

RESEARCH

# Rate-Distortion Optimized Multi-View Streaming in Wireless Environment using Network Coding

Claudio Greco<sup>1</sup>, Irina D Nemoianu<sup>2</sup>, Marco Cagnazzo<sup>3\*</sup>  
and Béatrice Pesquet-Popescu<sup>3</sup>

\*Correspondence:

[marco.cagnazzo@telecom-paristech.fr](mailto:marco.cagnazzo@telecom-paristech.fr)

<sup>3</sup>Institut Mines,

Telecom-ParisTech, LTCI-CNRS,  
Paris, France

Full list of author information is  
available at the end of the article

## Abstract

Multi-view video streaming is an emerging video paradigm that enables new interactive services, such as 3D video, free viewpoint television, and immersive teleconferencing. Because of the high bandwidth cost they come with, multi-view streaming applications can greatly benefit from the use of network coding, in particular in transmission scenarios such as wireless network, where the channels have limited capacity and are affected by losses. In this paper, we address the topic of cooperative streaming of multi-view video content, wherein users who recently acquired the content can contribute parts of it to their neighbors by providing linear combinations of the video packets. We propose a novel method for selection and network encoding of the transmitted frames based on the users' preferences for the different views, and the rate-distortion properties of the stream. Using network coding enables the users to retrieve the content in a faster and more reliable manner, and without the need for coordination among the senders. Our experimental results prove that our preference-based approach provides a high quality decoding even when the up-link capacity of each node is only a small fraction of the rate of the stream.

**Keywords:** Multi-View Video Coding; Network Coding; Video Streaming

## 1 Introduction

In recent years, the advances in video acquisition, compression, transmission, and rendering have made possible the development of technologies that can enhance the viewers' experience by including the third dimension. While traditional 2D video offers the viewer only a passive view of the scene, a more realistic experience can be obtained through applications such as 3D video or free view-point selection. 3D cinema productions have already generated big revenues, but other applications such as 3DTV and Free Viewpoint TV (FTV) [1, 2] are also becoming more desirable due to the increased affordability of 3D displays for home use.

*Multi-View Video* (MVV) is one of the key elements of these applications: it consists in the simultaneous representation of a scene captured by  $N$  cameras placed in different spatial positions, called points of view. By using more than two cameras during video acquisition, adjacent views act like local stereo pairs to guarantee stereoscopy to the viewer. This can be used to synthesize virtual views different from the acquired ones. This functionality is used in FTV where the user interactively controls the viewpoint in the scene. On the

other hand, since 3D video could not be deployed if the quality perceived by the user does not exceed the existing 2D quality standards, the bandwidth for storage and transmission of the multiple views is accordingly increased.

A first solution for multi-view video transmission, known as *simulcast* [3], is to compress and send each view independently [4]). While simple to implement and backward compatible with the existing infrastructures, this technique does not take into account the redundancy due to the similarities among the views that can be used to further compress the data. On the other hand, it allows for easier switching between views, as the lack of inter-view prediction makes the views independently decodable.

The *Multi-view Video Coding* (MVC) extension of the H.264/MPEG-4 AVC standard [5] exploits inter-view dependency in a simple, yet effective ways: images from other views (but at the same time instant) can be used as references for the current frame prediction (inter-view prediction). This is the only major change introduced in the MVC extension of H.264. The MVC extension of HEVC, referred to as MV-HEVC, is based on very similar principles [6]. With MVC two main coding schemes are particularly worth mentioning: view progressive and fully hierarchical. In the view progressive architecture, the first view, called the *base view*, is encoded independently from the others. In any other view, for each GOP, there is one frame, the V-frame, that is predicted using only inter-view prediction from the corresponding I-frame in the base view. For all other frames only temporal prediction is used. In the second architecture, both hierarchical temporal prediction and inter-view prediction are performed for all P/B-frames of all views except for the the base view. These tools allow a rate reduction, for the same subjective quality, estimated around 50% with respect to the case of independent view coding (Simulcast) [5].

Even though recently a relevant part of the attention of the research in 3D has been attracted by depth-based formats [7] (which allow virtual view-point synthesis), the interest in MVV coding is still very high, as witnessed by the activity of the ad-hoc group on free viewpoint TV and super-multiview video (i.e. video with more than 30 views, and holoscopic video) [8–10]. The quality of synthesized view generated with depth data is still questionable, at such a point that it is still not completely clear whether depth-based format have a clear advantage over MVV or super-MVV, above all when subjective quality is considered [11]. In summary, (super-) MVV seems still being a serious candidate for FTV and 3D video services [12].

Multi-view streaming becomes an even more challenging task in the context of mobile networking, where the high bitrate issue of multi-view adds on top of the existing problems of mobile networking. Even though streaming applications are nowadays commonplace, and the technology involved has greatly advanced in the past few years [13, 14], in a wireless network it is difficult to meet the inherent requirement of continuous delivery necessary for an uninterrupted presentation of the content, as the nodes move freely and independently in all directions –thus, the channel conditions of the links and the link themselves are unreliable and erratic– and individual nodes may connect and disconnect asynchronously [15].

Also, in the context of a streaming application, it would be desirable to have the quality of the received media degrade gracefully as the network environment and resources change and to tolerate losses to some extent. Even though techniques to provide graceful degradation and loss immunity exist, these usually require an increase in the bitrate of the stream, a condition that could be difficult to satisfy in a wireless network, where the nodes' uplink capacity is typically quite limited.

One positive aspect of wireless networks w.r.t. video streaming is the inherently broadcast nature of the medium. This makes more straight-forward for a sender the task of multicasting the content to several receiver, but also allows a single receiver to collect video packet from several servers.

Recently, good results have been achieved, in the context of mobile video streaming, by exploiting the broadcast nature of the medium through the construction of video packet delivery overlays [16, 17]. These logical networks, built on top of the actual wireless network through the cooperation of nodes, allow to provide a streaming service with good video quality and graceful degradation. However, these techniques were designed for single-view streams, and relied on the use of Multiple Description Coding (MDC) [18], a joint source-channel coding technique that does not lend itself well to be conjugated with multi-view, due to its additional bitrate cost, a cost already considerable for multi-view streams.

In this article, we propose to use network coding for the robust delivery of MVV and super-MVV over an unreliable network such as a wireless networks. In order to do so, we 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 minimize the expected video distortion measured at the receiver. This optimization will be performed by taking into account the preferences of the users in terms of required views, an approach already successfully exploited for video caching of single-view streams in mobile environment [19]. Being the wireless medium inherently broadcast, we exploit the fact that each receiver could be exposed to multiple senders. We thus ensure that senders transmit innovative packets (*i.e.* packets with novel information with respect to those already sent) even though they do not coordinate their actions.

The particularity of the coding structures of the multi-view representation reflects in a non-trivial impact of each coded frame on the overall quality of the reconstructed multi-view content. If this impact is properly captured, it can be used to design an intelligent transmission scheme that allocates the limited channel capacities in a rate-distortion optimized order (*scheduling*). In order to effectively disseminate the content to the end-users, an analogous scheme can be devised to schedule the frames for transmission [20].

*Network Coding* (NC) [21] has been proposed as an elegant and effective solution for multi-view transmission. In NC, instead of merely relaying packets, the intermediate nodes of a network send linear combinations of the packets they have previously received, with random coefficients taken from a finite field. The coding coefficients, needed to reconstruct the original packets, are typically sent along the combinations as headers [22–25], unless more advanced reconstruction schemes are implemented at the receiver side [26, 27]. Used as an alternative to traditional routing, NC has proved beneficial to real-time streaming applications, both in terms of maximization of the throughput and in terms of reduction of the effects of losses [28–33].

In a NC-based transmission system, rather than sending the data packets, the users send mixed packets. The advantage of this technique is that even though the users act independently from each other, with high probability each of them will contribute innovative information to the transmission [20, 34]. In the most common implementation of network coding, referred to as Practical Network Coding (PNC) [25], the content is divided into groups of packets known as *generations*, and only packets belonging to the same generation can be mixed together. In our system each packet contains only one encoded frame, and we only mix frames belonging to the same GOP. The set of packets actually used to generate a mixture is referred to as *coding window*.

One technique based on the network coding principles has been proposed by Wang *et al.* [35] for peer-to-peer video-on-demand applications. More recently, Kao *et al.* [36] proposed a general framework able to provide an interactive streaming service, *i.e.*, allowing random access operations to the users. However, neither of these techniques addresses the multi-view case, nor takes into account the rate-distortion properties of the stream, nor the users' preferences.

Other existing works have tackled the subject of distributed video services, achieving similar properties, by proposing to use rateless codes –conceptually similar to network coding– for video delivery [37, 38]. However, even though these techniques have been proposed for video delivery, only the delay requirements of video streaming have been exploited, while our method is tailored for multi-view video content and in particular it uses the prediction structure of the encoded sequence in its optimization algorithm. It should be noted that in our method a proper RDO-based scheduling is performed in order to provide the users with the best possible video quality given the limited channel capacity allocated to each node.

The rest of this article is organized as follows. In Sec. 2 we review some recent works closely related to our problem. Then, in Sec. 3 we present the system model, detailing and motivating our assumptions. In Sec. 4 we describe the selection method used to decide which frames will be included in the coding window of the transmitting nodes. In Sec. 5, we present the experimental validation of the proposed technique and analyze the results. Finally, in Sec. 6 we draw our conclusions and point out some directions for future work.

## 2 Related work

Unlike previous works on multi-view streaming rather than focusing on the source encoding of the content, and rather than considering each client as an independent agent, we study how the distribution of the stream can take advantage of an *a priori* knowledge about the different clients, and in particular the fact that they share common preferences—in this case, in terms of preferred view.

Examples of work in the context of multi-view streaming that take user preferences into account include the source rate allocation technique proposed in [39] and the joint source-channel coding scheme introduced in [40].

While these works consider similar applications as ours, we address here a substantially different problem, in which the multi-view video has been already encoded, and we must decide, at each sending opportunity, about which parts of the content have to be included in the coding window for transmission. We also consider the case when the preference estimation used to decide the packet scheduling does not perfectly correspond to the actual user preferences.

In our work, we also rely on a network coding scheme that allows for the prioritization of certain packets with respect to others. Several works exist that make use of similar schemes, in which the video stream is divided into layers of priority and unequal error protection is given to the different layers using PNC.

For instance, in [29] a receiver-driven network coding strategy is proposed, where the receiving peers request packets from classes with varying importance. Packet classes are constructed based on the unequal contribution of the various video packets to the overall quality of the presentation or in scalable video streams. Prioritized transmission is achieved

by varying the number of packets from each class that are used in network coding operations. The coding operations are driven by the children nodes that determine the optimal amount of coding allocated to each importance class of the data to which they subscribe.

The work in [29] has later been extended to the case of multi-view video in [41]. Cameras' streams are organized into layered subsets, with subsets organized based on their priority levels. These prioritized layers are transmitted in an UEP fashion, sending in a more reliable way more important subsets. Interview dependencies are built based on the subsets organization: views from a given subset can depend from views of the same subset or lower ones. In this way, since lower subsets are more likely to be received than higher ones, every time a view has to be decoded, most likely the reference view from which it depends has been already received.

This work is related to ours both for its use of network coding and its application to multi-view content. However, there are notable differences both in the model of the service provided to the user and, as a consequence, to the utility function that is maximized.

In the scenario envisioned in this work, users request viewpoints that are, in general, synthesized from camera views either by coinciding with one them, or by using depth-image based rendering on a couple of camera views bracketing the synthetic viewpoint. The distortion to minimize depends on the spatial distance between the synthetic view and each of the camera views used to reconstruct it. Priority, in the sense of a higher redundancy to insure reception in the face of losses, is defined based on the utility of camera view subsets in reconstructing the synthetic views requested by the users.

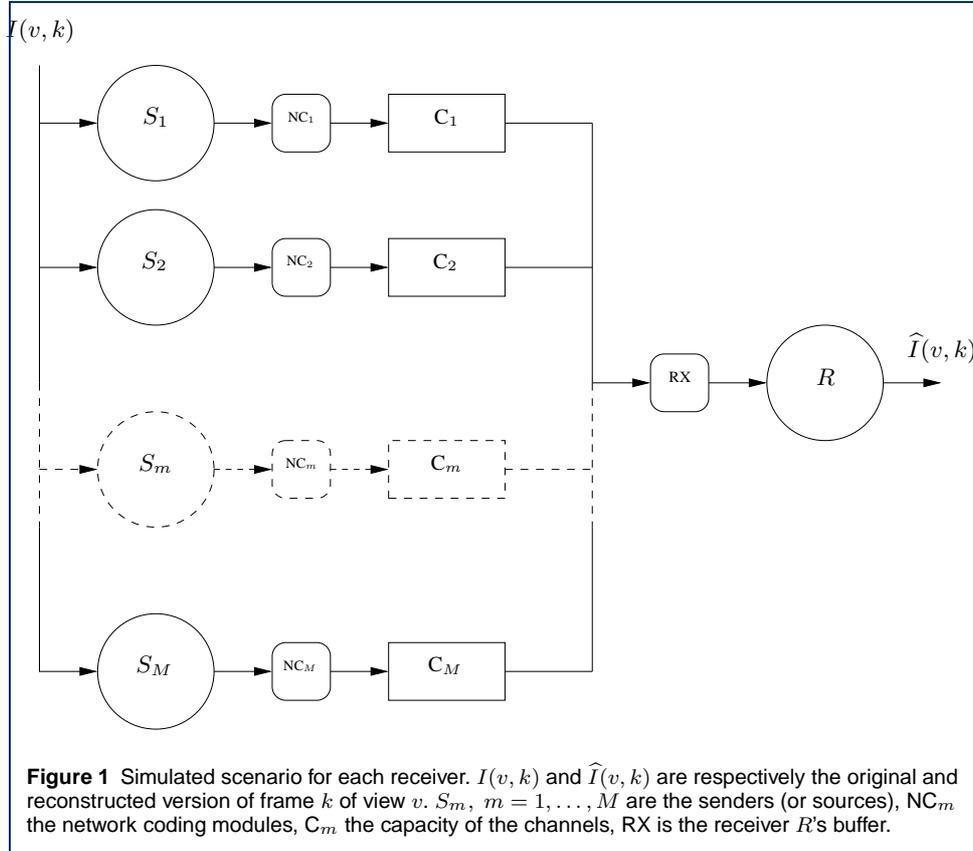
In our work, on the other hand, the users are only interested in camera views, *i.e.*, no view synthesis is used. This implies that, while in the above mentioned work, there are different combination of received camera views that can satisfy the view request of a user, with different levels of distortion depending on their distance, in our scheme only the exact camera view the user is interested in can increase its quality of experience.

