# A NETWORK CODING SCHEDULING FOR MULTIPLE DESCRIPTION VIDEO STREAMING OVER WIRELESS NETWORKS

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

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

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

## ABSTRACT

Network coding (NC) can achieve the maximum information flow in a 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 a framework for video delivery in wireless networks that combines Expanding Window Network Coding (EWNC), Multiple Description Coding (MDC), and a novel Rate-Distortion Optimised (RDO) scheduling algorithm that optimises the order in which the video packets are included in the coding window at the current sending opportunity. Results show that our approach consistently outperforms both EWNC applied on single description coding (SDC) and EWNC applied to MDC when a trivial scheduling is used.

*Index Terms*— Network coding, multiple descriptions, video coding, wireless networks.

## 1. INTRODUCTION

*Network Coding* (NC) [1] has recently been investigated by the research community as an alternative to classical routing for multicast streaming. Using NC, a multi-hop communication is relayed at intermediate nodes by sending combinations of the received messages, rather than mere copies. An interesting application of NC is to grant partial loss immunity to data streams in unreliable wireless networks [2]. Using *Random Linear Network Coding* (RLNC) [3], a technique in which nodes send *combination vectors*, i.e., random linear combinations of their received packets, with coefficients taken from a finite field of proper size, the communication can be routed in unreliable networks with dynamically varying connections with no need for node coordination.

A *practical* implementation of RLNC [4] (PNC) can be achieved segmenting the data into groups of packets called *generations* and combining only packets belonging to the same generation. All packets in a generation are jointly decoded as soon as enough linearly independent combinations have been received, by means of simple linear system solving. Since the coefficients are taken from a finite field, perfect reconstruction is assured regardless of the precision of the implementation.

Recently [5], it has also been proposed to apply NC to video content delivery, dividing the video stream into layers of priority and providing unequal error protection for the different layers via PNC. Layered coding requires that all users receive at least the base layer, hence all received packets must be stored in a buffer until a sufficient number of independent combinations are received, which introduces a decoding delay that is often unacceptable in real-time streaming applications. There exist several techniques aimed to reduce the decoding delay, proposed by both the NC and the video coding communities.

From a network coding perspective, a viable solution is to use *Expanding Window Network Coding* (EWNC) [6]. The key idea of EWNC is 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 packet. Using Gaussian elimination at the receiver side, this method provides *instant decodability* of packets. Thanks to this property EWNC is preferable over PNC in streaming applications. Even though PNC could achieve almost instant decodability using a small generation size, this would be ineffective in a 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 combinations will necessarily be linearly dependent. On the other hand, EWNC automatically adapts the coding window size allowing early decodability, and *innovativity* (i.e., linear independence) can be achieved if the senders include the packets in the coding window in a different order. However, these orders should take into account the RD properties of the video stream, as we shall discuss in detail in Sec. 2.

Another possibility is to employ NC jointly with *multiple description coding* (MDC). MDC [7] is a well established joint source-channel coding paradigm based on splitting a media content into $N$ sub-streams, referred to as *descriptions*. Any description can be independently decoded for representing the content, but the quality improves with the number of descriptions. Video MDC has been proven to be a valuable tool to cope with packet losses in wireless networks [8].

In this work, we propose to jointly use EWNC and video MDC, in order to provide a robust video delivery over an unreliable wireless network, without any need for centralised control or feedback channel. In order to do so, we design a *Rate-Distortion Optimised* (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 maximise the expected video quality perceived by the receiver. Since the wireless medium is inherently broadcast, we want to exploit the possibility of the receiver being exposed to multiple senders. In other words, we assure that the senders transmit innovative coding vectors even though they do not coordinate their actions.

## 2. PROPOSED APPROACH

In this section, we detail our proposed framework, whose objective is to provide a novel transmission strategy for lossy wireless networks able to guarantee a good trade-off between resiliency to losses and timely delivery.

In our scenario, a set of $M$ uncoordinated sources transmit the same encoded video sequence to a single receiver. This scenario could model, for instance, a single hop of a multi-hop transmission.

