Cross-Channel Collaborations in Peer-to-Peer Streaming

Hongchao Li[a] and Kang Li[a]

[a]Department of Computer Science, University of Georgia, Athens, GA 30602-7404, USA

ABSTRACT

Peer-to-Peer(P2P) streaming has become a very popular technique to realize live media broadcast over the Internet. Most previous research of P2P streaming focuses on the delivery of a single media stream (called a channel). The widely deployed implementations, however, all concurrently offer multiple channels through their P2P networks. This paper investigates the overlay organization for multi-channel P2P streaming systems through modeling and simulations. In particular, this paper examines the potential collaborations among nodes across multiple channels. Our investigation shows that collaboration among nodes across different channels can improve the overall performance of the multi-channel P2P streaming system. However, the collaboration strategies need to be carefully selected. Simple collaboration strategies, such as treating collaborative nodes (those "borrowed" from other channels) the same as a channel's native nodes (those playing the channel), tend to have marginal or even negative effects on the whole system performance. This result is contrary to common impression—the larger population the better performance of P2P system—and we found that this is caused by the differences between P2P streaming and traditional P2P file-sharing systems. Furthermore, this paper proposes a set of simple strategies that controls the upload-download ratio of collaborative nodes. We showed that this set of strategies produces a much better collaboration result for multi-channel P2P streaming systems. Although only a preliminary study, we believe the results will promote further investigation on the topic of multi-channel P2P streaming.

Keywords: Peer-to-Peer Streaming, Multi-Channel Collaboration

1. INTRODUCTION

Peer-to-Peer(P2P) streaming has become a popular technique to realize live Internet media broadcast such as online TV and radio. Although P2P streaming is still an active research topic, several pieces of P2P streaming software, such as PPLive[1] and PPStream,[2] have been deployed and widely used by millions of users as an alternative platform to watch TV programs. For example, PPLive reported having more than a million nodes concurrently streaming its programs on the night of December 31, 2005. Although some of these P2P streaming systems are derived from recent research (e.g. Coolstream[3]), we observed that: *previous research focuses on delivering a single live stream, but in practice multiple streams are often concurrently offered in one system.* For example, PPlive currently offers more than 300 channels [*] and users can select and switch channels in a similar way as watching TV. We classify the systems that delivering multiple live streams (usually from the same service provider) as multi-channel P2P streaming systems.

Similarly to the TV and cable companies, it is in the streaming service provider's best interest to offer multiple channels in order to increase the potential audience size. Because of this demand of offering multiple channels, an issue about overlay organization within a streaming system is raised and consequently studied in this paper, i.e. *should nodes of each channel be organized as an independent P2P overlay or should nodes across different channels collaborate?*

Two general approaches exist for the delivery of multiple channels over P2P systems. The natural approach is that only the nodes playing the same channel form a P2P overlay, and thus each channel is delivered through its independent P2P system. The alternative is to consider collaborations among nodes across channels. Assuming

Further author information: (Send correspondence to Hongchao Li)
Hongchao Li: E-mail: hongchao@uga.edu, Telephone: 1 706 542 0597
Kang Li: E-mail: kangli@cs.uga.edu, Telephone: 1 706 583 0395

[*]Although not all, many of these channels are live video streaming of sports and entertainment events.

Multimedia Computing and Networking 2007, edited by
Roger Zimmermann, Carsten Griwodz, Proc. of SPIE-IS&T Electronic Imaging,
SPIE Vol. 6504, 65040K, © 2007 SPIE-IS&T · 0277-786X/07/$15

a similar audience population and connection bandwidth, a channel that plays cartoons might have a lower bandwidth demand than a live sport channel. Letting some nodes of the cartoon ("rich") channel to simultaneously participate the delivery of the sport ("poor") channel could possibly enhance the overall perceived quality of the streaming services. In practice, since all users in the system are running the same software, it is possible for the service provider (or software developer) to build the collaboration strategies into the client software once collaboration is proved feasible and beneficial.

