Interactive Free Viewpoint Video Streaming Using Prioritized Network Coding

Laura Toni ⋆, Nikolaos Thomos ⋆,†, and Pascal Frossard ⋆

⋆ *Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*
† *Communication and Distributed Systems laboratory (CDS), University of Bern, Bern, Switzerland*
{laura.toni, nikolaos.thomos, pascal.frossard}@epfl.ch

***Abstract*—In free viewpoint applications, the images are captured by an array of cameras that acquire a scene of interest from different perspectives. Any intermediate viewpoint not included in the camera array can be virtually synthesized by the decoder, at a quality that depends on the distance between the virtual view and the camera views available at decoder. Hence, it is beneficial for any user to receive camera views that are close to each other for synthesis. This is however not always feasible in bandwidth-limited overlay networks, where every node may ask for different camera views. In this work, we propose an optimized delivery strategy for free viewpoint streaming over overlay networks. We introduce the concept of *layered quality-of-experience* (QoE), which describes the level of interactivity offered to clients. Based on these levels of QoE, camera views are organized into layered subsets. These subsets are then delivered to clients through a prioritized network coding streaming scheme, which accommodates for the network and clients heterogeneity and effectively exploit the resources of the overlay network. Simulation results show that, in a scenario with limited bandwidth or channel reliability, the proposed method outperforms baseline network coding approaches, where the different levels of QoE are not taken into account in the delivery strategy optimization.**

I. Introduction

Recent advances in multimedia technology and communication have pushed ahead the diffusion of new user-centric video services, such as interactive multiview (MV) video applications. These services endow clients with the possibility of freely changing their displayed viewpoint in realtime [1]. In such interactive scenarios, where only the views requested by the final users need to be transmitted, classical MV coding and streaming strategies become inefficient since they usually target the delivery of the full set of views to each client. The main challenge for effective delivery relies on the fact that the subset of selected views varies over time, which leads to an expensive view switching process in terms of delay and bandwidth.

A tradeoff between storage, bandwidth and quality of the interactive experience can be sought by free viewpoint streaming applications [2]. In such systems, an array of closely spaced depth and texture cameras acquire the same scene from different perspectives, but the viewpoints that can be displayed at the receiver side are not limited to the physically acquired camera views. With help of depth-image based rendering (DIBR), any intermediate view between two physical cameras can be synthesized at the decoder. The quality of the synthesized view increases with both the quality of the images used as reference (usually the closest camera views) and the correlation between the camera views and the synthesized view. The availability of many camera views for high quality synthesis however comes at a large price in terms of bandwidth. This has not escaped the attention of the scientific community and many research efforts have been made towards addressing the tradeoff between bandwidth and quality by novel source coding and data rendering strategies [3]–[6]. Differently from these works, we focus on the optimization of the delivery strategies, which are usually overlooked in the literature.

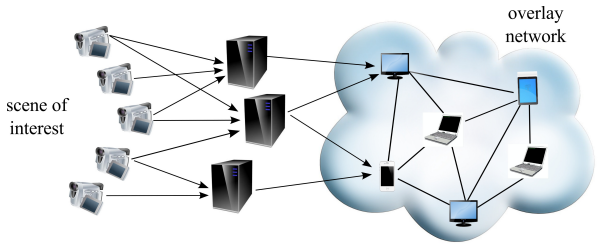

Figure 1. Investigated scenario.

A few works however propose solutions for delivery of interactive video data. The user's head position is tracked and predicted in [7], in order to estimate the views that most likely will be selected by the user. However, the streaming scheme becomes inefficient when several clients are considered together, possibly with conflicting requests. More distributed scenarios for interactive communications have been considered for pan/tilt/zoom functionality [8], for video-on-demand applications [9], and for interactive multiview scenarios [10], [11]. The latter investigates collaborative live free viewpoint applications, showing the benefit of sharing anchor views among peers, though this is associated with a quality reduction of the synthesized view. To the best of our knowledge, even if distributed scenarios have been investigated in interactive MV applications, an optimized streaming policy, which takes into account both the links constraints and users' requests in

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00

such a way that the level of interactivity offered to clients is adapted to individual channel constraints, is still missing.

