MULTI-VIEW VIDEO STREAMING OVER WIRELESS NETWORKS WITH RD-OPTIMIZED SCHEDULING OF NETWORK CODED PACKETS

Irina Delia Nemoianu, Claudio Greco, Marco Cagnazzo, Béatrice Pesquet-Popescu

TELECOM-ParisTech, TSI department, 46 rue Barrault, 75634 Paris Cedex 13, FRANCE

{nemoianu,greco,cagnazzo,pesquet}@telecom-paristech.fr

ABSTRACT

Multi-view video streaming is an emerging video paradigm that enables new interactive services, such as free viewpoint television and immersive teleconferencing. However, it comes with a high bandwidth cost, as the equivalent of many single-view streams has to be transmitted. Network coding (NC) can improve the performance of the network by allowing nodes to combine received packets before retransmission. Several works have shown NC to be beneficial in wireless networks, but the delay introduced by buffering before decoding raises a problem in real-time streaming applications. Here, we propose to use Expanding Window NC (EWNC) for multi-view streaming to allow immediate decoding of the received packets. The order in which the packets are included in the coding window is chosen via RD-optimization for the current sending opportunity. Results show that our approach consistently outperforms both classical NC applied on each view independently and transmission without NC.

Index Terms— Multi-view video coding, network coding, wireless networks.

1. INTRODUCTION

Multi-View (MV) video has drawn increasing attention due to the development of acquisition and rendering systems, and the public interest in applications such as 3DTV and Free Viewpoint TV (FTV) [1]. Multi-view video coding is a complex subject, which has been widely investigated in literature [2–4].

The idea of exploiting the statistical dependencies from both temporal and inter-view reference pictures for motion-compensated prediction was incorporated in the *Multi-view Video Coding* (MVC) extension of the popular H.264/MPEG-4 AVC standard [5].

Even though a call for proposals has recently been issued to provide an efficient compression of an even higher number of views with their depth (3D Video Coding, or 3DVC) [6], the traffic of multi-view video still remains several times larger than traditional video, exacerbating the existing problems of current networks, which can be unreliable or have to meet many client demands.

Network Coding (NC) [7] is an alternative to classical routing for network communications. In a network with coding capabilities, a multi-hop communication is relayed at intermediate nodes by sending combinations of the received messages, rather than mere copies.

It has been shown [8] that using linear combinations is sufficient to achieve the maximum network throughput for multicast communications, as long as an alphabet with a large enough size is used. Furthermore, under loose hypotheses, the combination coefficients, referred to as *coding vectors*, can be assigned randomly in a distributed fashion [9].

In their influential work on *Practical Network Coding* (PNC) [10], Chou *et al.* provided the first approach to NC that also takes into account the transmission of the coding vectors, with coefficients taken from a finite field of proper size. The data stream is segmented into groups of G data packets called *generations* and only packets belonging to the same generation are combined. The source generates G new data packets by prepending the G-dimensional unit vector to the corresponding data packet. In this way, when intermediate nodes generate a new packet, the G-dimensional overhead will correspond to the global encoding vector. All data packets in a generation are jointly decoded by performing Gaussian elimination as soon as enough linearly independent combination vectors have been received.

More recently [11], it has been proposed to apply PNC to video content delivery. The authors suggest to divide the video stream into layers of priority and to provide unequal error protection for the different layers via PNC. Such a hierarchical method needs to ensure that all end-users receive at least the base layer, therefore all received packets must be stored in a buffer until a sufficient number of independent coding vectors are received. This introduces in the decoding process a delay that is often unacceptable in real-time streaming applications.

In order to reduce the decoding delay, a viable solution is to use *Expanding Window Network Coding* (EWNC) [12] which means to increase the size of the *coding window* (*i.e.*, the set of packets in the generation that may appear in combination vectors) for each new data packet. Using Gaussian elimination at the receiver side, this method provides *instant*

decodability of data packets.

Even though PNC could achieve almost instant decodability using a small generation size, this would be ineffective in a multi-hop wireless network, where a receiver could be surrounded by a large number of senders, and if the size of the generation is smaller than the number of senders, some coefficient vectors will necessarily be linearly dependent. On the other hand EWNC allows early decodability while *innovativity* (*i.e.*, linear independence) can be achieved if the sources include the packets in the coding window *in a different order*. However, these orders should take into account the properties of the video stream, as we shall discuss in detail in Sec. 2.