This paper focuses on the study of feasibility and the challenges of using collaborations in a multi-channel P2P streaming system. In particular, we would like to answer the following questions: a) Is there any benefit for cross-channel collaboration? b) For a node that has spare resources to help another channel, what strategies should it take to utilize its resources? We answer these questions by building simulations for general P2P streaming systems and conducting experiments accordingly. We inspect both individual channel performance with various numbers of collaborative nodes and the overall system performance when one channel contributes collaborative nodes to another channel.

Although this is just an initial effort in our investigation of multi-channel P2P streaming, this study produces several interesting results. First, our investigation shows that, in general, collaboration among peers across different streams helps to improve the overall performance of the multi-stream system. However, the collaboration strategies need to be carefully selected. Our results show that a simple collaboration strategy of treating collaborative and native nodes the same tends to produce marginal or even negative effects on the whole system performance, but a strategy of maximizing the upload efficiency of collaboration nodes tends to produce better collaboration results.

Several questions are still left open for multi-channel P2P streaming. For example, collaboration across channels obviously means that some nodes would deliver data (download and upload) unrelated to the channel which the nodes are playing. Therefore, incentive mechanisms are required to compensate. As a first study of multi-channel P2P streaming, we ignore this incentive issue and expect it to be addressed in future research.

The rest of the paper is organized as follows: Section 2 reviews the related previous research on P2P streaming. Section 3 describes the experimental methodology used in this study. Section 4 presents our results, and Section 5 describes the future work and concludes the paper.

2. RELATED WORK

A common approach of live streaming is to use a tree-based overlay to organize the audience nodes (e.g.,Narada[4] and Zigzag[5]). New media segments are periodically pushed out by the source node and propagated in the tree. Some more sophisticated systems such as SplitStream[6] and CoopNet[7] maintain multiple trees simultaneously, and by using scalable coding techniques[8] each tree broadcasts only one layer of media data. Each node dynamically joins in an appropriate number of trees according to its own bandwidth limitation. The tree-based approach has an important benefit of a low buffering requirement at each node. However, the property that each node only has one parent from which to receive media segments results in some weaknesses for live streaming. For example, it can not handle well the dynamics of peer participation or adapt to network bandwidth variations.

Inspired by file-swarming mechanisms such as BitTorrent,[9] some mesh-based streaming frameworks (e.g.,DoNet[3] and PRO[10]) were proposed for live streaming. The general idea of mesh-based approach is that each node simultaneously maintains multiple neighbors. Each node, including the media source, gossips about what media segments it has in its buffer (called buffer map) to all its neighbors. A node schedules the requests on media segments based on buffer maps of its neighbors. The neighbor relationship can be updated according to their upload and download contributions and network dynamics. Recently, several works [11–13] studied the design of the best overlay structure to optimize the performance of single-channel live P2P streaming, and they contributed much to the success of single-channel live P2P streaming. This paper differs from all these previous works by focusing on the collaboration aspect in multi-channel P2P streaming.

3. EXPERIMENT METHODOLOGY

This section describes the performance metrics, the simulation setup as well as our assumptions used in the study of a multi-channel peer-to-peer streaming system.

3.1. Performance Metric

The goal of P2P live streaming is to continuously deliver the media content on time to individual audience nodes. A piece of media content is considered on time for a node when it arrives at the node before its playback deadline. Previous research[3,11] uses the *Continuity Index* (also called *on-time rate*) as the major metric to evaluate the performance of live streaming applications, and so does this paper.

The *Continuity Index* (CI) refers to "the number of segments, which arrive before or on playback deadlines, over the the total number of segments".[3] For a node that participate in multiple channels, only those media segments played back by the node are counted toward its CI calculation. We consider the CI of a channel to be the mean CI value of all the participating nodes in the channel. Similarly, the CI over multiple channels is measured by the mean CI over all the nodes in all the channels.

