# STANFORD PEER-TO-PEER MULTICAST (SPPM) – OVERVIEW AND RECENT EXTENSIONS

*Jeonghun Noh, Pierpaolo Baccichet*, Frank Hartung†, Aditya Mavlankar, and Bernd Girod*

Department of Electrical Engineering, Stanford University
350 Serra Mall, Stanford, CA 94305, USA
Email: jhnoh@stanford.edu, pier@dyyno.com, frank.hartung@ericsson.com, maditya@stanford.edu, bgirod@stanford.edu

*(Invited Paper)*

### ABSTRACT

We review the Stanford Peer-to-Peer Multicast (SPPM) protocol for live video streaming and report recent extensions. SPPM has been designed for low latency and robust transmission of live media by organizing peers within multiple complementary trees. The recent extensions to live streaming are time-shifted streaming, interactive region-of-interest (IRoI) streaming, and streaming to mobile devices. With time-shifting, users can choose an arbitrary beginning point for watching a stream, whereas IRoI streaming allows users to select an arbitrary region to watch within a high-spatial-resolution scene. We extend the live streaming to mobile devices by addressing challenges due to heterogeneous displays, connection speeds, and decoding capabilities.

***Index Terms*—** peer-to-peer streaming, time-shifting, region-of-interest, transcoding, mobile streaming

## 1. INTRODUCTION

Over the last few years, peer-to-peer video streaming has attracted increasing attention due to its scalability and its potentially lower cost for delivering media streams to a large population of users [1–5]. For live video streaming, we have developed the Stanford Peer-to-Peer Multicast (SPPM) system with focus on low-latency and cross-layer optimized video transport. Live contents are disseminated to users via multiple complementary multicast trees explicitly built among the peers. The degradation of video quality due to peer churn and network packet loss is alleviated by network-aware and video-aware packet handling algorithms.

In this paper, we provide an overview of the original design of SPPM (Section 2) and report its recent extensions towards time-shifted streaming (Section 3), interactive region-of-interest (IRoI) streaming (Section 4), and streaming to mobile devices (Section 5). Time-shifted streaming allows users to pause a live video stream, rewind (even to a scene before the one when they joined the session), and fast forward. With IRoI streaming, users can interactively watch a region of a high-spatial-resolution video. SPPM extension to stream to mobile devices involves multicasting to a heterogeneous peer population of fixed and mobile devices with different decoding and display capabilities.

*Now with Dyyno, Inc., Palo Alto, CA

†Work performed while on leave from Ericsson Eurolab, Herzogenrath, Germany

## 2. STANFORD PEER-TO-PEER MULTICAST

Similar to CoopNet [1] and SplitStream [2], SPPM organizes peers in an overlay of multiple complementary trees. Every tree is rooted at the video source. The media stream, originating from the video source, is packetized and distributed on different trees such that there are no duplicate packets across the trees. Peers subscribe to every tree in order to receive the media stream contiguously.

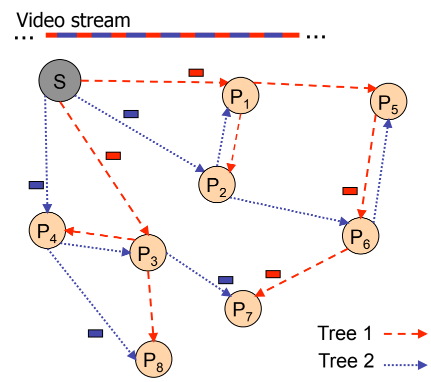


**Fig. 1**. A peer-to-peer overlay consisting of two complementary distribution trees. The encoded video is packetized and split into disjoint substreams, one for each distribution tree.

### 2.1. Overlay Construction and Management Protocol

Fig. 1 illustrates an example overlay in which two complementary trees are constructed among a small group of peers. Typically, we use 4 to 8 trees, and peer groups might be much larger. We have run experiments with up to 10,000 peers. As peers join the system, the trees are incrementally constructed in a distributed manner. When a new peer contacts the video source, the video source replies with session information, such as the number of multicast trees and the video bit rate. It also sends a list of candidate parents randomly chosen from the table of participating peers it maintains. The new peer then probes each candidate parent to know about their current status. After receiving probe replies, the best candidate parent is selected and contacted for each tree by minimizing the height of the distribution tree. Once the selected candidate parent accepts the attachment request, a data connection is established between the parent and the new peer. After data transmission starts, each child peer periodically sends *hello* messages to their parents. When a peer leaves the system ungracefully, its parents detect it by observing consecutive missing *hello* messages, and stop forwarding video to the child. The departing peer's children notice that neither video packets nor a response to *hello* messages arrive. Each abandoned child then initiates procedure to connect to a different parent node in the tree.

