2007

Network Architectures for Live Peer-to-Peer Media Streaming

Jagannath Ghoshal
University of Nebraska-Lincoln, jghoshal@cse.unl.edu

Lisong Xu
University of Nebraska-Lincoln, xu@cse.unl.edu

Byrav Ramamurthy
University of Nebraska-Lincoln, bramamurthy2@unl.edu

Miao Wang
University of Nebraska-Lincoln, mwangcse@gmail.com

J. Ghoshal, L. Xu, B. Ramamurthy, M. Wang, "Network Architectures for Live Peer-to-Peer Media Streaming"
UNL Department of Computer Science and Engineering, Technical Report TR-UNL-CSE-2007-020,
September 2007.

Network Architectures for Live Peer-to-Peer Media Streaming

Jagannath Ghoshal, Lisong Xu, Byrav Ramamurthy and Miao Wang
Department of Computer Science and Engineering,
University of Nebraska-Lincoln, Lincoln, NE 68588-0115
Email: {jghoshal, xu, byrav, mwang}@cse.unl.edu

Abstract

Peer-to-Peer (P2P) media streaming networks, motivated by the huge success of P2P file downloading networks, have recently attracted a lot of research interest. However, it is challenging to design P2P media streaming networks because of the stringent time constraints on the delivered media streams, which require more efficient and resilient overlay architectures. In this paper, we focus on *live* P2P media streaming networks, a promising application flourishing in the Internet and which requires the distribution of live (not stored) multimedia content to subscribers. We review the architectures for live P2P media streaming networks, and consider both overlay topologies and their construction.

1 Introduction

Prior to the deployment of peer-to-peer (P2P) networks, the client-server model and the content distribution network (CDN) along with IP multicast were the most desirable solutions to support media streaming. These options are gradually losing ground with the widespread deployment of P2P networks due to their unique characteristics. A major advantage in using a P2P network is that each peer contributes its own resources to the network. As a result, there is an increase in the amount of overall resources of the network, such as bandwidth, storage space, and computing power. Consequently, P2P networks overcome the bottleneck problem at the server in a client-server model, where the single server must have enough resources to support all simultaneous clients. A CDN alleviates the same bottleneck problem by introducing more dedicated servers at geographically different locations but that also results in expensive deployment and maintenance. Lastly, IP multicast has good scalability in theory; however, its actual deployment across the Internet is limited.

There are two major types of P2P network protocols: P2P file downloading protocols and P2P media streaming protocols. Gnutella, BitTorrent, Kazaa, etc are famous P2P file downloading networks. The downfall of Napster which was the initiator of the P2P revolution, eventually helped in stimulating the more sophisticated networks such as Gnutella, Kazaa, etc. P2P media streaming protocols are motivated by the work on P2P file downloading protocols. Both of them establish a P2P network of users; nevertheless, there are significant differences between them.

Media streaming has a tight time constraint in that the playback starts soon after the streaming begins and the stream should be played back continuously; whereas file downloading has no such requirement on the downloading order of different blocks of a file. In addition, the file is accessed by a user only after the whole file has been downloaded. These differences required improvements to the architectural design of P2P file downloading protocols to readily address the timing constraints and to provide good media quality for P2P media streaming protocols.

Live media streaming, and stored media streaming also known as Video on Demand (VoD) are the two broad categories in P2P media streaming networks. In the former a live media stream is encoded on the fly and sent to all users, whereas the latter allows a user to jump to any point in time of a pre-recorded media stream. This implies one major difference between the design of these two systems: the sending rate of a media source in live streaming is limited by both the media encoding rate and the available bandwidth, whereas the sending rate of a media source in stored streaming is limited only by the available bandwidth.