We propose to jointly use EWNC and video MDC, which we expect to provide loss resiliency to the video stream without affecting the delay. However, as mentioned in Sec. 1, the efficiency of EWNC highly depends on the order in which the packets are included in the coding window. The original EWNC method was proposed for layered video coding, therefore the priority of the packets was naturally imposed by the dependencies among layers. 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 prioritisation is optimal, it is also unique, and thus all the senders would always transmit dependent combinations, defeating the purpose of using NC. However, there exist frames with very similar RD-properties, hence we can generate a variety of scheduling slightly suboptimal, but with performances very close to the optimum.

For instance, the GOP structure of a video coding technique (such as H.264/AVC) leaves a certain degree of freedom in the scheduling, as frames on the same prediction level can be sent in any order (two examples of GOP structures are shown in Fig. 1), but this may not be enough to provide a sufficient number of different schedules for the different senders.

Using an MDC technique, it is possible to have multiple senders transmitting packets that refer to the same instant, but different nonetheless. Furthermore, corresponding packets of different descriptions are mutually refinable, therefore a node being served by multiple senders will perceive an enhanced
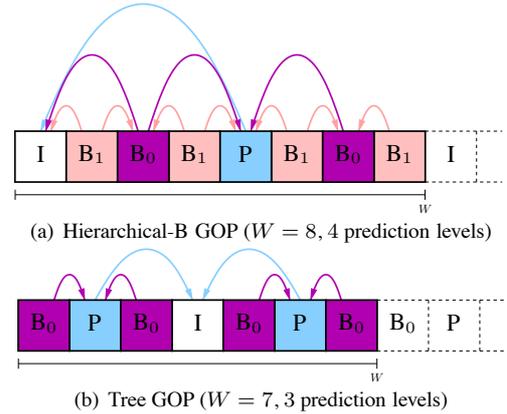


(a) Hierarchical-B GOP ($W = 8$, 4 prediction levels)



(b) Tree GOP ($W = 7$, 3 prediction levels)

**Fig. 1**. Two possible GOP structures in H.264/AVC. Arrows indicate prediction. Frames on the same prediction level can be sent in any order.

video quality. Using MDC, the pool of frames candidate for inclusion in the coding window is a bi-dimensional *multiple description GOP* (MD-GOP), i.e., a rectangular buffer of size $N \times W$, where $N$ is the number of descriptions and $W$ is the GOP size of each description. An example of MD-GOP is depicted in Fig. 2, for 4 descriptions and a GOP structure of each description as the one in Fig. 1(a), i.e., Hierarchical-B. Notice that in the buffer the frames are not ordered by their play-out date, but in the encoding order, so that frame dependencies are respected.
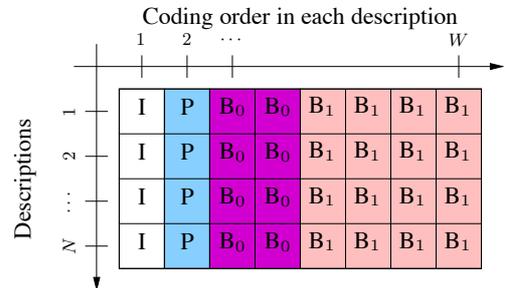


**Fig. 2**. MD-GOP for $N = 4$ descriptions and $W = 8$ frames in Hierarchical B-frame GOP. Frames are ordered by prediction level.

The task of the scheduler is to provide an order in which the frames in the MD-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 maximise the marginal benefit in terms of RD properties. In other words, at each step, we want the scheduling algorithm to select the frame that optimises an RD criterion for insertion in the coding window. However, the corresponding frames of different descriptions might have differences in their RD properties, which would still lead to a unique optimal policy of inclusion in the coding window.

In order to obviate this problem, we propose a *clustering* of the video frames. The clustering is a classification of the frames that takes place at video source, after the video encoding and before scheduling for transmission. Its purpose is to improve diversity by letting nodes transmitting, at each sending opportunity, a random frame within an optimal cluster. Clusters are decided once at the encoder, where rate and distortion are known with negligible computational overhead, with frames in the same prediction level. The average rate and distortion of the cluster $R(c)$ and $D(c)$ are then computed, possibly quantised, and added as a header to each frame in the cluster.

At each sending opportunity, among the clusters whose prediction level is compatible with the scheduling so far ($\mathcal{C}$), each sender chooses the cluster $c$ that minimises the cost function $J^* = \min_{c \in \mathcal{C}} \{J(c) = R(c) + \lambda D(c)\}$. Within this cluster, each sender *randomly* chooses one frame and schedules it for transmission. This frame is added to the encoding window, increasing its size by one. The size of the coding window is reset to one with the new GOP.