Furthermore, in our scheme priority is not intended in the sense of loss protection, but rather arrival order. In our scheme, the different treatment of layers is not intended to differentiate the likelihood of their reception, but rather the delay experienced by the user before they can start displaying it. For this reason, while the network coding scheme used in [41] varies the number of packets from each layer in the coding window, in our scheme all packets from lower layers are introduced in the coding window before any packet of a higher layer is introduced.

Notice that this work only consider the case of aligned and equally spaced cameras, so that correlation between views decreases with their distance. In a more recent work [42], the same authors extend this model to optimize other settings, but this work does not address the communication aspects.

Another relevant approach to video transmission from multiple senders is proposed in [43], wherein the authors jointly tackled the problem of defining an optimal schedule and an optimal network-coding strategy using a prioritizing network coding scheme. Unlike ours, this work only considers the case of single-view content, therefore there are no preferences to be taken into account, and the optimal schedule is unique. Furthermore, in order to find an optimal solution this technique requires some degree of coordination among the senders, whereas we assume that coordination is not feasible and rely on randomization in order to circumvent this limitation.

[41] [42] [43] [32] [40]



### 3 System Model

In order to optimize the rate-distortion performance of the transmitted content, we select the frames to be included in the coding window based on their popularity among the users. Before explaining in detail our proposed technique, in this section we list and justify some assumptions about the system that will be used in the design of the technique.

- From the point of view of the network, we assume that the users are connected in a (generally partial) mesh network in which each node can potentially receive from multiple servers. This reflects the case of wireless networks and in particular ad-hoc networks. Furthermore, we assume that the connectivity among the users can be modeled with a set of independent channels, each of them having a given capacity  $C$ , expressed as a fraction of the encoded video bit-rate. When  $C = 100\%$  each node is able to transmit all the packets of a GOP in the time allocated to a GOP. Still, these packets may be lost on the channels. We consider two models for these channels: a simple packet erasure channel (PEC) with loss rate  $\varepsilon$ , and a Gilbert-Elliott erasure channel (GE), characterized by loss rates in good and bad state ( $\varepsilon_G$  and  $\varepsilon_B$ ) and by transition probabilities ( $p_{GB}$  and  $p_{BG}$ ). Notice that each channel does not necessarily provide sufficient capacity for transferring the whole multi-view stream. Our study will focus on the video quality achieved by a generic receiver  $R$  exposed to  $M$  senders or sources  $S_1, \dots, S_M$ . This scenario is represented in Fig. 1.
- From the point of view of the content, we assume that the stream is encoded using H.264/MVC [5] or a similar inter-view prediction scheme, such as MV-HEVC [6]. In our experiments, the stream is encoded using the prediction structure depicted in

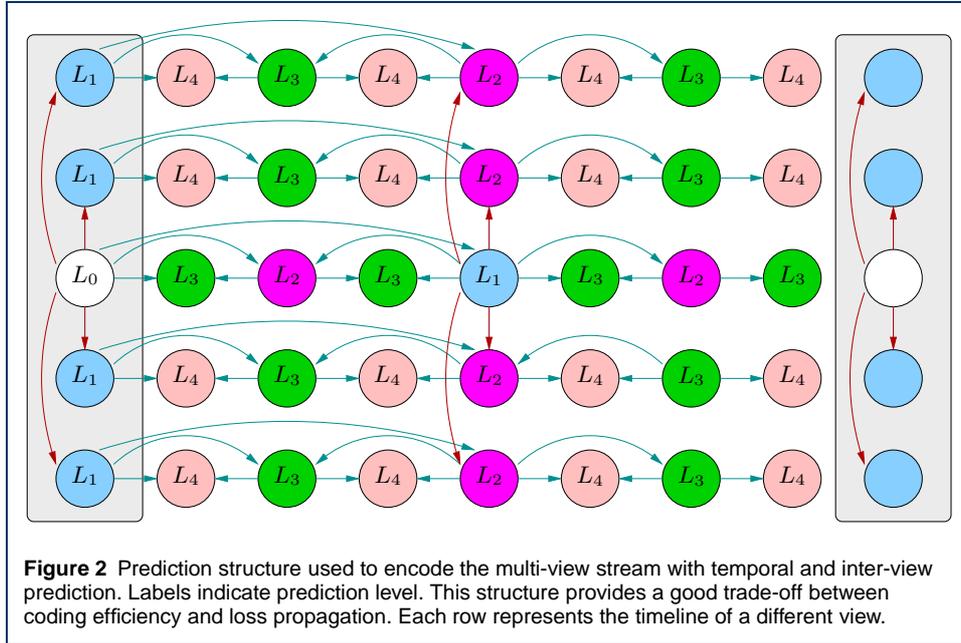
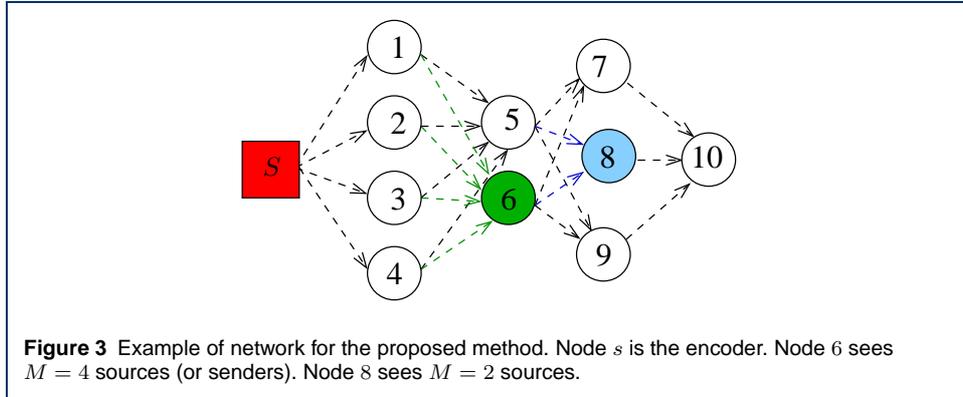


Fig. 2, with  $M = 5$  views and  $N = 8$  pictures per view in a GOP. This structure is a compromise between view progressive and fully hierarchical MVC that uses inter-view prediction in order to achieve a better coding efficiency, but is not fully hierarchical in order to reduce the dependencies among the frames, thus reducing the propagation of the effects of losses. However, it should be noted that our study can easily be extended to other coding techniques and prediction structures of multi-view content.

- For the user's preferences, we assume that the choice of the preferred view for each user follows the same, known distribution. Notice that, even though the proposed method could be applied to any preference model, how the learning and keeping track of the preference distribution is performed is outside the scope of this article, and shall not be addressed in the following. However, these preference may be easily learned and spread over the network with approaches similar to those shown in [17].
- We assume that the preference distribution does not change too fast over time, that is, we assume that it can be considered valid for at least the duration of a GOP, defined as an independently decodable set of  $N \times W$  frames, as depicted in Fig. 2. This implies that our system is able to work even when users' preferences change as frequently as once per GOP, which typically lasts less than one second. Any change in preferences during a GOP will be taken into account at the next GOP.

An example of the complete system is shown in Fig. 3. The video server  $S$  sends the encoded video packets together with side information about RD characteristics of the sequence. Nodes 1 to 9 relays the video using the proposed system.

We focus on a given node receiving the video sequence from  $M$  sources (or senders), performing network coding and relaying the video to downlink nodes. For example Node 6 sees  $M = 4$  sources, *i.e.* nodes 1 to 4. Node 8 sees  $M = 2$  sources, *i.e.* nodes 5 and 6. We propose an algorithm to decide the order of inclusion of frames in the coding window. We assume (for simplicity) that nodes do not compete for capacity but the available capacity



$N$	$\in \mathbb{N}$	No. of views
$W$	$\in \mathbb{N}$	No. of frames per view in a GOP
$\vec{B}$	(set)	Bi-dimensional frame buffer
$\vec{p}$	$\in [0, 1]^N$	Users' preferences distribution
$M$	$\in \mathbb{N}$	No. of senders
$C_m$	$[0, 1]$	Capacity of channel $m$ as a fraction of the stream rate
$\varepsilon_m$	$[0, 1]$	Packet loss rate of channel $m$
$\mathcal{W}$	(set)	Coding window
$r$	$\in \mathbb{N}$	Current size of the coding window
$D$	$\in \mathbb{R}$	Expected total distortion
$D_v$	$\in \mathbb{R}$	Distortion of view $v$

**Table 1** Summary of the notation used in this article.

may be less than the video coding bit-rate. We model each channel's capacity as a percentage of the encoded video bit-rate. and that each node has view preferences according to a given probability distribution.

## 4 Proposed Method

In this section we describe our proposed method of network encoding for a wireless streaming of multi-view video content based on the users' preferences.

As we mentioned in Sec. 1, most practical implementations of NC are achieved by segmenting the data flow into generations and combining only packets belonging to the same generation. Packets are made of the same length by padding. All packets in a generation are jointly decoded as soon as enough linearly independent combinations have been received, by means of linear system solving. Since the coefficients are taken from a finite field, perfect reconstruction is assured.

It has been proposed [29] 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 may be undesirable 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. In our technique, we use an implementation of random linear network coding referred to as *Expanding Window Network Coding* (EWNC) [28, 32]. 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 *innovation* (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. In our previous work, we already successfully applied EWNC principles to multi-view streaming in the context of wireless networking [44], but we did not take into account the preferences of the users in terms of displayed view.

As mentioned in Sec. 1, in other works user preferences were used to optimize the rate allocation in the encoding process. Here, we show how they can be used to decide which parts of the content have to be included in the coding window in order to optimize the rate-distortion properties of the transmitted stream.

We model the distribution of users' preferences with a probability vector  $\vec{p}$ , such that  $p_v$  is the probability that a member of the group chooses to watch view  $v \in \{1, \dots, N\}$  for the current GOP.

In our case, the transmitted packets will contain linear combinations of frames belonging to the same GOP. In order to select the order in which the frames will be included in the coding window, which we denote by  $\mathcal{W}$ , we proceed as follows. For each GOP, all the frames of the current GOP are stored in a bi-dimensional frame buffer  $\vec{B}$ , with  $N$  rows, and  $W$  columns, where  $N$  is the number of views and  $W$  is the per-view time-length of the GOP. For clarity, a summary of the notation used in this article is given in Tab. 1. The maximum possible size of the coding window, i.e., the generation size, will be the size of the GOP  $NW$ , while the current size of the coding window will be denoted  $r \leq NW$ .

The organization of the bi-dimensional buffer corresponding to the prediction structure described in Sec. 3 and depicted in Fig. 2 is shown in Fig. 4. Notice that the views are re-arranged to reflect the coding order, so the central view of Fig. 2 corresponds to view 1 in Fig. 4, as the other views are predicted upon it.

---

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

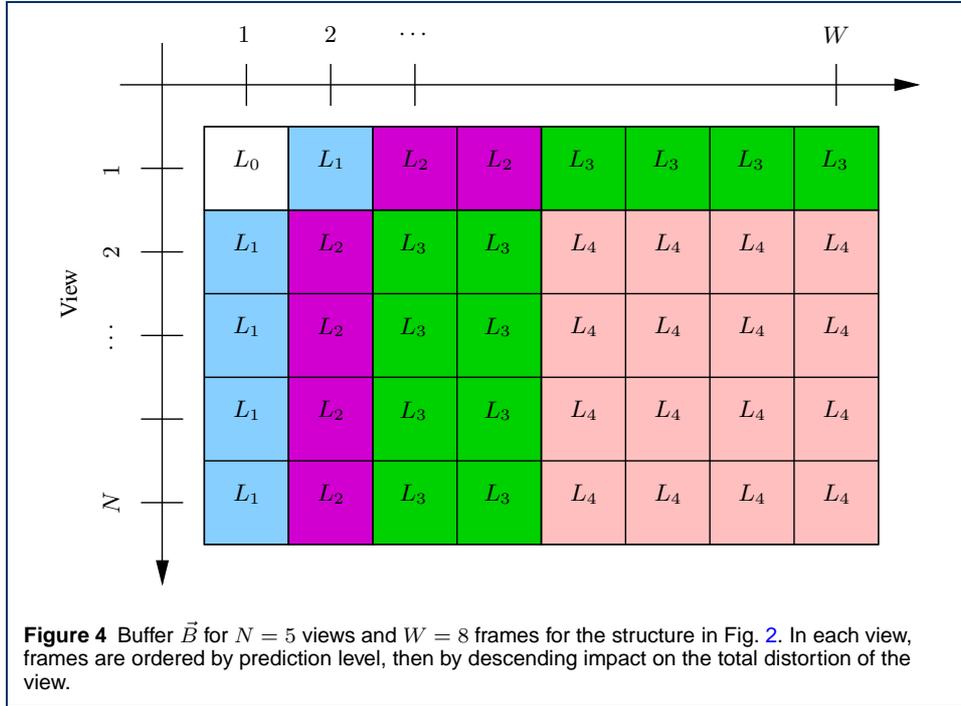
---

```

1: procedure SCHEDULEFRAMES
2:    $G \leftarrow N \times W;$  ▷ Size of the generation.
3:   for all MV-GOPs do
4:      $\mathcal{W} \leftarrow \emptyset;$  ▷ Coding window.
5:     for  $r \leftarrow 1$  to  $G$  do
6:        $\mathcal{F} \leftarrow \{f \in \vec{B} \mid \text{REF}(f) \subseteq \mathcal{W} \wedge f \notin \mathcal{W}\};$ 
7:        $J^* \leftarrow \min_{f \in \mathcal{F}} \{J_f = D(\mathcal{W} \cup \{f\}) + \lambda R(\mathcal{W} \cup \{f\})\};$ 
8:        $f^* \leftarrow$  a random frame in  $\{f \mid J_f = J^*\};$ 
9:        $\mathcal{W} \leftarrow \mathcal{W} \cup \{f^*\};$ 
10:    end for
11:  end for
12: end procedure