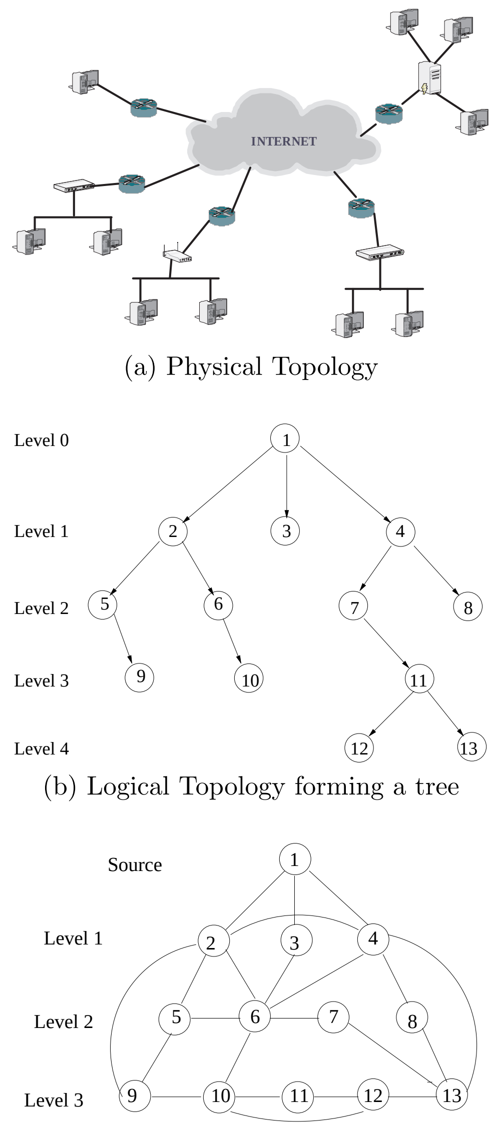
In this paper, we focus on live P2P media streaming networks, a promising application flourishing in the Internet. There are two important components in the design of live P2P streaming networks: overlay architecture and content delivery. The former deals with how to construct an efficient and resilient architecture of a P2P streaming network, and the latter considers how to efficiently distribute a media stream through the established architecture. This paper focuses on overlay architecture. The rest of the paper is organized as follows. We consider architecture topologies in Section 2, and architecture construction in Section 3. Section 4 presents discussions and possible research directions. Finally Section 5 concludes the paper.

2 Overlay Architectures

The efficiency of a live P2P streaming network relies on its overlay architecture, which is a logical topology built over the existing physical topology to transmit a media stream from the source to all participating peers.

There are two basic types of overlay architectures: tree-based, and mesh-based architectures (see Figure 1). The physical topology connecting all participating peers in a live P2P streaming network is shown in Figure 1(a). Without loss of generality, assume that peer 1 is the source broadcasting the live media stream. Figure 1(b) shows a directed tree-based logical topology, where there is only a single delivery path from the source (i.e., peer 1) to any other peer. Figure 1(c) shows a mesh-based logical topology, with bi-directional links between each pair of peers other than the broadcasting peer, which consists of multiple delivery paths from the source to any other peer. The mesh-based architecture has more links than the tree-based architecture, which provides it a stronger connectivity, and at the same time a more complicated content delivery scheme due to the complex structure.

Recently a surge of overlay architectures based on the above two basic types have been proposed. Below, we discuss in detail four types of overlay architectures: architectures based on a single tree, on multiple trees, on a mesh, and on tree-mesh (hybrid).



(a) Physical Topology

(b) Logical Topology forming a tree

(c) Logical Topology forming a mesh

Figure 1: Physical topology and two basic types of overlay architectures of a live P2P streaming network

2.1 Architectures Based on a Single Tree

The design of P2P streaming networks was initially motivated by IP multicast, where a multicast tree is built at the IP layer to deliver packets from a source node to all of the destination nodes. Following the same idea, early P2P streaming networks, such as ZIGZAG [7], built a multicast tree for media streaming at the application layer. The streaming media is pushed from the source peer (i.e., the root node) to all other peers following the multicast tree.