An example of frame clustering is presented in Fig. 3. There, the I-frames of the $4$ descriptions have roughly the same RD properties and are therefore assigned to a single cluster. On the P-frames, on the other hand, descriptions $1$ and $2$ have similar RD properties between them, but different from descriptions $3$ and $4$, which are in turn close to each other. In this case, two clusters are created containing the frames with similar properties. The same holds true for the $B_0$-frames, where descriptions $1$, $2$ and $3$ have been clustered together, while description $4$ was assigned to another cluster. Finally, all $B_1$ frames of all descriptions give similar contributions to distortion and have been assigned to a single cluster.

Large clusters increase the diversity of the scheduling among senders, thus reducing non-innovative packets. However, if clusters are chosen too large, the scheduler will randomly choose among frames with very different values of the objective function, resulting in a sub-optimal performance. Ideally, the size of the clusters should be chosen according with the expected number of senders that are going to transmit at the same time, 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 descriptions are actually frame-by-frame balanced. If the corresponding frames of different descriptions have slightly unbalanced properties, then a more sophisticated scheme can be employed, e.g., based on thresholding.

An example of two different scheduling orders compatible with the clustering of Fig. 3 is presented in Fig. 4. For the sake of clarity, only the scheduling for the first 16 packets is presented. We can observe that, if only a subset of a cluster
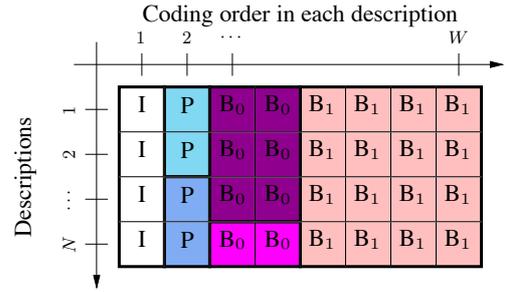


**Fig. 3**. Example of MD-GOP clustering. Frames marked by the same colour are in the same cluster and share similar RD properties.
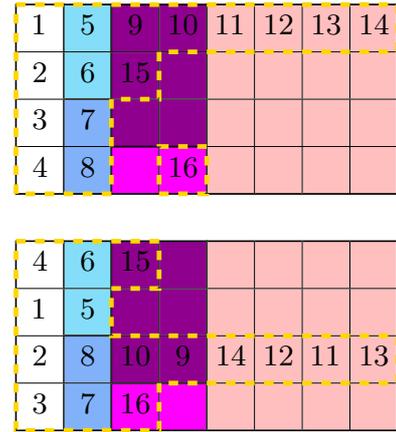


**Fig. 4**. Two possible schedules (first 16 packets). The numbers indicate the order in which the frame is included in the coding window. The dashed border identifies which frames have been selected for inclusion in the coding window at the 16-th packet.

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 via EWNC applied to an SD-coded stream and EWNC applied on an MD-coded stream, but ordered using a trivial schedule.

The transmission scenario we simulate is depicted in Fig. 5. In this scenario, $M$ sources $S_m$, $m = 1, \ldots, M$, intend to transmit the same video sequence, $I(k)$, $k = 1, \ldots, K$, to a single receiver $R$.

For SDC, the trivial strategy consists in including the frames in coding order, i.e., by prediction level and, within frames on the same level, play-out order. For MDC, we assume again that frames are included in coding order and, within frames with the same encoding time (i.e., corresponding frames of independently encoded descriptions), the descriptions are selected in a fixed order. To encode the video

sequences, we chose to use 4-descriptions *Polyphase Downsampling Multiple Descriptions* (PDMD) [9, 10], a technique where $N$ sub-streams are generated by splitting the original sequence via polyphase down-sampling along rows and columns by a factor of 2. To generate the descriptions, each sub-stream is independently encoded using an H.264/AVC reference encoder JM [11], version 17.0. The encoding algorithm uses the closed-GOP structure presented in Fig. 1(a). A closed-GOP was preferred in order to reduce error propagation in case of losses. The RD properties of each frame are exactly measured. Clustering is performed based on prediction level. The average rate and distortion for the frames in each cluster are computed, quantised on eight bits each, and sent along with the video data.