In this paper, we propose an optimization problem for live free viewpoint streaming techniques over distributed and bandwidth-limited networks. As depicted in Fig. 1, we study a scenario in which video sequences acquired from each camera are real-time encoded into separate streams. These streams are delivered to servers, which obtain part or all camera views. The servers distribute the data over an overlay network, in which each intermediate node is interested in navigating within the scene of interest. The network is characterized by a large diversity in terms of client capabilities, bandwidth, channel conditions, and views required by the nodes. The portion of camera views received by each client is limited by the network conditions. Thus, there is the need to optimize the MV delivering strategy, in such a way that each client is able to maximize its quality-of-experience (QoE) during the navigation. The QoE is here defined as the quality at which users can navigate between viewpoints, *i.e.*, the quality at which the view of interest is decoded (or virtually synthesized) and then displayed by each client.

We propose a network coding based *camera scheduling* optimization scheme, aimed at maximizing the user QoE. To reach this goal, we introduce the concept of *layered QoE* offered to users: we organize cameras in layered subsets, each of those enhancing the QoE with respect to the previous subsets. To allow each user to experience the QoE level that better fits its request and channel constraints, we propose a transmission scheme which combines layered camera sets with an unequal error protection (UEP) delivery schemes. In particular, since network coding (NC) naturally accommodates for network diversity and clients heterogeneity, we extend the concept of prioritized NC, introduced in [12], [13] to the free viewpoint scenario. With our definition of prioritized layered camera subsets, a receiver-driven scheduling strategy is proposed to optimize each node's coding scheme, such that the overall QoE in the overlay network is maximized. Simulation results show the gain achieved by the proposed scheme with respect to baseline network coding approaches, where the different levels of QoE are not taken into account in the delivery strategy optimization. The streaming scheme is optimized with a low-complexity algorithm able to effectively exploit the resources of the overlay network. The promising concept behind this work is that a scalable streaming scheme can be offered to heterogenous users by combining QoE levels in free viewpoint navigation with prioritized NC schemes.

Overall, the main contributions of this paper are the following: i) we introduce the concept of *layered QoE* in interactive MV streaming scenarios and we use this metric to evaluate the utility function; ii) we study interactive streaming in *heterogeneous scenarios* both in terms of network and clients' requests; iii) based on the concept of layered QoE, we construct *prioritized classes* to be used into prioritized *network coding* schemes.

The remainder of this paper is organized as follows. In Sec. II, we first detail the free viewpoint model and then we introduce the layered QoE. Sec. III describes how QoE levels are applied to prioritized network coding schemes and how the streaming strategy is optimized. Finally, results and conclusions are provided in Sec. IV and Sec. V, respectively.

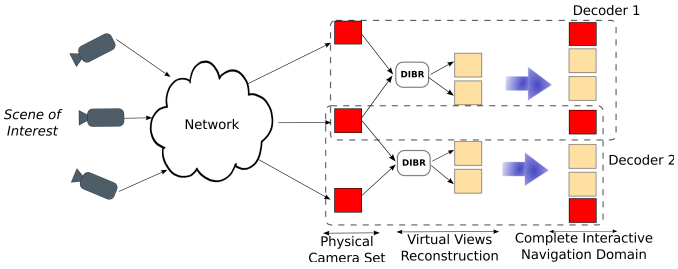

Figure 2. Multicamera scenario with $V = 3$ camera views and 2 virtually synthesized views from two adjacent cameras (*e.g.*, K=3).

II. Layered QoE in Interactive Streaming

In this section, we first detail the free viewpoint video model considered in our work; then, we introduce the concept of QoE layers in interactive MV applications.

A. Free Viewpoint Video Model

Let $\mathcal{V} = \{1, 2, \ldots, V\}$ be a discrete set of V cameras that acquire and encode the 3D scene of interest.[1] At the decoder side, a given view u can be virtually synthesized using texture and depth map of two camera views (*e.g.*, anchor views) via DIBR, as described in [14]. In short, each user can reconstruct any view of the discrete set $\mathcal{U} = \{1, 1+(1/K), 1+(2/K), \ldots, V-(1/K), V\}$ for some large K value, being $(K-1)$ the number of views synthesized between two adjacent anchor views. Note that, if $\mathcal{V}$ is the set of cameras that acquire the scene, $\mathcal{U}$ is the set of all possible viewpoints that the user can select, including both the actual camera views and the synthetic views, and $\mathcal{V} \subseteq \mathcal{U}$. In Fig. 2, a multiview scenario is illustrated, where 3 cameras acquire the scene. From each pair of cameras 2 views can be synthesized.