In live media streaming, the end-to-end delay from the source peer to a receiver peer is a critical metric, since it determines the user-perceived liveness of the streamed media. The longest delay in a multicast tree depends on the height of the tree; that is, the maximum number of logical links from the source peer to a receiver peer. For this reason, one major design goal of a tree-based overlay architecture is to shorten the end-to-end delay by reducing the height of the multicast tree. For example, ZIGZAG organizes the peers into a hierarchy of clusters by following a special set of rules, which guarantees that the height of the tree built is $O(log_k N)$ where N is the total number of peers and k is a constant.

However, there is a significant difference between an IP-layer multicast tree and an application-layer multicast tree. The former is built upon IP routers that are relatively stable in the Internet, whereas the latter is built upon participating peers that are very dynamic in that peers may join and leave the network at any time. This difference makes a tree-based overlay architecture inappropriate for P2P streaming networks, since a departing peer disconnects all its descendants from the source peer, resulting in poor media quality for all its descendants. As a whole, the impact of high churn rate in tree-based structures drastically affects the performance of the P2P system.

Apart from the above shortcomings, there are two other weaknesses of a tree-based overlay architecture. First, the amount of media data that can be delivered to a peer and thus the media quality experienced by the peer are limited by the minimum link capacity among all the links along the path from the source peer to the peer. Second, a tree does not use any of the uploading link capacity of a leaf peer.

2.2 Architectures Based on Multiple Trees

Multiple tree based (also known as multi-tree based or forest-based) overlay architectures, such as SplitStream [1], CoopNet [5] and ChunkySpread [8], are proposed to mitigate the strong dependency of a peer on all its ancestors in architectures based on a single tree. They are typically designed to work with more advanced video encoding techniques. For example, CoopNet uses multiple description coding (MDC), which encodes a media stream into multiple independent substreams (or descriptions).

An attractive property of MDC is that each substreeam can be independently decoded, and the overall quality of the decoded media is proportional to the total number of received substreams. CoopNet constructs multiple independent multicast trees, one for each substream. A peer can improve its media quality by joining more multicast trees under the constraint of its download link capacity. More importantly, the departure of one ancestor in a multicast tree does not severely degrade the media quality of a peer, since it can still receive the majority of substreams from other multicast trees.

There are several mechanisms implemented in multiple-tree based architecture, which helps in providing good media quality and efficient routing of data blocks among the peers. These mechanisms are targeted at alleviating the dependency and distributing the overall load of the network appropriately among participating peers. CoopNet addresses the problem of dependency by constructing diverse and bushy trees. Typically, a peer is a leaf node in most trees, and works as an interior node in the remaining few trees. For example, a peer in CoopNet can be an interior node in only one tree, and therefore, the departure of any peer disrupts only one tree and thus one substream.

Other mechanisms implemented in SplitStream and ChunkySpread addresses the load balancing problem of the network. ChunkySpread achieves load balancing by assigning greater number of neighbors to a peer if it has a higher load than other peers. SplitStream uses an underlying Distributed Hash Table (DHT) of the network to construct overlay routes and eventually assign neighbors to its peers. This mechanism of load balancing fails in a highly heterogeneous environment, which forces frequent disconnections of nodes, thus mitigating the advantage of using a DHT. Fundamentally, SplitStream addresses the problem of latency- versus-overhead tradeoff, where it succeeds in minimizing the latency. However, ChunkySpread manages to decrease the overhead of the network since it reacts efficiently to membership changes due to proper distribution of the peers and eventually succeeds over SplitStream in terms of latency as well.

Compared to architectures based on a single tree, architectures based on multiple trees are more resilient to peer departures and failures. In addition, they can more efficiently use the uploading link capacity of each peer, since each peer works as an interior node in at least one tree. However, they achieve these benefits at the cost of more complicated architectures and media encoding methods.