### 2.2. Cross layer optimization

To provide the best video quality despite congestion and peer churn, SPPM employs sender-driven packet scheduling [6] in conjunction with receiver-driven retransmission requests [7]. Acting as a relay, each peer schedules the next packet to forward by comparing the importance of packets in its output queue. As a receiver, each peer evaluates the importance of missing packets by computing the expected video quality improvement if the corresponding frame is received before its playout deadline. Retransmission requests of selected missing packets are then sent to alternative parents. Since packet losses typically occur in rare bursts, such feedback error control is generally superior to forward error control. Feedback implosion, prohibiting feedback error control in network-layer multicast, is not an issue for a P2P overlay because retransmissions are handled locally between a parent and a child peer.

## 3. TIME-SHIFTED STREAMING

Time-shifted streaming allows viewers to individually pause a live video stream, rewind (even to a scene before the one when they joined the session), and fast forward. In a P2P system like SPPM, this functionality can be provided by storing a portion of the received stream at each peer and serving it to other peers at a later time. Using each peer's local buffer for storage and relay while peers receive the stream can free the video source from most requests for previous-to-live scenes. The concept of peer-assisted time-shifting in a live session is related to asynchronous video streaming in peer-assisted video-on-demand (VoD) systems [3–5, 8].

Let $V(t)$ denote a live stream at time $t$. Suppose that a peer joins at time $t$, but desires to watch the video starting from $V(t-d)$. We distinguish between LS (live streaming) peers with $d = 0$, and TS (time-shifted streaming) peers with $d > 0$. For TS peers, we need to delay the live stream by $d$ somewhere between the source and the peers. The video source maintains the list comprising each peer's identifier, time of joining, beginning video time-stamp, and the estimated latest time-stamp of video contained in its buffer. When the source receives a join request from a peer, it returns a list of randomly chosen parent candidates whose video buffer contains the requested, possibly time-shifted video. The requesting peer then probes the candidates in the list. After the requesting peer receives probe replies, it selects the best parent candidate. For LS peers, the best candidate is selected by minimizing the number of logical hops to the source [9]. Using this criterion, peers attempt to reduce disruption of the live stream due to peer churn. However, TS peers are not susceptible to peer churn as long as they can avoid buffer underflow. Suppose that a child receives a time-shifted stream from its parent. When the parent has sufficient uplink bandwidth, the stream can be pushed faster than playout speed to the child. With *fast prefetching*, peer buffers can grow faster than playout speed, which reduces the urgency for acquiring later packets. It reduces buffer under-run and video disruption when the peer is disconnected from the overlay.

Fast prefetching can also help the server to lower its uplink data volume. Since peers disseminate video among themselves more rapidly, the data availability among peers increases. The higher data availability among peers allows more viewers to obtain video from other peers. The effects of fast prefetching are illustrated in Fig. 2. Child 1 receives the stream faster than the playback speed until Child 2 connects to the same parent. The uplink of the parent is twice the video bitrate. The trajectory of the received video over time is depicted for each peer. The dotted line illustrates the trajectory of Child 1's received video over time in case of no fast prefetching.

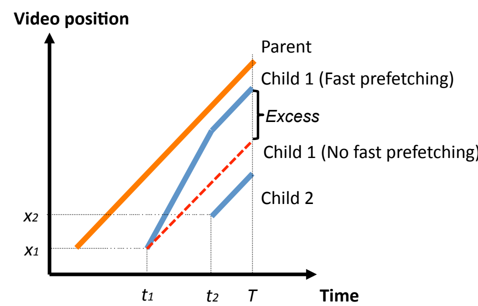


**Fig. 2**. Fast prefetching for time-shifted streams. At time $T$, the excess amount of video stream acquired by Child 1 due to fast prefetching is shown.

We performed experiments using the NS2 simulator. In the experiments, 150 peers join and leave repeatedly with an average lifetime of 120 seconds. On joining at time $t$, half of the peers start watching the stream from the live position. The rest of them begin watching the stream from a past position randomly selected between 0 and $t$. We show results for the *Mother and Daughter* sequence, encoded at 420 kbps. The video server's uplink bandwidth was set to $20R$ (video bitrate denoted by $R$) and the peer uplink bandwidth was $2R$. In Fig. 3, the number of direct children of the server is plotted over time for one simulation instance. As the live session proceeds, more TS peers connect to the server because less peers are found to have cached the requested video portion. With fast prefetching, however, server load (number of direct children at the server) was reduced by 44% on average. Peers experience 1.4 dB higher video quality on average compared to no prefetching. The results are summarized in Table 1.

