Near-Optimal Content Replication for Interactive Multiview Video Streaming

Huan Huang S.-H. Gary Chan
Dept. of Comp. Sci. & Eng.
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: {huangzunhuan, gchan}@cse.ust.hk

Gene Cheung
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo, Japan 101-8430
Email: cheung@nii.ac.jp

Pascal Frossard
École Polytechnique Fédérale de Lausanne
EPFL-STI-IEL-LTS4, Station 11
CH-1015 Lausanne, Switzerland
Email: pascal.frossard@epfl.ch

***Abstract*—In interactive multiview video streaming (IMVS), a client can watch the multiview video and interact with it by switching to different viewing angles at frozen moments. To provide scalable IMVS services, a content provider often deploys distributed content servers with heterogeneous storage capacities in order to collaboratively replicate video segments for their clients. IMVS presents a new challenge in content replication: which video segments to replicate to support interactive view-switching during an IMVS session. In this paper, we first formulate the content replication problem as an integer linear programming (ILP) problem, which is proven to be NP-hard. We then propose a fast algorithm to solve it with bounded approximation error. In this algorithm, we first solve the ILP problem as a relaxed LP problem, and then heuristically round the resulting fractional LP solution to integer towards ILP feasibility. Simulation results show that our replication strategy can substantially reduce access costs compared to a commonly used replication scheme and a state-of-the-art replication scheme.**

I. Introduction

A multiview video is a set of 2D images of the same 3D scene captured synchronously by a large array of closely spaced cameras from different viewpoints [1]. In *interactive multiview video streaming* (IMVS), a client can play back the captured video of a single view and, in addition to random access in time, may perform *inter-view switching*, i.e., pause the video in time and switch to nearby viewpoints to observe the 3D scene from different viewing angles (the so-called *frozen moment* [2]). Active selection of viewpoints by a client can potentially enhance a perception of depth in the observed 3D scene, enriching visual experience.

In an IMVS network, there is a remote repository storing all the pre-encoded multiview videos. In order to support large number of users, distributed content servers are deployed close to user pool. They collaboratively replicate video segments for their clients given their limited storage capacities. A client sends inter-view or temporal switching requests of segments to its local server. The local server fulfills the request directly if the data has been replicated locally; otherwise, it contacts a neighboring server or remote repository to retrieve the missing data with some access cost. One of the most challenging problems in IMVS is content replication among the distributed servers, specifically, which video segments should be replicated at each server to support interactive view-switching but minimize network access cost.

In this paper, we consider a typical IMVS network where the inter-server transmission cost is negligible compared to the cost between the remote repository and the servers. Our work is based on a redundant coding structure proposed in our previous work, which facilitates temporal and inter-view switching in IMVS [3]. Beyond inter-view switching, the proposed coding structure can also be used to enable *indirect hit* among neighboring content servers: even if an exact requested view is not available in neighboring servers, a different but *correlated* view can be first fetched locally; then, the remote repository only needs to subsequently transmit the pre-encoded view difference. Using this coding structure, we formulate the content replication problem as an integer linear programming (ILP) problem and show that it is NP-hard. We then propose a content replication strategy based on linear programming relaxation and integer programming with near-optimal performance. The strategy is to solve the ILP problem as a relaxed LP problem first, and then round the resulting fractional LP solution to integer using a rounding procedure with linear execution time. Simulation results in typical network conditions show that our replication strategy can reduce access cost substantially compared to random replication and to a state-of-the-art scheme.

The outline of the paper is as follows. We first discuss the related work in Section II. We then review the redundant coding structure used for IMVS in Section III.

This work was supported, in part, by the General Research Fund from the Research Grant Council of the Hong Kong Special Administrative Region, China (611209) and Proof-of-Concept Fund at the HKUST (PCF.005.09/10 & PCFX05-15C00610/11ONA).

We formulate the content replication problem as an ILP problem and show that it is NP-hard in Section IV, and provide a LP-based solution in Section V. Experimental results and conclusions are provided in Section VI and VII, respectively.

