Optimal Server Bandwidth Allocation for Streaming Multiple Streams via P2P Multicast

Aditya Mavlankar, Jeonghun Noh, Pierpaolo Baccichet, and Bernd Girod

Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford CA 94305, USA
{maditya, jhnoh, bacci, bgirod}@stanford.edu

***Abstract*—We consider the general scenario where content hosted by the server comprises streams and each peer can subscribe one or more streams. Multiple multicast trees are built to deliver the streams to respective peers while exploiting the overlap of their interests for efficient and scalable delivery. We propose an optimization framework for allocating server bandwidth to minimize distortion across the peer population. We apply the framework to a novel application, peer-to-peer (P2P) multicast live video streaming with virtual pan/tilt/zoom functionality. In this application, each user can watch arbitrary regions of a high-spatial-resolution scene yet the system exploits overlapping interests by building multicast trees. Experimental results indicate that optimal server bandwidth allocation enhances the delivered quality across the peer population.**

I. Introduction

Peer-to-peer (P2P) multicasting is appealing as it requires much less server resources compared to a content delivery network (CDN) and is self-scaling as the resources of the network increase with the number of users. Recently, numerous academic and commercial Internet P2P video streaming systems have become available, for example [1]–[4]. Many protocols proposed in the literature [5]–[9] build one or more complementary multicast trees to push a video stream to interested peers.

We consider the general scenario where content hosted by the server comprises streams and each peer can subscribe one or more streams. A server hosting multiple video channels, for instance, falls under this category. To deliver each stream to respective peers, one or more distribution trees rooted at the server are built. Without loss of generality, we assume that a peer needs to connect to only one of these trees to receive the stream. The server has to expend more bandwidth for feeding more trees. Nevertheless, having more trees reduces the impact of peer-churn on the delivery of the stream. We propose an optimization framework to allocate server bandwidth for forming the optimal number of trees for each stream. Although we focus on minimizing the total distortion across the peer population, other goals such as min-max fairness can also be accommodated in the proposed framework.

Our approach takes into account the following properties that are characteristic of each stream; the membership size, the churn resulting from peers joining and leaving, and the rate-distortion tradeoff associated with the content. In this paper, we apply the framework to interactive region-of-interest (IROI) P2P video multicast.

IROI P2P video multicast allows users to watch arbitrary regions-of-interest (ROI) from the scene with arbitrary spatial resolutions (zoom factors). The system offers interactive virtual pan/tilt/zoom whereas the interactive features of proposed P2P multicast video systems in the literature are generally limited to video-on-demand and/or VCR-like functionality. We build a P2P overlay using a distributed protocol. The goal is to exploit the overlap in the ROIs of the peers by adapting the topology of the overlay according to the changing ROIs. This enables the server to host a live IROI video session with modest uplink capacity and the system scales well with increasing number of peers. Our distributed IROI P2P protocol is based on the Stanford P2P Multicast (SPPM) protocol [10], [11].

The optimization framework for the general scenario is presented in Section II. Section III describes the IROI P2P video multicast system and analyzes how the diversity in the requested streams grows with the size of the peer population. Section IV analyzes the gains by applying the framework to the system compared to alternative simpler allocation strategies.

II. Server Bandwidth Allocation

Let $\mathbf{P}$ denote the set of all peers. Let $\mathbf{S}$ denote the set of all streams. The server bandwidth is denoted by R_S.

Assume that we are given the following quantities related to stream j, $\forall j \in \mathbf{S}$:

- m_j: Number of peers interested in stream j.
- r_j: Bitrate of stream j.
- d_j: Distortion reduction per frame-interval associated with successful delivery of stream j.
- μ_j: Exponential distribution parameter for stream j. The contiguous period for which a peer requires stream j is an exponentially distributed random variable. A peer can subscribe and unsubscribe a stream multiple times.