In this work, we propose to use EWNC for the robust delivery of multi-view video over an unreliable network such as a wireless network. At the best of the authors' knowledge, the problem of providing an efficient strategy for inclusion of video packets in the coding window has not been addressed yet. We therefore design a *Rate-Distortion Optimized* (RDO) scheduling algorithm that, at each sending opportunity, selects which video packet has to be added to the coding window in such a way as to maximize the expected video quality measured at the receiver. Since the wireless medium is inherently broadcast, we want to exploit the possibility of the receiver being exposed to multiple sources. We thus want to ensure that the sources send innovative coding vectors even though they do not coordinate their actions.

The rest of this paper is organized as follows: in Sec. 2, we introduce our proposed approach to timely transmission of compressed video streams over lossy channels. Then, in Sec. 3, we present some of our experimental results, with a detailed illustration of the simulation environment and a comparison with the state-of-the-art techniques. Finally, in Sec. 4 we draw conclusions and outline future work.

2. PROPOSED APPROACH

In this section, we detail our proposed framework, whose objective is to provide a novel transmission strategy for multi-view streams over lossy networks, such as wireless networks, with a good trade-off between resiliency to losses and timely delivery.

Recent work [11] has shown that providing a prioritized video delivery via path diversity and random linear network coding substantially outperforms baseline network coding and rate-less codes with inherent UEP properties.

In order to provide such a prioritization, we propose to use EWNC [12], which we expect to provide loss resiliency to the video stream without affecting the delay.

As mentioned in Sec. 1, the efficiency of EWNC highly depends on the order (or priority) in which the data packets are included in the coding window. The original EWNC method was proposed for a single view layered video, therefore the priority of the packets was naturally imposed by the dependencies among layers.

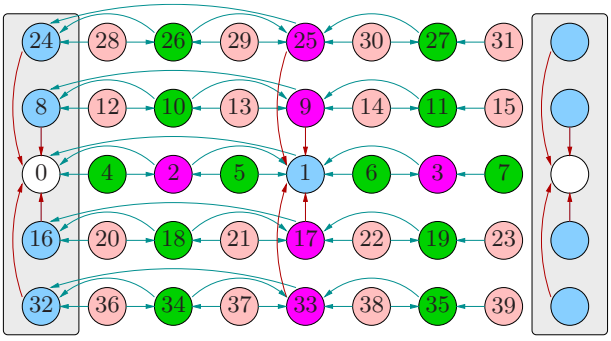

Fig. 1. Example of MVC prediction structure with intra- (horizontal arrows) and inter-view (vertical arrows) prediction. The numbers indicate the coding order.

Such a strategy is unfeasible in our scenario, as we deal with multiple uncoordinated senders sharing a broadcast medium, and if they all were to choose the same order of packets (*i.e.*, the one imposed by the layered structure), at any given sending opportunity they would send *non-innovative* combinations.

In general, if a prioritization is optimal, it is also unique, thus all the senders would always transmit dependent combinations, defeating the purpose of using NC. In order to take advantage of the benefits of NC in terms of loss resiliency, we need to generate a variety of schedules, possibly slightly sub-optimal, but with acceptable performances.

The GOP structure of a multi-view video coding technique (such as H.264/MVC) leaves a certain degree of freedom in the scheduling, as frames on the same prediction level can be sent in any order. However, this degree of freedom may not be enough to provide a sufficient number of different schedules for the different senders.

In a multi-view context, the pool of frames candidate for inclusion in the coding window is a bi-dimensional *multi-view GOP* (MV-GOP), *i.e.*, a rectangular buffer of size $N \times W$, where N is the number of views and W is the temporal size. An example of a multi-view GOP structure is shown in Fig. 1. Notice that the frames in this buffer are not ordered by their play-out date, but in encoding order, so that frame dependencies are respected.

The task of the scheduler is to provide an order in which the frames in the MV-GOP are included in the coding window. Since wireless networks are affected by churn and mobility, and the video stream can be interrupted at any moment, it is desirable that any new combination maximizes the marginal benefit in terms of RD properties. In other words, at each step, we want the scheduling algorithm to select the frame that optimizes an RD criterion for insertion in the coding window.

However, corresponding frames of different views have differences in their RD properties, which would lead to a unique optimal policy of inclusion in the coding window.