II. Related Work

There has been much research in multiview video coding (MVC), focusing on compression of all captured frames across time and view, and exploiting both temporal and inter-view correlation to achieve maximal coding gain [4]. For an IMVS application [2], only a single requested view per client is needed at one time. Although MVC is suitable for compact storage of all multiview data (e.g., on a DVD disc), using MVC directly for IMVS means more than one video view must be transmitted so that a single view can be correctly decoded and displayed, leading to an increase in streaming rate.

In previous work on multiview video transport, non-interactive stereoscopic video data (two views for left and right eyes) is broadcasted / multicasted to a large group of clients [5]. In contrast, our IMVS is interactive, where a user can periodically select one out of a large number of views (100 capturing cameras were used in [1]) as the video is played back [2]. Our work extends the optimization further by considering both the frame structure *and* content replication strategy to achieve optimal overall IMVS system performance.

Previous work on content replication in video streaming has focused only on the single-view case [6]–[8]. Generally, other work adapts Bit-Torrent protocol to address video caching problem [9]. None of these works considers taking advantage of correlation among views of the same video for caching gain. Our previous work [3] propose a pure *heuristic-based* solution for the *single* movie case. In this work, we propose a *LP-based* content replication strategy to minimize network transmission cost for the *multiple movies* case.

III. Review of IMVS Coding Structure

A. Coding Structure in Details

To support inter-view and temporal switching while achieving good compression efficiency, the following frame structure was proposed [3] to pre-encode a given multiview video content of length NK frames in time. K consecutive captured video frames in time, $nK, \ldots, (n+1)K-1$, and of same view i, $i = 1, \ldots, U$, are encoded into a *segment*[1] $B_n(i)$. K is the inter-view switching period, determining how often in time view-switch can be requested by client. In other words, segment $B_n(i)$ of view i, $n \geq 0$, contains leading picture $F_{nK}(i)$ (*head* of $B_n(i)$) of time instant nK of view i, and trailing $K-1$ pictures $F_{nK+1}(i), \ldots, F_{(n+1)K-1}(i)$ (*tail* of $B_n(i)$). $F_{nK}(i)$ has a *redundant representation*, so that inter-view correlations among nearby viewpoints are exploited. Specifically, for a given *redundant window* δ, head $F_{nK}(i)$ contains up to 2δ pre-encoded differentials using heads $F_{nK}(j)$'s of nearby view j's as predictors, $j \in \{\max(1, i-\delta), \ldots, \min(U, i+\delta)\}$, so that a view-switch from view j to i only requires transmission of the corresponding pre-encoded differential. See Fig. 1 for an illustration of a multiview video frame structure for five views $U = 5$ and redundant window $\delta = 2$.

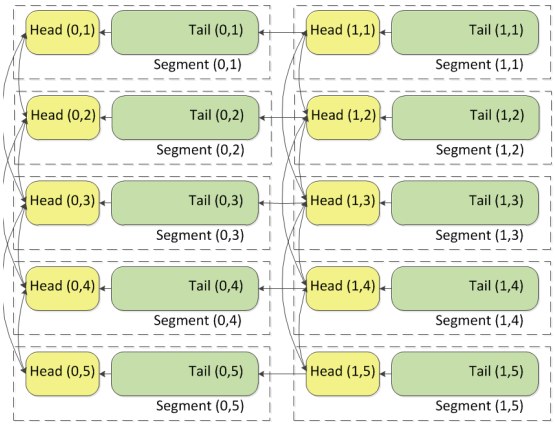

Fig. 1. Dependencies among segments in proposed multiview video frame structure. Arrows among heads indicate feasible view switches using pre-encoded differentials.

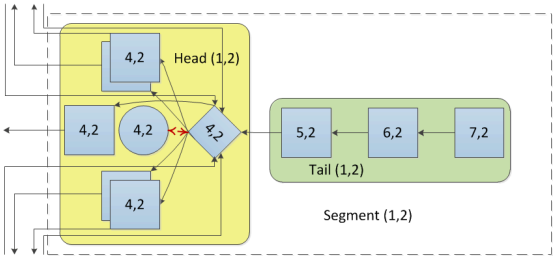

Fig. 2. An implementation of a segment using I-, P- and DSC frames (denoted by circle, squares, and diamonds, respectively).