At the decoder side, all the descriptions are independently decoded in order to obtain the $N$ sub-streams, which the receiver interleaves to reconstruct the central sequence. When some descriptions are lost, the receiver oversamples the available sub-streams, interpolating the missing pixels to obtain a good low-resolution frame (side decoding). When none of the descriptions is available, the loss is concealed using the closest decoded frames.

In order to compare the performance of the method under a variety of inputs, we selected a set of 10 QPs (in Tab. 1) and 8 video sequences (in Tab. 2) with CIF spatial resolution at 30 frames per second.

| High Bitrate | 16 | 19 | | |
|---|---|---|---|---|
| Medium Bitrate | 22 | 25 | 28 | 31 |
| Low Bitrate | 33 | 36 | 39 | 42 |

**Table 1**. QPs used in encoding the video sequences.

| akiyo | hall | foreman | city |
|---|---|---|---|
| coastguard | football | stefan | bus |

**Table 2**. Video sequences used in simulations.

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 $S_m$ sends exactly one packet from a predetermined transmission buffer $TX_m$. Each channel $C_m$ between transmission buffer $TX_m$ and the receiver buffer RX 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 RX, generating a reconstructed sequence $\widetilde{I}(k)$. This simple scenario is well suited to model a wireless ad-hoc network where a channel reservation mechanism is enforced [12], which provides both discrete-time transmission and channel isolation.

In our simulations, the proposed approach has proven to be able to deliver an acceptable video quality to the receiver in a shorter number of rounds than the reference techniques.
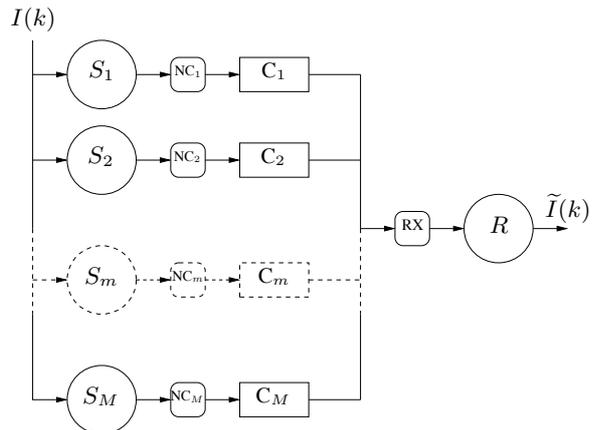


**Fig. 5**. Simulated scenario. $I(k)$ and $\widetilde{I}(k)$ are respectively the original and reconstructed frames, $S_m$, $m = 1, \ldots, M$ are the senders, $NC_m$ the network coding modules, $C_m$ the channels, RX is the receiver $R$'s buffer.

As an example, in Fig. 6, we report a comparison with the reference techniques under a few different simulation conditions. We observe that, thanks to the variety in the scheduling, our technique is able to reduce the number of linearly dependent coding vectors, and is therefore able to provide a better video quality (in terms of Y-PSNR) in fewer rounds. It should be noted that the final value of the Y-PSNR for the SD-based technique is slightly higher (about $0.5$ dB) than that of both MD-based ones, which is a direct consequence of the inherent redundancy among the descriptions of the MD encoding. However, this happens after a long enough time (i.e., about 30 rounds), during which MDC/NC has already achieved its final Y-PSNR. We can also observe that the performance of the method benefits from a higher number of sources, whereas it is of course negatively affected by the loss rate.

## 4. CONCLUSIONS

In this work, we presented a novel technique for video streaming over unreliable channels using a combination of multiple description coding and network coding.

The key idea in this technique is to use Expanding Window Network Coding in order to guarantee instant decodability to the flow. The frames are included in the coding window in an order determined by an RD-optimised scheduler. In order to reduce the probability of generating non-innovative packets, the sources operate a classification of the frames (clustering) that provides them with a degree of freedom in the choice of the schedule.