3.2. Simulation Setup

Our simulation took similar parameter settings as those in a previous study of P2P streaming.[11] The physical topology was generated with Brite,[14] using the following configuration parameters: 20 AS with 10 routers per AS in top-down mode. The delays on the access links between routers and user nodes are randomly selected between [5ms,25ms]. We assume homogeneous nodes in the system, and we choose one common ADSL setting (512Kbps/3Mbps) as the bandwidth of local access links. Core links among routers have high bandwidth and thus all connections experience bottlenecks only on the access links. For all the experiments in this paper, each simulation was run for one hour. To eliminate errors caused by random overlay organization, each simulation was run 30 times with different random seeds and the average values were taken as the final results.

3.3. Overlay Organization

Without losing generality, we picked a simple mesh-based overlay organization that is commonly seen in mesh-based overlay streaming. Specifically, when a node joins a channel's overlay, it contacts a bootstrapping node who will randomly select a fixed number (called *peer degree*) of nodes as its peers. Data are disseminated from a source node (hosted by the streaming service provider), which evenly distributes data to the source's immediate neighbors. Once a peer gets a piece of media data (called a segment), it advertises this information to its neighbors immediately. A peer always requests a segment from the neighbor from whom the peer gets the first advertisement about the segment. Each peer has a buffer which is used to cache segments for requests from its peers. At this initial stage of the investigation, we did not model churn and focused on steady-state content deliveries.

Although this above overlay organization is very simple, it captures the basic mechanism of most mesh-based overlay organization and it works well for a streaming system under the condition of homogeneous peers and the absence of churns. We expect the results drawn from the cross-channel collaboration based on this simple delivery strategy to also be applicable to more advanced overlay organizations.

In addition to the overlay organization, several other parameters affect the performance of a mesh-based streaming system. Early research of single channel P2P streaming system[11] shows that the media bit-rate, the peer degree and the buffer size within each node are the key parameters. We select these parameters based on the previous research as well as experiment results from our simulations. For the experiments presented in this paper, the media rate of each channel is in the range of 300Kbps to 600Kbps. The peer degree is set to 12, and each node buffers up to 120 seconds of video data.

4. EXPERIMENT RESULTS

This section presents the simulation results of cross-channel collaborations for multi-channel P2P streaming. The results are organized in the following way: The first subsection presents the performance of the channel that "borrows" nodes, followed by a simple model that explains the effect of collaborative nodes at different modes. Later, the second subsection presents the overall effect of collaboration on multiple channels, which include the performance of the channels from which the collaborative nodes are "borrowed".

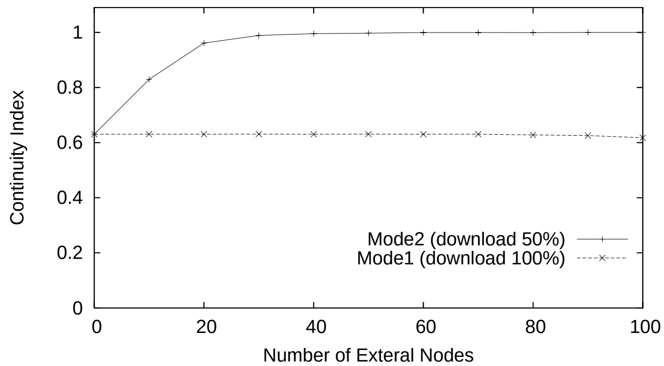

Figure 1. The Effect of External Nodes (with 100 native nodes)

4.1. Effect of External Nodes on a Single Channel

The way we study the effect of adding additional nodes is to measure the performance of a channel with a fixed number of audience nodes (call *native nodes*) and different numbers of nodes "borrowed" from other channel (called *external nodes*). Specifically, the simulation presented in this section considers a channel with 100 native nodes with 0 to 100 external nodes.

We choose 550Kbps as the data rate played by the channel, which is slightly higher than the upload ability of each native nodes. This data rate is chosen so that the channel's CI value is low (below 0.65) and thus potential improvement can be demonstrated. These external nodes are assumed to have all their bandwidth available for lend. Although this is usually unrealistic because these nodes have to participate the delivery of their own channel, this assumption simplifies this first experiment considerably. The experiments with different available resources from collaborative nodes are shown in the next subsection.