In more details, head of $B_n(i)$ is represented by multiple compressed versions of the same picture $F_{nK}(i)$: i) one independently coded I-frame $I_{nK}(i)$, ii) multiple differentially coded P-frames $P_{nK}(i)$'s, and iii) one *distributed source coding* (DSC) frame $W_{nK}(i)$. First, a temporal P-frame $P_{nK|nK-1}(i)$ is motion compensated using a P-frame $P_{nK-1}(i)$ of previous time instant $nK-1$ and same view i as predictor. Then, inter-view P-frames $P_{nK}(i|j)$'s are disparity compensated, each using I-frame $I_{nK}(j)$ of a nearby view j of the same time instant nK as predictor. Temporal P-frame $P_{nK|nK-1}(i)$ is for video playback in time in the same view i. Inter-view P-frames $P_{nK}(i|j)$'s are for inter-view switching.

Graphically, DSC frame $W_{nK}(i)$ is shown using the multiple P-frames $P_{nK}(i)$'s as predictors. In details, DSC frame is constructed as follows. I-frame $I_{nK}(i)$ is the encoding target for $W_{nK}(i)$. Each $P_{nK}(i)$ provides *side information* (SI) to help decode $W_{nK}(i)$. Because SI $P_{nK}(i)$ and target $I_{nK}(i)$ are both a representation of the same

[1]We will adopt the convention that superscripts denote movie and/or server, subscripts denote time, and brackets denote view in the sequel.

picture $F_{nK}(i)$, the frequency contents in $P_{nK}(i)$ and $I_{nK}(i)$ in Discrete Cosine Transform (DCT) domain are mostly the same, except for a few least significant bits (LSB) in certain frequencies. If we now view each SI $P_{nK}(i)$ as a channel-corrupted version of target $I_{nK}(i)$, then $W_{nK}(i)$ only needs to deploy a channel code (low-density parity check code (LDPC) is used in [10]) that is strong enough to overcome the largest "channel noise" in all SI $P_{nK}(i)$'s, no matter which SI $P_{nK}(i)$ is actually available at decoder, recovering target $I_{nK}(i)$ is perfectly.

Thus, by DSC frame construction, $W_{nk}(i)$ can be perfectly reconstructed as long as *one* of the predictors $P_{nK}(i)$'s is available at decoder. Functionally, $W_{nK}(i)$ serves as a *merge* operator to reconciliate minor differences due to motion/disparity compensation and quantization among P-frames $P_{nK}(i)$'s to target $I_{nK}(i)$. This is done so that other frames in turn can simply use the one unified version of $F_{nK}(i)$, I-frame $I_{nK}(i)$, as predictor for differential coding. $W_{nK}(i)$ in practice is much smaller than independently coded I-frame $I_{nK}(i)$ [10]. Continuing with our example, Fig. 2 shows an example frame structure for segment $(1, 2)$.

B. Coding Structure in Usage

We now discuss how the redundant frame structure previously described is used in IMVS. If a viewer requests a view j after observing view i, where $|i - j| \leq \delta$, only inter-view P-frame $P_{nK}(j|i)$ and DSC frame $W_{nK}(j)$ need to be transmitted. On the other hand, if $|i - j| > \delta$, then the much larger independently coded I-frame $I_{nK}(j)$ must be transmitted.

The cost of repository transmission is generally much more expensive than a local server transmission; the goal of content replication is to avoid repository transmission as much as possible. Thus, to avoid repository transmission of $I_{nK}(j)$ during an inter-view switch from i to j when $|i - j| > \delta$, if a neighboring server has replicated I-frame $I_{nK}(l)$ locally, $|l - j| \leq \delta$, then the server can first forward $I_{nK}(l)$ to the client, while the repository transmits smaller P-frame $P_{nK}(j|l)$ and DSC frame $W_{nK}(j)$. This is called *indirect hit*: a local server does not have the requested view j replicated locally, but has a correlated view l that can help to lower repository transmission cost from $I_{nK}(j)$ to $P_{nK}(j|l)$ and $W_{nK}(j)$.