2.3 Architectures Based on a Mesh

Many recently proposed P2P streaming networks, such as CoolStreaming [10], AnySee [3], PRIME [4], PRO [6], and DagStream [2], construct an overlay architecture based on a mesh as illustrated in Figure 1(c). A mesh-based architecture is very similar to a multi-tree-based architecture, in that both of them adopt a multi-parent and multi-children approach for media streaming. That is, a peer in both architectures receives media data from multiple parent peers, and at the same time sends media data to multiple child peers.

However, there is a fundamental difference between a mesh-based architecture and a multi-tree-based architecture. Just as in an architecture based on a single tree, a packet in an architecture based on multiple trees still has only one static delivery path from the source peer to any other peer, even though packets in different substreams may take different paths. On the other hand, a packet in an architecture based on a mesh may take different paths in response to the changes of peers and network conditions. The reason is that in an architecture based on both a single tree and multiple trees, a packet is pushed by a peer to all its child peers in a multicast tree, and there is a static parent-child relation between a peer and its child peers. On the other hand a mesh-based architecture adopts a swarm-like design, where a peer periodically exchanges data availability information with its neighboring peers, proactively pulling packets from other peers that have the requested data, and pushing packets to other peers that are expecting the data. This difference considerably improves the resilience of mesh-based architectures to peer dynamics and network bandwidth dynamics.

One of the design goals of mesh-based architectures is to construct a robust mesh network resilient to peer and network condition dynamics. There are two general types of mesh network architectures; unstructured mesh networks, and structured mesh networks. An unstructured mesh network such as PRIME, and CoolStreaming connects a peer with a large number of randomly selected peers, with the purpose of providing more neighbors and more diverse paths. A structured mesh network typically groups peers into clusters based on their physical locations, and makes connections between peers within a cluster more likely, in order to reduce the propagation delay of packets. However, such a locality-aware network may suffer from the shared bottleneck problems, where the media quality of all peers in a cluster strongly depends on the available bandwidth at a shared bottleneck, and the single point of failure problem, where the departure of a peer connecting two clusters may disconnect other peers in the cluster from the source peer. For these reasons, a locality-aware approach constructs a mesh with some special structure in order to achieve good network connectivity. For example, DagStream builds a directed acyclic graph with the property that every peer has at least K neighbors, and the departure of less than K neighbors does not disconnect other peers from the source peer.

However, mesh-based overlay networks have various problems, which lead to the rise of a hybrid architecture as discussed in the next section. The major disadvantage of mesh-based overlay networks is that since a mesh overlay does not maintain a parent-child relationship during routing of the data blocks, these data blocks have to be pulled from neighbors. This would result in incorporating more delays and escalating the control overhead. Also, sending control notifications for every data block received at a peer will undoubtedly increase the overhead. Therefore using buffer maps helps in reducing the excessive overhead at the cost of increasing latencies. This problem stated by the authors of mTreebone [9] as efficiency-latency tradeoff problem is one of the major problems in mesh-based overlay networks, due to which in some situations multi-tree based architectures are preferred.

2.4 Architectures Based on a Tree-Mesh (Hybrid)

The hybrid network proposed for mTreebone [9] is a novel research contribution to the P2P live streaming research area. It addresses most of the problems encountered by the above architectures. It is a complex architecture where a tree-based network is built to create a backbone, which monitors the majority of live streaming traffic, and a mesh structure is built on top of this backbone to complete the overlay network. The backbone is mainly composed of stable nodes and the mesh structure consists of non-stable nodes that are available within the network. Nevertheless the information about the stability and lifetime of a single node is provided by the mesh structure to the overlay network.

Initially a threshold time is computed within which the availability of the stable nodes that have long lifetime values are obtained and the tree is constructed with these nodes. The stability of the network depends mainly on the backbone where the routing of the packets is done through parent-child relationship. Live streaming is carried out by implementing push/pull switching mechanism. Thus the push mechanism is performed by deactivating the pull mechanism along the tree network to distribute the traffic to the descendant peers and also to the mesh network. If there is any block of media data missing at a node during the distribution, then the node uses the pull mechanism by deactivating the push mechanism, in order to obtain the missed block.