```

---



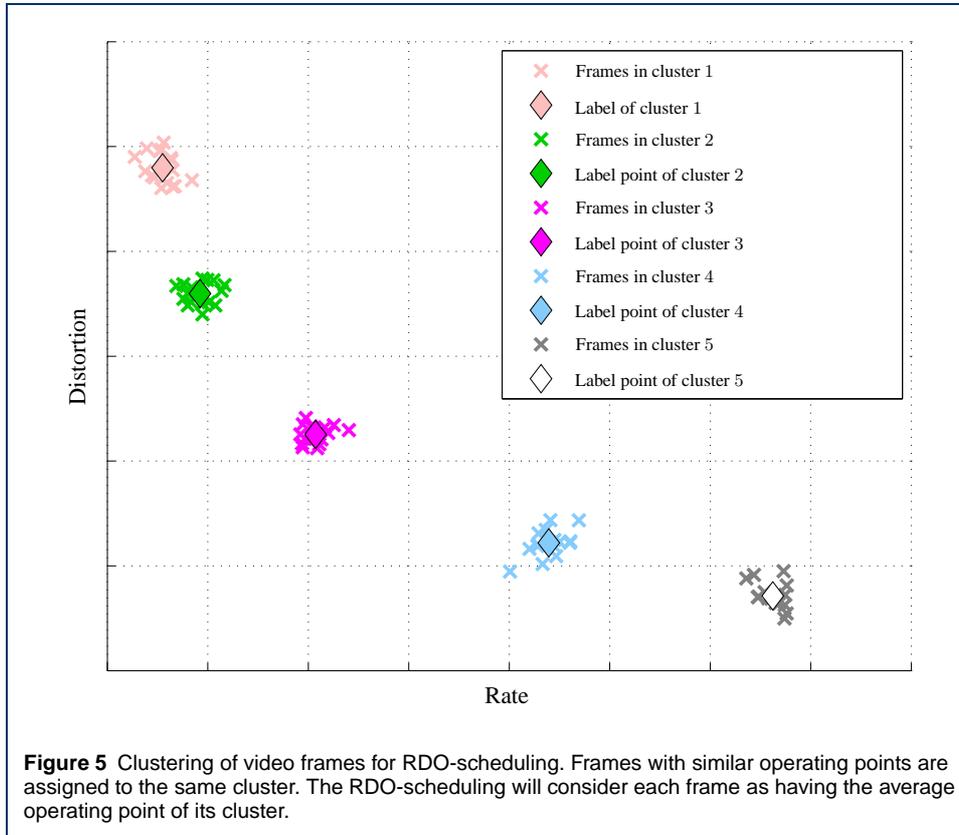
The scheduling algorithm 1 aims to minimize the expected total distortion given a number  $r \leq NW$  of frames to be included in the coding window  $\mathcal{W}$ . It works in an iterative fashion, starting with an empty coding window ( $\mathcal{W} \leftarrow \emptyset$ ), and adding at each iteration the most suitable frame to  $\mathcal{W}$ , given the frames already included at the previous iteration. More precisely, let  $D_v(\mathcal{W})$  be a function that computed the distortion of the view  $v$  when the frames selected in  $\mathcal{W}$  are available. Note that, due to the inter-view prediction, the functions  $D_v(\cdot)$  depend on all the selected frames. For a generic user, the expected total distortion  $D$  is expressed as:

$$D(\mathcal{W}) = \sum_{v=1}^N p_v D_v(\mathcal{W}) = \vec{p}^T \vec{D}(\mathcal{W}), \tag{1}$$

where vector  $\vec{D}(\mathcal{W})$  is such that its  $v$ -th component is  $D_v(\mathcal{W})$ . The optimization problem can therefore be stated as:

$$\mathcal{W}^*(r) = \arg \min_{\mathcal{W}} \left\{ \vec{p}^T \vec{D}(\mathcal{W}) \right\} \quad \text{s.t.: } \|\mathcal{W}\| \leq r \tag{2}$$

Algorithm 1 provides an heuristic way to compute  $\mathcal{W}^*(r)$  for all  $r \in \{1, 2, \dots, NW\}$  with the additional constraint that for all  $r$ ,  $\mathcal{W}^*(r - 1)$  is a subset of  $\mathcal{W}^*(r)$ , that is,  $\mathcal{W}^*(r)$  is build by adding a frame to  $\mathcal{W}^*(r - 1)$ . In general, the optimal solution to this problem is unique. This means that all the senders would always compute exactly the same scheduling order. As a consequence, the "randomness" of NC would be lost: all the senders always transmit dependent combinations. Even if a node receives packets from  $M > 1$  senders, they will be identical, 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.



In order to solve this problem, we propose a *clustering* of the video frames: the clustering is a classification of the 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 labeled with the average rate and distortion of its cluster, possibly quantized. Notice that clustering is performed using the definition of distortion given in Eq. (1), *i.e.*, by taking the preferences into account. An illustration of the clustering is given in Fig. 5. We observe that frame or packet classification based on RD properties has been used in the literature, for example by Chakareski and Frossard [45]. However, beside the differences in computing or estimating rate and distortion, we use classification in a totally different way, *i.e.* for achieving a scheduling diversity to be used in network coding. This concept is original in scientific literature.

The labels are decided *only once* at the encoder side, where rate and distortion are known with negligible computational overhead and where a best estimation of the receiver's preferences is more likely to be available. Since the encoder knows the rate and distortion characteristics of the frames, it can send them to the users with a very little overhead, since this information amount to a few bytes per frame.

Let us now describe in detail how the algorithm 1 works in an intermediate node. For all MV-GOPs, the node computes the coding window, starting with an empty set and adding at each sending opportunity a new frame. For a new value of  $r$ , first we compute the set of eligible frames  $\mathcal{F}$ . It is made up of those whose references for prediction, if any, are already in the coding window. For example, when  $\mathcal{W} = \emptyset$ , only the Intra frame of the GOP is eligible. Therefore for all the nodes, when  $r = 1$ ,  $\mathcal{W}$  only contains the Intra frame. In

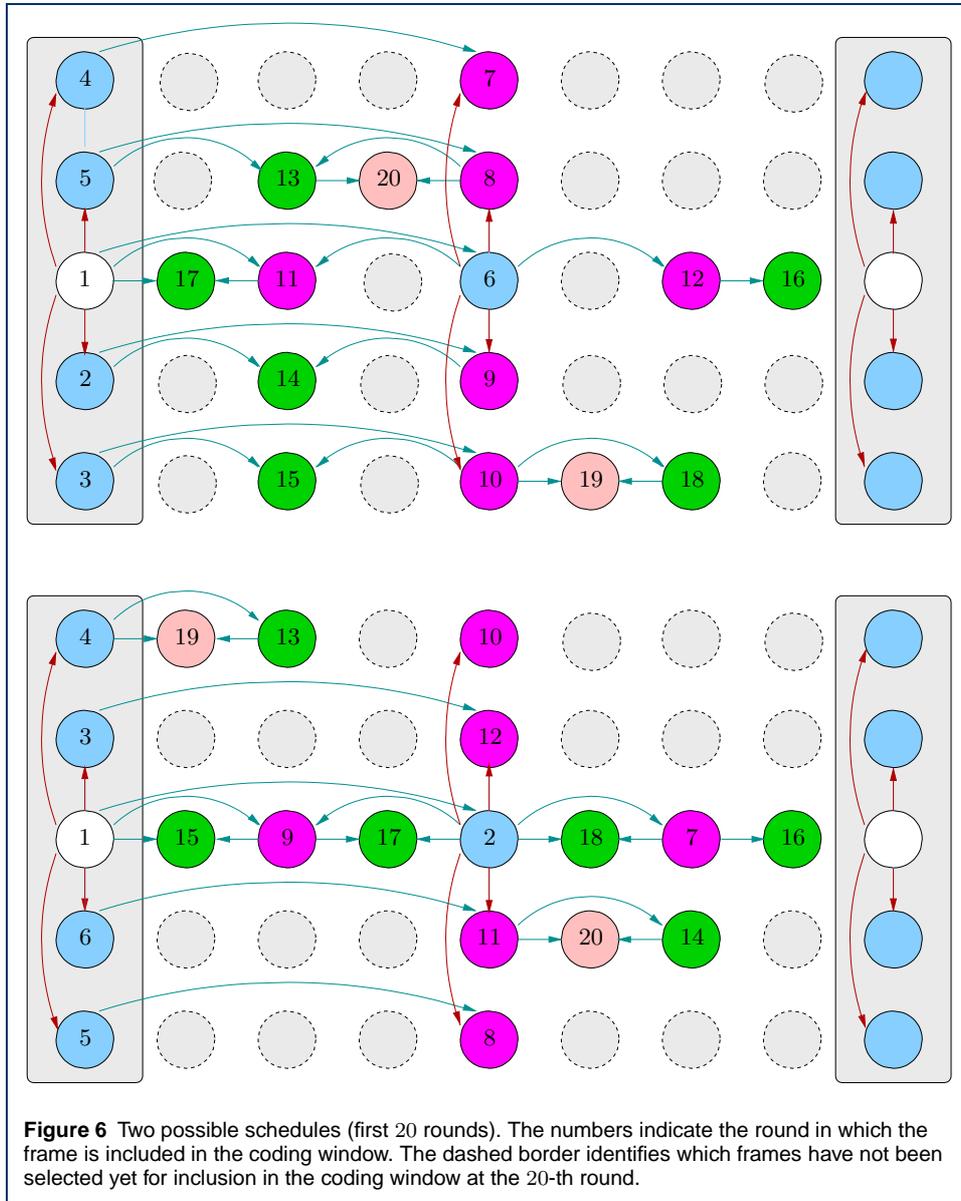
general however,  $\mathcal{F}$  contains all the frames that are decodable using only the frame in  $\mathcal{W}$ , and that are not in  $\mathcal{W}$  (algorithm 1, line 6), that is, for the second iteration, all the frames of level  $L_1$  with respect to Fig. 4. For each frame in  $\mathcal{F}$ , the algorithm computes the coding cost function  $J_f$  obtained by adding  $f$  to  $\mathcal{W}$ . Without clustering, generally speaking a unique frame  $f$  would minimize  $J_f$ , making it impossible to produce different scheduling at different nodes. However, with clustering, several frames are labeled with the same, fictive values of rate and distortion, even though they do not correspond to the actual rate and distortion, see Fig. 5. These frames will produce the same value of  $J_f$  (algorithm 1, line 7). Therefore, the set of frames that achieve the minimal value of  $J_f$  will in general be composed of several frames. As a consequence, each node can pick a random frame in this set (line 8) to be added into  $\mathcal{W}$  (line 9). This step introduces the scheduling diversity needed by NC.

As far as the choice of the value for  $\lambda$  (step 7) is concerned, as in classical RD optimization problems, it depends on the target coding rate [46]. In principle each node could adjust this value according to its knowledge about the downlink channel capacities. However, in our simulation we assume for simplicity that each node uses the same Lagrangian parameter used by the encoder (this value is deduced from the QP and do not need to be transmitted).

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.

As far as the computational complexity of the scheduling Algorithm 1 is concerned, we observe that, for a given MV-GOP, steps 4 to 10 are executed. The complexity of this part is dominated by the minimization of the cost function  $J$  (step 7), which is executed  $G = NW$  times. This minimization is performed by exhaustion: for any candidate frame  $f \in \mathcal{F}$ , we compute the cost  $J_f = D + \lambda R$ . As mentioned before, the rate distortion characteristics of the sequence are computed once at the encoder as side-product of the compression process, and may be sent as side information to nodes with negligible overhead. Therefore, the complexity of step 7 is dominated by one multiplication per candidate frame. Since at any iteration over  $r$  the number of candidate frames cannot be larger than  $NW$ , the minimization complexity is at most  $NW$  per value of  $r$  and per MV-GOP. Since  $NW$  values of  $r$  are considered, the complexity of the scheduling algorithm is dominated by at most  $N^2W^2$  multiplications per MV-GOP. With the configuration used in our simulation setup, this amount to 5000 multiplications per second, which is assumed to be negligible with respect to other tasks of each node (e.g. video decoding for display).

A key point in this algorithm is the labeling of frames with fictive rate and distortion values. If we cluster many frames with the same label, we increase the chance of different nodes selecting different schedules, thus reducing the case of linear dependent packets in NC. On the other hand, large clusters increase also the chances of having RD labels that differ significantly from the actual RD values. This implies an RD-suboptimal scheduling. In conclusion, the clustering must be carefully performed, taking into account the expected similarity of RD values among different frames. A simple clustering scheme is to assign all the frames on the same prediction level to the same 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, and is the approach that we have followed in our experiments. If the corresponding frames of different views have unbalanced properties, then a more sophisticated scheme can be employed.



#### 4.1 A running example

An example of two different scheduling orders is presented in Fig. 6. For the sake of simplicity, only the scheduling for the first 20 packets is presented. We observe that since clustering has been performed at prediction level, when Algorithm 1 is run, at step 8 any frame of a given prediction level can be selected.

In this example we show how the algorithm could run within nodes 5 and 6 in Fig. 3. Their coding windows are depicted respectively in the top and bottom parts of Fig. 6. In this case, a receiver such as node 8 would see  $M = 2$  sources (or senders).

Let  $f(v, k)$  be the  $k$ -th frame in display order of view  $v$ , with the views denoted top to bottom as  $1, 2, \dots, 5$  so that  $v = 3$  is the central view.