To summarize, there are four possible transmission costs during a requested inter-view switch from i to j, depending on available replicated content in servers. In order of increasing costs, they are:

1) *Direct hit*: when a neighboring local server has the exact segment $B_n(j)$ requested by a client. Replicated I-frame $I_{nK}(j)$ can be forwarded locally.
2) *Differential transmission*: when the repository transmits pre-encoded differentially coded P-frame $P_{nK}(j|i)$ and DSC frame $W_{nK}(j)$.
3) *Indirect hit*: when a neighboring server y has replicated a *correlated* frame $I_{nK}(l)$, which is forwarded locally, and the repository transmits only P-frame $P_{nK}(j|l)$ and DSC frame $W_{nK}(j)$.
4) *Replicate miss*: where servers do not have exact or correlated frames, and the repository transmits independently-coded I-frame $I_{nK}(j)$.

Because there are no differentially coded P-frames $P_{tK|nK}(i)$, $n \neq t$, there are only two possible costs for temporal switching, direct hit or replicate miss, similar to conventional caching mechanisms.

IV. Problem Formulation

In this section we present the distributed content replication problem as an integer linear programming (ILP) problem. We first define the decision variables, followed by the constraints and optimization objective.

A. Decision Variables

Each server must decide which segment(s) of movie m to replicate given its (limited) storage size. Let $\phi_n^{x,m}(i) \in \{0, 1\}$ be a binary variable denoting the decision to replicate segment $B_n^m(i)$ of movie m at server x. After watching a segment $B_n^m(i)$ of movie m, users can request an inter-view or temporal switch, and IMVS network has to decide from whom to get the segment. For inter-view switching, we first define $\xi_n^m(i, j) \in \{0, 1\}$ to be the binary variable denoting the decision to directly pull requested view j from a local server if an inter-view switch is requested from view i to j of instance n (direct hit). Further, we define $\zeta_n^m(i, j) \in \{0, 1\}$ to be the binary variable denoting the decision to pull a correlated view from a local server (and pre-coded differential from repository) if an inter-view switch is requested from view i to j of instance n (indirect hit). $\xi_n^m(i, j) = \zeta_n^m(i, j) = 0$ would mean requested view j must be pulled entirely from repository (replicate miss).

For temporal switching, we define $\theta_{n,t}^m(i) \in \{0, 1\}$ as the binary variable denoting whether to pull segment $B_t^m(i)$ from a server if a temporal switch is requested from time instance n to t.

B. Linear Constraints

Let $S_n^m(i)$ be the size of segment $B_n^m(i)$ of movie m. Because of finite storage capacity c^x of server x, we have the following capacity constraint for each local server x:

$$\sum_m \sum_n \sum_i \phi_n^{x,m}(i) S_n^m(i) \leq c^x, \quad \forall x. \tag{1}$$

For temporal switch variable $\theta_{n,t}^m(i)$, it can be 1 only if there is at least one server x replicating segment $B_t^m(i)$ locally. Thus, we can write:

$$\theta_{n,t}^m(i) \leq \sum_x \phi_t^{x,m}(i), \quad \forall m, n, t, i. \tag{2}$$

Similarly, for direct replicate inter-view switch variable $\xi_n^m(i, j)$, it can be 1 only if there is a server x replicating

segment $B_n^m(j)$:

$$\xi_n^m(i,j) \leq \sum_x \phi_n^{x,m}(j), \quad \forall m,n,i,j. \tag{3}$$

For indirect replicate inter-view switch variable $\zeta_n^m(i,j)$, it is more complicated. It can be 1 only if there is a server x replicating segment $B_n^m(l)$, where view l is in the window $j-\delta \leq l \leq j+\delta$, so that repository can send only P-frame $P_n^m(j|l)$ and DSC frame $W_n^m(j)$. Thus, we can write:

$$\zeta_n^m(i,j) \leq \sum_x \sum_{l=j-\delta, l\neq j}^{l=j+\delta} \phi_n^{x,m}(l), \quad \forall m,n,i,j. \tag{4}$$

For a given inter-view switch from i to j, to ensure that indirect and direct hits are not selected simultaneously, we write:

$$\xi_n^m(i,j) + \zeta_n^m(i,j) \leq 1. \tag{5}$$

We can see that all constraints are linear with respect to the decision variables.