There is a high degree of coordination between the two structures (tree and mesh). The continuity of the streaming is not affected even if a stable node leaves the backbone, since it is replaced by the next stable node immediately. This is done with the help of the mesh network, which keeps the information about the neighboring peers during their lifetime so that it can readily provide stable nodes when the network becomes highly dynamic. The major advantage of such a network is that the height of the backbone is short as a result of which it provides better efficiency and lower latency and thus better performance over the other types of architectures. Although the architecture is complex, it succeeds in overcoming the limitations of a tree and mesh network. The disadvantage of such an architecture is that, the initial time taken to build the backbone can vary, depending on the availability of the stable nodes.

Thus the hybrid architecture helps in constructing a resilient overlay structure with low overhead and short delay. Unlike other architectures, when the churn rate is high and there are too many disconnections of the peers from the backbone, the mesh overlay takes charge of the streaming traffic temporarily until the backbone is reconstructed. Therefore both the structures deliver their best performance to address the demands in the P2P network.

3 Architecture Construction

In this section, we discuss three important design issues in constructing an overlay architecture for P2P streaming networks: peer discovery, peer selection, and peer replacement.

3.1 Peer Discovery

The purpose of peer discovery is to find the information about other peers in a P2P streaming network, when a peer initially joins the network. Since the obtained peer information will be used to select appropriate neighbor peers for a peer, the desired information includes: which peers are currently in the network, what the bandwidth usage of a specific peer is, what the workload of a specific peer is, etc.

There are two general peer-discovery methods: one maintains the global peer information, and the other one keeps only the information about a small list of peers. With global peer information, a P2P streaming network such as CoopNet, can find the best neighbor peers and construct a more efficient overlay architecture. However, when the number of peers becomes very large, the cost to maintain the global peer information would be very high. Therefore, many P2P streaming networks such as ZIGZAG, CoolStreaming, and AnySee, keep only the information about a small list of peers, as described next.

The peer-discovery methods which maintain the information for only a small list of peers can be further classified into two categories: the flooding-like method and gossip-based method, according to the way in which they obtain peer information.

In networks based on the flooding-like method such as ZIGZAG, a peer sends a probe message first to a starting peer, and then to its neighbors and their neighbors until finally finding a suitable point to join the network. Compared to the pure flooding method where a peer sends a probe message to all other peers at the same time, the flooding-like method generates far fewer number of probing messages. However, when the scale of the network is large, it may take a long time for a peer to join the network.

In networks using the gossip-based method such as CoolStreaming, and AnySee, a peer maintains the information about a small list of randomly selected peers, initially obtained from a starting peer, and the information is updated periodically by exchanging messages with other peers in the list. The gossip-based method has good scalability, and is suitable for large-scale P2P streaming networks. However, a weakness of the gossip-based method is that the media quality cannot be guaranteed, since a group of randomly selected peers may not have enough resources to provide the desired media quality.

In summary, peer discovery that maintains the information about peers in a P2P network is the basis for constructing an efficient and scalable architecture. However, when choosing a peer-discovery mechanism for a P2P streaming network, we have to balance the efficiency and the management cost.

3.2 Peer Selection

Peer selection is essential for a P2P streaming network, in that it directly determines the performance of individual peers and the overall network. Peers are usually selected to achieve one or multiple of the following goals [6]: 1) to minimize packet delay, 2) to achieve a minimum of total streaming rate at a peer, and 3) to be more resilient to peer and network dynamics.

Since packet delay is relatively easy to measure, many P2P streaming networks use packet delay to select appropriate peers. A peer may select geographically nearby peers in order to minimize the packet delay between its parents and itself as implemented in DagStream, or may select those peers that can minimize the total delay from the source peer to itself as in ZIGZAG and AnySee. However, the delay between two peers cannot reflect the available bandwidth between them. In other words, it is possible that the delay of a path is short, however, its available bandwidth is not high enough for media streaming.