We assume that multiple multicast trees are built to serve the streams to respective peers. For stream j, the server allocates bandwidth to grow up to α_j trees. Each tree uses a different peer as the direct descendant of the server. Associated with stream j, random variable D_j represents total distortion reduction over all peers that subscribe stream j. The goal of

This work has been supported, in part, by the Max Planck Center for Visual Computing and Communication, Deutsche Telekom Laboratories, and Sharp Laboratories of America.

the optimization problem is to determine α_j, the number of trees, rooted at the server, formed for stream j.

$$\max_{\alpha_j,\ \forall j \in \mathbf{S}} E \sum_{j \in \mathbf{S}} D_j\left(d_j, m_j, \mu_j, \alpha_j\right)$$

$$\text{subject to}$$

$$\sum_{j \in \mathbf{S}} \alpha_j r_j \leq R_S,$$

$$\alpha_j \in \mathbb{W},\ \alpha_j \leq m_j \tag{1}$$

During the multicast session, the optimization problem can be solved repeatedly, each time for a short time-horizon T. The probability $\tilde{p_j}$ that a peer unsubscribes stream j in this interval is given by $\tilde{p_j} = \left(1 - e^{-\mu_j T}\right)$.

The distortion reduction per frame-interval for stream j over all peers that subscribe stream j can be written as

$$D_j\left(d_j, m_j, \mu_j, \alpha_j\right) \quad \geq \quad \sum_{i \in \mathbf{P_j}} d_j\left(1 - Y_{ij}\right) \tag{2}$$

where

- $\mathbf{P_j}$ is the set of peers interested in stream j; $|\mathbf{P_j}| = m_j$.
- Y_{ij}: Indicator random variable which is 1 if peer i suffers disconnection due to an ancestor leaving the tree or if peer i itself unsubscribes the stream in time-interval T; it indicates that peer i fails to contribute to the total distortion reduction.

The inequality sign in (2) accommodates the fact that peers unsubscribing or suffering disconnection might receive the stream for part of the time-horizon T. Assuming that T is short enough, we can write

$$\begin{aligned} E\left[D_j\left(d_j, m_j, \mu_j, \alpha_j\right)\right] &\approx m_j d_j - d_j \sum_{i \in \mathbf{P_j}} E\left(Y_{ij}\right) \\ &= m_j d_j - d_j \sum_{i \in \mathbf{P_j}} \Pr\left\{Y_{ij} = 1\right\} \end{aligned}$$

Let random variable $\tilde{M_j}$ denote the number of peers that fail to contribute to the total distortion reduction associated with stream j; i.e., $\tilde{M_j} = \sum_{i \in \mathbf{P_j}} Y_{ij}$. In general, the lower the average tree-height for stream j the lower the value of $E\left(\tilde{M_j}\right)$. Given the structure of the trees for stream j, $E\left(\tilde{M_j}\right)$ can be obtained by summing the probabilities $\Pr\{Y_{ij} = 1\}$. If peer i is l hops away from the server, then $\Pr\{Y_{ij} = 1\} = 1 - \left(1 - \tilde{p_j}\right)^l$. These probabilities have been analyzed in detail, for example in [12], also for scenarios where forward error correction (FEC) is used for error-resilient P2P multicast video streaming. Consider the example shown in Fig. 1. We can compute $E\left(\tilde{M_j}\right)$ for increasing α_j assuming that regular degree-bounded trees are constructed. We observed that for regular degree-bounded trees, for a wide range of $(\tilde{p_j}, m_j)$, the expected number of peers that fail to contribute to the distortion reduction can be modeled as $m_j e^{-\frac{a\alpha_j}{(\tilde{p_j})^b (m_j)^c}}$ with a suitably-fit constant triplet of model parameters (a, b, c). Figure 2 shows an example of the model.

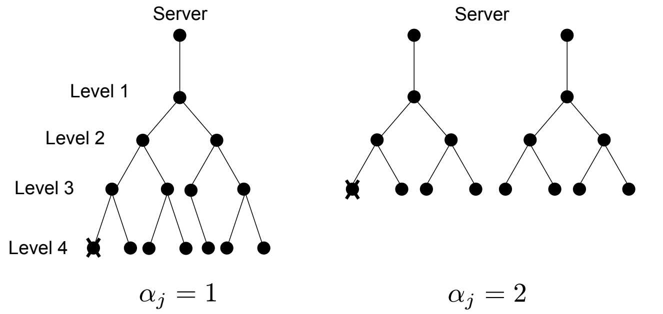


Fig. 1. Example with $m_j = 14$ peers. On the left, $\alpha_j = 1$ tree is constructed and on the right, $\alpha_j = 2$ trees are constructed. More trees lower the average tree-height, however, the server has to spend more bandwidth.

