

RANDOMIZED NETWORK CODING FOR UEP VIDEO DELIVERY IN OVERLAY NETWORKS

Nikolaos Thomos, Jacob Chakareski, and Pascal Frossard

Signal Processing Laboratory (LTS4)

Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

{nikolaos.thomos,jacob.chakareski,pascal.frossard}@epfl.ch

ABSTRACT

This paper presents a receiver-driven video delivery algorithm that exploits a novel Randomized Network Coding (RNC) scheme for unequal error protection (UEP). The main idea of our approach is to account for the unequal importance of media packets in the network coding algorithm for efficient stream delivery in lossy overlay networks. Based on the requests from their neighbours, the network nodes properly combine packets and forward them to their children nodes. The network coding operations at every node are formulated as a log-concave optimization problem, which is solved with a greedy algorithm in only a few iterations. Our experimental results demonstrate that the proposed scheme permits to respect the priorities between the different packet classes. It further outperforms baseline network coding techniques for video streaming in overlay networks.

Index Terms— Network coding, rate allocation, unequal error protection, overlay networks

1. INTRODUCTION

Network coding for multimedia communication in overlay networks has recently attracted a lot of attention due to its resilience to random node departures and non ergodic link failures. Furthermore, this technique also effectively deals with bandwidth variations and packets' duplications that typically arise in such networking environments. While computational complexity still represents an important issue, a few algorithms have been proposed recently that successfully apply network coding principles in multimedia streaming applications [1, 2]. An overview of first attempts to make practical network coding is presented in [3].

We propose in this paper a network coding algorithms that consider the specific importance of media packets in order to prioritize the delivery of the most important packets and improve the quality of service. We consider that the nodes in an overlay networks perform Randomized Network Coding (RNC) on data packets in order to improve the robustness to failures or erasures in the delivery process without the need for any centralized control function. In particular, when packets are requested by its neighbours, a network node combines some of its own packets and sent to the children nodes. As media packets can typically be grouped into classes of different importance, Unequal Error Protection (UEP) is typically achieved at each node by varying the number of packets from each class that are used in the network coding algorithm. We formulate an optimization problem that describe the optimal network coding choices

in each node. As the problem is shown to be log-concave, we propose a low-complexity algorithm that determine locally 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 the packets. It typically outperforms baseline network coding algorithms for the delivery of layered media streams.

A few works have recently addressed the prioritization of packets in network coding algorithms, but the available solutions are usually computationally complex and difficult to extend to distributed settings. For example, priority random linear codes [4] 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 [5] for defining unequal amount of protection to scalable data. GEK allows a decomposition of the network graph into connected line graphs where different coding operations are applied. The optimization is defined as *minmax* and solved exhaustively. RNC in conjunction with the coding scheme from [6] is presented in [7] for scalable video streaming over multicast networks. The authors propose to organize the layered video data into large data arrays, and the unequal error protection is defined at the servers, based on the network conditions. All these works provide limited adaptivity to the systems dynamics, as the coding strategy is generally pre-defined at the server, based on global knowledge about the network. The work in [8] addresses the problem of MBMS 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 and the optimization is performed in intermediate network nodes. However, the related coding decisions are complex to compute due to the high number of dependencies between the video packets.

2. RNC WITH UEP PROPERTIES

We consider overlay network topologies where the nodes perform random coding operations with the received packets before forwarding them. In particular, each packet forwarded by a node is built by a randomly weighted combination of the packets available at the node. We further consider that packets are organized into classes, depending on their importance. In particular, we define as class i the set of packets that are linear random combinations of native (unencoded) packets from the first i classes. In practice, the class of a packet is identified by a small header which is appended to the packet.