In the first round, senders 5 and 6 each consider prediction level  $L_0$  only. As a result, they have an identical coding window containing only the I-frame of the central view —which

is the only frame in the cluster of prediction level  $L_0$ :  $\mathcal{W}_{1,5} = \mathcal{W}_{1,6} = \{f(3, 0)\}$ , where  $\mathcal{W}_{r,k}$  is the coding window of sender  $k$  at round  $r$ .

In the second round, senders 5 and 6 each randomly select a frame from prediction level  $L_1$ , likely a different one. Let us for example assume that  $\mathcal{W}_{2,5} = \{f(3, 0), f(4, 0)\}$  and  $\mathcal{W}_{2,6} = \{f(3, 0), f(3, 4)\}$ .

In the following rounds, both senders keep adding a random frame from the cluster of prediction level  $L_1$  to their coding windows, until no frame is left to be selected:

$$\begin{aligned} \mathcal{W}_{3,5} &= \mathcal{W}_{2,5} \cup \{f(5, 0)\} & \mathcal{W}_{3,6} &= \mathcal{W}_{2,6} \cup \{f(2, 0)\} \\ \mathcal{W}_{4,5} &= \mathcal{W}_{3,5} \cup \{f(1, 0)\} & \mathcal{W}_{4,6} &= \mathcal{W}_{3,6} \cup \{f(1, 0)\} \\ \mathcal{W}_{5,5} &= \mathcal{W}_{4,5} \cup \{f(2, 0)\} & \mathcal{W}_{5,6} &= \mathcal{W}_{4,6} \cup \{f(5, 0)\} \end{aligned}$$

Eventually, both senders will have included the whole cluster of frames of prediction level  $L_1$  in their coding windows, which would therefore be again identical:  $\mathcal{W}_{6,5} = \mathcal{W}_{6,6} = \{f(1, 0), f(2, 0), f(3, 0), f(3, 4), f(4, 0), f(5, 0)\}$ .

On the receiver side (node 8), let us consider the set  $\mathcal{U}_r$  of the decodable frames received by the end of round  $r$ .

In the first round, since the coding windows of the two senders are identical, the receiver only obtains one decodable frame,  $\mathcal{U}_1 = \{f(3, 0)\}$ .

Then, since the schedule of 5 and 6 diverge, the receiver starts obtaining on average more than one new decodable frame per round:

$$\begin{aligned} \mathcal{U}_2 &= \mathcal{U}_1 \cup \{f(4, 0), f(3, 4)\} \\ \mathcal{U}_3 &= \mathcal{U}_2 \cup \{f(5, 0), f(2, 0)\} \\ \mathcal{U}_4 &= \mathcal{U}_3 \cup \{f(1, 0)\} \end{aligned}$$

Eventually, the receiver is able to decode the whole prediction level  $L_1$  (and prediction level  $L_0$ , which is composed of  $f(3,0)$  alone). As a consequence, the following packets received from 5 and 6 will not be innovative, meaning that they are linear combinations of the packets in  $\mathcal{U}_4$  and do not increase its rank:  $\mathcal{U}_6 = \mathcal{U}_5 = \mathcal{U}_4$ . However this redundancy is effective against packet losses.

The same algorithm is applied for subsequent prediction levels until the whole GOP is transmitted.

More in general, once the order of inclusion has been selected, each node generates a set of  $NW$  mixed packets by applying EWNC, while the original stream will be discarded. When a node receives a request from one of its neighbors to stream the content, it will answer with as many combination packets as its capacity allows. The receiver will then collect all the packets it receives from its neighbors and try to decode as many video frames as possible. It will then select a view of the content, and display the relative decoded frames, achieving a video quality depending on the frames it received and the view it selected. The node will also in turn generate new combinations to contribute to future requests.

In this description of the scheduling algorithm, we assumed for the sake of simplicity that each view frame fits in one packet. However, the algorithm is immediately generalized to the case when any other data structure is used, provided that it is possible to determine its rate, distortion and coding dependencies. For example, slices could be used; or more than one frame can be included in the same packet. Given these three pieces of information, Algorithm 1 can be run on any coded data structure.

## 5 Experimental Results

In this section, we present the results of the proposed technique and compare them with three different reference techniques. The simulation scenario is the one depicted in Fig. 1 of Sec. 3.

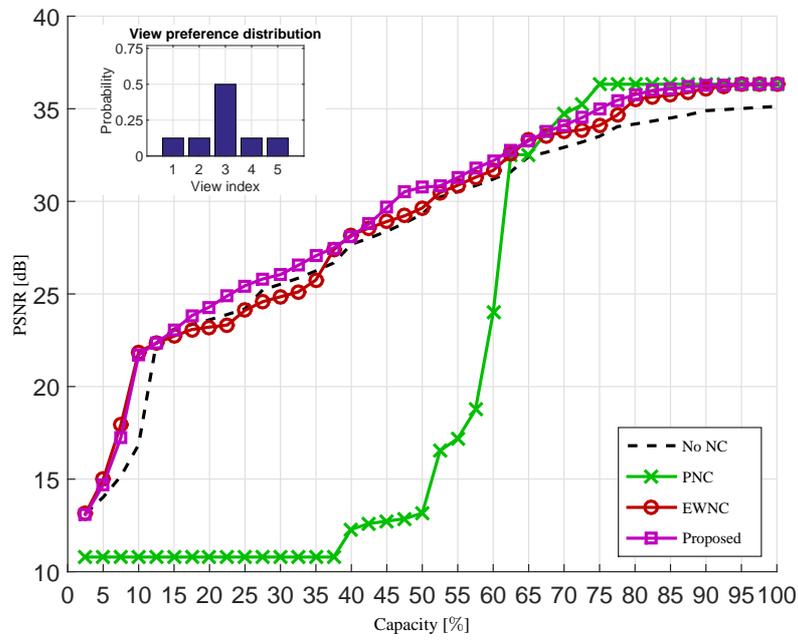
The receiver  $R$  is trying to obtain the multi-view content from its neighbors  $S_m, m \in \{1, \dots, M\}$ , referred to as senders or sources, each connected to the receiver with a channel having capacity  $C_m$ . The channel may be a PEC with packet loss rate  $\varepsilon_m$  or a GE characterized by  $\{\varepsilon_G, \varepsilon_B, p_{GB}, p_{BG}\}$ . The receiver, at each GOP, randomly selects a view according to the probability distribution of  $\vec{p}$ . In a first experiment we will consider a PEC channel with a perfect knowledge of user preferences. Later we will show the results when a GE model is employed and when preference estimation is not perfect.

We have selected three reference techniques to be compared with our proposed technique.

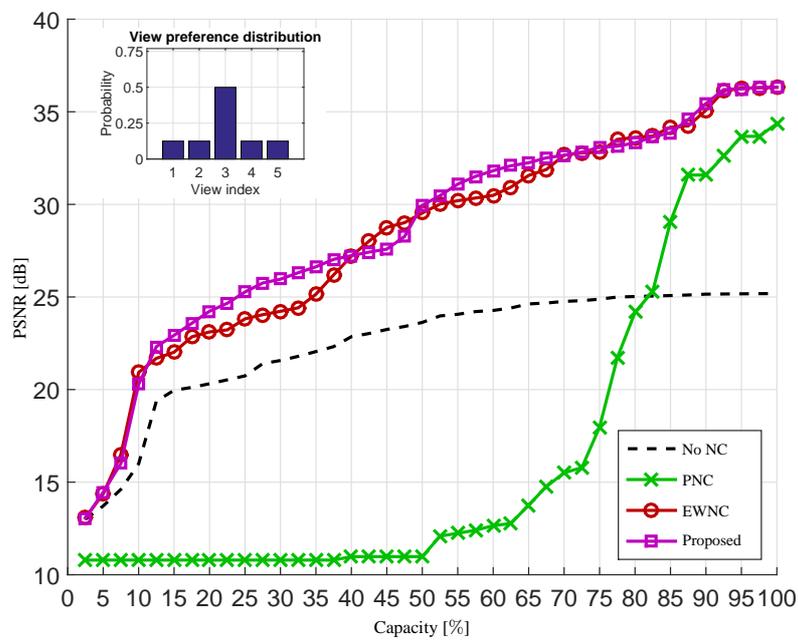
- The first reference uses EWNC and an RD-optimized scheduling to select the order of inclusion in the coding window. However, unlike our proposed technique, the preferences of the receiver are not taken into account and the expected distortion is measured simply as the mean of the distortion on the views. This is equivalent to assume a uniform distribution for  $\vec{p}$  independently from the actual distribution. This technique is practically equivalent to the one previously proposed by the authors in [44], and is labeled *EWNC* in the figures.
- The second reference uses Practical Network Coding [25] to transmit the stream. Since the senders are uncoordinated, they are not aware of the number of other senders or the capacities of their channels. Therefore, they use a coding window of the same size of the generation (*i.e.*, the same size as the GOP). This technique is labeled *PNC* in the figures. In our scenario each of the senders generates as many packets as it is the rank of the input generation *i.e.*, no redundancy is added by the senders. However, from the receiver side, the redundancy is inherent in having  $M > 1$  uncoordinated senders transmitting linear combinations of the same sender generation. So, in each of our scenarios, the redundancy is  $r = (M - 1)/M$ .
- The last reference does not use network coding, nor it is aware of the users preferences. This technique is inspired by classical replication schemes, such as the one proposed in [47], and is labeled *No NC* in the figures.

We used four common multi-view video sequences: “Ballet”, “Bookarrival”, “Break-dancers”, and “Doorflowers”. They have  $1024 \times 768$  pixels and 25 frames per second. We used 100 frames per view and the first 5 views per sequence, for a total of 2000 frames. They have been encoded in H.264/MVC using the GOP structure described in Sec. 3, and depicted in Fig. 2, with QPs 31, 34, 37, and 40. The corresponding coding rates range from 280 to 1570 kbps per view. The results presented are obtained averaging over at least 100 runs and over all the sequences.

We tested the system using four models of view preferences. In the first one, called “peaky” distribution, the central view has a given probability  $p_c$  and the other views share uniformly the residual probability. In a second one, called “triangular”, probabilities increase linearly from the left-most view to the central, then they decrease symmetrical up to the right-most view. A third model uses a discrete Gaussian-like distribution, where probability of view  $k$  is proportional to  $e^{-\frac{(k-c)^2}{2\sigma^2}}$ , where  $c$  is the index of central view. Finally we consider a “bimodal” distribution where two views, symmetrical with respect to the central



(a)  $M = 2$   $\epsilon = 10\%$   $p_c = 50\%$



(b)  $M = 2$   $\epsilon = 30\%$   $p_c = 50\%$

**Figure 7** Comparison of the average PSNR of the decoded sequences (2 sources, 50% probability of the receiver displaying the central view). Packet loss rates are 10% (left) and 30% (right). The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the PSNR is computed as the average over the views weighted by the preference probabilities.

View	1	2	3	4	5
Peaky	0.1250	0.1250	0.5000	0.1250	0.1250
Peaky	0.0625	0.0625	0.7500	0.0625	0.0625
Gaussian	0.0216	0.2284	0.5000	0.2284	0.0216
Gaussian	0.0006	0.1244	0.7500	0.1244	0.0006
Triangular	0.0000	0.2500	0.5000	0.2500	0.0000
Triangular	0.1000	0.2250	0.3500	0.2250	0.1000
Bimodal	0.0000	0.5000	0.0000	0.5000	0.0000
Bimodal	0.1000	0.3500	0.1000	0.3500	0.1000

**Table 2** View preference distributions used in the experiments.

one, have the same probability, and the remaining share the residual probability. We consider at least two cases for each of the models, ending up with the probability distributions shown in Tab. 2. All these distribution are characterized by a single parameter, the probability of the preferred view. We refer to this parameter as  $p_c$  even though in the Bimodal case this is not the probability of the central view. For each GOP, each user randomly selects a view according to the distribution of  $\vec{p}$ , decodes the corresponding frames, and measures the PSNR as

$$\text{PSNR}_w = 10 \log_{10} \frac{255^2}{\sum_v p_v D_v} \quad (3)$$

that is, the distortion is the weighted MSE described in Sec. 4). This PSNR is reported as a function of the channel capacity, which in turn is expressed as a percentage of the video stream rate.

The interesting use case is when the channel capacity is intermediate between a very low value (where the only possible strategy is to send the I-frame of the GOP) and high values, where any solution would work quite well. The results of these experiments are reported in the following.

We start by considering the peaky distribution. In Figure 7, we report a comparison with the reference techniques for a two senders scenario, PEC channel with packet loss rate of 10 % and 30 %. The probability of the receiver displaying the central view is  $p_c = 50$  %, while the other views are equally probable.

First of all, we observe that our proposed technique outperforms all of the reference techniques for the majority of the values of channel capacity, and has very similar performances in the remaining cases.

We also observe that, if no network coding is used, each received packet increases the PSNR. However, the transmission cannot recover from losses, thus the maximum quality is not achieved. The EWNC technique follows the same trend as the No NC technique, but with slightly better performance, due to the effects of NC that partially compensates for the losses.