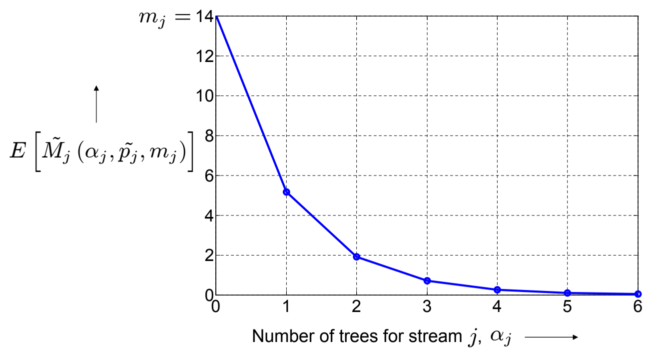


Fig. 2. The larger the number of trees the smaller the expected number of peers that fail to contribute to the total distortion reduction for stream j.

The expected distortion reduction by allocating α_j trees (i.e., rate $\alpha_j r_j$) to stream j is

$$E\left[D_j\left(d_j, m_j, \mu_j, \alpha_j\right)\right] = m_j d_j - m_j d_j e^{-\frac{a\alpha_j}{(\tilde{p_j})^b (m_j)^c}} \tag{3}$$

The expected distortion reduction per unit rate by adding the α_j^{th} tree, given that $\alpha_j - 1$ trees have already been allocated, is given by

$$\Theta_j\left(\alpha_j\right) = \frac{E\left[D_j\left(d_j, m_j, \mu_j, \alpha_j\right)\right] - E\left[D_j\left(d_j, m_j, \mu_j, \alpha_j - 1\right)\right]}{r_j} \tag{4}$$

Note that $\Theta_j\left(\alpha_j\right) > \Theta_j\left(\alpha_j + 1\right)$. Hence our optimization problem (1) can be cast as a classic knapsack problem. A greedy solution can be obtained by sorting all $\Theta_j(1) \cdots \Theta_j(m_j)$, $\forall j \in \mathbf{S}$ and allocating trees till the bit-budget is exhausted.

To illustrate the generality of the framework, various modifications to the above optimization problem can be mentioned: min-max fairness, distortion-oblivious maximization of expected number of peers that receive a stream, adaptation of model of $E\left(\tilde{M_j}\right)$ to better suit the distributed P2P protocol, e.g., speed of rejoin procedure compared to the time-horizon T, etc. Although we focus on multicast trees, the approach is also applicable to mesh-based topologies.

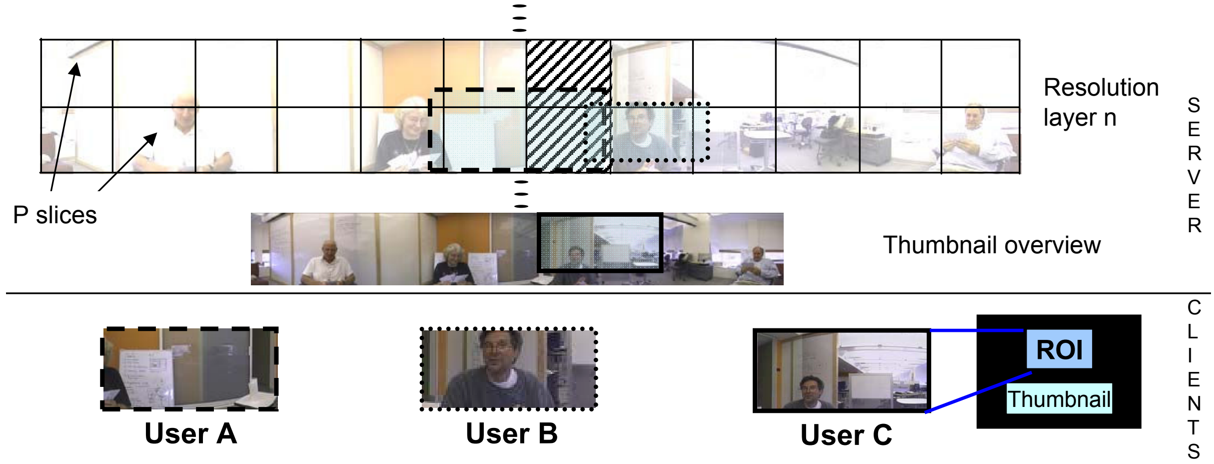


Fig. 3. An ROI is rendered from a portion of the multi-resolution representation. The multi-resolution representation consists of dyadically spaced resolution layers and the base layer. The base layer is displayed as the thumbnail. The ROIs of three users, illustrated within the multi-resolution representation, appear to be of different sizes due to arbitrary non-dyadic zoom factors. The slices shown shaded are required by two users. The video screenshot is taken from the *Cardgame* sequence used for our experiments.