C. Inter-view & Temporal Switch Model

Before defining the objective, we first describe the probabilistic model we use to describe the likelihood of a user choosing different inter-view and temporal switches. Let $p_{n,t}^m$ be the probability that a user chooses a temporal switch from a segment of time instant n of *any view* to segment of instant t of the same view. Further, let P_n^m be the relative *temporal popularity* of segments in instant n, i.e., $\sum_n P_n^m = 1$. P_n^m can be derived from $p_{n,t}^m$'s easily, if we assume a user starts an IMVS session in the first instant and has a variable lifetime (in terms of number of temporal switches before leaving the IMVS session) with known probability mass function.

Similarly, we can define $\pi^m(i,j)$ as the probability that a user switches from view i to j in *any time instant*. Further, we define $\Pi^m(i)$ as the steady state probability of view i, where $\sum_i \Pi^m(i) = 1$. Finally, let Ω be the probability that a client performs temporal switching; i.e., he switches to a different view with probability $1-\Omega$ at any time from any view.

D. Linear Objective

We first define $s_n^m(i)$ and $u_n^m(i)$ to be the expected cost of temporal and inter-view switching at segment $B_n^m(i)$ of movie m, respectively. Let Q^m be the probability that movie m is selected for observation. We can now write the expected user access cost S as

$$S = \sum_m \sum_i \sum_n Q^m P_n^m \Pi^m(i) \left[\Omega\ s_n^m(i) + (1-\Omega)\ u_n^m(i)\right]. \tag{6}$$

Our objective is to minimize expected user access cost S by deciding, which server to replicate each segment $B_n^m(i)$ (ϕ's), and where to pull content during a temporal switch (θ's) or an inter-view switch (ξ's and ζ's), i.e.,

$$\min_{\{\phi\},\{\theta\},\{\zeta\},\{\xi\}} S. \tag{7}$$

We can write temporal switch cost $s_n^m(i)$ as the sum of all time-to-time switch costs $C_{n,t}^m(i)$'s to some time t:

$$s_n^m(t) = \sum_t\ p_{n,t}^m\ C_{n,t}^m(i). \tag{8}$$

Similarly, we can write inter-view switch cost $u_n^m(i)$ as the sum of all view-to-view costs $T_n^m(i,j)$ to some view j:

$$u_n^m(i) = \sum_j\ \pi^m(i,j)\ T_n^m(i,j). \tag{9}$$

Each time-to-time switch cost $C_{n,t}^m(i)$'s can be written simply in terms of temporal switch variable $\theta_{n,t}^m(i)$:

$$C_{n,t}^m(i) = \theta_{n,t}^m(i)\ \epsilon\ + (1-\theta_{n,t}^m)\ S_t^m(i). \tag{10}$$

Equation (10) says that the time-to-time switch cost is a small ϵ if segment $B_t^m(i)$ is replicated in a neighboring server, and size of the segment $S_t^m(i)$ if $B_t^m(i)$ must be pulled from the repository.

