, the peers obviously need less time to gather a sufficient number of packets for decoding. We only consider results where all the clients are able to decode all the video information in the baseline schemes. It can be further observed that the *UEP-NC* is very adaptive to the network capacities. It shows comparable decoding times to each of the baseline network coding solutions. Specifically, for low link capacities, the *UEP-NC* scheme transmits mostly packets from the first layer and its performance coincides with the *Class-1* scheme. For medium link capacities, the *UEP-NC* show higher decoding times, as some of the clients also decode the second video layer. It reaches performance similar to the *Class-2* scheme when all the clients are able to decode the first two layers. Then, the

decoding time decreases again as the necessary time to gather enough packets of the first two video layers decreases with increasing link capacities. Note that we do not provide a comparison with the *SV-EWF* scheme here, since the delay for this scheme rapidly becomes very large, as nodes have to successively decode and recode the video information. In this case, the delay grows with the size of the network and a comparison with the *UEP-NC* is not meaningful.

We provide another perspective on the performance under timing constraints in Fig. 7(b). We constrain the playback delay to values between 250 and 2000 ms, which are reasonable for real-time streaming applications. Packets that are late are discarded by a decoder. The deadline constraints are given by the packet time-stamps that depend on the generation and the playback delay value. When the timing constraints are very tight, the *UEP-NC* scheme does not perform well as the playback delay is too small to gather enough packets for decoding. When the playback delay increases, the *UEP-NC* algorithm is able to de-
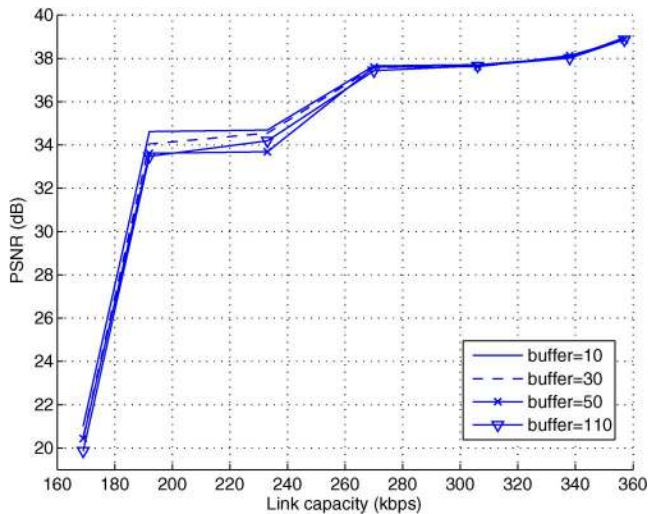
Fig. 8. PSNR evaluation of the *UEP-NC* scheme for various buffer sizes.

code the video only when the bandwidth is high enough, since the delivery is faster in this case. For higher playback delays, which are actually closer to the typical constraints imposed in practical systems, the *UEP-NC* scheme is able to provide high video quality for medium to high bandwidth networks.

Next, we analyze the influence of the buffer size at the peer nodes. We consider four different buffer sizes, i.e., 10, 30, 50, and 110 packets. Note that the last value basically corresponds to an infinite buffer in our setup, as it is much larger than a GOP (generation) size in packets. The performance of the proposed scheme with constrained buffers is illustrated in Fig. 8, where the playback delay constraint is set to 250 ms. We can observe that buffer size is not in fact a critical parameter in our network coding scheme. This can be explained by the high symbol diversity provided by the servers and the peer nodes. In addition, the *UEP-NC* scheme is further able to generate a high symbol diversity even if only a small number of packets is available in a intermediate node due to the limited buffer size. The proposed scheme is able to achieve a sustained performance even with small peer buffers.

VI. RELATED WORK

While initially network coding research has mainly focused on throughput benefits, nowadays, many researchers investigate the application of network coding to error resilient video communication. This research is driven by similarities between network coding and traditional channel coding techniques.

A testbed called "Lava" that applies network coding concepts to streaming systems was developed in [28]. Lava is based on a standard pull-based peer-to-peer streaming protocol. The randomized network coding is implemented as a plugin component. Prior to transmission, the streams are divided in segments of predefined duration. These are further divided into blocks. In Lava, the peers periodically exchange messages to announce the availability of certain segments. The peers possessing a certain segment are used as seeds for the other peers requiring this segment. Progressive decoding is achieved by a Gauss-Jordan elimination. The evaluation is encouraging and shows that the scheme is resilient to network dynamics, maintains stable buffering levels, and limits playback skips. Motivated by the success of [28], Wang *et al.* proposed a novel architecture known by the acronym

R^2 [29]. In this work, RNC is combined with a randomized push algorithm to take full advantage of coding operations at peer nodes. Whenever R^2 detects a transmission opportunity, it chooses at random a segment [30] that the downstream peer has not downloaded yet and it generates a corresponding network coded block. Frequent buffer map exchanges are required to identify segments that have not been downloaded. The buffer maps are sent together with previously requested segment. To avoid an explosion of overheads sent between the peers, larger size segments are employed.