We consider two modes of organizing an overlay for a channel with both native and external nodes. The simplest mode (Mode1) is to treat all nodes (native and external) the same and form an overlay in which every node tries to download the whole media and exchange data with its neighbors. However, considering that the external nodes do not actually play the channel, we picked a second mode (Mode2), in which an external node only downloads half of the data in the channel (i.e. download and distribute only those frames with odd or even sequence number) but still try to upload as much as possible to its neighbors. Accordingly, two sets of experiments are conducted, one for each mode. The experiment results are presented in Figure 1.

The simulation results indicate that the effects of external nodes on the original channel performance are in fact not necessary positive. In Mode 1, as the number of external nodes increases, the original channel performance does not improve and even slightly decreases. This result is counter-intuitive in the sense that P2P system performance is generally believed to improve as the population increases. In contrast, in Mode2 the channel's CI value dramatically improves from 0.65 to 1 as the number of external nodes increases. The CI reaches 1 as more than about 30 external nodes are borrowed.

By inspecting the total amount of bandwidth consumed by the native nodes and external nodes, we found that the external nodes, although contributing more upload bandwidth, introduced additional download requirements to the P2P overlay. In the case of Mode1, an external node introduced considerable loads to other nodes because it unnecessarily acts as a native node. The external node's consumption ends up higher than the amount of upload bandwidth it contributes†.

†In general, introducing a new node to an overlay could improve the overlay system streaming capacity in two ways: by increasing the capacity to serve other peers with the new node's upload bandwidth resource, and by increasing the connectivity of the overlay, i.e. connecting nodes that were originally partitioned or with poor connections. Our simulation does not take the second factor (connectivity improvement) into account, and thus the benefit of the external node comes solely from the increment of total uploading bandwidth.

In the case of Mode 2, however, an external node does not consume as much as a native node. Specifically, an external node downloads only half of the media stream. If scheduled correctly, an external node can still contribute as much as its upload bandwidth permits. Consequently, this external node's contribution is larger than the load that it introduced. Therefore, the external nodes in Mode2 help improve the channel performance significantly compared to the one in Mode1. Certainly Mode2 (download half of the data stream) in this experiment is just one instance of variable modes of the external nodes and is not necessarily the optimal one. Therefore, we proposed a simple model to explain the effects of different modes and expected to find guidelines for design better collaboration modes.

4.1.1. A Simple Model for P2P Live Streaming

This section uses a simple model to explain the results of various collaboration strategies. The model is based on the following observation of P2P live streaming: most of nodes in a channel are synchronized or close-to-synchronized in terms of the data being played, i.e. a given frame is played at about the same time (with differences no more than a few minutes) on all nodes. The close-to-synchronized playback is a significant difference between live and non-live streaming. The latter, and in an extreme case the P2P file sharing, can tolerant more asynchronous playback. The result of this close-to-synchronized effect is that data buffered at each node does not need to be large because data expires (not useful for other nodes) fast. The amount of data being buffered at any time is very small compared to the total stream size. Non-live streaming or file sharing, on the other hand, usually buffers large amounts of data or even entire objects at each node.

Based on this observation, we made an assumption about the live streaming: all nodes in a channel play media data at the same pace. To further simplify the modeling, we also assume that upload bandwidth is less than or equal to download bandwidth for all nodes except the server. This is the common case of Internet connections at the end user side. Finally, we assume single channel P2P streaming systems are well designed, and all data being download and or upload are not wasted.

Based on these assumption, if there is a source node p_0 and n audience nodes $p_1, p_2 ..., p_n$ in a single P2P streaming channel, the overall downloading rate should equal the total uploading rate of all the nodes. That is,

$$\sum_{i=0}^{n} u_i = \sum_{i=0}^{n} d_i \,, \tag{1}$$

where u_i is the uploading rate of node p_i and d_i the downloading rate of p_i.