We compared the performance of our technique with Expanding Window Network Coding applied on both on Single Description and Multiple Description coding, assuming a trivial scheduling order, and (in the case of MDC) limiting the combinations within the same description. We observe that
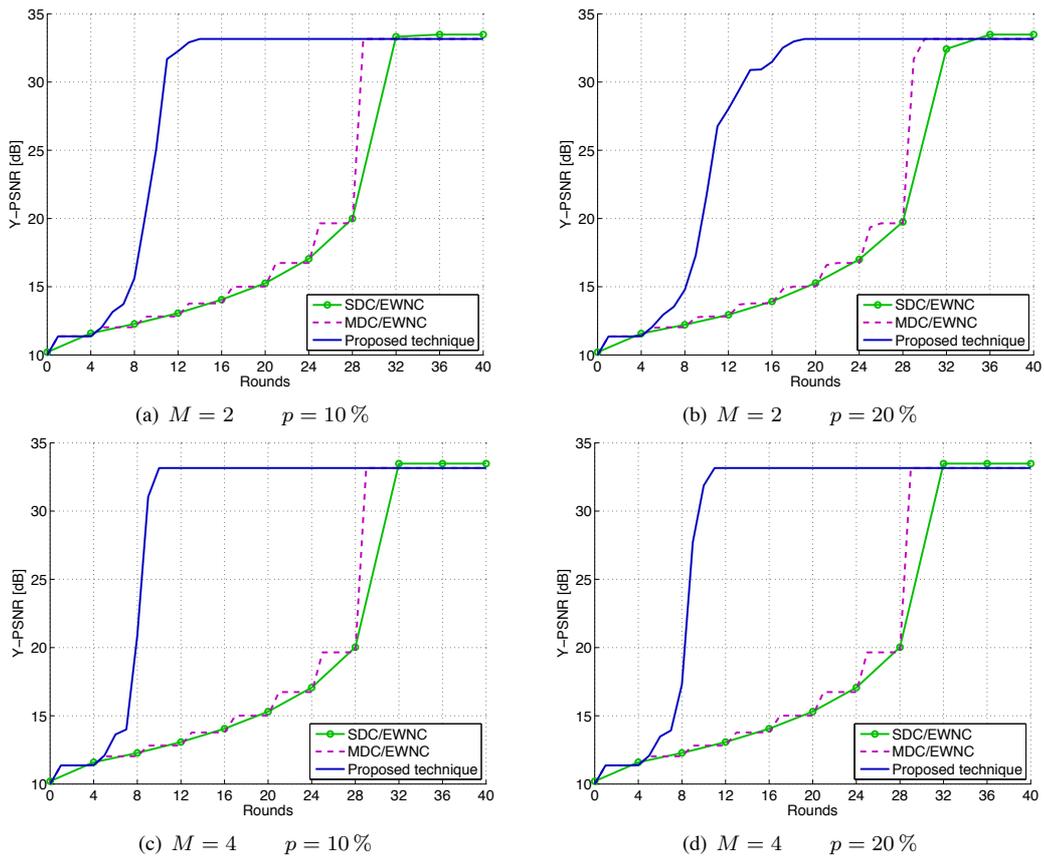
(a) $M = 2$     $p = 10\%$

(b) $M = 2$     $p = 20\%$

(c) $M = 4$     $p = 10\%$

(d) $M = 4$     $p = 20\%$

**Fig. 6**. Comparison of the average Y-PSNR of the decoded sequences, for $M$ sources and packet loss probability $p$.

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

The result we obtained suggests 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 relay the stream.

# References

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

[2] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft, "XORs in the air: practical wireless network coding," *ACM SigComm Computer Commun. Rev.*, vol. 36, no. 4, pp. 243–254, Aug. 2006.

[3] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

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

[5] 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.

[6] D. Vukobratović and 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, pp. 280–285.

[7] Vivek K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 74–93, Sept. 2001.

[8] Y. Wang, A.R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005, Invited Paper.

[9] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, "Multiple description video coding for scalable and robust transmission over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 321–334, Mar. 2005.

[10] M. Caramma, M. Fumagalli, and R.C. Lancini, "Polyphase downsampling multiple-description coding for IP transmission," in *Proc. of SPIE*, 2000, vol. 4310, p. 545.

[11] Karsten Sühring, "JM reference software release 17.0," Source Code, Jan. 2011.

[12] Claudio Greco and Marco Cagnazzo, "A cross-layer protocol for cooperative content delivery over mobile ad-hoc networks," *Inderscience Int. J. of Commun. Networks and Distrib. Syst.*, vol. 7, no. 1–2, pp. 49–63, June 2011.