For any view u to be synthesized, a left (v^l) and right (v^r) camera view are required, with $v^l, v^r \in \mathcal{V}$ and $v^l \leq u \leq v^r$. The clients reconstruct the requested view at a distortion which depends on the level of spatial correlation that subsists between the anchor views and the virtual one. More in details, we consider aligned and equally spaced cameras such that the correlation level decreases with the distance between views. Hence, the distortion of the synthesized view depends on the selected camera views as follows [11]

$$d_u(v^l, v^r) = D_{\min} + \gamma e^{\alpha_u (v^r - v^l)} \left[e^{(\beta_u \min\{u-v^r, u-v^l\})} - 1 \right] \quad (1)$$

where γ, α_u and β_u are multiplicative coefficients that depend on the video sequence and drive the increasing rate of the distortion with the distance to camera views[2]. Note that $D_{\min}$ is

[1] Both texture and depth map of the 3D scene are encoded.

[2] We remind the reader to [11] for further details on the distortion model and for the specific meaning of each parameter.

the distortion at which each camera view can be decoded when actually received (*e.g.*, if $u \in \mathcal{V}$). From Eq. (1), we observe that the larger is the distance between u and the anchor views, the larger is the distortion. The key intuition behind Eq. (1) is that, when DIBR is adopted, the error in the disparity map (between the reference view and the virtual synthesized one) is given by $(k\,f)/\Delta Z$, where k is the distance between the camera view and the synthetic view, ΔZ is the error in the depth map and f is the rectified focal distance length of the cameras. Thus, for f and ΔZ constant, the error is proportional to k.

It is worth noting that the optimization of the RD function for DIBR methods is beyond the scope of this paper. The model in Eq. (1) has been chosen because is quite simple and yet accurate enough to build groups of cameras views as proposed next. Our interactive MV live streaming framework however is general and other source distortion function models can be used.

B. Prioritized Cameras Streams

Equipped with the above notations, we now introduce the concept of prioritized streams in interactive MV systems. We consider a scenario in which each user has the possibility of freely selecting a view $u \in \mathcal{U}$ for navigation. We assume that the popularity q_u of view u (that relates to the probability for a client to select the view u) is known. Note that the popularity can be described by either a uniform distribution, which is typical for static scenes (*e.g.*, museums), or by an exponential or non-uniform distributions, for dynamic scenes where most of the clients focus their attention on the same viewpoints (*e.g.*, soccer game) [15], [16]. For any camera popularity model, we define the interactive QoE level offered to the user as the ability of switching to any view anytime and still preserving the video quality. In other words, the QoE level is described by the distortion at which the viewpoints in the navigation domain (*e.g.*, $u \in \mathcal{U}$) can be virtually synthesized, given a set of received cameras streams $\mathcal{V}' \subseteq \mathcal{V}$. This is given by

$$D(\mathcal{V}') = \sum_{\substack{u\in\mathcal{U},\\ u:v_u^l,v_u^r\in\mathcal{V}'}} q_u\ d_u(v_u^l, v_u^r) + \sum_{\substack{u\in\mathcal{U},\\ u:v_u^l,v_u^r\notin\mathcal{V}'}} q_u\ D_{\max} \quad (2)$$

where v_u^l (v_u^r) is the left (right) camera in $\mathcal{V}'$ closest to u such that $v_u^l \le u \le v_u^r$, and $D_{\max}$ is the maximum distortion achieved when the viewpoint cannot be virtually synthesized. In particular, each virtual view can be synthesized by a left camera view v_u^l such that $v_u^l \le u$ and a right camera view v_u^r such that $v_u^r \ge u$, with both v_u^r and v_u^l available at the receiver. When this conditions are not met, the view cannot be synthesized. This happens when either views in $\mathcal{V}'$ are all smaller than u ($v' < u, \forall v \in \mathcal{V}'$) or when views in $\mathcal{V}'$ are all larger than u ($v' > u, \forall v \in \mathcal{V}'$). We denote this case by $u : v_u^l, v_u^r \notin \mathcal{V}'$. Usually, this event is experienced by the lateral views that cannot be synthesized when only central cameras are received.