Using network coding techniques in conjunction with Raptor codes [31] has shown in [21] and [32] that they resolve problems related to strict timing constraints and reduce significantly the computational cost. The optimal rate allocation is determined by the least reliable user which can be a limiting factor when we have to cope with heterogeneous networks and users. In such networks, network coding with built-in UEP can be beneficial as it can offer a variety of quality-of-service levels without employing expensive control protocols in the network for achieving the same goal.

To date, only a few works have addressed the prioritization of packets in network coding algorithms. Furthermore, the proposed solutions are usually computationally complex and difficult to extend to distributed settings. For example, priority random linear codes [33] are proposed for data dissemination in peer-to-peer and sensor networks, where improved data persistence is achieved due to the fact that the most important video data represents a combination of fewer source packets. The global encoding kernel (GEK) approach is proposed in [34] for defining unequal amount of protection to scalable data. GEK permits a decomposition of the network graph into connected line graphs where different coding operations are applied like in [35]. The optimization is defined as *minmax* and solved exhaustively. In [36], the problem is considered as an inter-session network coding problem [37]. In inter-session network coding, various sources access the network and mixing of packets from different sources is allowed when the clients receiving the mixed packets are interested in the content of all mixed sources.

Practical distributed network coding algorithms are largely inspired by the work of Chou *et al.* [10] that adapts RNC [9] to streaming applications. For UEP protection, the authors have employed a modified version of the well-known PET algorithm [38]. A complete overview of this system can be found in [7]. The recent work in [39] applies similar concepts, but replaces the PET algorithm in the UEP strategy by an MD-FEC scheme [17]. It computes the optimal source and channel rate allocation so that the average distortion is minimized for given channel conditions. In [40], another rate allocation algorithm is employed for scalable video streaming over multicast networks. All these works provide limited adaptivity to the system dynamics, as the coding strategy is generally predefined at the server based on global knowledge about the network. The work in [18] provides finer adaptivity in addressing the problem of streaming of H.264/AVC encoded video content where frame dependencies are taken into account for determining the optimal network coding operations for each video quality layer. This scheme bases its decisions on estimating the number of innovative packets received by each client. The optimization is performed at the intermediate network nodes. However, the related coding decisions are complex to compute due to the high number of dependencies between the video packets. The

coding decisions are much simpler when the peers implement embedded network coding strategies as proposed in our novel *UEP-NC* algorithm.

A scheme called *SV-EWF* has also been proposed for scalable video multicasting [23], [24]. It employs EWF codes [25] that consist of several LT codes [26] applied separately to expanding windows of data. This corresponds to some form of embedding coding as packet combinations are generated only with packets from the same window, which actually coincide with the importance classes of the source data. Even if this scheme shares a few concepts with our algorithm, it is less efficient in terms of adaptivity and end-to-end delay as shown in the previous section. Finally, other UEP rateless codes have been proposed recently [41], [42]. They, respectively, deal with the problem of minimizing the overhead of point-to-point communication [41] and with the scenario where different source packet sets are available at network nodes that employ in turn different degree distribution functions [42]. However, both schemes cannot be applied to content delivery in peer-to-peer networks as they suffer from the same disadvantages as SV-EWF in terms of end-to-end delay.

VII. CONCLUSIONS

We have proposed a novel receiver-driven RNC technique with built-in UEP properties. The technique considers the unequal importance of the various packet classes and implements different random network coding protection levels. The randomized coding strategy permits to keep a simple code design and avoids the use of expensive policies at the intermediate network nodes. The UEP properties are achieved simply by choosing the proper rate allocation among the different classes. Each client node periodically requests different shares of network coded packets from each importance class. The requests are computed independently at each node such that the expected distortion is minimized. We exploit the properties of the objective function to propose a simple iterative search algorithm that finds the optimal rate allocation at each node. We then demonstrate through simulation results that the proposed solution outperforms baseline network coding strategies for peer-to-peer delivery of scalable video content. By properly handling the different video classes and providing adaptivity to local network statistics, our framework achieves efficient distributed video delivery over heterogeneous and unreliable overlay networks.