In order to solve this problem, we propose a *clustering* of the video frames: the clustering is a classification of the

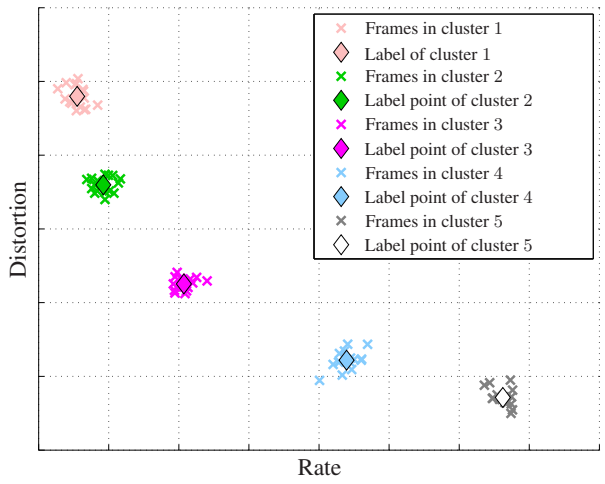

Fig. 2. Clustering of video frames for RDO-scheduling. Frames with similar operating points are assigned to the same cluster. The RDO-scheduling will consider each frame as having the average operating point of its cluster.

frames based on their RD properties that takes place at the video source, after the video encoding and before scheduling for transmission. Frames with similar RD points are assigned to the same cluster; each frame is labelled with the average rate and distortion of its cluster, possibly quantized. An illustration of the clustering is given in Fig. 2.

The labels are decided *only once* at the encoder, where rate and distortion are known with negligible computational overhead.

The purpose of clustering is to increase diversity. At the intermediate nodes, at each sending opportunity, the scheduler selects for inclusion in the coding window a frame among the eligible ones, *i.e.*, those whose references for prediction, if any, are already in the coding window. The selected frame minimizes a rate-distortion cost function. However, nodes use the rate and distortion values reported on the labels to evaluate the cost function, rather than the actual values. It is therefore very likely that several frames appear to have the same cost. In this case, a node would randomly select one of them. This frame is added to the coding window, increasing its size by one. The size of the coding window is reset to zero with the new MV-GOP. A summary of the operations performed by the nodes is reported in Algorithm 1.

Notice that large clusters increase the chance of different nodes selecting different frames, thus reducing non-innovative packets. On the other hand, if clusters are chosen too large, the scheduler will randomly choose among frames with very different values of the cost function, resulting in a sub-optimal performance.

Ideally, the size of the clusters should be chosen according to the expected number of senders that are going to transmit at the same time (so that each sender could select a different

Algorithm 1 Algorithm used by the nodes to include the frames in the coding window.

procedure SELECTFRAME
  $G \leftarrow N \times W$; ▷ Size of the generation.
  for all MV-GOPs **do**
    set size of coding window to zero;
    for $i \leftarrow 1$ **to** G **do**
      $\mathcal{F} \leftarrow \{\text{Pool of eligible frames}\}$;
      $J^* \leftarrow \min_{f \in \mathcal{F}} \{J(f) = R(f) + \lambda D(f)\}$;
      $f^* \leftarrow$ a random frame in $\{f | J(f) = J^*\}$;
      increase size of coding window by one;
      include f^* in coding window;
    end for
  end for
end procedure

frame of the same cluster), which can be roughly estimated with the node density of the network.

In practice, clustering can be performed in several ways. For instance, a coarse but simple scheme is to assign all the frames on the same prediction level to a single cluster. This scheme is independent from the actual RD properties of the sequence and can be easily implemented; nevertheless, it can be quite efficient if the views have frame-by-frame similar RD properties. If the corresponding frames of different views have unbalanced properties, then a more sophisticated scheme can be employed.

An example of two different scheduling orders is presented in Fig. 3. For the sake of clarity, only the scheduling for the first 20 packets is presented. We can observe that, if only a subset of a cluster is chosen, the two schedulers choose different frames within it. If the whole cluster is chosen, then the frames still differ in the order they are included in the coding window.

3. EXPERIMENTAL RESULTS

In the following, we present the results of the proposed technique and compare them with the results achievable using PNC on each view independently. For reference, the results achieved without using NC are also presented.

In our scenario, M sources S_m, $m \in \{1, \cdots, M\}$, transmit a multi-view video sequence $I(k)$, $k \in \{1, \cdots, K\}$ to a single receiver R. In the simulations, the video sequences "ballet", "bookarrival", "breakdancers", and "doorflowers" (5 views, 1024×768, 100 frames) have been encoded in H.264/MVC using the GOP structure in Fig. 1 with QP 31, 34, 37, and 40.