Given the above definition, we can now organize the cameras' streams into layered subsets, each one offering an incremental level of QoE. More in details, we divide the finite set of cameras into C subsets such that $\mathcal{V}'_1 \cup \mathcal{V}'_2 \cup \ldots \cup \mathcal{V}'_C = \mathcal{V}$, with $\mathcal{V}'_i \cap \mathcal{V}'_j = \emptyset, i \neq j$. Subsets are organized based on their priority level, where $\mathcal{V}'_1$ and $\mathcal{V}'_C$, respectively, are the most and the least important subsets. These prioritized layers are transmitted in an UEP fashion, sending in a more reliable way more important subsets. We consider a prioritized transmission which guarantees that the c-th subset is received only if the $(c-1)$-th is already available at the decoder side. This means that when the frames from the c most important subsets of camera are received and decoded, the quality of the interactive navigation is

$$\begin{aligned} D_c &= D\left(\bigcup_{i=1}^{c} \mathcal{V}'_i\right) \\ &= \sum_{\substack{u\in\mathcal{U},\\ u:v_u^l,v_u^r\in\bigcup_{i=1}^{c}\mathcal{V}'_i}} q_u\ d_u(v_u^l, v_u^r) + \sum_{\substack{u\in\mathcal{U},\\ u:v_u^l,v_u^r\notin\bigcup_{i=1}^{c}\mathcal{V}'_i}} q_u\ D_{\max} \end{aligned} \quad (3)$$

with $D_c \ge D_{c+1}$ since we assume that each camera views subset is a refinement of the quality experienced by the interactive user.

III. Prioritized Network Coding

Due to distributed and heterogeneous structure of the network, a scalable mechanism for delivering views to clients can be reached by employing the prioritized network coding strategy proposed in [12]. In short, source packets are organized in classes, sorted by their priority levels and a receiver-driven prioritized random network coding (PRNC) method is proposed to achieve UEP. The UEP strategy is obtained by varying the number of packets from each class that are used in the embedded network coding operations performed at each node. The coding optimization is performed locally (in a distributed manner) and every node requests from the parent nodes the best rate allocation among different classes. In this way, each node is able to experience the best QoE offered by the overlay network.

The class c is defined as the set of packets that are linear random combinations of packets from the c most important subsets of camera views $\mathcal{V}'_1 \cup \ldots \cup \mathcal{V}'_c$. Each client node n needs to optimize the coding strategy that should be implemented at the parent nodes, based on the available network bandwidth, the expected loss probability and the distortion gain associated to each class. This can be formulated as follows. Let $\boldsymbol{w} = [w_1, w_2, \ldots, w_C]$ be the rate distribution vector to be optimized, where w_c indicates the portion of packets from class c among the requested packets. The optimized distribution vector is the one that minimizes the expected distortion evaluated as follows

$$\overline{D}(n) = D_0\ p_0 + \sum_{c=1}^{C} D_c\ p_d(c) \quad (4)$$

where $D_0 = D_{\max}$ is the maximum distortion achieved when no classes are received, p_0 is the probability of decoding no classes, and $p_d(c)$ is the probability of decoding c video

classes (*e.g.*, the probability of decoding frames within the c most important subsets), which is derived in [12]. In short, each node n optimizes the optimal class distribution $\boldsymbol{w}^\star$ (*i.e.*, the number of packets that the node requests from its parent nodes for each packet class) computed as the distribution that minimizes the expected distortion (or that maximizes theQoE in the navigation). Formally,

$$\begin{aligned}\boldsymbol{w}^\star &= \arg\max_{\boldsymbol{w}} \overline{D}(n) = \\ &= \arg\max_{\boldsymbol{w}} \left\{ D_0\ p_0 + \sum_{c=1}^{C} D_c\ p_d(c) \right\} \\ \text{s.t. } &\sum_{c=1}^{C} w_c = 1 \text{ and } w_c \geq 0, \forall c \in [1, C]\,. \end{aligned} \tag{5}$$

The above distributed resource allocation problem is optimized with the iterative method proposed in [12], [13].