View-to-view cost $T_n^m(i,j)$ is only slightly more involved. If it is a direct hit in server ($\xi_n^m(i,j) = 1$), then like previous (10), it is a small cost ϵ. If is is an indirect hit ($\zeta_n^m(i,j) = 1$), then repository must transmit a P-frame and a DSC frame, resulting in cost $\epsilon + d_n^m$; we approximate transmission of a P-frame $P_n^m(j|l)$ and DSC frame $W_n^m(j)$ for any intermediate view l, $j-\delta \leq l \leq j+\delta$, to be d_n^m. Finally, if it is a replicate miss ($\xi_n^m(i,j) = \zeta_n^m(i,j) = 0$), then repository must transmit all requested content, resulting in cost $D_n^m(i,j)$:

$$\begin{aligned} T_n^m(i,j) \quad = \quad & \xi_n^m(i,j)\ \epsilon\ +\ \zeta_n^m(i,j)(\epsilon + d_n^m)\ + \\ & [1-\xi_n^m(i,j)-\zeta_n^m(i,j)]\ D_n^m(i,j). \end{aligned} \tag{11}$$

Repository cost $D_n^m(i,j)$ depends whether the target view j is within the prediction window $[i-\delta, i+\delta]$ or not:

$$D_n^m(i,j) = \begin{cases} |P_n^m(j|i)| + |W_n^m| & \text{if } |i-j| \leq \delta; \\ |I_n^m(j)| & \text{if } |i-j| > \delta\ . \end{cases} \tag{12}$$

Since ϵ, $S_n^m(i)$, d_n^m and $D_n^m(i,j)$ are fixed, it is clear that time-to-time switch cost $C_{n,t}^m(i)$ and view-to-view switch cost $T_n^m(i,j)$ are linear in the decision variables. Thus, expected temporal and inter-view switch cost $s_n^m(i)$ and $u_n^m(i)$ are also linear, and objective function S is also linear. Since all the constraints are also linear, our problem is an Integer Linear Programming problem (ILP).

E. NP-Hardness Proof

We briefly outline a NP-hardness proof of our ILP problem by showing a special case can be mapped to the known NP-complete problem *bin packing* [11]: given a bin capacity W, a list of items of sizes $a_1, \ldots, a_Z$,

and integer B, is there a capacity-preserving item-to-bin assignment so that B or fewer bins are required?

Consider a special case of our problem where there are B servers in the IMVS network, each of storage size W, and the single multiview video only has one view. Thus, client can only perform temporal switching. Suppose each segment B_n of Z segments has size a_z, and each segment is requested with equal likelihood. If there is a content replication strategy to fit all segments in the servers (reflected in the resulting cost S since no repository transmission is required), then there is a capacity-preserving item-to-bin assignment to fit all items in B or fewer bins. Thus, our optimization problem is at least as hard as the NP-complete binary decision bin packing problem, and hence is NP-hard.

V. LP Relaxation and Rounding Algorithm

We discuss how a LP relaxation of the ILP problem can be easily solved. Given the LP-relaxed solution, we finally discuss a heuristic to round the fractional LP solution to integers to find a feasible solution.

A. LP Relaxation

The ILP problem posed earlier have linear constraints and objective; it is hard because of the additional integer constraints. If we remove these integer constraints, we can solve the resulting LP problem using one of several known algorithms like Simplex in polynomial time [11]. The resulting objective function value S^* represents a *super-optimal* solution value; i.e., $S^* \leq S^o$, where S^o is the true optimal solution value to the original ILP problem. The reason is that LP problem is a relaxed version of the original ILP problem with fewer constraints.

If we now perform rounding to the LP solution so that the integer constraints are satisfied, we have a (likely sub-optimal) solution that is feasible with objective value S^a. It is then easy to see that the approximation error ϵ away from the true optimal solution we suffer from our rounded solution is bounded as follows:

$$\epsilon = |S^a - S^o| \leq |S^a - S^*|. \tag{13}$$

Thus, the LP solution provides us with an *a posteriori* approximation bound to quantify the quality of our rounded solution. We discuss next how the LP solution also provides additional information so we can perform integer rounding to a good approximate solution.