III. IROI P2P VIDEO MULTICAST

A. User Interface and Video Coding Scheme

We have developed a graphical user interface which allows the user to select the ROI while watching the video. The ROI location and zoom factor are controlled by operating the mouse. The application supports continuous zoom to provide smooth control of the zoom factor. In addition to the ROI, we also display a thumbnail overview with a rectangular box overlaid to show the location of the ROI.

We require a video coding scheme that efficiently supports random access to arbitrary ROIs, while keeping the transmission rate as low as possible. We have proposed such a video coding scheme in our earlier work [13]. The coded representation of the scene consists of the thumbnail version and versions with different spatial resolutions that are dyadically spaced.

The thumbnail overview, which we will also refer to as the base layer video, is coded with H.264/AVC using I, P and B pictures. The reconstructed base layer video frames are upsampled by a suitable factor and used as prediction for encoding the video corresponding to the higher resolution layers. Each frame belonging to a higher resolution layer is coded using a grid of rectangular P slices. Employing only upward prediction enables efficient random access to local regions within any spatial resolution. For every frame interval, the display of the client is rendered by transmitting the corresponding frame from the base layer and few P slices from exactly one higher resolution layer. We transmit slices from that resolution layer which corresponds closest to the user's current zoom factor. At the client's side, the corresponding ROI from this resolution layer is resampled to correspond to the user's zoom factor. If some enhancement layer P slices are unavailable, we perform error concealment by upsampling portions of the thumbnail video signal.

B. Distributed P2P Multicast Protocol

Users participating in the video multicast request different regions of the video at different zoom factors according to individual ROIs. Fig. 3 illustrates an example of the system serving three users. To exploit overlaps of ROIs for efficient and scalable delivery, we build one multicast tree for the base layer and one multicast tree each for every slice of the higher resolution layers, also called enhancement layer slices. Every client subscribes to the base layer tree for the entire session. Depending on its current ROI, each client further subscribes to the trees corresponding to required enhancement layer slices. Clients also dynamically unsubscribe slices that are no longer required for the current ROI. Notice that in case the user's zoom factor corresponds closer to the base layer than any other resolution layer, then the peer needs to subscribe only to the base layer; i.e., the ROI would be rendered using part of the thumbnail, for example user C in Fig. 3.

In our earlier work [14], we have proposed ROI predictors for predicting the user's ROI ahead of time and initiating the join-process on new trees in advance. This enables low latency of interaction. Whenever the ROI prediction indicates a change of ROI, the peer sends an ROI-switch request to the server. This consists of the top-left and bottom-right slice IDs of the old ROI as well as the new ROI. The server maintains a database of slices that each peer is currently subscribed to. In response to the ROI-switch request, the server sends a list of potential parents for every new slice that the peer needs to subscribe. The server also immediately updates its database assuming that the peer will be successful in updating its subscriptions. Upon receiving the list, the peer tries to connect to the new trees by following the remaining steps of the six-way handshake that is employed in the SPPM protocol [10], [11]. If a peer is informed of a parent's unsubscription or if it detects that the parent has left, then it requests a potential parents' list from the server for that distribution tree. A detailed description of the IROI P2P video multicast system can be found in [15].

C. Diversity among Requested Slices

The base layer is coded as one slice and can be considered as one stream. Similarly, each enhancement layer slice is also a stream. A peer thus subscribes to a few streams to render its ROI[1]. In the following analysis, we determine the expected number of distinct slices streamed by the server. This helps tell how much bandwidth should be provisioned at the server to cater slices required for a certain size of peer population.

Let $|\mathbf{P}| = P_t$ peers and $|\mathbf{S}| = S_t$ slices. Let the Bernoulli random variable X_{ij} be 1 if peer $i \in \mathbf{P}$ needs slice $j \in \mathbf{S}$. Let $\Pr\{X_{ij} = 1\} = p_j, \forall i \in \mathbf{P}$. Let indicator random variable $X_j^{P_t}$ be 1 if *at least* one out of P_t peers needs slice j.