IV. Simulation Results

A. Simulation Setups

For our simulations, we consider $V = 7$ cameras equally spaced between each others and $K = 3$, which means that two views are virtually synthesized for every pair of cameras. For each view $u \in \mathcal{U}$, we evaluate the reconstructed distortion from Eq. (1) and assume that a view is synthesized by DIBR if either it corresponds to a virtual view, or it corresponds to a camera view that is not available at the receiver. We consider a uniform distribution of the views popularity, such that each view has a probability of being selected by users of $1/|\mathcal{U}|$. In this case, the priority between cameras is assigned based on their spatial distance, as shown in Fig. 3. The first subset (the most important one) is the set of cameras which guarantees the synthesis of all views in $\mathcal{U}$, i.e., the set includes the external views. Then, higher classes are constructed such that the distance between camera views and synthesized views is reduced.[3] This leads to the following organization when three subsets are considered: $\mathcal{V}'_1 = \{0, 6\}$, $\mathcal{V}'_2 = \{2, 4\}$ $\mathcal{V}'_3 = \{1, 3, 5\}$.

Results are carried out for a scenario in which each server stores the streams from all cameras and multi-view video coding (MVC) can then be performed. In particular, we consider a MVC with an inter-view dependency scheme that does not affect the switching cost. 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. In our scenario, we have three classes encoded into $30, 30$, and 23 packets per GOP, respectively, when the packet size including the network coding header is set to 1500 bytes. The values of the QoE experienced in the navigation are here provided in terms of PSNR[4]. The QoE achieved with the reception of the first c classes is 29.52 dB, 37.77 dB, and 38.16 dB, for $c = 1, 2$, and 3, respectively, for the "Ballet" video sequence. The distortion of the virtually synthesized views is given by Eq. (1). In the following, rather than focusing on timing aspects (*e.g.* , switching delay), we provide simulation results in terms of expected quality. The reason is that, thanks to the DIBR, any user is constantly able to responsively display the requested view (with negligible switching delay). So rather than focusing on the delay after which the desired viewpoint can be displayed by the user, we look at quality at which the requested view is displayed.

Network coding operations are performed on $\mathbb{F}_{2^8}$. The sources transmit network coded packets according to the rate distribution vector $\mathbf{w}^\star$ that their children nodes request. The considered networks are overlay mesh networks where each node i has upload capacity U_i that is equally distributed to its children nodes. Furthermore, each node is connected with D_{in} parent and D_{out} children nodes.

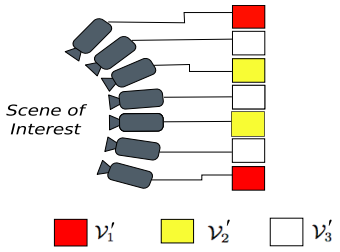

Figure 3. Construction of prioritized subsets of camera.

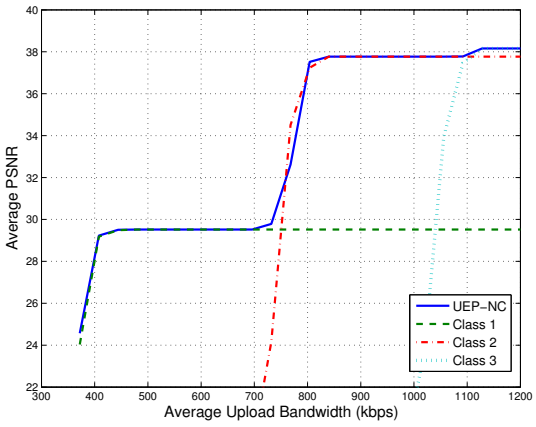

Figure 4. Comparison of the average PSNR (dB) curves for different values of average upload capacity in the network.

B. Results

We first study the impact of network nodes upload capacity U_i. Specifically, we uniformly change the average upload capacity of all network nodes in the range $[350, 1200]$ *kbps*.