The network coding operations can be written as follows. If a

This work has been supported by the Swiss National Science Foundation, under grant PZ00P2-121906.

node u generates M packets by RNC, then the m^{th} network coded packet c_m is of the form

$$c_m = \sum_{p_i(u) \in l_{in}(u)} f_{i,m} \cdot p_i(u) = \sum_{i=1}^N f'_{i,m} \cdot n_i(u)$$

where $l_{in}(u)$ corresponds to the set of packets available at node u , $p_i(u)$ denotes either a network coded packet or a native (uncoded) packet in class i , and $f_{i,m}$ is a random coefficient over the galois field $\text{GF}(q)$. Furthermore, N is the total number of native packets that are originally used to generate network coded packets at the video source. The basis of the galois field is typically set to $q = 256$, as it has been shown in [6] that this guarantees high symbol diversity and low probability of building duplicate packets. Finally, $n_i(u)$ and $f'_{i,m}$ represent respectively the native packets and their corresponding coding coefficients. A packet is therefore augmented with another $N \cdot \log_2(q)$ -bit header containing the vector of coding coefficients $f'_j = [f'_{1,j}, \dots, f'_{N,j}]$, which does not grow with the number of hop transmissions. To control latency, packet coding operations are restricted to (native or combinations of native) packets from the same generation only, as in [6]. In our system, a generation corresponds to a Group of Pictures (GOP) in the video encoding. Hence, N denotes the number of video packets in a GOP.

A node u employing our scheme requests packets of the different classes from its parent nodes. It may ask for a different number of packets from each class. Let w_i denote the fraction of requested packets in class i with respect to the total number of requested packets, $N_{in}(u)$. Then, the expected number of packets in class i received by u is given as

$$r_u(i) = \lfloor N_{in}(u) \cdot (1 - \pi_{in}(u)) \cdot w_i \rfloor \quad (1)$$

where $\pi_{in}(u)$ is the average packet erasure rate on the incoming links of node u . It holds that $\sum_{i=1}^L w_i = 1$ and $w_i \geq 0$, where L is the number of packet classes in the network. The number of requested packets $N_{in}(u)$ is determined based on the overall bandwidth of the incoming links at u . Furthermore, the node u is interested in selecting the vector of coefficients $\mathbf{w} = [w_1, \dots, w_L]$ such that the network coding strategy at the parent node maximizes the overall quality. It should be mentioned here that a sending node receives coefficient vectors \mathbf{w} from each of its children. This is in contrast to the related work [8] where a parent sends the same distribution of network coded packets to all of its children, which permits to deal with different network conditions for each node.

We assume now that the packets are divided into L quality layers with the first layer containing the most important information, while the L^{th} layer is the less important. The distortion reduction experienced by the video presentation after the successful decoding of the i^{th} layer is denoted as d_i and the number of packets in the first i layers as $\beta_i = \sum_{j=1}^i \alpha_j$, where α_j stands for the number of video packets in layer j . The total number of native packets is $\beta_L = N$. A client recovers the i^{th} layer as soon as it receives β_i innovative network coded packets that represent linear combinations of source (native) packets from the first i video layers. A packet is called innovative when it increases the rank of the equation system constructed by the received set of packets at a node.

The expected reduction in video distortion that the node u experiences is a function of the number of video layers it can decode. Formally, this can be written as

$$D(u) = \sum_{i=0}^L d_i \cdot P_{NC}(i) \quad (2)$$

where $P_{NC}(i)$ denotes the probability that the node u recovers i video layers after network decoding. Furthermore, let $\epsilon(u)$ be the probability that an innovative packet arrives at node u . It can be computed as $\epsilon(u) = (1 - \pi_{in}(u)) \cdot P_{in}$, where the probability P_{in} that a network coded packet is innovative is lower bounded by $P_{in} \geq (1 - 1/q)$ [9]. Therefore, the Eq. (2) can be written as

$$D(u) = \sum_{i=0}^L \sum_{l_1=0}^{r_u(1)} \dots \sum_{l_i=0}^{r_u(i)} \beta_{i+1} - \beta_i - \sum_{j=i+1}^i l_j \quad \beta_L - \beta_i - \sum_{j=i+1}^L l_j \quad (3)$$

$$\sum_{l_{i+1}=0} \dots \sum_{l_L=0}$$

$$\prod_{j=1}^L \binom{r_u(j)}{l_j} \cdot (1 - \epsilon(u))^{l_j} \cdot \epsilon(u)^{r_u(j) - l_j} \cdot d_i.$$

Note that the sequence of summation terms in the first line of the right side of Eq. (3) counts the events of receiving enough packets to decode up to class i packets, while not being able to decode packets of subsequent classes ($i + 1, \dots, L$). This latter condition is described with the second sequence of sums in the second line of Eq. (3). Finally, the product term in the third line of Eq. (3) denotes simply the likelihood (probability) of one such joint event described by the two conditions above.