Conversely, PNC eventually achieves the maximum quality, and can provide, for low loss rates slightly better video quality than the proposed technique within a range of channel capacities (about 70 % to 80 % of the stream rate for up to a 10 % loss rate ) but the receiver cannot decode almost any frame if the capacity of the senders is below a threshold of about 50 % of the stream rate. For higher loss probabilities, the PNC approach is even

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	0.86	1.10	4.95	6.08
PNC	7.16	7.76	9.93	12.29
EWNC	0.50	0.47	0.63	0.43

**Table 3** PSNR gain of the proposed technique with respect to the references, averaged over the channel capacity.  $M = 2$  sources, Peaky distribution,  $P_c = 50\%$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	0.75	1.15	5.10	6.05
PNC	7.02	7.78	9.71	12.06
EWNC	0.30	0.39	0.33	0.07

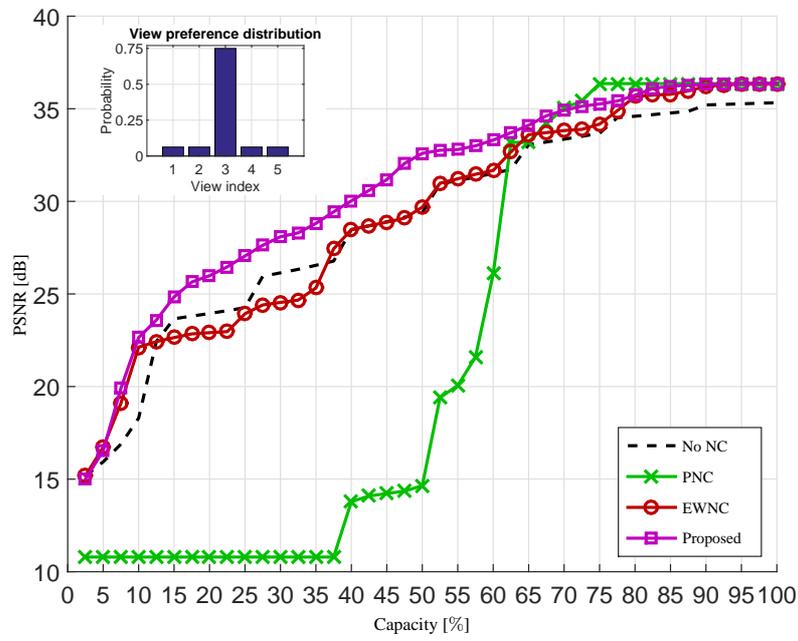
**Table 4** PSNR gain of the proposed technique with respect to the references, averaged on the channel capacity. Triangular preference distribution,  $P_c = 35\%$ .  $M = 2$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	1.61	1.91	5.65	5.33
PNC	7.77	8.42	10.48	11.59
EWNC	1.47	1.44	1.63	1.08

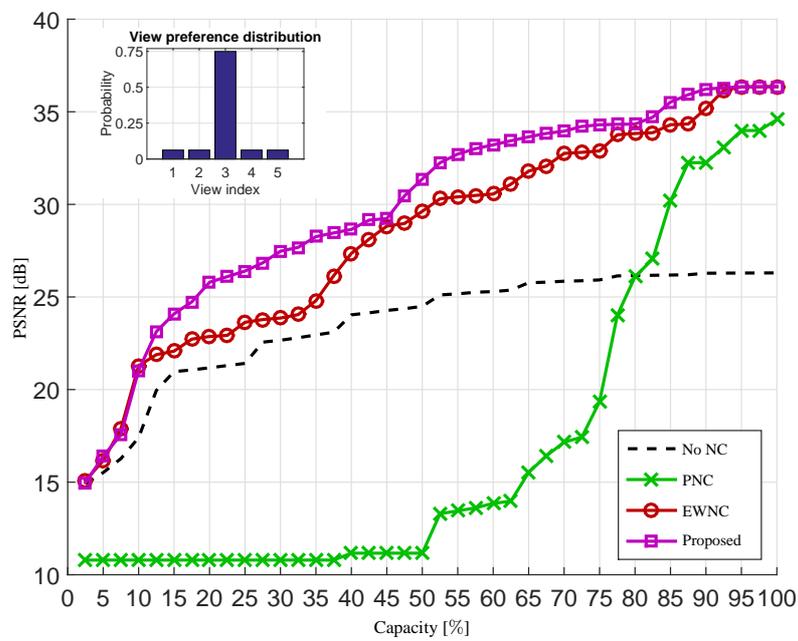
**Table 5** PSNR gain of the proposed technique with respect to the references, averaged on the channel capacity. Gaussian preference distribution,  $P_c = 50\%$ .  $M = 2$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	1.07	1.47	5.87	6.63
PNC	7.29	8.07	10.08	12.39
EWNC	0.38	0.46	0.50	0.19

**Table 6** PSNR gain of the proposed technique with respect to the references, averaged on the channel capacity. Bimodal preference distribution,  $P_c = 35\%$ .  $M = 2$



(a)  $M = 2 \quad \epsilon = 10\% \quad p_c = 75\%$



(b)  $M = 2 \quad \epsilon = 30\% \quad p_c = 75\%$

**Figure 8** Comparison of the average PSNR of the decoded sequences (2 sources, 75% probability of the receiver displaying the central view). Packet loss rates are 10% (left) and 30% (right). The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the PSNR is computed as the average over the views weighted by the preference probabilities.

more impaired, and is practically useless unless the channel capacity approaches the stream rate. The necessity of a high minimum capacity to achieve any acceptable quality is a very undesirable property in a wireless environment, as in a mobile scenario the channel conditions could rapidly become very harsh, leaving then the node with no useful data. Also, it is worth noticing that, as mentioned in Sec. 1, the rate of a multi-view stream can be several times larger than that of traditional, single-view, stream. The mobile nodes are therefore likely to have an uplink capacity that is only a small fraction of the multi-view stream rate.

In Tab. 3 we reported the PSNR gain of the proposed technique with respect to the three reference techniques averaged along the channel capacity and for different values of packet loss probability. We observed that the proposed technique outperforms in average all the references, even though in this configuration EWNC achieves a close performance. In the following tables from 4 to 6, we show the result for similar experiments where we just change the preference probabilities. Gains are even larger for distribution other than the Peaky one.

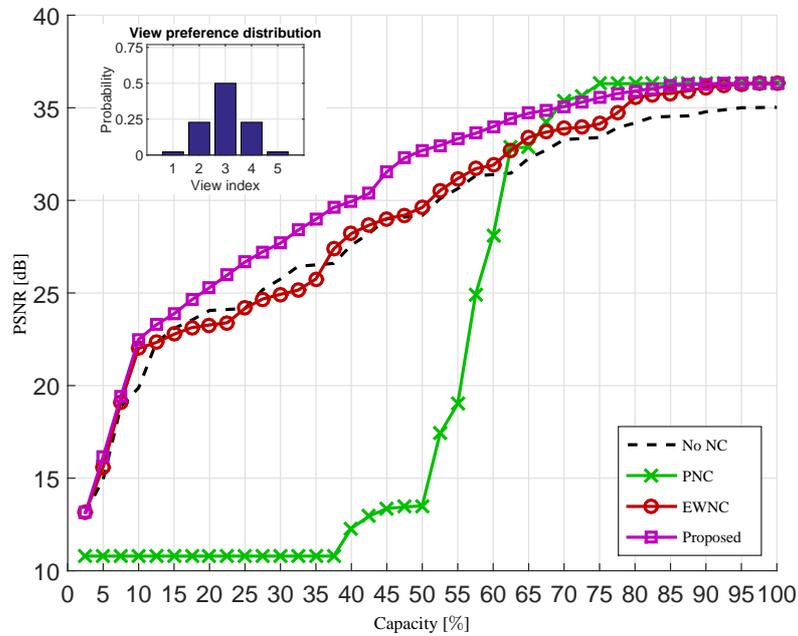
In Figure 8, we present the results for the same number of senders and the same packet loss rates, when the probability of the receiver of displaying the central view is  $p_c = 75\%$ , while the other views are equally probable. We also reported in Tab. 7 the average PSNR gains of the proposed technique with respect to the references. Tables from 8 to 10 report results for the other distributions, again in the case where the preferred view has a large probability. The proposed techniques reports consistent gains in all these configurations.

As we can see both from the table and the figure, while the performance of the proposed technique and of PNC stay almost unaltered, the performance of the EWNC and of the No NC techniques drop visibly. This can be explained by the following observations: the proposed technique adapts its coding window inclusion order to the distribution of the preferences, thus producing a scheduling quasi-optimal with respect to the preferences no matter what these are. On the other hand, the PNC technique imposes joint decoding of the whole generation, so the order of inclusion is irrelevant. Finally, both No NC and EWNC do use an RD optimized scheduler to decide their order of transmission and inclusion (respectively), but since they do not take into account the receiver's preferences, their estimation of the expected distortion is incorrect, resulting in a sub-optimal order. In fact, by averaging the PSNR over the views, these two models implicitly assume a uniform distribution of preferences. We can therefore expect that their performance will be the less effective the less the preference distribution resemble a uniform distribution, which is what we observed experimentally.

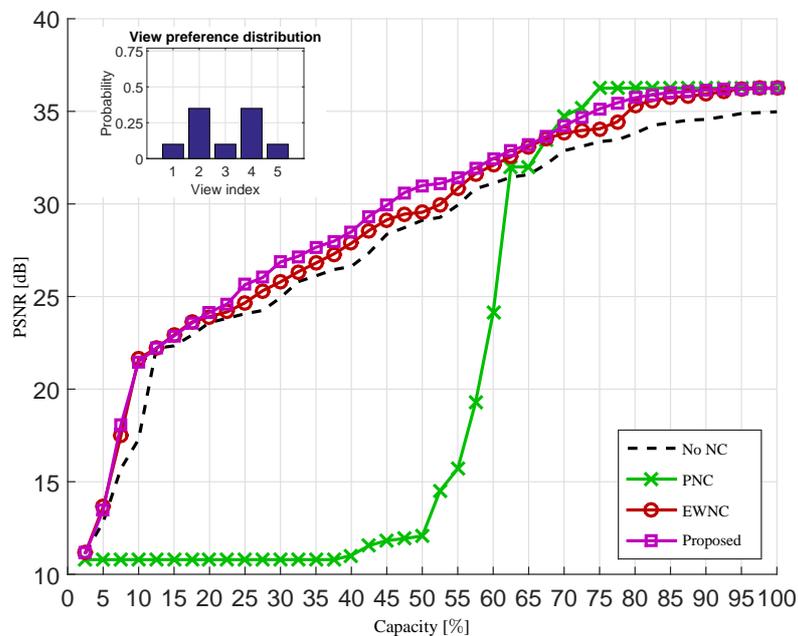
This is confirmed by using the other preference distributions, as also shown in Figures 9(a) and 9(b).

In conclusion, with  $M = 2$  sources, the proposed technique performs largely better than PNC and No NC, especially when the channel conditions are harsh (high loss rate, small capacity) and the preferences are skewed. It keeps a smaller gain over EWNC, around 0.5 dB when  $p_c = 50\%$  and 1.5 dB when  $p_c = 75\%$  for the Peaky distribution, and higher for others.

In Figures 10 and 11 we present analogous results for  $M = 4$  sources and for respectively  $p_c = 50\%$  and  $p_c = 75\%$ . Likewise, Tab. 11 and 12 present the averaged PSNR over channel capacity at several loss rates. Similar results are obtained for the other distributions. We do not report them for the sake of brevity. However, as for the previous case, the Peaky distribution is the least favorable to our technique.



(a)  $M = 2$   $\epsilon = 10\%$   $p_c = 50\%$



(b)  $M = 2$   $\epsilon = 30\%$   $p_c = 35\%$

**Figure 9** Comparison of the average PSNR of the decoded sequences (2 sources, packet loss rate 10%). View preference distributions are Gaussian and Bimodal, with maximum probability  $p_c$  equal to 50% (top) and 35% (bottom). The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the PSNR is computed as the average over the views weighted by the preference probabilities.

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	1.48	1.72	4.85	6.27
PNC	7.78	8.36	10.48	12.91
EWNC	1.45	1.47	1.60	1.53

**Table 7** PSNR gain [dB] of the proposed technique with respect to the references, averaged on the channel capacity. Peaky preference distribution,  $P_c = 75\%$ ,  $M = 2$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	2.51	2.87	5.09	2.92
PNC	8.14	8.98	9.56	9.02
EWNC	2.48	2.52	1.45	0.86

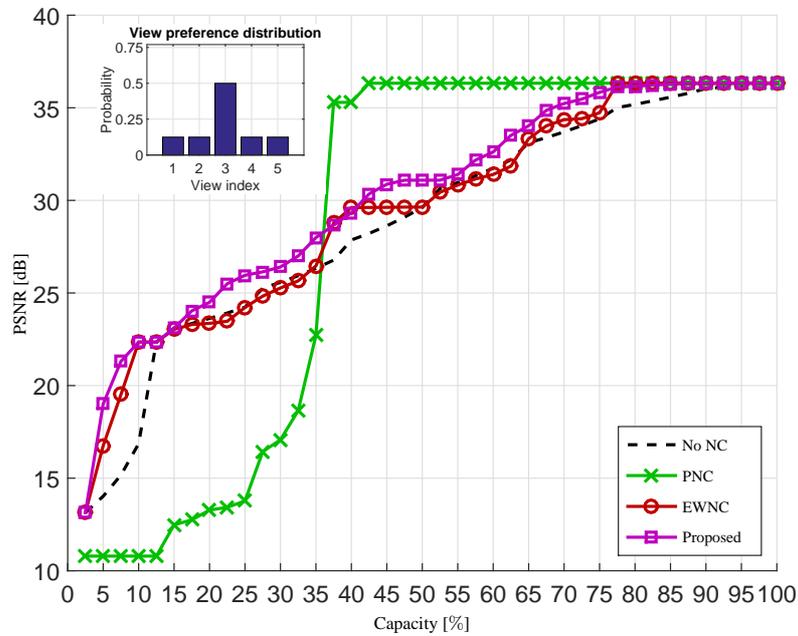
**Table 8** PSNR gain of the proposed technique with respect to the references, averaged on the channel capacity. Triangular preference distribution,  $P_c = 50\%$ ,  $M = 2$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	2.76	2.73	4.94	4.22
PNC	8.48	8.87	10.05	10.67
EWNC	2.94	2.66	2.11	0.23