[3]Note that in the case of non-uniform popularity, layers would be constructed in such a way that the interview distance is minimized among the most requested views first.

[4]PSNR=$10 \log_{10}(255^2/\overline{D})$, where $\overline{D}$ is derived from Eq. (4).

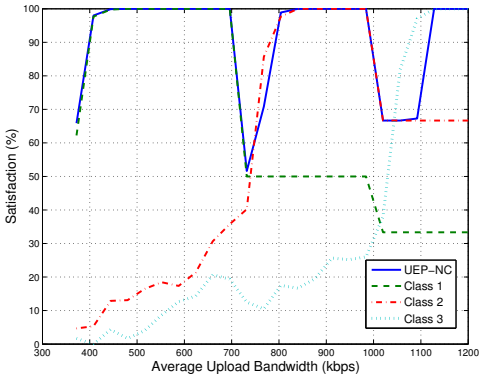

Figure 5. Satisfaction comparison for different values of average upload capacity in the network.

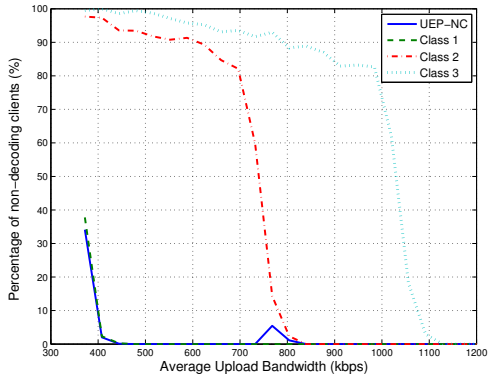

Figure 6. Percentage of nodes unable to decode any packet for different values of average upload capacity in the network.

We compare the proposed approach *UEP-NC* that optimizes the request allocation vectors according to Eq. (5) with three basic network coding schemes called *Class 1*, *Class 2* and *Class 3*. In *Class 1* scheme only packets that belong to most important anchor views (*i.e.*, $\mathcal{V}'_1$) are communicated to the network. Similarly, in *Class 2* and *Class 3* the network nodes transmit packets that are combinations of all the packets from views subset $\mathcal{V}'_1 \cup \mathcal{V}'_2$ and $\mathcal{V}'_1 \cup \mathcal{V}'_2 \cup \mathcal{V}_3$ respectively. The examined networks consist of three servers and 18 peer nodes. All the nodes have the same upload capacity that changes homogeneously in the range $[350, 1200]$ *kbps*. Without loss of generality each network node has $D_{in} = D_{out} = 3$. All the network links experience the same average loss rate 5%. The channels are modeled as Gilbert Elliot with burst length of nine packets.

The results of the evaluation are presented in Fig. 4, where the average PSNR is depicted with respect to the average upload capacity measured in *kbps*. From the evaluation, we observe that when U_i is less than 700 *kbps*, *UEP-NC* and *Class 1* schemes perform equally well in terms of PSNR. In this range of capacity values the resources are limited and sufficient only for transmitting packets that are combinations of packets from set of views $\mathcal{V}'_1$. As the upload bandwidth increases, *UEP-NC* takes advantage of the additional resources and transmits also packets from *Class 2* and rapidly the schemes achieves the PSNR that corresponds to the views in the set $\mathcal{V}'_1 \cup \mathcal{V}'_2$. From this comparison is obvious that *Class 2* scheme when bandwidth is larger than 700 *kbps* has non zero probability to decode the set $\mathcal{V}'_2$. Thus, PSNR increases and *Class 2* performs equally well to *UEP-NC*. For this range of bandwidth values, *Class 1* scheme is not anymore competitive to *UEP-NC* and *Class 2* as it cannot benefit from the increased bandwidth since only network coded packets from set $\mathcal{V}'_1$ are transmitted. When the link capacity grows to values higher than 1000 *kbps*, *Class 3* scheme performs equally well to *UEP-NC*, as there are is enough bandwidth for transmitting packets from set set $\mathcal{V}'_1 \cup \mathcal{V}'_2 \cup \mathcal{V}'_3$. *Class 2* and *Class 3* cannot profit from this excess of bandwidth resources. A very