Random variable N_S denotes the number of distinct slices streamed by the server

$$\begin{aligned} N_S &= \sum_{j\in\mathbf{S}} X_j^{P_t} \\ E\{N_S\} &= \sum_{j\in\mathbf{S}} E\left\{X_j^{P_t}\right\} \\ &= \sum_{j\in\mathbf{S}} \Pr\{X_j^{P_t} = 1\} \\ &= \sum_{j\in\mathbf{S}} 1 - \Pr\{X_j^{P_t} = 0\} \\ &= \sum_{j\in\mathbf{S}} 1 - (1-p_j)^{P_t} \end{aligned} \quad (5)$$

This uses the fact that random variables $X_{ij}, \forall i \in \mathbf{P}$ are independent *given* p_j. They represent independent coin tosses with a coin having bias p_j. For any peer, say peer i, we do not assume independence/uncorrelatedness between random variables $X_{ij}, \forall j \in \mathbf{S}$ since the knowledge of X_{ij} affects the probability of peer i subscribing slices neighboring to slice j.

Let random variable N_i denote the number of slices needed by peer i. Let $E(N_i)$ be denoted by κ. Its value depends on the slice sizes and the dimensions of the ROI display and can be obtained as shown in [13].

$$\begin{aligned} E\{N_i\} &= \sum_{j\in\mathbf{S}} E\{X_{ij}\} \\ &= \sum_{j\in\mathbf{S}} p_j \end{aligned} \quad (6)$$

Under the constraint $\sum_{j\in\mathbf{S}} p_j = \kappa$, it can be shown that

$$\begin{aligned} \max_{p_j,\, \forall j\in\mathbf{S}} E\{N_S\} &= \sum_{j\in\mathbf{S}} 1 - \left(1 - \frac{\kappa}{S_t}\right)^{P_t} \\ &= S_t\left[1 - \left(1 - \frac{\kappa}{S_t}\right)^{P_t}\right] \end{aligned} \quad (7)$$

The maximum occurs when $p_j = \frac{\kappa}{S_t}, \forall j \in \mathbf{S}$; i.e., when the slices are equally popular. A plot with $S_t = 141$ slices

[1]In the following, we use slice and stream interchangeably.

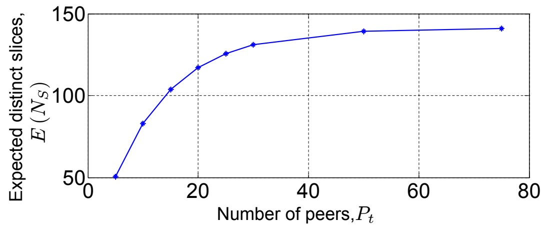


Fig. 4. Expected number of distinct slices streamed by the server saturates at S_t. Here shown for $S_t = 141$ slices and $\kappa = 12$ slices.

and $\kappa = 12$ slices is shown in Fig. 4. However, it should be noted that our proposed optimization solution is applicable irrespective of how much aggregate bandwidth is provisioned by the server.

IV. Experimental Results

We use the *Cardgame* sequence[2] having 3584x512 pixels and 25 frames/sec. Fig. 3 shows a frame of the sequence. It is a 360° panoramic video sequence stitched from several camera views. The camera setup is stationary and only the four card players move. We encode the thumbnail version at resolution 896x128 with an intraframe period of 15 frames using two consecutive B frames between anchor frames. The ROI display is 480x240 pixels. The two resolution layers in the coded representation have 1752x256 pixels (matches zoom factor of 1) and 3584x512 pixels resolution. These are encoded using slice sizes 64x256 and 128x128 respectively; i.e., the first layer has 28 slices horizontally and 1 slice vertically, whereas the second layer has 28 slices horizontally and 4 slices vertically. Every frame of the base layer is coded as one slice, hence, the server hosts 141 streams corresponding to 141 slices. The user's zoom factor is restricted between 1 and 6. The PSNR @ bitrate for the thumbnail is about 39.1 dB @ 162 kbps. The total data rate for the thumbnail and the ROI required by each peer is about 900 kbps on average. The uplink and downlink capacities of each peer are set to 2 Mbps. We implemented the distributed IROI P2P protocol within the NS-2 network simulator [16]. We recorded 100 ROI trajectories constituting navigation paths of as many peers. Each 1-minute-long trajectory starts at a random location. The 1-minute-long video sequence is obtained by looping a set of 298 frames. Peers are on for the entire session.