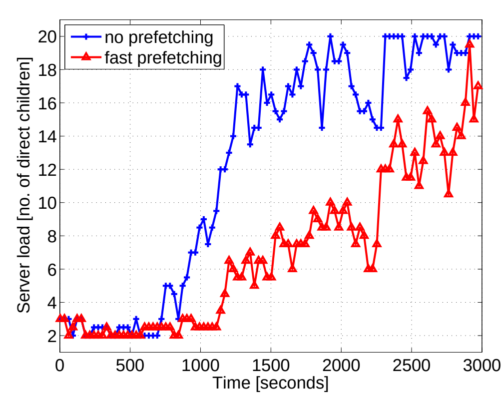


**Fig. 3**. Server load [number of direct children] over time.

|  | no prefetching | fast prefetching |
|---|---|---|
| Average video quality [dB] | 40.3 | 41.7 |
| Average server load [number of direct children] | 12.32 | 6.98 |

**Table 1**. Effects of fast prefetching.

## 4. REGION-OF-INTEREST STREAMING

With interactive region-of-interest (IRoI) streaming, different users can watch different portions of the high-spatial-resolution scene with arbitrary zoom factors [10]. The thumbnail overview, which serves as a base layer, is coded using I, P and B pictures of H.264/AVC. Each peer subscribes to this base layer multicast regardless of its current region-of-interest (RoI). Each higher-resolution layer [1] frame is divided into a two-dimensional raster of P slices as shown in Fig. 4. At the video source, the P slices are assigned to independent multicast trees. According to its RoI, each peer subscribes to the appropriate set of multicast trees to receive the relevant P slices. When the RoIs of the peers exhibit high overlap due to common interests in the scene, peers can share common P slices in a peer-to-peer manner to reduce server load.

We have extended SPPM to deliver the base layer and the P slices. When a peer changes its RoI, it triggers unsubscription from slices that are no longer required and subscription to new slices that now intersect the RoI. The peer sends RoI switch requests to the source, and the source replies with a list of parent candidates for each new slice. To display the RoI immediately after user input, it is predicted in advance and subscription to new slices is initiated beforehand ("Early Join"). However, the peer delays sending leave messages to its parent(s) as well as its children for slices it no longer requires ("Delayed Departure"). In [10], we have shown that SPPM can achieve significant server load reduction while delivering acceptable video quality despite RoI movements. The RoI movements produce a significant amount of short-lived slice subscriptions, which are a dominant source of overlay dynamics. Fig. 5 depicts the cumulative distribution function (cdf) of slice subscription duration, obtained from the empirical RoI trajectories of 200 users used in the experiments in [10]. The average slice subscription duration was only 8.76 seconds. Therefore, short join time and low end-to-end transmission delay are essential in an interactive P2P streaming system.

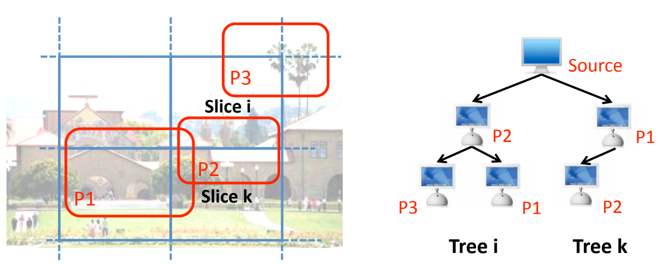


**Fig. 4**. Left: A 2D raster of P slices for one of the higher-spatial-resolution layers is shown. The boxes represent the RoIs of peers, P1, P2 and P3. Slices $i$ and $k$ are shared by more than one peer. Right: Trees consisting of more than one peer are shown. Each slice is distributed by its own tree.

[1]Multiple higher-resolution layers are hosted by the server to support a wide range of zoom factors.