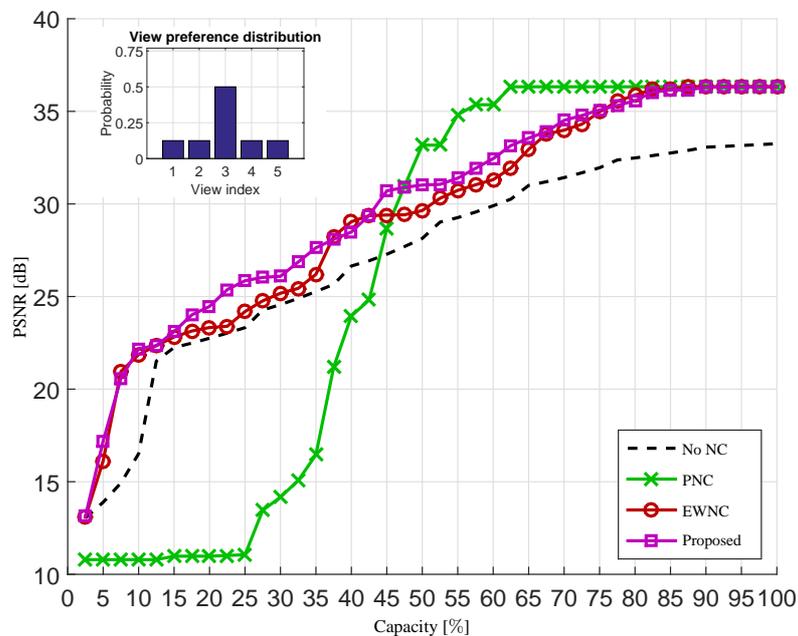
**Table 9** PSNR gain of the proposed technique with respect to the references, averaged on the channel capacity. Gaussian preference distribution,  $P_c = 75\%$ ,  $M = 2$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	3.05	2.49	5.29	4.58
PNC	8.57	8.63	9.04	10.38
EWNC	2.49	1.69	0.49	0.20

**Table 10** PSNR gain of the proposed technique with respect to the references, averaged on the channel capacity. Bimodal preference distribution,  $P_c = 50\%$ ,  $M = 2$

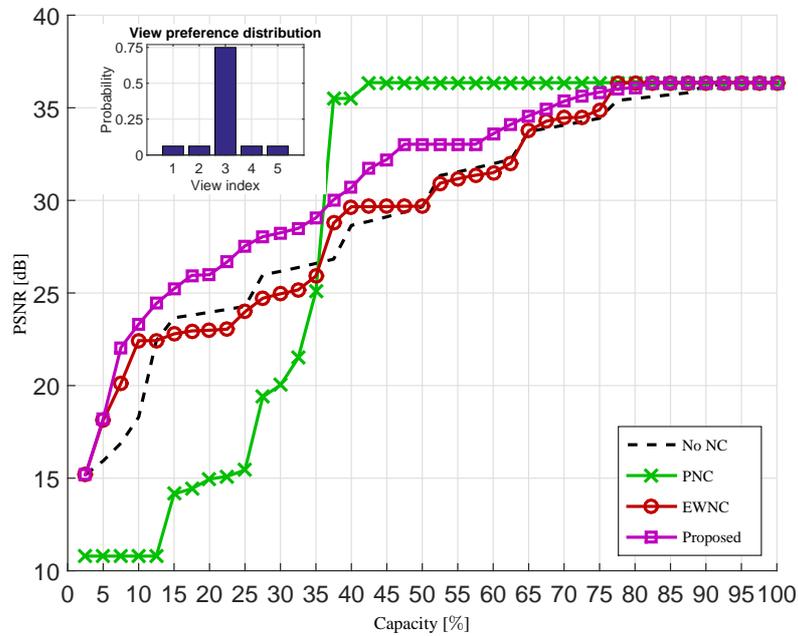


(a)  $M = 4$   $\epsilon = 10\%$   $p_c = 50\%$

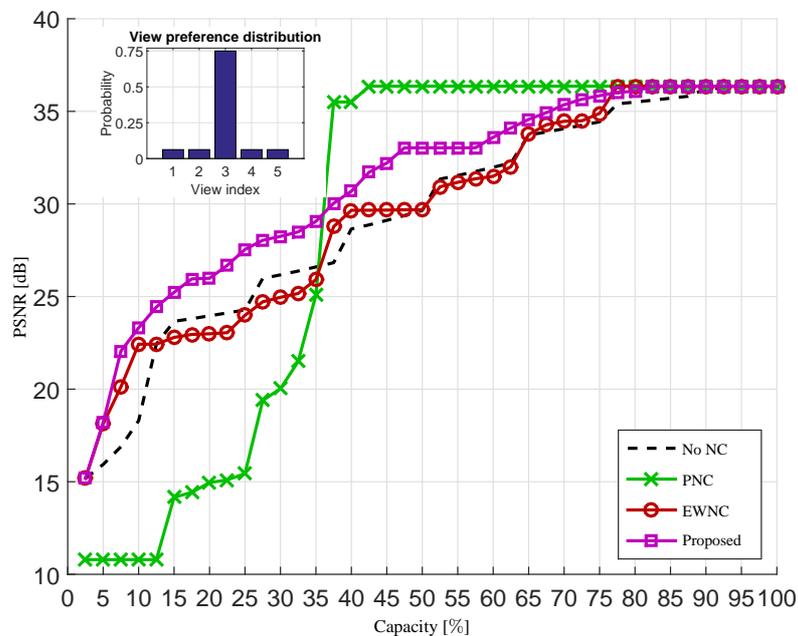


(b)  $M = 4$   $\epsilon = 30\%$   $p_c = 50\%$

**Figure 10** Comparison of the average PSNR of the decoded sequences (4 sources, 50 % probability of the receiver displaying the central view). Packet loss rates are 10 % (left) and 30 % (right). The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the PSNR is computed as the average over the views weighted by the preference probabilities.



(a)  $M = 4$   $\varepsilon = 10\%$   $p_c = 75\%$



(b)  $M = 4$   $\varepsilon = 30\%$   $p_c = 75\%$

**Figure 11** Comparison of the average PSNR of the decoded sequences (4 sources, 75% probability of the receiver displaying the central view). Packet loss rates are 10% (left) and 30% (right). The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the weighted PSNR is computed as the average over the views weighted by the preference probabilities.

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	1.22	1.32	1.20	2.63
PNC	0.95	1.78	2.46	3.60
EWNC	0.41	0.72	0.61	0.53

**Table 11** PSNR gain of the proposed technique with respect to the references, averaged over the channel capacity.  $M = 4$ ,  $P_c = 50\%$

	$\varepsilon = 5\%$	$\varepsilon = 10\%$	$\varepsilon = 20\%$	$\varepsilon = 30\%$
NO NC	1.85	1.76	1.77	3.02
PNC	1.59	2.14	2.98	4.07
EWNC	1.42	1.52	1.54	1.46

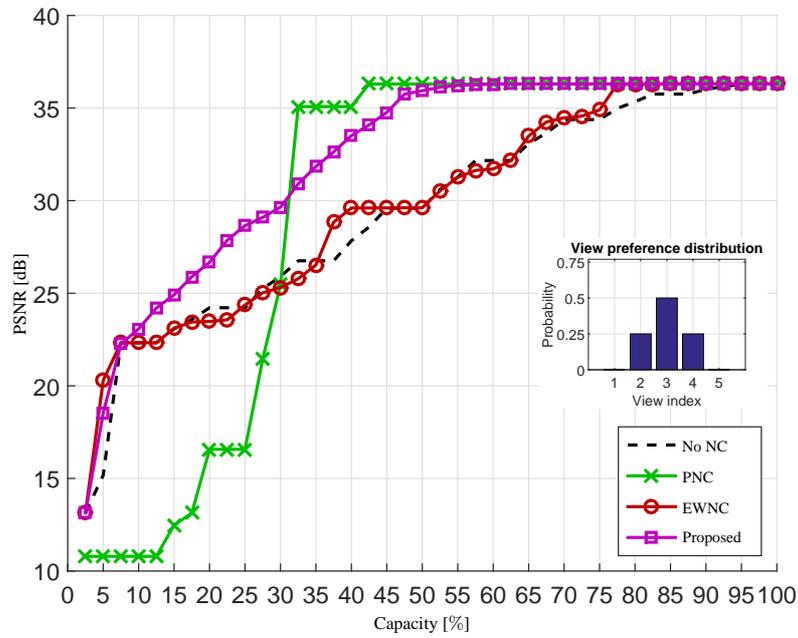
**Table 12** PSNR gain of the proposed technique with respect to the references, averaged over the channel capacity.  $M = 4$ ,  $P_c = 75\%$

We observe that increasing the number of sources improves the performance of all techniques. However, the greatest effect is visible in the PNC technique as, having the largest coding window for any value of capacity, it is the one that benefits most from the diversity of the received packets. This translates in a reduction of the minimum capacity needed to achieve an acceptable quality and the minimum capacity needed to achieve the same quality as the proposed technique (and eventually surpass it). These minimum capacities of course are affected negatively by the packet loss rates, and the range of capacities in which PNC can outperform the proposed technique narrows further when the distribution of the receiver's preferences is less uniform (Fig. 11). Similar results are obtained for other distribution, as shown in Fig. 12. Moreover, the proposed technique still provides globally better performance than the reference ones, as shown in Tab. 11 and 12. As expected, the gains are larger for less uniform preference distributions, and the gain with respect to PNC increases with the packet loss ratio  $\varepsilon$ .

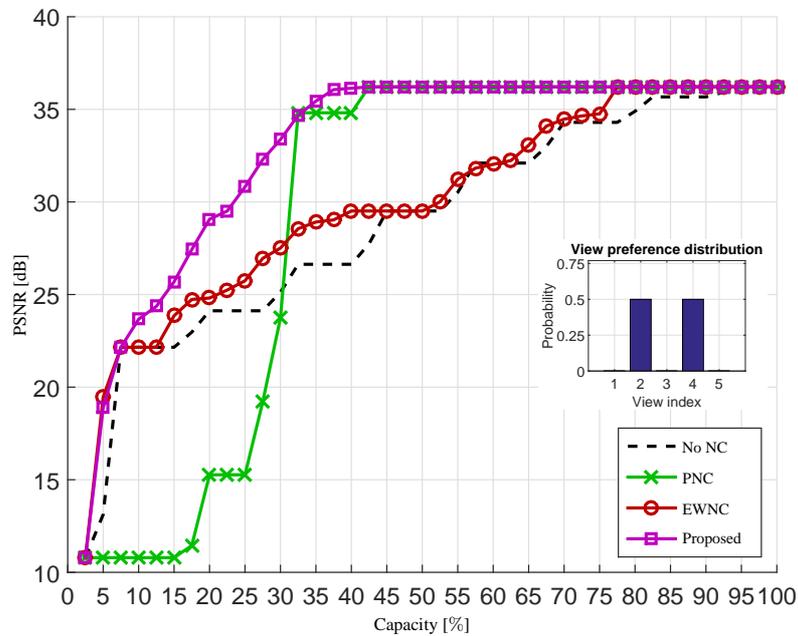
If the number of available sources increases, the PNC approach will finally provide the best performance. In our experiments we found that the threshold is  $M = 6$  for a packet loss ratio  $\varepsilon = 10\%$ . However we underline that all the parameters of these simulations (senders uplink capacities, packet loss rate, user preferences and in particular number of senders per receiver) are in general not under control of the service provider. Thus, even though there exist specific scenarios in which PNC could provide a better service than the proposed technique, the latter has a more stable and predictable behavior, much less influenced by these factors, and in particular by the network-dependent factors, which –in a wireless network– could change frequently and abruptly.

In Fig. 13 we show the distribution of the PSNR's per view in the case of Gaussian preference distribution,  $p_c = 50\%$ , loss rate =10%, and  $M = 2$  sources. We observe that when the channel capacity is small our technique allocate the resources to the central view, in such a way that the PSNR distribution has a behavior similar to the view preference distribution. On the contrary, the other strategies have a quite uniform per-view PSNR. Of course, when the capacity is high, all the strategies achieve very high PSNR over all the views. Lower PSNR for PNC and NoNC are explained as before, i.e. these technique are less robust to losses.

Even though, as we stated in Sec. 3, the estimation of the preference distribution is outside the scope of this article, it is worth mentioning the effects of an incorrect preference model on the performances of the technique. The effect on our technique is that