We observed that, for this sequence, stream-membership sizes have low variance, as can be seen in Fig. 5. We assume an oracle for predicting the user's future ROI and focus on the comparison of three server bandwidth allocation schemes: 1) our optimal allocation proposed in Section II, 2) round-robin allocation starting from lowest slice ID, and 3) having the same cap on number of trees for each slice. For the same-cap scheme, if the server runs out of bandwidth, then it does not form new trees until one of the direct descendants leaves. The optimal allocation is computed using T equal to 2 frame-intervals, the stream-membership sizes are averaged over the

[2]Sequence provided by the Stanford Center for Innovations and Learning (SCIL), Stanford University, Stanford, CA.

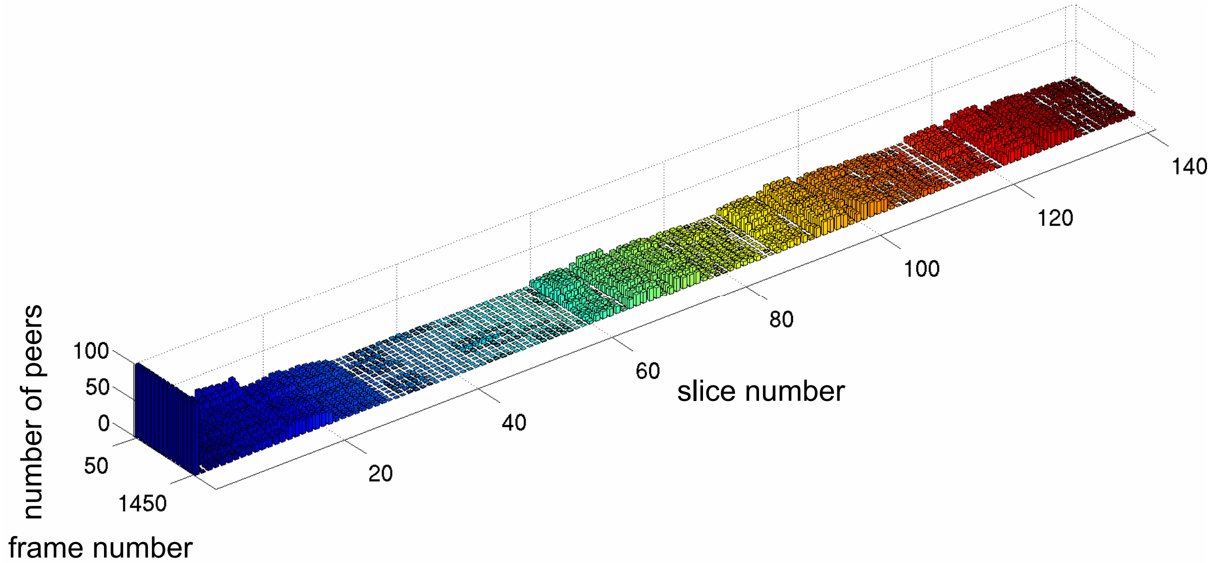
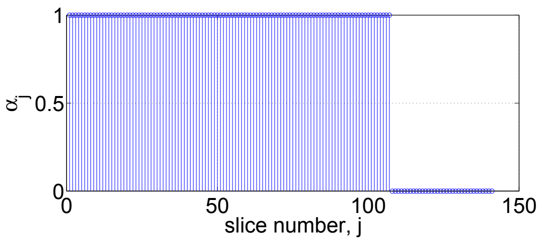
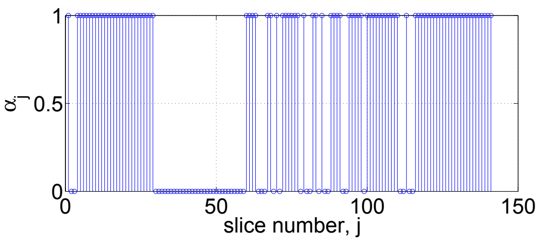


Fig. 5. In spite of churn due to changing ROIs of peers, stream-membership sizes have low variance over the interactive streaming session.