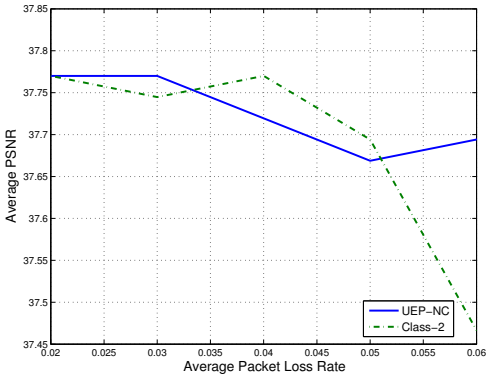

Figure 7. Comparison of the average PSNR (dB) curves for different values of average upload capacity in the network.

interesting observations is that *UEP-NC* is able to achieve the best performance in all the range of bandwidth changes, which shows the adaptability of the proposed approach.

For the same network setting, in Fig. 5 we present comparisons regarding the users' satisfaction. Satisfaction is measured as rate of the set a node decodes, *e.g.* $\mathcal{V}'_1$, $\mathcal{V}'_1 \cup \mathcal{V}'_2$ and $\mathcal{V}'_1 \cup \mathcal{V}'_2 \cup \mathcal{V}'_3$ over the theoretical maximum that the node may decode given its incoming capacity. A node when decodes $\mathcal{V}'_1$ receives one credit, while when decodes the $\mathcal{V}'_1 \cup \mathcal{V}'_2$ two credits *etc*. From the results, we can see that *UEP-NC* achieves always the best performance that is no lower than 50%. *Class 1* achieves high satisfaction when the upload capacity is less than 700 *kbps*, however for higher capacity values the satisfaction drops as it cannot exploit the additional bandwidth resources. Results for *Class 2* scheme show that when the resources are enough for decoding subset $\mathcal{V}'_1 \cup \mathcal{V}'_2$ the satisfaction is high. Satisfaction level becomes lower for more than 1000 *kbps* while is very low for less than 700 *kbps* as only few nodes close to the servers are able to decode the data. Note that there is a drop in the satisfaction curve experienced by *UEP-NC* for uploading capacities in the range $[700, 800]$ *kbps*.

In this transition region, the available resources are larger than the ones needed to decode $\mathcal{V}_1'$ but not always enough to successfully decode the subset $\mathcal{V}_1' \cup \mathcal{V}_2'$. Thus, users asking for $\mathcal{V}_1' \cup \mathcal{V}_2'$ might not be able to decode the requested views. This can be observed in Fig. 6, which depicts the percentage of nodes that are unable to decode any class in different cases. The results are carried out for the same network settings of before.

For the sake of completeness, we illustrate in Fig. 7 PSNR results for the same network setting as before. All nodes have outgoing capacity equal to 770 *kbps*. We consider that the loss rate in each link varies from 2% to 6%. We compare the proposed *UEP-NC* with *Class 2* as for the above setting *Class 1* and *Class 3* are not competitive in terms of PSNR. From the results is obvious that as the average packet loss rate increases, an increasing number of nodes is unable to decode the subset $\mathcal{V}_1' \cup \mathcal{V}_2'$. However, we can see that *UEP-NC* scheme offers the possibility to downgrade the decoded quality. This is not possible for *Class-2* scheme as it shows on-off performance, *i.e.* either decode the quality that corresponds to subset $\mathcal{V}_1' \cup \mathcal{V}_2'$ or decode nothing. Overall, we can conclude that *UEP-NC* is more robust to the loss rate changes that *Class-2* scheme.

C. Discussion

Results provided above demonstrate the benefit of combining the concept of layered camera sets with client-based network coding strategies with UEP built-in property. In this way, a scalable delivery scheme of the MV packets is provided, opportunistically adapting the subset of cameras included in the network coding scheme to the local network conditions and user's request. The proposed scheme is able to exploit the network resources, leading the rate allocated to highest classes to either decrease in limited network conditions, or increase when good channel conditions are experienced by the peer. Compared to baseline algorithms, the proposed *UEP-NC* scheme is able to achieve the largest QoE across different bandwidth availability, Fig. 4, and different packet erasure probabilities, Fig. 7. This leads the *UEP-NC* scheme to offer the highest satisfaction of users in the interactive scenario under investigation, Fig. 5.