Suppose, the given media bit-rate of a channel is b. If a node wants to receive the media with its highest quality, the node needs to get media content at rate b. Then all the n nodes need to download at a rate $n * b$ in total. On the other hand, all the nodes, including the source node, can upload at a rate $\sum_{i=0}^{n} u_i$. Then, the best average *continuity index* that could be given by all possible overlays equals the total uploading rate over the total required downloading rate. That is,

$$\overline{CI} = \frac{\sum_{i=0}^{n} u_i}{nb} \,. \tag{2}$$

Suppose there are m external nodes $p'_1, p'_2 ..., p'_m$, which are idle or have extra uploading bandwidth, in the multi-channel streaming system. If we let these nodes participate in the first channel, the CI changes to:

$$\overline{CI} = \frac{\sum_{i=0}^{n} u_i + \sum_{j=1}^{m} (u'_j - d'_j)}{nb} \,, \tag{3}$$

where b and n are the bit-rate and the original population size of the channel, u_i is the uploading rate of original node p_i, and, u'_j and d'_j are the uploading and downloading rates of external node p'_j spent on this channel.

Based on this simple derivation, it is easy to conclude that the best strategy for these collaborative nodes is to avoid adding load to the original channels (by minimizing d'_j) but contributing as much as upload bandwidth permits (by maximizing u'_j). Unfortunately, the upload rate u'_j is not independent to the d'_j. For example, if d'_j is set to 0, nothing can be uploaded to other nodes even if spare uploading capacity is available. Thus, the best strategy is to minimize d'_j as long as u'_j is fully used. Overall, the media quality of a channel will be improved by the participation of these external nodes if external nodes download less than their uploading in the channel. This explains the effect seen early in this section.

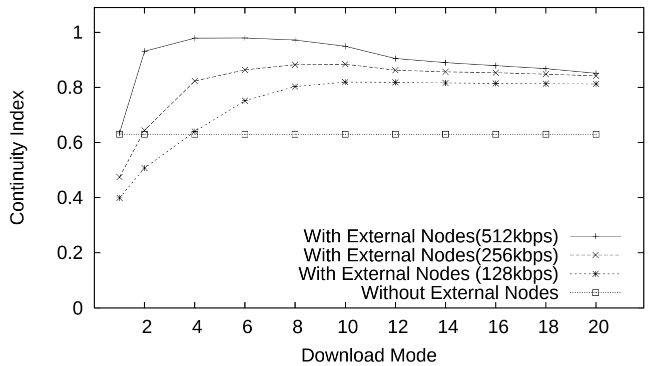

Figure 2. The Effect of Various Consuming Modes and Available Bandwidths of External Nodes

4.1.2. The Effects of Various Collaboration Strategies

The previous experiments demonstrate that the participation of external peers can improve the streaming quality if these external nodes download less than they upload. Naturally, the question to be answered is the following: what is the best mode for external nodes to help improve the streaming quality of the native channel?

To answer this question, we use the following simulation setup. As in the previous experiments, we choose a channel with 100 native nodes. Instead of varying the number of external nodes, we only consider 20 external nodes but with various download modes. We choose a set of download modes by simply varying the amount of data in external downloads: mode-i refers to the case that a node only download i-th share of the total stream. We also consider various uploading capacity of the external nodes, from 128Kbps to 512Kbps. The channel performance with external nodes under these various conditions is presented in Figure 2.

From Figure 2, we get the following three observations. First, the channel added with external nodes of larger available uploading bandwidth always achieves better streaming quality at each consuming mode. This observation matches intuition.

Second, although external nodes at most of the modes help improve the channel performance, that is not always the case. (i.e. nodes with 128Kbps upload bandwidth do not help when download modes are below 4; neither does the case of 256Kbps with modes below 2) We found out that external nodes always improve the channel performance when the production of their available bandwidth and consuming mode is greater than 520Kbps, the media bit-rate. This can be explained by the Equation 3. Suppose, the consuming mode of external nodes is i, their available uploading bandwidth is u' and the streaming bit-rate is b. When the uploading rate of external nodes is greater than their downloading rate ($u' > b/i$), the streaming quality of the channel will be improved by the participations of the external nodes.