B. Rounding Heuristic : Minimum Eviction

Given an LP solution, we can classify the storage variables $\phi_n^{x,m}(i)$'s into two classes: 1) *Primary variables*: a segment $B_n^m(i)$'s fractions $\phi_n^{x,m}(i)$'s across servers sum to one, i.e. $\sum_x \phi_n^{x,m}(i) = 1$; 2) *Secondary variables*: segment $B_n^m(i)$ where $\sum_x \phi_n^{x,m}(i) < 1$. LP solution tells us that primary variables are more important than secondary ones, because they are stored in entirety in servers. Heuristics *Minimum Eviction* essentially tries to fit as many primary variables in server storage as possible by iteratively considering the fractional segment with the largest size first:

1) Identify storage variables $\phi_n^{x,m}(i)$'s that equal 1. These are *stable* assignments and will not be changed further.
2) Find a target fractional primary variable $\phi_n^{x,m}(i) < 1$ representing the largest fractional segment. Round this fraction up, and corresponding variables $\phi_n^{y,m}(i)$'s in other servers y down.
3) If rounding up in step 2 results in server x storage constraint violation, evict secondary variables in order of decreasing fractional segment sizes until: i) constraint is met, or ii) no more secondary variables are left.
4) If server x storage constraint is still violated, evict the unstable primary variable $\phi_n^{x,m}(i) < 1$ in server x in order of decreasing fractional segment sizes until: i) constraint is met, or ii) no more unstable primary variables are left.
5) If server x storage constraint is still violated, evict the target unstable $\phi_n^{x,m}(i)$ instead.
6) Go to step 1. If no variables after another iteration, round down all remaining fractional variables.

The key idea is that by attempting to round up storage variable with the largest fractional segment size, it is either kept at server x, or it is removed from the servers, but never moved from one server to another.

VI. Illustrative Simulation Results

We now present simulation setup and results to demonstrate the performance of our replication strategy over competitor schemes.

Unless otherwise stated, we use the following parameters in all experiments: number of servers $\mathcal{V} = 3$, number movies $M = 3$, number of views $U = 7$, number of segments in time $N = 7$, server storage size $c^x = 3840$ units, $\forall x \in \mathcal{V}$, I-, P- and DSC frame sizes are 48 units, 4 units and 8 units, segment size 144 units, view switch tendency $\Omega = 0.3$.

We first solve the LP-relaxed version of the ILP problem, resulting in a "super-optimal" (SUP) solution. Based on the solution of the LP relaxation, we use LP relaxation with integer rounding (LPRR) to find an approximate solution. We compare with random replication (RAN), which randomly replicates movie segments at each server. We also compare with a state-of-the-art scheme called Local Greedy (LG), which divides the movies into three categories: those popular ones which all servers store, those medium popular ones which only one server store, and those unpopular ones which only the repository stores. To show the performance of LPRR, we compare the segment access cost of different solutions.

Figure 3 plots the access cost versus view switch tendency for different schemes. View switch tendency

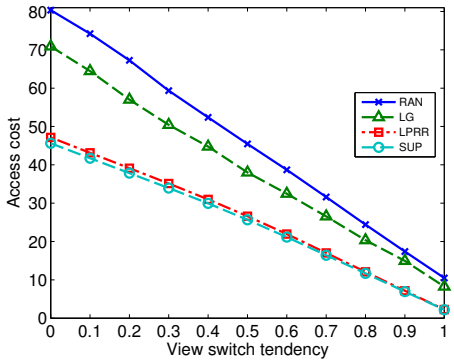

Fig. 3. Access cost versus view switch tendency given different schemes.

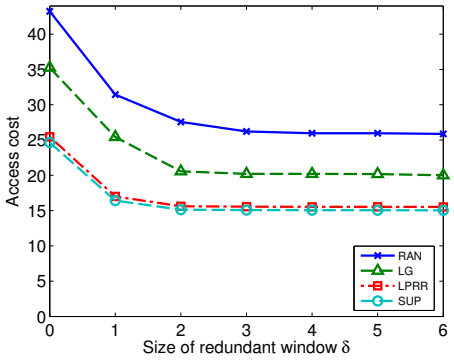

Fig. 4. Access cost versus size of redundant window given different schemes.