Because the media quality experienced by a peer strongly depends on the total streaming rate received by it, peer selection in most P2P streaming networks is designed to achieve a minimum acceptable level of total streaming rate for every peer. Since it is not easy to quickly and accurately measure the link capacity or available bandwidth, many networks just select a large enough number of parent peers for each peer. A few P2P streaming networks, such as Coop-

Net and PRO estimate and monitor the available bandwidth between two peers, and then select parent peers based on the estimated bandwidth.

Peer selection based on delay or bandwidth may achieve good performance, if the network and peer conditions do not change. However, it is possible that the constructed overlay may have a large number of shared links or shared parent and children peers, and it performs poorly in dynamic network environments. For this reason, some P2P streaming networks such as PRO randomly select parent peers with the purpose of providing more diverse paths and diverse parents. During peer selection routine, peers should consider selecting some resilient nodes as their parents according to some standards. Note that the goal to be more resilient to peer and network dynamics does not necessarily conflict with other goals. For example, PRO randomly selects a parent peer according to a probability proportional to its available bandwidth.

3.3 Peer Replacement

When a peer leaves a P2P streaming network, the streaming of the media is disrupted at the descendant peers of the missing peer and the routing structure of the network is weakened, especially in tree-based networks. The extent of disruption depends on the number of descendant peers connected to the missing peer. In this case, a new parent peer may be selected for each child node of the departed peer. In addition, when the available bandwidth between a parent peer and a child peer has been reduced significantly for a long time, a new parent peer should be selected to replace the current parent peer.

Two approaches are available to replace a peer, namely reactive and proactive. In a reactive approach, after a peer leaves the network, or a peer failure has occurred, the connectivity of the network is retained by assigning a new parent to each child of the missing peer. In a proactive approach, the restoration plan for the descendant peers of the missing peer is carried out beforehand. The limitation of a reactive approach is that, it consumes a lot of time in order to repair the network, since several peers have to be contacted to locate and select a new parent. In case of proactive approach, which is followed by AnySee, an *Inter-overlay Optimization Manager* maintains two sets of active streaming paths, including the current streaming path and the pre-computed backup paths, of all the peers in the network. So when a peer fails or leaves the network selfishly, eventually introducing a broken link (path), a new path is selected from the backup sets to replace the broken link thus restoring the connectivity of the network. This mechanism is advantageous because the neighboring peers are not swarmed with requests due to a peer's departure; instead the overlay manager just replaces a lost link by referring to the backup set and thus replacing a peer efficiently. Therefore the approach in AnySee restores the connectivity of the network faster than the reactive approach.

4 Discussion and Future Directions

Overlay architectures are an essential part of a P2P network. Since the characteristics of the peers willing to join a P2P network for media streaming are not known beforehand, deployment of any of the architectures should be performed wisely. Each of these structures is suitable for specific environments where they can outperform the others in their performance. In a P2P network with low bandwidth and stable peer membership, a tree-based architecture is preferable to a mesh-based architecture, since it requires less control overhead (e.g., neighbor peers do not exchange their media data information). In a P2P network with high bandwidth and a high peer churn rate, a mesh-based architecture is preferable to a tree-based architecture, because a mesh-based architecture is more resilient to peer churn. Lastly, the idea of constructing a hybrid model is a perfect step towards a new architecture for P2P streaming networks. The hybrid model can be extended to build multiple-trees instead of a single-tree for the backbone which will provide better resilience in handling the dynamic nature of the network. This idea has not yet been fully investigated in the implementation of P2P live media streaming networks.