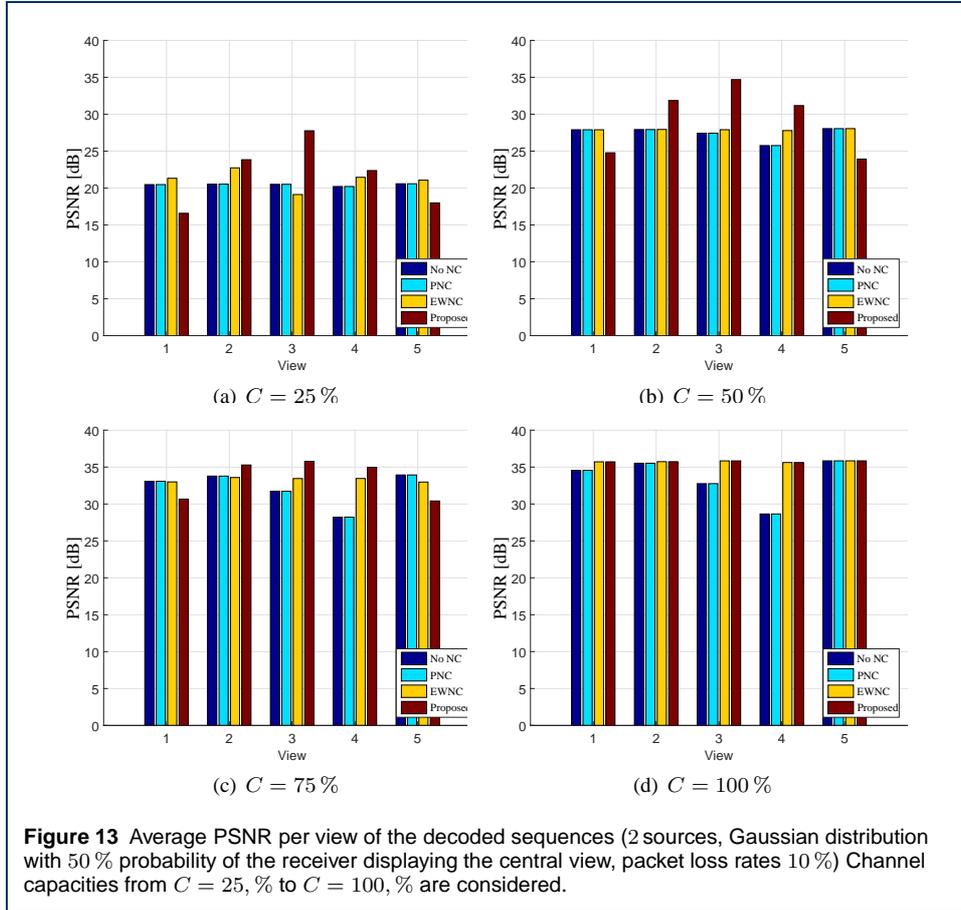


(a)  $M = 4$   $\epsilon = 10\%$   $p_c = 50\%$



(b)  $M = 4$   $\epsilon = 10\%$   $p_c = 50\%$

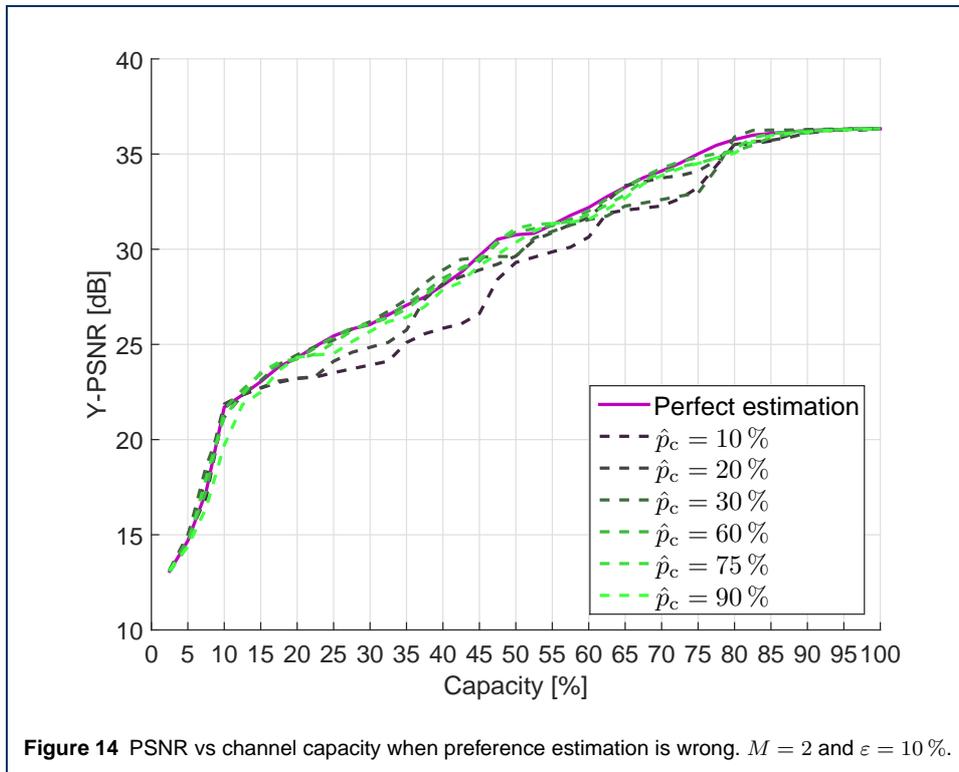
**Figure 12** Comparison of the average PSNR of the decoded sequences (4 sources, packet loss rate 10%). View preference distributions are Triangular and Bimodal, with maximum probability  $p_c$  equal to 50%. The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the PSNR is computed as the average over the views weighted by the preference probabilities.



the performance are negatively affected, and become closer to those of the EWNC technique. Since the EWNC technique is equivalent to ours when an uniform probability model is assumed, our technique still outperforms this reference as long as the model used is a better approximation of the real distribution than the uniform model, *e.g.*, in terms of Kullback–Leibler divergence. In order to prove this, we performed an experiment in which the proposed technique uses an estimation of the central view probability,  $\hat{p}_c \in \{10\%, 20\%, 30\%, 50\%, 60\%, 75\%, 90\%\}$ , while the true user preference is  $p_c = 50\%$ . In order to keep things simple, the probability on the views other than central is always uniform. Then we computed the PSNR as a function of the channel capacity for the different preference estimations. The results are reported in Fig. 14 for 2 sources and  $\varepsilon = 10\%$ . The average  $\Delta$ PSNR with respect to the perfect estimation are reported in Tab. 13, together with the relative entropy of the estimated distribution with respect to the real one. We observe that the PSNR losses are larger when the estimated probability is further from the real one. However, as we can see in Tab. 13 and in Fig. 15 (that reports the same results graphically), it is better to overestimate  $p_c$  than to underestimate it. We also observe that, unless we use a very bad estimation of  $p_c$  (*i.e.*,  $\hat{p}_c = 10\%$ ), the global performance are still better than EWNC, which corresponds to the point  $\hat{p}_c = 20\%$ . In conclusion, even when the preference probabilities are not very precisely estimated, the proposed technique can provide better performance than the reference ones, provided that the estimated preferences are not much worse than the implicit estimation of EWNC.

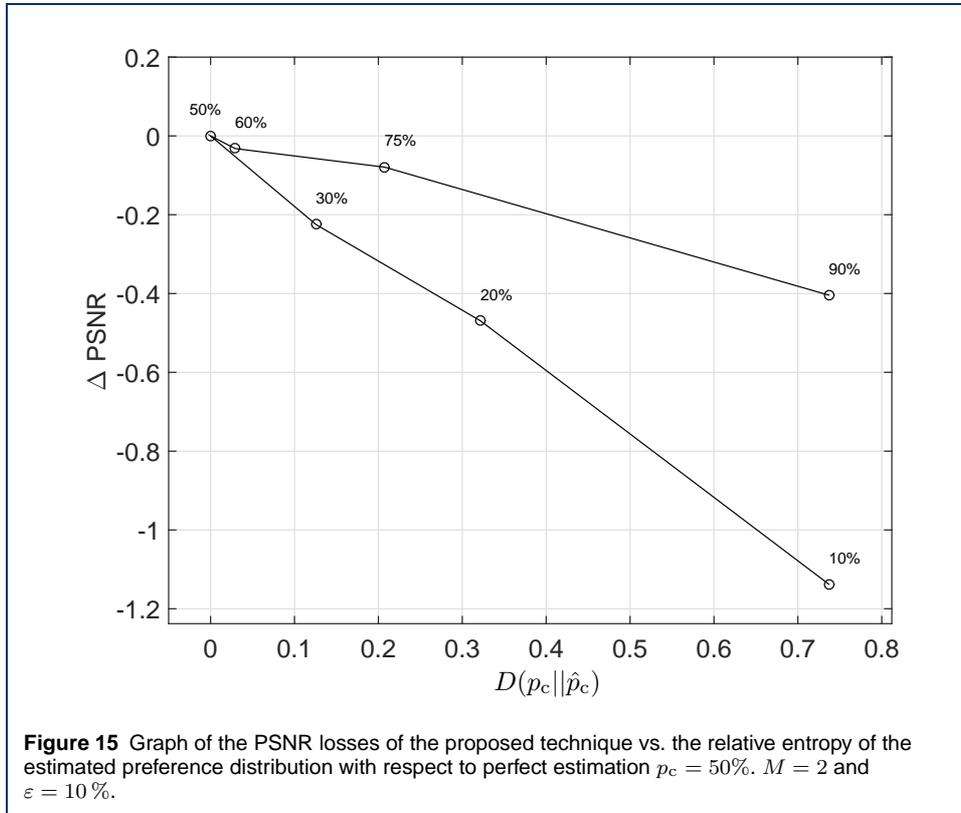
$\hat{p}_c$	$D(p  \hat{p})$	$\Delta$ PSNR
10%	0.74	-1.14
20%	0.32	-0.47
30%	0.13	-0.23
60%	0.03	-0.03
75%	0.21	-0.08
90%	0.74	-0.40

**Table 13** PSNR losses of the proposed technique and relative entropy of the estimated preference distribution with respect to the perfect estimation  $p_c = 50\%$ .  $M = 2$  and  $\varepsilon = 10\%$ .



In a further experiment, we considered two new clustering structures, to assess how this impact on the rate-distortion performance of the system. We used the same conditions as in Fig. 7(a) and Fig. 10(a), i.e., peaky distribution,  $p_c = 50\%$ , error probability  $\varepsilon = 10\%$ , and  $M = 2$  or  $M = 4$  sources.

- 1 In a first case, we use a larger number of clusters: this means that the labels are better representative of the clusters, but the “randomness” of the approach is limited. For  $M = 2$  sources we found a practically identical PSNR result with respect to the initial clustering, while for  $M = 4$  we registered a very small loss ( $\Delta$ PSNR=-0.06 dB). This is reasonable: having small clusters makes ineffective having multiple sources, since the senders are obliged to pick frames in the small sets.
- 2 In a second case, we used a smaller number of clusters, i.e. we merged predictions levels two by two. This means that we have large clusters, and the hypothesis of similar rates and distortions within a cluster is less reasonable. On the other hand, we improve the “randomness” of the algorithm. We observed a slightly larger losses both for  $M = 2$  ( $\Delta$ PSNR=-0.07 dB) and for  $M = 4$  ( $\Delta$ PSNR=-0.12 dB): we are



$\varepsilon_G$	10%
$\varepsilon_B$	30%
$p_{GB}$	5%
$p_{BG}$	20%

**Table 14** Gilbert-Elliot model parameters used in the experiments.

not able to take advantage of the improved randomness, and we pay the fact of less representative cluster labels.

As a conclusion from these new tests, we observe that the structure of the cluster has some impact on the global RD performance, related to the trade-off between representation and randomness. However, in all cases, the performance is better than the references.

In the last experiment we changed the channel model and used a more realistic Gilbert-Elliot model, whose parameters are described in Tab. 14. This corresponds to a global packet loss ratio of 14.0%. We performed the same experiment as the one in Fig. 7, using  $M = 2$  sources and respectively  $p_c = 50\%$  and  $p_c = 75\%$ . The PSNR's as a function of the channel bandwidth for the proposed and reference techniques are shown in Fig. 16, while the average PSNR gains are reported in Tab. 15. We observe results similar to those obtained for the PEC same global loss probability, for both values of  $p_c$ . Moreover the proposed technique outperforms the others for practically all the values of the channel capacity. In conclusion, changing the channel model does not modify a lot the ranking among the tested techniques, and the proposed one confirms being the best when the transmission conditions are the most difficult.

	$p_c = 50\%$	$p_c = 75\%$
NO NC	3.36	3.94
PNC	8.42	9.02
EWNC	0.44	1.41

**Table 15** PSNR gain of the proposed technique with respect to the reference,  $M = 2$ . Gilbert-Elliot channel with parameters in Tab. 14.

To summarize, we can conclude that our approach, thanks to the early decodability offered by EWNC and the *a priori* on the users in terms of distribution of the preferred views, is able to provide both a more consistent video quality, and –in the vast majority of scenarios– better than PNC. Other approaches, such as using EWNC without exploiting the preference distribution, or not using NC at all, are always outperformed by the proposed technique. Even when the estimation of preferences is not perfect, the proposed technique is worth using, provided that this estimation is not too wrong. With respect to PNC, we observe that the higher gains are achieved when the number of senders is small, the capacities of the channels are low, and the packet loss rate is high. In other words, the benefits of the proposed technique are more visible when the transmission conditions are harsher (e.g., small capacity, high error probability), which is a particularly interesting case for wireless services.

## 6 Conclusions and Future Work

In this work we presented a novel technique for distributed video streaming of multi-view content over a wireless network. The challenge consists in transmitting a multi-view stream, with the associated high bitrate, in a mobile network where the channel capacities are limited and the packet loss rates may be high. To address this challenge, we propose to use Network Coding as a transmission technique, which –thanks to its unique property of allowing uncoordinated cooperation of the nodes– is able to fully exploit the available capacity. In particular, in order to dynamically adapt to the network conditions, we propose to use Expanding Window Network Coding, a network coding scheme that allows instant decodability of the combined packets.

In order to be able to adapt the transmission to the channel capacity, the frames are included in the coding window of each sender in an order determined by an RD-optimized scheduler. The key idea is to use the users' preferences to identify the parts of the content more likely to be needed by the receiver in order to display its selected view. This induces a definition of the expected distortion of the stream as a weighted average of the distortion of the views. Using this metric, in order to reduce the probability of generating non-innovative packets, the senders generate a simplified probabilistic RD model that provides them with a degree of freedom in the choice of the schedule. Thus, the senders are able to generate a multitude of different, close to the optimum, network coded flows that, when decoded jointly by the receiver, allow for a high and stable expected video quality.

We compared the performance of our proposed method with several techniques, in a wide range of scenarios, in terms of number of senders, channel models and capacity, user preferences and quality of the preference estimation. We observed that the introduction of the preferences, jointly with the constraint imposed on the instant decodability of the selection, significantly improves the performance w.r.t. the reference techniques in the vast majority of the scenarios, in terms of video quality (PSNR) for a given capacity. Possible

future work includes the development of a large-scale interactive multi-view distribution system, 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. This optimization could be performed using topological information inferred from the packet exchanges of neighboring nodes.

#### Competing interests

The authors declare that they have no competing interests.

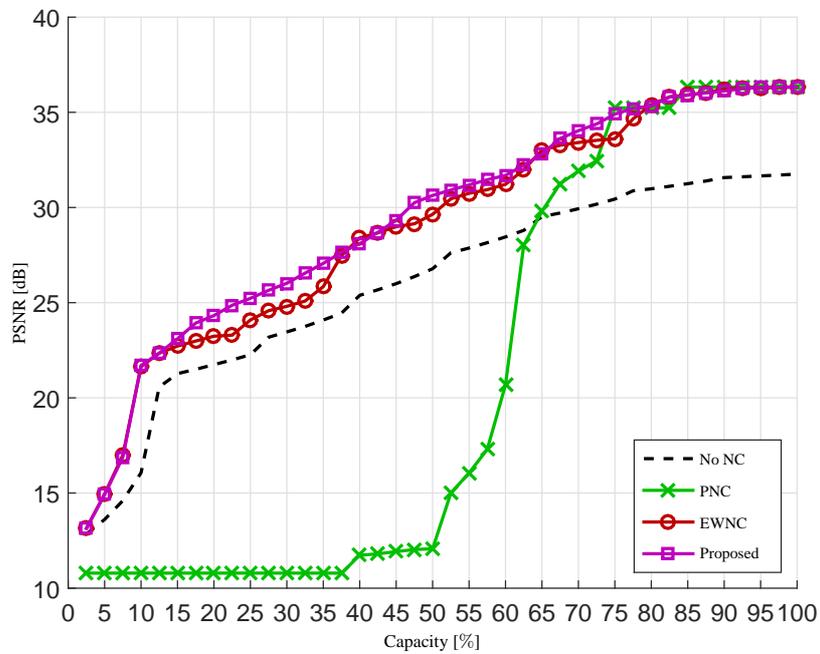
#### Author details

<sup>1</sup>DxO, Paris, France. <sup>2</sup>I3S CNRS/UNSA, Sophia Antipolis, France. <sup>3</sup>Institut Mines, Telecom-ParisTech, LTCI-CNRS, Paris, France.