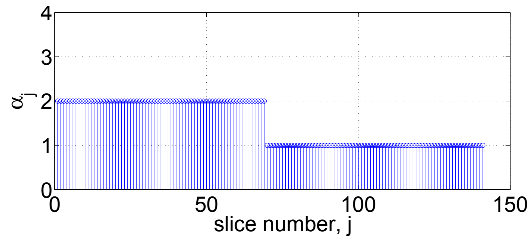


(a) Round-robin allocation.

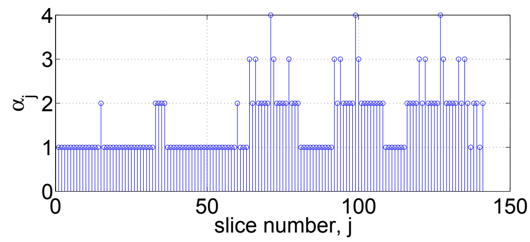


(b) Optimal allocation.

Fig. 6. Round-robin and optimal allocations with server bandwidth of 8 Mbps. The Y-axis shows the number of trees for which bandwidth is reserved at the server.



(a) Round-robin allocation.

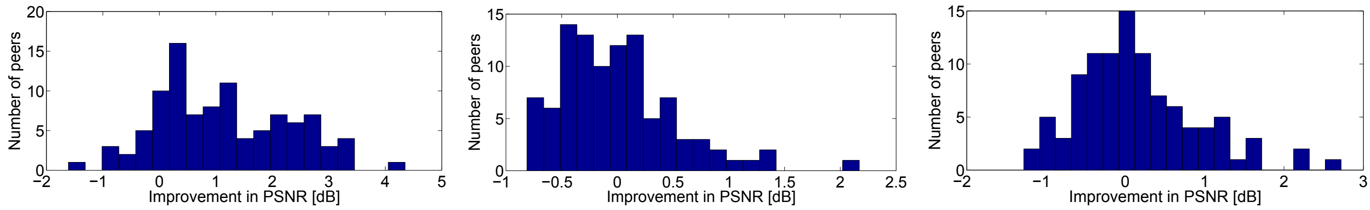


(b) Optimal allocation.

Fig. 7. Round-robin and optimal allocations with server bandwidth of 15 Mbps. The Y-axis shows the number of trees for which bandwidth is reserved at the server.

entire trajectory-length, and the allocation is held constant. Since the focus is on exploring the gain potential, we assume that average rate and distortion reduction over all frames is available for every slice. Figs. 6 and 7 show the optimal and round-robin allocations for server bandwidth of 8 Mbps and 15 Mbps respectively.

The average lower bound for the PSNR is 33 dB and the average upper bound is 39.8 dB. The lower bound is the PSNR that results when no enhancement layer slices reach the peer, whereas the upper bound corresponds to the case when all required enhancement layer slices reach the client in time. The reference for the PSNR calculation is the ROI rendered from the original uncompressed multi-resolution video. Table I lists the average drop in PSNR from the upper bound for the three schemes. The largest gain of the proposed allocation method can be seen when the server bandwidth is 8 Mbps; i.e., less than the rate of the multi-resolution representation, which is about 10 Mbps. The optimal scheme allocates a cap of zero for the least important slices, whereas the round-robin scheme wastes resources by reserving bandwidth for these slices. With the same-cap scheme, freed bandwidth due to a direct descendant leaving is recycled to serve new slices, and hence its performance is in between. However, if the server bandwidth is more than the rate of the multi-resolution representation but less than twice of it, then the same-cap scheme fails to make use of the additional resources. The round-robin scheme allocates all available resources, however, it might waste resources in serving unimportant slices. This



(a) Server bandwidth 8 Mbps (b) Server bandwidth 12 Mbps (c) Server bandwidth 15 Mbps

Fig. 8. Histograms of improvement in PSNR with the optimal allocation over the round-robin allocation.

is the reason why its performance does not improve when the server bandwidth is increased from 12 Mbps to 15 Mbps.

TABLE I
AVERAGE DROP IN PSNR FROM UPPER BOUND SHOWN FOR THE THREE ALLOCATION SCHEMES.