We can now formulate the optimization problem solved by node u , which tries to determine the number of packets it should request from each class, so that the total video distortion is minimized or alternatively the cumulative distortion reduction, as contributed by the requested packets, is maximized. Formally, the problem can be written as follows:

Rate Allocation Problem:

$$\max_{\mathbf{w}} D(u), \text{ s.t. } \sum_{i=1}^L w_i = 1 \text{ and } w_i \geq 0 \text{ for } i = 1, \dots, L \quad (4)$$

The above problem formulation can be modified to take into account that intermediate nodes are not actually interested for the transmitted content, rather simply participate to the transmission process. Instead of the overall incoming bandwidth at a node, we consider the maximum of the incoming and outgoing link capacities. These are more important factors in maintaining a high packet diversity in the network. In particular, the factor $N_{in}(u) \cdot (1 - \pi_{in})$ in Eq. (1) is replaced with $\Phi(u)$ defined as

$$\Phi(u) = \max\{N_{in}(u) \cdot (1 - \pi_{in}(u)), N_{out}(u) \cdot (1 - \pi_{out}(u))\}$$

where $N_{out}(u) \cdot (1 - \pi_{out}(u))$ and $\pi_{out}(u)$ are respectively the overall number of packets forwarded by the node u and the average packet erasure rate on the corresponding outgoing links. Thus, Eq. (1) becomes

$$r_u(i) = \lfloor \Phi(u) \cdot w_i \rfloor \quad (5)$$

3. OPTIMIZATION OF CLASS DISTRIBUTION VECTOR

In the previous section, we have defined the optimization problem that every network client has to solve independently, based on local information. Since the search space is huge, exhaustive algorithms can not be used even for a small number of packets. Fortunately, we can show that the objective function given in Eq. (3) is a log-concave

function. According to [10] we can show that Eq. (3) is log-concave by proving that all terms

$$f(w_i) = \prod_{j=1}^L \binom{r_u(j)}{l_j} \cdot (1 - \epsilon(u))^{l_j} \cdot \epsilon(u)^{r_u(j)-l_j} \cdot d_i$$

are log-concave. The terms $f(w_i)$ represent the convolution of functions of the form

$$g(w_i) = (1 - \epsilon(u))^{l_j} \cdot \epsilon(u)^{r_u(j)-l_j} \cdot d_i$$

These functions are log-concave as they represent a product of log-concave binomial coefficients and exponential terms and we know that log-concavity is maintained under multiplication. Note that the factor d_i in $g(w_i)$ is simply a constant multiplier that is not a function of w_i . Overall, the objective function in Eq. (3) is therefore log-concave.

We propose now a greedy algorithm which seeks for the optimal distribution vector w . The algorithm starts from a pivotal packet distribution among classes and refines the distribution iteratively. In every step of the algorithm, we examine the neighbors of the distribution vector w that is reached at previous iterations. A neighbor distribution is obtained by changing the rate allocation when a packet is removed from one class, to be added to the neighbor class. If one of the neighbor distribution decreases the distortion, we include it into the list of possible solution candidates. This procedure is repeated as long as packet exchanges could decrease the distortion. When there is no further beneficial packet exchanges, the algorithm stops. This process is summarized with the following pseudocode.

Pseudocode

- 1: **Initialization:** For $t = 1$, include the pivotal distribution vector $w^t(l)$ in the list \mathcal{L}^t and set $l = 1$. Calculate the corresponding expected distortion reduction $D(u, l)^t$ and set $maxD = D(u, l)^t$
- 2: **while** $\mathcal{L}^t \neq \emptyset$ **do**
- 3: Examine the four neighboring distributions of $w^t(l)$ whether their distortion reduction is greater than $D(u, l)^t$. Whenever this is true we insert the corresponding neighbour vector into the list \mathcal{L}^{t+1} and we set $l = l + 1$. Let the corresponding distortion reduction be denoted $D(u, l)^{t+1}$. Then, if it further holds $D(u, l)^{t+1} > maxD$ then we store the corresponding distribution vector as the present optimal solution.
- 4: **end while**
- 5: $t = t + 1$
- 6: If $\mathcal{L}^t \neq \emptyset$ go to step 2