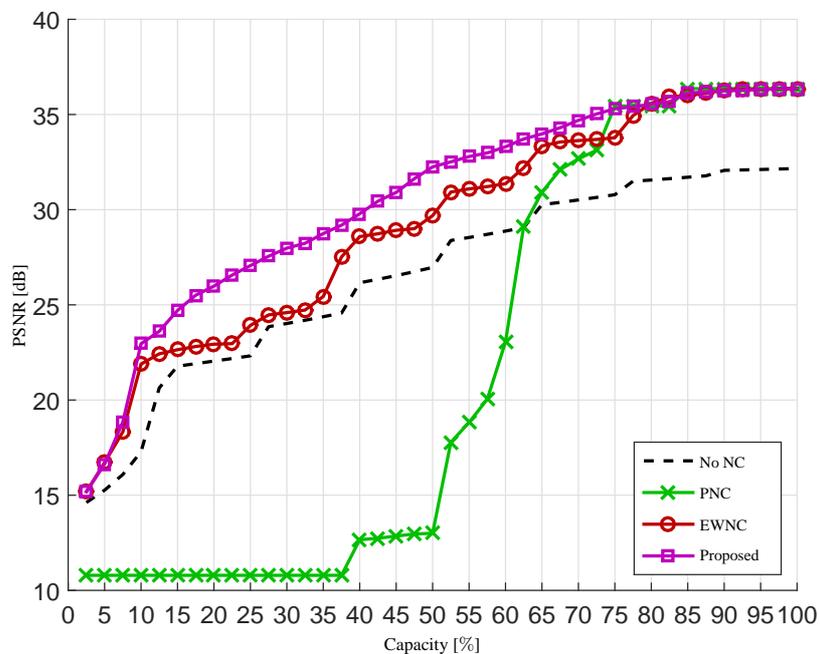
#### References

- Dufaux, F., Pesquet-Popescu, B., Cagnazzo, M. (eds.): *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*. John Wiley & Sons, Ltd, UK (2013)
- Tanimoto, M., Tehrani, M.P., Fujii, T., Yendo, T.: Free-viewpoint TV. *IEEE Signal Processing Magazine* **28**(1), 67–76 (2011)
- Merkle, P., Smolic, A., Muller, K., Wiegand, T.: Efficient prediction structures for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology* **17**(11), 1461–1473 (2007). Invited Paper
- Sullivan, G.J., Ohm, J.R.: Recent developments in standardization of high efficiency video coding (HEVC). In: *Proceedings of SPIE Conference on Applications of Digital Image Processing*, San Diego, CA, USA (2010)
- Vetro, A., Wiegand, T., Sullivan, G.J.: Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard. *Proceedings of the IEEE* **99**(4), 626–642 (2011). Invited Paper
- Sullivan, G.J., Boyce, J.M., Chen, Y., Ohm, J.-R., Segall, C.A., Vetro, A.: Standardized extensions of high efficiency video coding (HEVC). *Selected Topics in Signal Processing*, *IEEE Journal of* **7**(6), 1001–1016 (2013)
- Schwarz, H., Bartnik, C., Bosse, S., Brust, H., Hinz, T., Lakshman, H., Merkle, P., Muller, K., Rhee, H., Tech, G., Winken, M., Marpe, D., Wiegand, T.: Extension of high efficiency video coding (HEVC) for multiview video and depth data. In: *Proceedings of IEEE International Conference on Image Processing* (2012)
- Tehrani, M.P., Senoh, T., Okui, M., Yamamoto, K., Inoue, N., Fujii, T.: [m31095][FTV AHG] Use cases and application scenarios for super multiview video and free-navigation. In: *International Organisation For Standardisation* (October 2013). ISO/IEC JTC1/SC29/WG11
- Tehrani, M.P., Senoh, T., Okui, M., Yamamoto, K., Inoue, N., Fujii, T.: [m31103][FTV AHG] Introduction of super multiview video systems for requirement discussion. In: *International Organisation For Standardisation* (October 2013). ISO/IEC JTC1/SC29/WG11
- Tehrani, M.P., Senoh, T., Okui, M., Yamamoto, K., Inoue, N., Fujii, T.: [m31261][FTV AHG] Multiple aspects. In: *International Organisation For Standardisation* (October 2013). ISO/IEC JTC1/SC29/WG11
- Dricot, A., Jung, J., Cagnazzo, M., Dufaux, F., Pesquet-Popescu, B.: Subjective evaluation of Super Multi-View compressed contents on high-end light-field 3D displays. *Elsevier Signal Processing: Image Communication*, **15** (2015)
- Dricot, A., Jung, J., Cagnazzo, M., Pesquet-Popescu, B., Dufaux, F.: Full parallax super multi-view video coding. In: *IEEE International Conference on Image Processing*, vol. 1. Paris, France, pp. 135–139 (2014)
- Savas, S.S., Gürlür, C.G., Tekalp, A.M., Ekmekcioglu, E., Worrall, S., Kondoz, A.: Adaptive streaming of multi-view video over {P2P} networks. *Signal Processing: Image Communication* **27**(5), 522–531 (2012). doi:[10.1016/j.image.2012.02.013](https://doi.org/10.1016/j.image.2012.02.013). {ADVANCES} {IN} 2D/3D {VIDEO} {STREAMING} {OVER} {P2P} {NETWORKS}
- Ozcinar, C., Ekmekcioglu, E., Kondoz, A.: Adaptive 3d multi-view video streaming over p2p networks. In: *Image Processing (ICIP), 2014 IEEE International Conference On*, pp. 2462–2466 (2014). doi:[10.1109/ICIP.2014.7025498](https://doi.org/10.1109/ICIP.2014.7025498)
- Abdul-Hameed, O., Ekmekcioglu, E., Kondoz, A.: State of the art and challenges for 3d video delivery over mobile broadband networks. In: *Advanced Video Communications over Wireless Networks*, pp. 321–354. CRC Press, ??? (2013)
- Greco, C., Cagnazzo, M.: A cross-layer protocol for cooperative content delivery over mobile ad-hoc networks. *Inderscience International Journal of Communication Networks and Distributed Systems* **7**(1–2), 49–63 (2011)
- Greco, C., Cagnazzo, M., Pesquet-Popescu, B.: Low-latency video streaming with congestion control in mobile ad-hoc networks. *IEEE Transactions on Multimedia* **14**(4), 1337–1350 (2012)
- Goyal, V.K.: Multiple description coding: compression meets the network. *IEEE Signal Processing Magazine* **18**(5), 74–93 (2001)
- Parate, P., Ramaswamy, L., Bhandarkar, S., Chattopadhyay, S., Devulapally, H.: Efficient dissemination of personalized video content in resource-constrained environments. In: *Proceedings of IEEE International Conference on Collaborative Computing*, Washington, DC, USA, pp. 1–9 (2009)
- Gkantsidis, C., Rodriguez, P.R.: Network coding for large scale content distribution. In: *Proceedings of IEEE International Conference on Computer Communications* (2005)
- Ahlsvede, R., Cai, N., Li, S.-Y.R., Yeung, R.W.: Network information flow. *IEEE Transactions on Information Theory* **46**(4), 1204–1216 (2000)
- Li, S.-Y.R., Yeung, R.W., Cai, N.: Linear network coding. *IEEE Transactions on Information Theory* **49**(2), 371–381 (2003)
- Koetter, R., Médard, M.: An algebraic approach to network coding. *IEEE/ACM Transactions on Networking* **11**(5), 782–795 (2003)

24. Ho, T., Médard, M., Shi, J., Effros, M., Karger, D.R.: On randomized network coding. In: Proceedings of IEEE International Symposium on Information Theory, Kanagawa, Japan (2003)
25. Chou, P.A., Wu, Y., Jain, K.: Practical network coding. In: Proceedings of Allerton Conference on Communication Control and Computing, Monticello, IL, USA (2003)
26. Nemoianu, I.D., Greco, C., Cagnazzo, M., Pesquet-Popescu, B.: On a practical approach to source separation over finite fields for network coding applications. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada (2013)
27. Greco, C., Nemoianu, I., Cagnazzo, M., Pesquet-Popescu, B.: On a hashing-based enhancement of source separation algorithms over finite fields with network coding perspectives. *IEEE Transactions on Multimedia* **16**(7), 2011–2024 (2014)
28. Vukobratović, D., Stanković, V.: Unequal error protection random linear coding for multimedia communications. In: Proceedings of IEEE Workshop on Multimedia Signal Processing, Saint-Malo, France (2010)
29. Thomos, N., Chakareski, J., Frossard, P.: Prioritized distributed video delivery with randomized network coding. *IEEE Transactions on Multimedia* **13**(4), 776–787 (2011)
30. Nemoianu, I.D., Greco, C., Cagnazzo, M., Pesquet-Popescu, B.: A framework for joint multiple description coding and network coding over wireless ad-hoc networks. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan (2012)
31. Greco, C., Nemoianu, I.D., Cagnazzo, M., Pesquet-Popescu, B.: A network coding scheduling for multiple description video streaming over wireless networks. In: Proceedings of European Signal Processing Conference, Bucharest, Romania (2012)
32. Vukobratovic, D., Stankovic, V.: Unequal error protection random linear coding strategies for erasure channels. *IEEE Transactions on Communications* **60**(5), 1243–1252 (2012)
33. Magli, E., Wang, M., Frossard, P., Markopoulou, A.: Network coding meets multimedia: a review. *IEEE Transactions on Multimedia* **PP**(99), 1–44 (2013)
34. Huang, S., Ramamoorthy, A., Medard, M.: Minimum cost mirror sites using network coding: Replication versus coding at the source nodes. *IEEE Transactions on Information Theory* **57**(2), 1080–1091 (2011)
35. Wang, H., Zhang, Y., Li, P., Jiang, Z.: The benefits of network coding in distributed caching in large-scale P2P-VoD systems. In: Proceedings of IEEE Global Telecommunications Conference, Miami, FL, USA, pp. 1–6 (2010)
36. Kao, Y.-C., Lee, C.-N., Wu, P.-J., Kao, H.-H.: A network coding equivalent content distribution scheme for efficient peer-to-peer interactive VoD streaming. *IEEE Transactions on Parallel and Distributed Systems* **23**(6), 985–994 (2012)
37. Golrezaei, N., Dimakis, A.G., Molisch, A.F., Caire, G.: Wireless video content delivery through distributed caching and peer-to-peer gossiping. In: Proceedings of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, pp. 1177–1180 (2011)
38. Golrezaei, N., Shanmugam, K., Dimakis, A.G., Molisch, A.F., Caire, G.: Wireless video content delivery through coded distributed caching. In: Proceedings of IEEE International Conference on Communications, Ottawa, ON, Canada, pp. 2467–2472 (2012)
39. Fiandrotti, A., Chakareski, J., Frossard, P.: Popularity-aware rate allocation in multiview video. In: Proceedings of IEEE International Symposium on Visual Communications and Image Processing, vol. 7744 (2010). Invited Paper
40. Chakareski, J., Velisavljevic, V., Stankovic, V.: View-popularity-driven joint source and channel coding of view and rate scalable multi-view video. *Selected Topics in Signal Processing, IEEE Journal of* **9**(3), 474–486 (2015)
41. Toni, L., Thomos, N., Frossard, P.: Interactive free viewpoint video streaming using prioritized network coding. In: *Multimedia Signal Processing (MMSP)*, 2013 IEEE 15th International Workshop On, pp. 446–451 (2013). IEEE
42. De Abreu Goes, A.K., Toni, L., Maugey, T., Thomos, N., Frossard, P., Pereira, F.: Multiview video representations for quality-scalable navigation. In: *IEEE Visual Communications and Image Processing Conference*, pp. 295–298 (2014)
43. Thomos, N., Kurdoglu, E., Frossard, P., Van der Schaar, M.: Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers. *IEEE Transactions on Multimedia* **17**(6), 893–906 (2015)
44. Nemoianu, I.D., Greco, C., Cagnazzo, M., Pesquet-Popescu, B.: Multi-view video streaming over wireless networks with RD-optimized scheduling of network coded packets. In: Proceedings of IEEE International Symposium on Visual Communications and Image Processing (2012)
45. Chakareski, J., Frossard, P.: Distributed collaboration for enhanced sender-driven video streaming. *Multimedia, IEEE Transactions on* **10**(5), 858–870 (2008)
46. Sullivan, G.J., Wiegand, T.: Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine* **15**(6), 74–90 (1998)
47. Huang, H., Chan, S.-H.G., Cheung, G., Frossard, P.: Near-optimal content replication for interactive multi-view video streaming. In: Proceedings of IEEE Packet Video Conference, Munich, Germany, pp. 95–100 (2012)



(a)  $M = 2$  Gilbert – Elliot model  $p_c = 50\%$



(b)  $M = 2$  Gilbert – Elliot model  $p_c = 75\%$

**Figure 16** Comparison of the average PSNR of the decoded sequences (2 sources) using a Gilbert-Elliot channel model (see Tab. 14. The probabilities of the receiver displaying the central view are 50 % (left) and 75 % (right). The capacity of the channels is expressed as a ratio of the rate of the stream. For each sequence the PSNR is computed as the average over the views weighted by the preference probabilities.