REFERENCES

[1] Approaching the Zettabyte Era in Cisco Visual Networking Index, Cisco Inc., Jun. 2008.

[2] The State of the Internet, Akamai Inc., 2008, Tech. Rep., Q4. [Online]. Available: http://www.akamai.com/stateoftheinternet/.

[3] PPLive. [Online]. Available: http://www.pplive.com/.

[4] PPStream. [Online]. Available: http://www.ppstream.com/.

[5] X. Zhang, J. Liu, B. Li, and T.-S. Yum, "CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming," in *Proc. INFOCOM*, Miami, FL, Mar. 2005, vol. 3, pp. 2102–2111, IEEE.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[7] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 77–85, Sep. 2007.

[8] M. Wang and B. Li, "Network coding in live peer-to-peer streaming," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1554–1567, Dec. 2007.

[9] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE Int. Symp. Information Theory*, Kanagawa, Japan, Jul. 2003.

[10] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Allerton Conf. Communication Control and Computing*, Monticell, IL, Oct. 2003.

[11] N. Thomos, J. Chakareski, and P. Frossard, "Randomized network coding for UEP video delivery in overlay networks," in *Proc. Int. Conf. Multimedia and Expo 2009*, New York, Jun. 2009.

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

[13] *Advanced Video Coding for Generic Audiovisual Services, Amendment 3: Scalable Video Coding*, ITU-T and ISO/IEC JTC 1, Apr. 2005, *Draft ITU-T Recommendation H.264-ISO/IEC 14496-10(AVC)*.

[14] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Asilomar, CA, Oct. 1999, vol. 1, pp. 342–346, IEEE.

[15] P. A. Chou and K. Ramchandran, "Clustering source/channel rate allocations for receiver-driven multicast with error control under a limited number of streams," in *Proc. IEEE ICME*, New York, Jul. 2000, vol. 3, pp. 1221–1224.

[16] J. Chakareski and P. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," *IEEE Trans. Commun.*, vol. 52, no. 10, pp. 1675–1687, Oct. 2004.

[17] R. Puri, K. Ramchandran, and K. W. L. V. Bharghavan, "Forward error correction (FEC) codes based multiple description coding for Internet video streaming and multicast," *Signal Process.: Image Commun.*, vol. 16, no. 8, May 2001.

[18] X. Liu, G. Cheung, and C. N. Chuah, "Structured network coding and cooperative local peer-to-peer repair for MBMS video streaming," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, Cairns, Queensland, Australia, Oct. 2008.

[19] S. Deb, M. Médard, and C. Choute, "On random network coding based information dissemination," in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 278–282.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univ. Press, 2004.

[21] N. Thomos and P. Frossard, "Collaborative video streaming with Raptor network coding," in *Proc. Int. Conf. Multimedia and Expo*, Hannover, Germany, Jun. 2008, pp. 497–500.

[22] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, Jan. 1972.

[23] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. F. Stankovic, and Z. Xiong, "Scalable data multicast using expanding window fountain codes," in *Proc. 45th Annual Allerton 2007 Conf. Communication, Control and Computing*, Monticello, IL, Sep. 2007.

[24] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1521–1532, Apr. 2007.

[25] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. S. , and R. Piechocki, "Expanding window fountain codes for unequal error protection," in *Proc. 41th Annual Asilomar 2007 Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2007.

[26] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS '02)*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.

[27] The Network Simulator—ns3. [Online]. Available: http://www.nsnam.org.

[28] M. Wang and B. Li, "Lava: A reality check of network coding in peer-to-peer live streaming," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[29] M. Wang and B. Li, " R^2: Random rush with random network coding in live peer-to-peer streaming," *IEEE J. Select. Areas Commun.*, vol. 25, no. 9, pp. 1655–1666, Dec. 2007.

[30] P. Maymoukov, N. J. A. Harvey, and D. Lun, "Methods for efficient network coding," in *Proc 44th Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Sep. 2006.

[31] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[32] N. Thomos and P. Frossard, "Raptor network video coding," in *Proc. 1st ACM Int. Workshop Mobile Video (in Conjunction With ACM Multimedia 2007)*, Augsburg, Germany, Sep. 2007.

[33] Y. Lin, B. Li, and B. Liang, "Differentiated data persistence with priority random linear codes," in *Proc. 27th Int. Conf. Distributed Computing Systems of Contents*, Toronto, ON, Canada, Jun. 2007, pp. 47–55.