4. SIMULATION RESULTS

We first compare the solution determined by the proposed algorithm with the optimal solution found by an exhaustive search. The randomized network coding scheme works on $GF(2^8)$. We considered three quality layers with $\alpha_i = 20$, while $\epsilon(u)$ was set to 5%. Figures 1(a) and (b) illustrate respectively the optimal solution and the one found by the proposed algorithm. The small differences that can be observed between the graphs shown in the two figures are due to rounding problems encountered when the binomial coefficients are computed. We can see that the proposed algorithm is able to find the optimal coding strategy, but for the rounding effects.

Then, we investigate the performance of the proposed scheme for the transmission of the Foreman CIF sequence encoded by the JM8.5 of the H.264/SVC standard in overlay networks. The video sequence

is encoded into three quality layers. The corresponding number of packets per layer is $a_1 = 38$, $a_2 = 15$, and $a_3 = 20$. The GOP size is set to 30 frames and the frame rate is 30 *fps*. We use packets of 1500 bytes which are augmented by the TCP/IP and the network coding headers. All operations are performed in $GF(2^8)$. We evaluate our scheme in seven-stage irregular overlay mesh topologies that have been generated from regular topologies with three nodes per stage by random pruning and shifting of some of the links [2]. Pruning simply consists in removing a link from the regular topology. Shifting consists in randomly changing the destination of some links, while making sure that cycles are avoided. The proposed UEP scheme is compared with three baseline algorithms that do not exploit the layered structure of the video. The first scheme considers only video packets from the first layer and is denoted as class-1. The other two schemes assume only packets from the first two and three layers, respectively, and are denoted as class-2 and class-3.

We first set the pruning and shifting probabilities P_{pr} and P_{sf} to 5%. The link capacities are varied from 160 *kbps* to 360 *kbps* and the packet loss rate is 5%. We examine in Fig. 2(a) the average performance of the different schemes over 100 simulations. We observe that the proposed scheme performs better than the other schemes. Class-1 scheme has comparable performance; however it overprotects the data as it considers only packets from first layer to guarantee the decoding of a minimum video quality. This performance gap increases as the link capacity increases since our UEP scheme is able to take advantage of the increased bandwidth in order to transmit data from other layers. The other two schemes perform poorly for low link capacities as they do not transmit enough packets for the successful decoding of layers two and three. Only for high link capacities these schemes become competitive with the proposed one. Even in these conditions they have still inferior performance as they suffer from on-off performance characteristics (*i.e.* they are either able to decode a layer or do not decode anything) while our UEP scheme can provide more adaptivity.

Finally, we consider a scenario with irregular topologies and homogeneous links capacities that are equal to 360 *kbps*. The loss rate is 5% on each link. The pruning and shifting probabilities P_{pr} and P_{sf} are set to be equal, but vary in the range $[0, \dots, 10]$ %. The results of the evaluation are illustrated in the Fig. 2(b). It can be seen that, when P_{pr} and P_{sf} are low then the class-3 scheme performs equally well with the proposed algorithms, as both are able to exploit the large network resources. However, when P_{pr} and P_{sf} increase, UEP scheme degrades smoothly while class-3 scheme has significantly lower performance. The other two schemes seem also to be robust to network variations, but are limited by a smaller number of video layers.

5. CONCLUSIONS

In this paper we have proposed a novel RNC technique with built-in unequal error protection properties. The technique combines the unequal importance of the various media classes with different random network coding protection levels. RNC permits to keep a simple code design and avoid the use of expensive policies at the intermediate nodes in the network. The UEP properties are achieved simply by choosing the proper rate allocation among the different classes. We design a greedy optimization algorithm that is able to determine the optimal coding operation as the expression that we optimize is log-concave. This algorithm is applied distributively at every node in the network. Layered video delivery demonstrate significant performance gains over traditional network coding methods that do not consider the layered data structure of the video presentation.