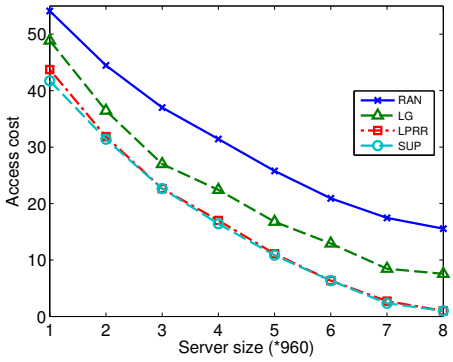

Fig. 5. Access cost versus number of servers given different schemes.

equals to 1 means users only perform inter-view switch at every switch opportunity. We observe that the access cost decreases with view switch tendency; higher view switch tendency means the repository has a high probability to transmit I-frame or P-frame-plus-DSC-frame instead of the whole segment. LPRR and SUP have very similar performance, showing the near-optimality of our proposed scheme. LPRR has much smaller access cost than RAN and LG, partly because LPRR optimizes content replication exploiting available indirect cost in the designed redundant frame structure whenever possible.

Figure 4 plots access cost versus size of redundant window size δ. (Note that $\delta = 0$ corresponds to the case when only I-frames are used to encode heads of coding units for view-switching.) We observe that the access cost decreases with δ; a larger redundant window means that the likelihood of indirect hit and differential transmission increases, so the repository just needs to transmit the pre-encoded differentials instead of a large I-frame, leading to a lower access cost. LPRR can reduce access cost substantially as compared to LG and RAN.

Figure 5 plots access cost versus number of servers. We observe that access cost decreases with the capacity of servers, because servers can replicate more views to serve the request directly. We can see that LRRR outperforms RAN and LG significantly in access cost.

VII. Conclusion

In this paper, we consider the problem of optimally replicating content for an IMVS network with distributed servers to support users watching multiview video while periodically requesting inter-view or temporal switches. We show that by using a redundant frame structure, the repository access cost can be lowered via "indirect hit". By replicating and locally forwarding correlated views in servers that are different from the requested views, the repository only needs to transmit pre-encoded frame differentials between the replicated views and the requested views. We propose a LP-based strategy with integer rounding to replicate segments of different movies among multiple servers, so that users can access the requested data with low access cost. Our simulation results show that our proposed scheme significantly outperforms a commonly used replication scheme and a state-of-the-art replication scheme in terms of access cost, and that it is very close to the LP-relaxed optimal solution.

References

[1] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound—100 camera and microphone system," in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, July 2006.

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

[3] H. Huang, B. Zhang, S.-H. G. Chan, G. Cheung, and P. Frossard, "Coding and caching co-design for interactive multiview video streaming," in *Proc. of the 31th Annual IEEE Conference on Computer Communic ations (INFOCOM'12) mini-conference*, 2012.

[4] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1461–1473.

[5] Z. Chen, M. Zhang, L. Sun, and S. Yang, "Delay-guaranteed interactive multiview video streaming," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, may 2009, pp. 1795 –1798.

[6] S.-H. Chan and F. Tobagi, "Distributed servers architecture for networked video services," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 125–136, Apr 2001.

[7] W.-P. K. Yiu, X. Jin, and S.-H. G. Chan, "VMesh: Distributed segment storage for peer-to-peer interactive video streaming," *IEEE Journal on Selected Areas in Communications (JSAC) special issue on Advances in Peer-to-Peer Streaming Systems*, vol. 25, no. 9, pp. 1717–31, Dec. 2007.

[8] S. Borst, V. Gupta, and A. Walid, "Self-organizing algorithms for cache cooperation in content distribution networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 2, pp. 71–72, 2009.

[9] J. Lv, X. Cheng, Q. Jiang, J. Ye, T. Zhang, S. Lin, and L. Wang, "LiveBT: Providing video-on-demand streaming service over bittorrent systems," *International Conference onParallel and Distributed Computing Applications and Technologies*, pp. 501–508, 2007.

[10] N.-M. Cheung, A. Ortega, and G. Cheung, "Distributed source coding techniques for interactive multiview video streaming," in *27th Picture Coding Symposium*, Chicago, IL, May 2009.

[11] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.