Finally, the CI of the channel decreases as the amount of download decreases (i.e. mode number larger than 12). In this case the uploading contribution is limited by the amount being downloaded at the external nodes. As the mode number (i) increases, less data is downloaded by the external nodes, and eventually it throttles the ability for the nodes to upload to its neighbors. We made a simple effort to derive the best contribution modes. If we assume each external peer always tries to contribute upload bandwidth as much as possible and sends every received segment to all its k child peers. Then, the uploading rate of an external node equals $min(u', k\frac{b}{i})$. The overall contribution c of an external node, i.e., the difference of its uploading and downloading rate, can be roughly calculated with the following formula:

$$c = \{ \begin{array}{lll} u' - \frac{b}{i} & \text{when} & (kb/i) \geq u' \\ (k-1)\frac{b}{i} & \text{when} & (kb/i) < u' \end{array} \tag{4}$$

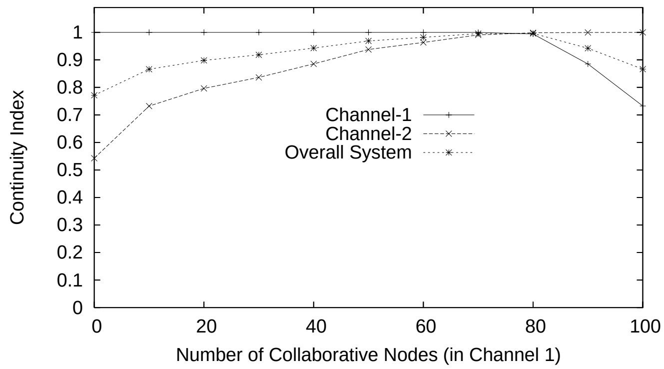

Figure 3. Effect of Collaboration in a 2-Channel System

4.2. Effect of Collaborations on Multiple Channels

This section presents our result on the overall effect of collaborations on multiple channels. We consider the case of a system with two concurrent streaming channels. The basic condition to launch two-channel collaboration is that one channel is streaming in good-quality but still has extra bandwidth while the other one is suffering from the lack of bandwidth.

To study the effect of collaboration, we set up the following simulation: we select the same population size for 2 channels: 100 nodes each. All nodes have the same up/download bandwidth (512Kbps/3Mbps). Channel-1 plays in a data rate of 300Kbps, and thus has some spare uploading bandwidth; while channel-2 plays a data rate of 600Kbps, which is higher than its average upload capacity. We control the collaboration by allowing some nodes in Channel-1 to concurrently participate Channel-2's delivery but only with their spare bandwidth. In implementation, this is achieved by reserving part of the uploading bandwidth of each of these nodes, 256Kbps, to participate in the streaming of Channel-2.

Figure 3 demonstrates the performance (in terms of CI values) of Channel-1, Channel-2 and the whole system. The quality of Channel-1 kept about 100% in CI value until more than 80 users are involved in collaboration. The streaming quality of Channel-2 was drastically improved as the number of collaborative nodes from Channel-1 increased. The average CI of the whole system reaches its peak (close to 100% CI) when 80 Channel-1 nodes help Channel-2. It means the quality of Channel-2 was improved to 100% in CI with the help from Channel-1 while the quality of Channel-1 had not been hurt.

However, after the number of collaborative nodes was greater than 80, more bandwidth borrowed from Channel-1 caused the upload bandwidth shortage of Channel-1 and resulted in a sharp downturn of the Channel-1 quality. When the total remaining upload bandwidth in Channel-1 becomes less than the product of the bit-rate and the Channel-1 user number, more bandwidth borrowing will hurt the quality of the channel. In the above experiment, when the number of collaborative nodes is 80, the remaining uploading bandwidth in Channel-1 equals $(80 \times 256 + 512 \times 21) = 30,976$Kbps which is little bit bigger than $(100 \times 300) = 30,000$Kbps because overhead also costs some bandwidth in streaming.