Churn rate is an important parameter in P2P networks, which necessitates efficient algorithms to alleviate its effect on the connectivity of the network. These algorithms are concerned with discovering peers, selecting appropriate peers and peer replacement after a peer failure, so that a robust connectivity is maintained within the network. As discussed earlier, peer discovery algorithms can be implemented in mesh or tree-based structures. Since individual peers may wish to keep information about their neighboring peers, a mesh-based network might contain a large amount of data and thus there might be information redundancy within the network. In case of peer selection algorithms, they can be implemented in either of the structures efficiently. However, implementation of peer selection algorithms on a mesh-based network might be advantageous, since a mesh-based network can provide rich information about the availability of a peer's resources. Lastly, peer replacement algorithms are more important for a tree-based architecture than a mesh-based architecture, since a departing peer in a tree-based architecture may significantly disrupt all its descendant peers.

Due to delay and bandwidth sensitive property of a P2P live media streaming network, future research should focus on faster peer failure recovery; in other words, we should maintain a relatively stable overlay structure when churn occurs. The goal might be achieved either by establishing a more resilient overlay structure or by keeping a list of available peers whose information is updated periodically. Another challenging issue is how to locate the required resources efficiently. This problem is more difficult to solve than in the file downloading systems, in that in P2P streaming systems, the order and the timeliness are critically important; contents that arrive at a specific peer late are treated as lost data. Additionally, the optimal peer selection methods and mTreebone require further study, and these need to be applied to real systems and we need to perform real world experiments to quantify their performance and for validation.

Apart from the operation of a P2P network, the power of its technology has given rise to several other issues, such as privacy, security, bandwidth usage and copyright/legal issues. There has not been enough research addressing these issues.

5 Conclusions

Live P2P streaming networks, as a promising Internet application, have attracted a lot of research interest. In this paper, we reviewed the overlay architectures for live P2P streaming networks, from the perspective of architecture topology and construction. Because of the stringent time constraint of live media, a live P2P streaming network has a strict requirement on the constructed overlay architecture, on top of which a media stream should be quickly and efficiently distributed to all participating peers. Considering the dynamic nature of large scale P2P networks, it is important that the constructed architecture be resilient to peer and network dynamics. We also highlighted some challenges including both technical and non-technical issues, before P2P streaming networks can be widely deployed to provide users with satisfactory media quality. Due to our specific perspective and limited space, this paper does not cover content delivery, which considers how a media stream is distributed through the established overlay architecture, and is another important component of a live P2P streaming network.

References

[1] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in a cooperative environment. In *SOSP'03*, Lake Bolton, New York, October 2003.

[2] J. Liang and K. Nahrstedt. DagStream: Locality aware and failure resilient peer-to-peer streaming. *Proceedings of MMCN*, January 2006.

[3] X. Liao, H. Jin, Y. Liu, L. M. Ni, and D. Deng. AnySee: Peer-to-peer live streaming. In *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006.

[4] N. Magharei and R. Rejaie. PRIME: peer-to-peer receiver-driven mesh-based streaming. In *Proceedings of IEEE INFOCOM*, Alaska, May 2007.

[5] V. Padmanabhan, H. Wang, and P. Chou. Supporting heterogeneity and congestion control in peer-to-peer multicast streaming. In *Proceedings of IPTPS*, San Diego, CA, February 2004.

[6] R. Rejaie and S. Stafford. A framework for architecting peer-to-peer receiver-driven overlays. In *Proceedings of NOSSDAV*, Cork, Ireland, June 2004.

[7] D. A. Tran, K. A. Hua, and T. Do. ZIGZAG: An efficient peer-to-peer scheme for media streaming. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, March 2003.

[8] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous unstructured tree-based peer to peer multicast. In *Proceedings of IEEE ICNP*, Santa Barbara, CA, November 2006.

[9] F. Wang, Y. Xiongand, and J. Liu. mtreebone: A hybrid tree/mesh overlay for application-layer live video multicast. In *The 27th IEEE International Conference on Distributed Computing Systems (ICDCS'07)*, Toronto, Canada, June 2007.

[10] X. Zhang, J. Liu, B. Li, and T. P. Yum. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. In *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.