	Scheme		
server bandwidth (in Mbps)	optimal (drop in dB)	round-robin (drop in dB)	same-cap (drop in dB)
8	1.9	3.0	2.2
12	1.6	1.6	1.9
15	1.4	1.6	1.9

Figure 8 shows the histograms of improvement in PSNR with the optimal allocation over round-robin. Although some peers experience a drop in quality, the drop is always less than the biggest improvement in PSNR among all peers.

V. CONCLUSIONS

We consider the broad scenario where the server hosts multiple streams and a peer can subscribe one or more streams. We propose an optimization framework to allocate the server bandwidth among the streams. The generality of the framework makes it applicable to other scenarios apart from the one considered in this paper; examples include a server hosting multiple multicast video sessions, mesh-based instead of tree-based topology, etc. The framework is amenable to a change of metric as well as replacement of the model of number of peers suffering disconnection.

Mesh-based approaches often lend more robustness at the cost of longer playout delay and redundancy in received data. However, due to the strict latency constraint of the interactive application, we prefer the tree-based push approach for IROI P2P. Application of the optimization framework yielded a gain of up to 1 dB in mean PSNR of rendered ROI compared to round-robin resource allocation. We found that the gains are particularly high when the resources are constrained. When the server provisions enough bandwidth to support diverse demand, we get lower gains because our IROI P2P protocol effects speedy rejoin to combat effects of churn. Nevertheless, churn-aware rate allocation still yields gain because when a peer loses a parent, it is bound to suffer disconnection till it finds an alternate parent. This is especially true since we apply the push approach and also the strict latency constraint does not leave much time for local retransmissions.

For both the IROI application as well as multiple video sequences, the following improvements are desirable. A dynamic video sequence with large change in the stream-memberships might result in a large change in the optimal allocation. However, changing the number of multicast trees for the streams on-the-fly might entail active topology rearrangement and induce churn. Optimal decision-making at the server as well as P2P protocol for topology rearrangement are required. Also, some parameters like bitrate and distortion should be ideally estimated online. Future work will focus on these improvements.

REFERENCES

[1] "SopCast," online: http://www.sopcast.org/.
[2] "PPLive," online: http://www.pplive.com/.
[3] "Coolstreaming," online: http://www.coolstreaming.us/.
[4] "GridMedia," online: http://www.gridmedia.com.cn/.
[5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 205–217, 2002.
[6] Y.-H. Chu, A. Ganjam, T. S. E. Ng, S. G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," in *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*, Berkeley, CA, USA, 2004, pp. 12–12.
[7] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast (keynote address)," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 28, no. 1, pp. 1–12, 2000.
[8] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *IEEE International Conference on Network Protocols (ICNP)*, Atlanta, USA, 2003, pp. 16–27.
[9] D. A. Tran, K. A. Hua, and T. T. Do, "Zigzag: An efficient peer-to-peer scheme for media streaming," in *IEEE INFOCOM*, San Francisco, CA, USA, 2003.
[10] E. Setton, J. Noh, and B. Girod, "Rate-distortion optimized video peer-to-peer multicast streaming," *Proc. of Workshop on Advances in Peer-to-Peer Multimedia Streaming at ACM Multimedia*, pp. 39–48, Nov. 2005, invited paper.
[11] E. Setton, P. Baccichet, and B. Girod, "Peer-to-peer live multicast: A video perspective," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 25–38, Jan. 2008.
[12] G. Dan, I. Chatzidrossos, V. Fodor, and G. Karlsson, "On the performance of error-resilient end-point-based multicast streaming," in *Proc. of 14th IEEE International Workshop on Quality of Service (IWQoS)*, Jun. 2006, pp. 160–168.
[13] A. Mavlankar, P. Baccichet, D. Varodayan, and B. Girod, "Optimal slice size for streaming regions of high resolution video with virtual pan/tilt/zoom functionality," *Proc. of 15th European Signal Processing Conference (EUSIPCO), Poznan, Poland*, Sept. 2007.
[14] A. Mavlankar, D. Varodayan, and B. Girod, "Region-of-interest prediction for interactively streaming regions of high resolution video," *Proc. of 16th Intl. Packet Video Workshop, Lausanne, Switzerland*, pp. 68–77, Nov. 2007.
[15] 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. of IEEE International Conference on Image Processing (ICIP)*, San Diego, CA, USA, Oct. 2008, in press.
[16] "The Network Simulator - ns-2," online: www.isi.edu/nsnam/ns.