In order to allow a clear evaluation of our technique, a discrete-time transmission model is assumed: the time is segmented in *transmission rounds* wherein each source sends exactly one packet from a predetermined transmission buffer.

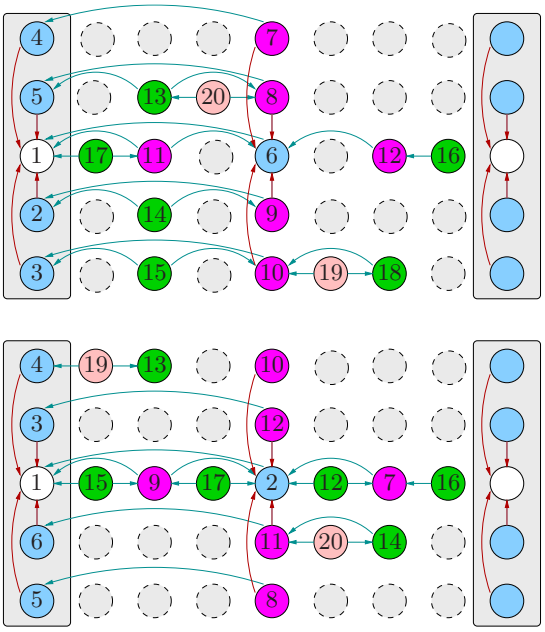

Fig. 3. Two possible schedules (first 20 rounds). The numbers indicate the round in which the frame is included in the coding window. The dashed border identifies which frames have not been selected yet for inclusion in the coding window at the 20-th round.

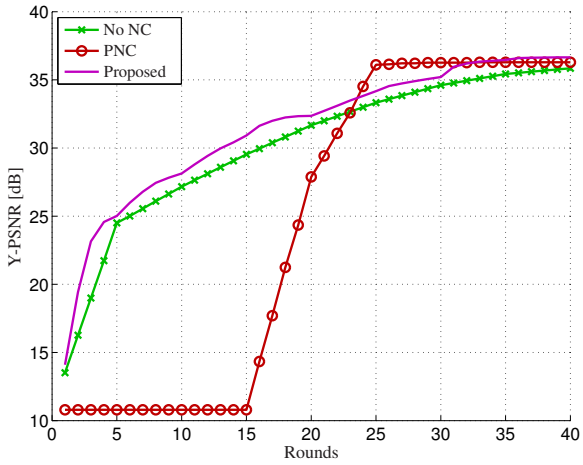

Fig. 4. Comparison of the average Y-PSNR of the decoded sequences, for 2 sources and packet loss probability 5 %.

Each channel between a transmission buffer and the receiver buffer is in general lossy, with independent uniform packet loss probability p_m; the transmissions on different channels do not interfere with each other. At the end of each round, the receiver decodes all the frames available in its buffer, generating a reconstructed sequence.

In our simulations, the proposed approach has proven to be able to deliver an acceptable video quality in terms of PSNR to the receiver in a shorter number of rounds than the reference techniques. We observed that the performance of the method benefits from a higher number of sources, whereas it is of course negatively affected by the loss rate.

In Fig. 4–7, we report a comparison with the reference techniques for a two senders scenario with several packet loss probabilities.

We observe that, if no network coding is used, each received packet increases the Y-PSNR. However, the transmission cannot recover from losses, thus in scenarios with high loss probability, the maximum quality is not achieved.

Conversely, PNC eventually achieves the maximum quality, but the receiver cannot decode any frames in the first rounds. This is undesirable in a wireless environment as the communication could be interrupted at any moment, leaving the node with no useful data.

Our approach, thanks to the early decodability offered by EWNC and the variety in the scheduling provided by the clustering, is able to both provide an acceptable video quality in the first rounds and to achive the maximum video quality in the long term, even in the presence of losses.

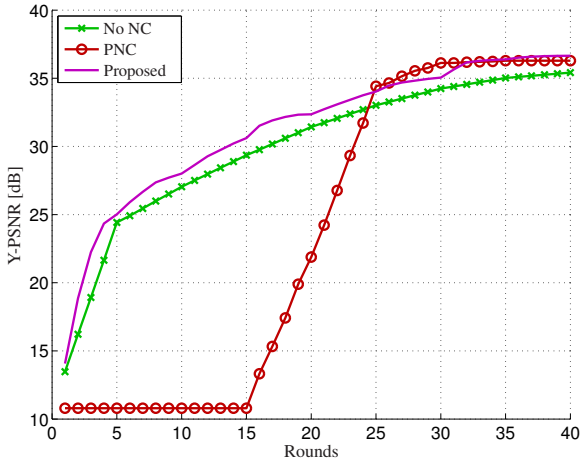

Fig. 5. Comparison of the average Y-PSNR of the decoded sequences, for 2 sources and packet loss probability 10 %.