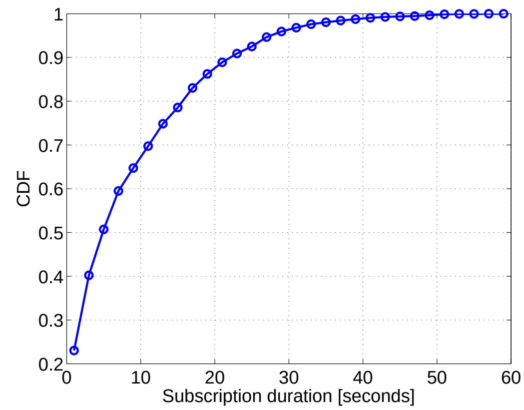


**Fig. 5**. CDF of P slice subscription duration.

## 5. STREAMING TO MOBILE PEERS

We are currently extending SPPM to stream video to a heterogeneous peer population of fixed peers and mobile peers (e.g. mobile phones). Because mobile devices are typically limited in their battery, display size, and wireless link speed, SPPM treats the mobile devices as leeches, i.e., peers that only receive packets, but do not relay the packets to other peers. Transcoding is required to deliver video streams to individual mobile devices to suit display size, decoding capability, and download speed of the cellular network [11].

To reduce an individual peer's transcoding overhead and video degradation due to peer churn, we employ distributed transcoding, under which $K$ parents collaboratively transcode the original stream into $K$ disjoint substreams. These substreams are transmitted and then assembled as if they were a single stream at the mobile. One possible solution, *interleaved distributed transcoding*, is illustrated in Fig. 6. In this scheme, the original frames are first downsampled to smaller frames in the spatial domain. The first frame in each GOP is coded as an I frame, and the following frames are coded as P frames predicted from the frame immediately preceding it in the substream. Parent $i$ codes Frame $i$, $K+i$, $2K+i$, $\ldots$, so that each parent transmits every $K^{th}$ frame in a disjoint manner. The I-frames are coded by all parents, yet transmitted by only one parent to avoid duplicate frames. Note that this scheme codes no B-frames and utilizes multiple reference frames for coding P-frames. This ensures that any typical H.264 decoder conforming to baseline profile [12], found in many high-end mobile phones, can decode the transcoded bitstream.

Fig. 7 shows the simulation results for conventional transcoding (1 parent) and interleaved distributed transcoding (2 and 4 parents). The *Foreman* sequence with CIF resolution was used as the original stream, coded at 590 kbps using H.264 main profile. The stream was transcoded using H.264 baseline profile to QCIF resolution. The solid curves show the rate-distortion performance with no packet loss. In the lossy scenario, the wireless channel is modeled by the Gilbert-Elliot model, as a discrete-time Markov chain with two states. Table 2 shows the Gilbert model transition probabilities based on the GSM network, presented in [13]. During the good state there is no packet loss, whereas during the bad state the channel produces packet loss with probability 0.5. In addition, a burst of packets is lost when a parent leaves. The lifetime of parents is exponentially distributed with an average of 90 seconds. The recovery time from a

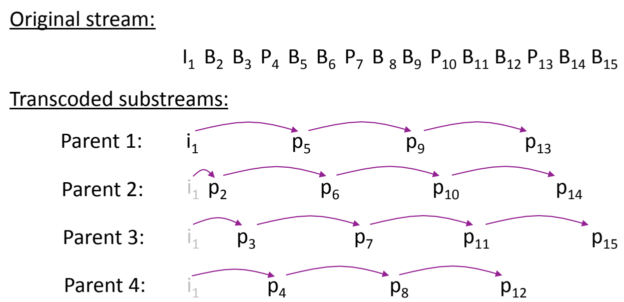


**Fig. 6**. Interleaved distributed transcoding: an example of 4 parents (transcoders) with a GOP of 15 frames. The high bitrate original video is transcoded into multiple disjoint substreams. Although the I-frame ($I_1$) is coded at all parents, it is transmitted only by Parent 1.

parent disconnect is 3 seconds. If a frame is either fully or partially not decodable, the last fully decoded frame is displayed in lieu of the current frame. The dashed curves show the degraded video quality in the lossy scenario, indicating that the rate-distortion performance of distributed coding is comparable to conventional transcoding while computational workload is distributed.

| State $i$ | Pr $(i)$ | Pr $(1\|i)$ | Pr $(0\|i)$ |
|---|---|---|---|
| 0 (Good) | 0.9449 | 0.0087 | 0.9913 |
| 1 (Bad) | 0.0551 | 0.8509 | 0.1491 |