[34] A. Limmanee and W. Henkel, "UEP network coding for scalable data," in *Proc. 5th Int. Symp. Turbo Codes and Related Topics, Turbo-Coding*, Lausanne, Switzerland, Sep. 2008, pp. 333–337.

[35] C. Fragouli, E. Soljanin, and A. Shokrollahi, "Network coding as a coloring problem," in *Proc. Conf. Information Science and Systems (CISS07)*, Princeton, NJ, Mar. 2004.

[36] Y. Wu, "Distributing layered content using network coding," in *Proc. 5th IEEE Annu. Commun. Soc. Conf. Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, San Francisco, CA, Jun. 2008, pp. 1–4.

[37] Y. Wu, "On constructive multi-source network coding," in *Proc. IEEE Int. Symp. Information Theory*, Seattle, WA, Jul. 2008, pp. 1349–1353.

[38] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.

[39] A. K. Ramasubramanian and J. W. Woods, "Video multicast using network coding," in *Proc. SPIE VCIP*, San Jose, CA, Jan. 2009.

[40] H. Wang, S. Xiao, and C. C. J. Kuo, "Robust and flexible wireless video multicast with network coding," in *Proc. IEEE Global Telecommunications Conf.*, Adelaide, Australia, Nov. 2007, pp. 2129–2133.

[41] A. Talari and N. Rahnavard, "Distributed rateless codes with UEP property," in *Proc. IEEE Int. Symp. Information Theory*, Austin, TX, Jun. 2010, pp. 2453–2457.

[42] Y. Cao, S. Blostein, and W.-Y. Chan, "Unequal error protection rateless coding design for multimedia multicasting," in *Proc. IEEE Int. Symp. Information Theory*, Austin, TX, Jun. 2010, pp. 2438–2442.

Jacob Chakareski completed his doctoral thesis in the Information Systems Laboratory at Stanford University, Stanford, CA.

He is an Associate (Senior) Scientist at EPFL, where he conducts research, lectures, and supervises students. His general interests lie at the intersection of computer networks, multimedia systems, optimization and control, and wireless communications. His present investigations involve, but are not limited to, online social networks, multi-camera arrays, and cognitive quality of experience in multi-hop wireless systems. Chakareski also actively explores ultrasound applications in telemedicine, remote sensing, and biomedicine. He has held research positions with Microsoft and Hewlett-Packard. He has authored one monograph, three book chapters, and over 100 international publications, and has nine pending or approved patent applications.

Dr. Chakareski is a recipient of a Graduate Research Fellowship from Texas Instruments and the Ambizione Career Development Award (2009-2012) from the Swiss NSF. He actively participates in technical and organizing committees of several IEEE conferences and symposia on a yearly basis. He was a publicity chair for the Packet Video Workshop in 2007 and 2009 and for the Workshop on Emerging Technologies in Multimedia Communications and Networking at ICME 2009. He organized and chaired a special session on telemedicine at MMSP 2009. He won the best student paper award at the IS&T/SPIE VCIP 2004 conference.

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

Between 2001 and 2003, he was a member of the research staff at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on media coding and streaming technologies. Since 2003, he has been a Professor at EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include image representation and coding, visual information analysis, distributed image processing and communications, and media streaming systems.

Dr. Frossard has been the General Chair of IEEE ICME 2002 and Packet Video 2007. He has been the Technical Program Chair of EUSIPCO 2008, and a member of the organizing or technical program committees of numerous conferences. He has been an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2004-), the IEEE TRANSACTIONS ON IMAGE PROCESSING (2010–), and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2006–). He is an elected member of the IEEE Image and Multidimensional Signal Processing Technical Committee (2007–), the IEEE Visual Signal Processing and Communications Technical Committee (2006–), and the IEEE Multimedia Systems and Applications Technical Committee (2005–). He has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee (2004–2006) and as a member of the IEEE Multimedia Signal Processing Technical Committee (2004–2007). He received the Swiss NSF Professorship Award in 2003, the IBM Faculty Award in 2005, and the IBM Exploratory Stream Analytics Innovation Award in 2008.

Nikolaos Thomos (S'02-M'06) received the Diploma and the Ph.D. degrees from the Electrical and Computer Engineering Department of the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2000 and 2005, respectively.

Currently, he is postdoctoral research fellow with the Signal Processing Laboratory of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Previously, he was a postdoctoral and graduate research fellow with the Informatics and Telematics Institute/Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include network coding, multimedia communications, joint source and channel coding, and distributed source coding.

Dr. Thomos has been awarded in 2008 the highly esteemed Ambizione career award from the Swiss National Science Foundation targeted to prospective researchers. He is a member of the Technical Chamber of Greece.