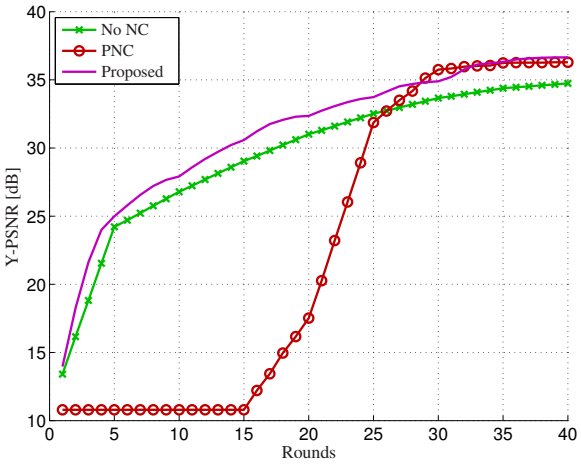

Fig. 6. Comparison of the average Y-PSNR of the decoded sequences, for 2 sources and packet loss probability $15\,\%$.

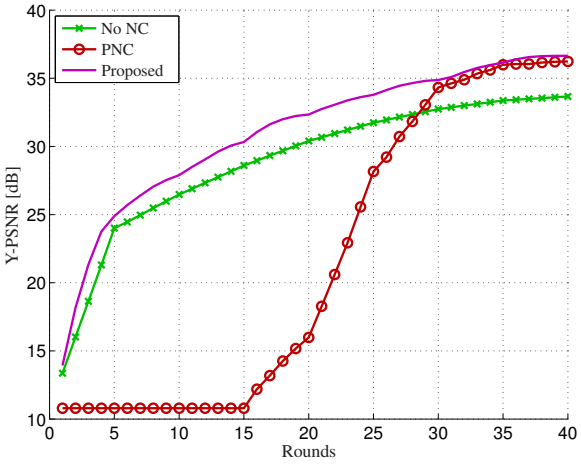

Fig. 7. Comparison of the average Y-PSNR of the decoded sequences, for 2 sources and packet loss probability $20\,\%$.

4. CONCLUSIONS

In this work, we presented a novel technique for streaming multi-view video content over unreliable channels using network coding. The key idea is to use Expanding Window Network Coding in order to guarantee instant decodability of the flow. The frames are included in the coding window in an order determined by an RD-optimized scheduler. In order to reduce the probability of generating non-innovative packets, the sources operate with a simplified probabilistic RD model that provides them with a degree of freedom in the choice of the schedule.

We compared the performance of our technique with Practical Network Coding applied on each view independently, and transmission without network coding, both assuming a trivial scheduling order.

We observe that the introduction of the scheduling, jointly with the possibility of mixing packets across views, significantly improves the performance w.r.t. the reference techniques, in terms of video quality perceived by the user.

The results we obtained suggest that further research in this direction could be promising, in particular in the direction of a joint design of an overlay management protocol that could select which nodes of the network should rely the stream.

References

[1] M. Tanimoto, M. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Proc. Mag.*, vol. 28, no. 1, pp. 67–76, Jan. 2011.

[2] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Free viewpoint switching in multi-view video streaming using wyner-ziv video coding," in *Proc. of SPIE*, vol. 6077, 2006, pp. 298–305.

[3] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV – A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1606–1621, Nov. 2007, Invited Paper.

[4] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007, Invited Paper.

[5] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011, Invited Paper.

[6] I. J. MPEG2011/N12036, "Call for proposals on 3D video coding technology," Mar. 2011.

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

[8] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[9] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Proc. of IEEE Int. Symp. Inf. Theory*, Kanagawa, Japan, Jun. 2003.

[10] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conf. on Commun. Control and Comput.*, Monticello, IL, USA, Oct. 2003.

[11] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized distributed video delivery with randomized network coding," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 776–787, Aug. 2011.

[12] D. Vukobratović and V. Stanković, "Unequal error protection random linear coding for multimedia communications," in *Proc. of IEEE Workshop on Multimedia Sign. Proc.*, Saint-Malo, France, Oct. 2010.