It is worth noting that the *UEP-NC* scheme performs the rate allocation optimization locally with limited *a priori* information. In this way, the coding scheme is able to responsively adapt to any variation in the system (*e.g.*, clients' requests, channel capacity, cameras available at the source), Moreover, the computational complexity of the optimization algorithm is reduced, and network resources are used in an efficient manner.

V. Conclusion

We have proposed a network coding based camera scheduling optimization problem, aimed at maximizing the user QoE for interactive multiview streaming in overlay networks. We have introduced the concept of layered QoE, which is associated to the different levels at which any user can navigate within the scene. Cameras are then organized into prioritized layers, each one enhancing the QoE. A prioritized network coding delivery strategy is optimized, by choosing the best allocation rate between prioritized classes. By properly handling different priorities, network conditions, and users' requests, the proposed streaming strategy is able to offer most important source packets to clients when network resources are scarce, and the entire camera set for smoother navigation to better connected clients. Future works will be conducted to extend the optimization of both camera subsets and prioritized NC strategy to overlay networks in which users are organized into social groups, each one characterized by its own views popularity distribution.

Acknowledgements

The authors would like to thank Dr Thomas Maugey for fruitful discussions, and for his initial guidance on DIBR algorithms.

This work has been supported by the Swiss National Science Foundation, under grant PZ00P2-137275 and the Hasler Foundation funded project "Adaptive Network Coding for Video Communications".

References

[1] A. Tekalp, E. Kurutepe, and M. Civanlar, "3DTV over IP," *IEEE Sig. Proc. Mag.*, vol. 24, no. 6, pp. 77–87, 2007.

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

[3] T. Maugey, I. Daribo, G. Cheung, and P. Frossard, "Navigation domain partitioning for interactive multiview imaging," *ArXiv:1210.5041*, vol. abs/1210.5041, 2012.

[4] M. Tanimoto, M. Tehrani, T. Fujii, and T. Yendo, "FTV for 3-D spatial communication," *Proc. IEEE*, vol. 100, no. 4, pp. 905–917, 2012.

[5] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Trans. Image Processing*, vol. 20, no. 3, pp. 744–761, 2011.

[6] Q. Wang, X. Ji, Q. Dai, and N. Zhang, "Free viewpoint video coding with rate-distortion analysis," *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 22, no. 6, pp. 875–889, 2012.

[7] E. Kurutepe, M. Civanlar, and A. Tekalp, "Client-driven selective streaming of multiview video for interactive 3DTV," *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 17, no. 11, pp. 1558–1565, 2007.

[8] A. Mavlankar, J. Noh, P. Baccichet, and B. Girod, "Peer-to-peer multicast live video streaming with interactive virtual pan/tilt/zoom functionality," in *Proc. IEEE Int. Conf. on Image Processing*, 2008, pp. 2296–2299.

[9] W.-P. Yiu, X. Jin, and S.-H. Chan, "VMesh: Distributed segment storage for peer-to-peer interactive video streaming," *IEEE J. Select. Areas Commun.*, vol. 25, no. 9, pp. 1717–1731, 2007.

[10] X. Xiu, G. Cheung, and J. Liang, "Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1109–1126, 2012.

[11] D. Ren, S.-H. G. Chan, G. Cheung, H. V. Zhao, and P. Frossard, "Collaborative P2P streaming of interactive live free viewpoint video," *ArXiv:1211.4767*, vol. abs/1211.4767, 2012.

[12] 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.

[13] E. Kurdoglu, N. Thomos, and P. Frossard, "Scalable video dissemination with prioritized network coding," in *Proc. of StreamComm (in conjunction with ICME)*, July 2011.

[14] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *Proc. of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2009.

[15] C. Zhang and D. Florencio, "Joint tracking and multiview video compression," in *Visual Communications and Image Processing 2010*. International Society for Optics and Photonics, 2010.

[16] A. Fiandrotti, J. Chakareski, and P. Frossard, "Popularity-aware rate allocation in multi-view video," in *Proc. of SPIE VCIP*, July 2010.