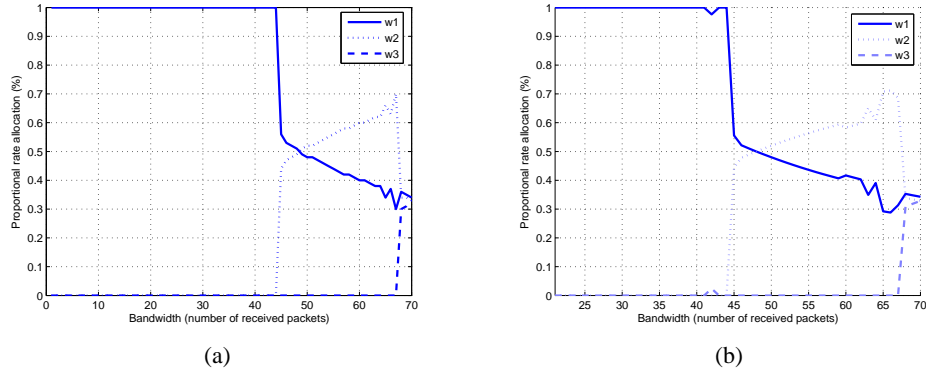


Fig. 1. Evolution of optimal class distribution given by the coefficients w_i , with respect to the available bandwidth: (a) exhaustive search and (b) the proposed optimization algorithm.

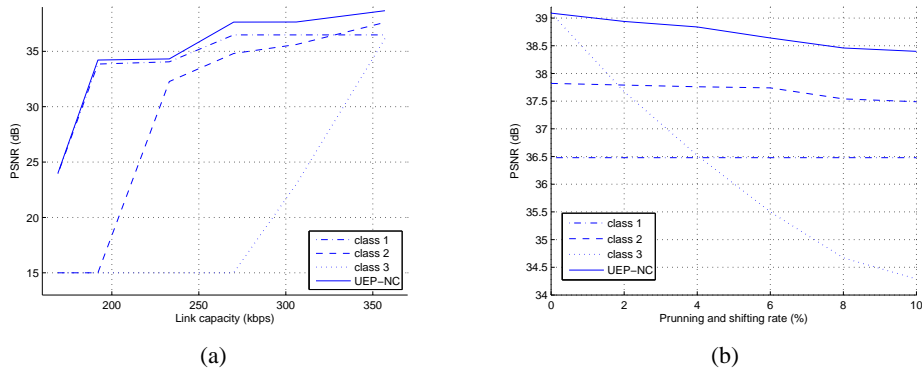


Fig. 2. PSNR comparison of the proposed UEP scheme with baseline NC solutions: (a) influence of the available bandwidth and (b) influence of the network regularity (pruning and shifting probabilities).

6. REFERENCES

- [1] 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.
- [2] N. Thomos and P. Frossard, "Collaborative video streaming with Raptor network coding," in *Proc. Int. Conf. on Multimedia and Expo*, Hannover, Germany, June 2008, pp. 497–500.
- [3] P. A. Chou and Y. Wu, "Network Coding for the Internet and Wireless Networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, Sept. 2007.
- [4] Y. Lin, B. Li, and B. Liang, "Differentiated Data Persistence with Priority Random Linear Codes," in *Proc. 27th Int. Conf. on Distributed Computing Systems of Contents*, Toronto, ON, Canada, June 2007, pp. 47–55.
- [5] A. Limmanee and W. Henkel, "UEP Network Coding for Scalable Data," in *Proc. 5th Int. Symp. on Turbo Codes and Related Topics, Turbo-coding*, Lausanne, Switzerland, Sep. 2008, pp. 333–337.
- [6] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. 41st Allerton Conf. on Communication Control and Computing*, Monticell, IL, USA, Oct. 2003.
- [7] H. Wang, S. Xiao, and C. C. J. Kuo, "Robust and Flexible Wireless Video Multicast with Network Coding," in *Proc. IEEE Global Telecommunications Conference*, Adelaide, Australia, Nov. 2007, pp. 2129–2133.
- [8] 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 on Multimedia Signal Processing*, Cairns, Queensland, Australia, Oct. 2008.
- [9] S. Deb, M. Médard, and C. Choute, "On Random Network Coding Based Information Dissemination," in *Proc. IEEE Int. Symp. on Information Theory*, Adelaide, Australia, Sep. 2005, pp. 278–282.
- [10] S. Boyd and L. Vandenberghe, *Convex Optimization*, New York: Cambridge University Press, 2004.