From this experiment, we can draw the following conclusion: Collaboration between two channels can improve overall quality of the whole system. Certainly, too much bandwidth borrowed from a channel will hurt the quality of the channel. To avoid affecting the channel from which nodes are borrowed, the total remaining upload bandwidth in the channel should be kept at more than the product of the bit-rate and the number of the users in the channel.

5. CONCLUSION AND FUTURE WORK

This paper presents our initial effort to model and study multi-channel peer-to-peer streaming. The initial work focused on the streaming channel collaborations when each channel is in its static state and all the peers are homogeneous in their properties. Many questions are still left unanswered. These questions include: 1) How does the system know a channel can potentially benefit from borrowing resources from other channels? Although the media rate is known to the server, the delivery capacity of an overlay is usually hard to estimate, especially considering the effects of churn. 2) Which nodes should be picked from one channel to collaborate with another channel? and 3) What scheduling strategy should a node use when concurrently participate multiple P2P overlay?

The preliminary results have already demonstrated the complexity of the resource allocation within a multi-channel streaming system. Our result also demonstrates the importance of managing download and upload bandwidth for the collaboration and we propose a guideline of maximizing the uploading to achieve high collaboration benefit. Although answering the above questions is crucial for the design of effective collaborations for multi-channel P2P streaming, the results shown in this paper have already shed light on resolving some of the issues.

In near future, we will investigate the effect of bandwidth heterogeneity, peer localities and churn on the collaborations of multiple streaming channels. We will investigate the collaborations at two levels. At a higher level, we plan to study the issue of organizing multiple overlays when collaboration is enabled in the multi-channel streaming system. At a lower level, we plan to study the scheduling detail of each node when they participate multiple overlays. In addition, we are working on a prototype implementation over PlanetLab so that we can investigate the collaborations in real networking environments.

REFERENCES

1. PPLive Corporation, *PPLive Software, Free TV on the Internet.* http://www.pplive.com/en/index.html.
2. PPStream Development Team, *PPStream, Peer-to-Peer Streaming Internet TV.* http://all-streaming-media.com/peer-to-peer-TV/p2p-streaming-Internet-TV-PPStream.htm.
3. X. Zhang, J. Liu, B. Li, and T. S. Yum, "Coolstreaming/donet: A data-driven overlay network for live media streaming," *Proc of 24th IEEE INFOCOM*, pp. 2102–2111, March 2005.
4. Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selection Areas In Communications* **20**, pp. 1456–1471, 2002.
5. D. A. Tran, K. A. Hua, and T. T. Do, "A peer-to-peer architecture for media streaming," *IEEE Journal of Selected Areas in Communications* **22**, pp. 121–133, 2004.
6. M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth multicast in a cooperative environment," *Proc of SOSP'03*, 2003.
7. V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," *Proc of IEEE ICNP'03*, November 2003.
8. V. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Magazine* , pp. 74–93, September 2001.
9. B. Cohen, "Incentives build robustness in bittorrent," *First Workshop on the Economics of Peer-to-Peer Systems* , June 2003.
10. R. Rejaie and S. Stafford, "A framework for architecting peer-to-peer receiver-driven overlays," *Proc of NOSSDAV 04*, 2004.
11. N. Magharei and R. Rejaie, "Understanding mesh-based peer-to-peer streaming," *Proc of NOSSDAV'06*, pp. 56–61, 2006.
12. J. Liang and K. Nahrstedt, "Dagstream: locality aware and failure resilient peer-to-peer streaming," *Proc of MMCN'06*, 2006.
13. T. Silverston and O. Fourmaux, "Source vs data-driven approach for live p2p streaming," *Proc of ICN'06*, April 2006.
14. A. Medina, A. Lakhina, I. Matta, and J. Byers, *BRITE: Boston university representative internet topology generator.* http://www.cs.bu.edu/brite/.