**Table 2**. Probabilities used in the Gilbert-Elliot model.

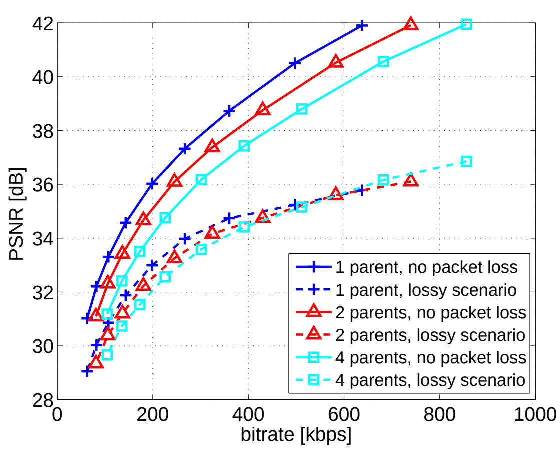


**Fig. 7**. Rate-distortion curves with and without packet loss for conventional transcoding (1 parent) and interleaved distributed transcoding (2 and 4 parents).

## 6. CONCLUSIONS

We provide an overview of SPPM, the Stanford Peer-to-Peer Multicast protocol for live video streaming, and report recent extensions that considerably expand its functionality. We show that a low-latency live multicast overlay can be easily extended to support time-shifted streaming. This is in contrast to extending non-real-time P2P file sharing system towards live multicasting, something that has been attempted repeatedly, but generally has not led to low-latency live streaming. IRoI P2P streaming is the spatial counterpart to the temporal interactivity provided by time-shifted streaming. As the P2P overlay network topology changes rapidly in response to user input, this problem tests the limits of fast overlay management. Finally, we show how heterogeneous peer populations, including mobile peers that require transcoding, can be accommodated. This approach achieves three objectives. Firstly, it provides robustness against unavoidable peer churn by assigning multiple parents to a mobile peer. Secondly, it harvests excess computing power in fixed nodes for transcoding. Thirdly, the load of transcoding can be split across multiple fixed nodes.

## 7. REFERENCES

[1] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," *Proc. of ACM NOSSDAV, Miami Beach, FL*, pp. 177 – 186, May 2002.

[2] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth content distribution in a cooperative environment," *Proc. of IPTPS'03, Berkeley, CA*, pp. 298 – 313, Feb. 2003.

[3] Y. Cui, B. Li, and K. Nahrstedt, "oStream: Asynchronous streaming multicast in application-layer overlay networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 91 – 106, 2004.

[4] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "P2Cast: peer-to-peer patching scheme for VoD service," *Proceedings of the 12th international conference on World Wide Web*, pp. 301–309, May 2003.

[5] T. Do, K. A. Hua, and M. Tantaoui, "P2VoD: providing fault tolerant video-on-demand streaming in peer-to-peer environment," *Proc. of IEEE International Conference on Communications (ICC)*, vol. 3, pp. 1467–1472, Jun. 2004.

[6] E. Setton, J. Noh, and B. Girod, "Congestion-distortion optimized peer-to-peer video streaming," *International Conference on Image Processing (ICIP), Atlanta, Georgia*, pp. 721 – 724, Oct. 2006.

[7] ——, "Low-latency video streaming over peer-to-peer networks," *Proceedings International Conference on Multimedia and Expo (ICME), Toronto, Canada*, pp. 569 – 572, Jul. 2006.

[8] C. Huang, J. Li, and K. Ross, "Peer-assisted VoD: Making internet video distribution cheap," *Proc. of IPTPS '07*, Feb. 2007.

[9] P. Baccichet, J. Noh, E. Setton, and B. Girod, "Content-aware P2P video streaming with low latency," *IEEE Int. Conference on Multimedia and Expo, ICME, Beijing, China*, pp. 400 – 403, Jul. 2007.

[10] A. Mavlankar, J. Noh, P. Baccichet, and B. Girod, "Peer-to-peer multicast live video streaming with interactive virtual pan/tilt/zoom functionality," *Proc. of International Conference on Image Processing (ICIP), San Diego, USA*, pp. 2296–2299, Oct. 2008.

[11] A. Warabino, S. Ota, D. Morikawa, and M. Ohashi, "Video transcoding proxy for 3G wireless mobile internet access," *Communications Magazine*, pp. 66–71, Jan. 2000.

[12] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[13] A. Konrad, B. Zhao, A. Joseph, and R. Ludwig, "A Markov-based channel model algorithm for wireless networks," *Wireless Networks*, vol. 9, no. 3, pp. 189–199, 2003.