Performance Study of Unstructured P2P Overlay Streaming Systems

Purvi Shah　　Jawad Rasheed　　Jehan-François Pâris

Department of Computer Science
University of Houston, Houston, TX 77204-3010
{purvi, jrasheed, paris}@cs.uh.edu

***Abstract*—One of the major differences among current P2P solutions for multimedia streaming is the way they organize the exchange of multimedia data among their peers. We compare the performance of two of these organizations, namely, an *unrestricted mesh* using a tit-for-tat incentive policy and a *directed mesh* that allows peers to dynamically replace poorly performing parents. We first observe that both organizations performed fairly well under ideal circumstances where all peers were cooperating and connections never failed. We then considered how the same two organizations would perform in the presence of connection failures and observe that the sole organization capable of providing a high quality of service was an unrestricted mesh organization that allowed at least eight concurrent downloads per peer. The same was not true for the directed mesh whose performance actually decreased when we allowed more than four concurrent downloads.**

1. Introduction

The peer-to-peer (P2P) approach offers a promising solution to the problem of distributing multimedia data from an original server to a large number of peers in a scalable manner. Since peers not only download content from the server, but also forward it to other peers, the serving capacity of the network grows as the number of peers increase. They can handle *flash crowds*, that is, very large and sudden surges of demand, thereby reducing the bandwidth cost burden on the server. In addition, P2P systems do not require any special support from the network, whether it be IP multicast or any specific content distribution infrastructure.

Our work focuses on the problem of application level multimedia streaming to a large group of users. We are primarily concerned with the performance of unstructured P2P streaming systems in terms of scalability, robustness and resilience. We performed simulations of the system's behavior for two different *overlay organizations* and investigated their perspective performances varying the video streaming rate, the peer connectivity level and the system size. In addition, we investigated how the two organizations would perform in the presence of connection failures.

We found that both mesh organizations performed fairly well under ideal conditions where all peers were cooperating and no connections were lost. The same conclusions did not hold in the presence of lost connections where the unrestricted mesh organization was the sole organization capable of providing a high quality of service.

The rest of the paper is structured as follows: Section 2 we discuss related work in P2P streaming. In Section 3, we present a quick overview of our unstructured P2P streaming system. Sections 4 and 5 describe our experimental setup and present our results. Finally Section 6 has our conclusions.

2. Related Work

There have been significant efforts both in academia and industry to address the challenges presented in designing P2P streaming systems. We give a short survey below.

2.1. Initial Work

Narada [CRZ00] focuses on multi-sender multi-receiver streaming applications, maintains a mesh among the peers and establishes a tree whenever a sender wants to stream multimedia to a set of receivers. Due to heavy control overhead resulting from intensive interactions between peers, Narada does not scale well to large groups of peers. Xu et al. [XH+02] were among the first to propose the concept of P2P streaming. Their work mainly focuses on the analysis of the capacity of P2P systems for streaming.

2.2. Structured Streaming Systems

2.2.1. Single Tree Streaming

Nice [SBK02] and *ZIGZAG* [THD03] propose constructing an overlay tree over the network and pushing data over that tree in order to minimize the playback delay and maximize the utilization of the bandwidth of the peers. Banerjee et al. [BL+03] observed that any traditional overlay tree-based protocol can be made resilient by duplicating chunks along a small number of randomly chosen additional overlay links. This way they utilize all the peers in the network. The major issue with these single overlay tree-based protocols is to build a scalable overlay tree with high efficiency.

2.2.2. Multi-Tree Streaming

In contrast *multi-tree* organizations, such as *SplitStream* [CD+03] and *CoopNet* [PW+02], build multiple distribution trees for streaming and assume the presence of nodes that are not willing to forward traffic. Their goal is to balance the forwarding load on different peers. Unfortunately, multi-tree overlay protocols cannot provide backup streaming services when peers in the upper layers of the tree fail or leave the tree. As a result, these systems are vulnerable to peer failures and prone to disrupted services.

2.2. Unstructured Streaming Systems

To overcome these limitations, various mesh-based P2P streaming systems have been proposed [JD+03, HH+03, Li04, LN06, PK+05 and ZLL05]. *Gnustream* [JD+03] is built upon the *Gnutella* system [Gnu07]. Most of these protocols such as *Napster* [Nap07], Gnutella and *FastTrack* [Fas07], were designed to share MP3 or image files that are a few megabytes in size, where search times are more crucial than distribution times. Hence they are not suited for streaming applications.

BitTorrent (BT) [Coh03] relies on *swarms*, that is, unstructured overlay mesh networks of peers, to distribute as efficiently as possible large files. In BT, as in multimedia streaming, the content to be distributed is divided into multiple *chunks* or segments to be downloaded from the server. This interesting property makes BT an attractive choice for P2P streaming.

More recent streaming systems such as *Chainsaw* [PK+05], [NR07] and *DagStream* [LN06] use such a mesh organization to limit the impact of peer dynamics and network dynamics. However, these systems do not provide a mechanism to enforce a fair resource contribution. For instance, Chainsaw allows peers to define its own maximum uploading bandwidth and fails to deter free riding. Unlike Chainsaw our barter based BT approach, uses the tit-for-tat policy to enforce fairness among peers and is also resilient against different peer arrival rates.

Unlike a tracker-based solution, *CoolStreaming* [ZLL05] has each peer periodically exchange the availability information of the media stream with different neighbors. The construction and maintenance of the systems require a much higher overhead, which may result in peers experiencing longer playback delays. In addition these solutions [JD+03, HH+03, Li04, LN06, PK+05 and ZLL05] use different internal policies to deal with the real-time requirements of multimedia streaming.

Previous studies have suggested that BT can be adapted for streaming purposes [SP07a, VIF06]. Tewari and Kleinrock [TK07] present an analytical basis to previous studies of BT-like streaming system in an existing swarm of peers.

TABLE I. SIMULATION PARAMETERS

File size (S)	150 MB
Chunk size	256 KB
Link bandwidths	10 Mbps
Rechoking interval	5 s
Optimistic unchoking interval	15 s

We extend our previous work [SP07a] that applied BT technology to multimedia streaming. While that work focused on the use of more efficient chunk to ensure on-time delivery of the streaming data, we investigate this time the impact of other parameters such as active neighbor connections, overlay organizations and system size.

Possibly the work more closely related to our is [BB07] where the authors restrict their comparison to tree-based streaming systems. In this work we decided to compare the mesh approach to a *directed mesh* approach because this approach encompasses the approach used by *Chunkyspread* [VFC06], a multi-tree streaning protocol viewed as the `best of breed' among tree-based P2P streaming solutions.

3. System Description

Our proposed system is a collaborative P2P unstructured streaming system. Incentive policies help peers improve their streaming quality by selecting cooperative and reciprocal neighbors.

Our P2P streaming system is based on BT. When a new peer registers to the system; it receives from the *tracker* a list of other peers in the system. The new peer then contacts these peers and receives from each of them a map of the chunks they own. It then selects a subset of peers as *neighbors* and requests chunks from them.

Peers download chunks that meet the playback deadline, that is, lie in the current playback window of interest, and are rare amongst their neighbors. We selected this combination of a rarest first policy within a window of interest because it was found to perform much better than BT’s rarest first policy in streaming application [SP07a].

The source uploads to a subset of peers downloading at the highest rate. Peers upload to other peers that provide them with the highest downloading rates. Thus, the incentive policy used for overlay organization is similar to the tit-for-tat policy used in BT which is executed at the start of each *rechoking* interval [Coh03]. According to the *tit-for-tat* policy peers typically upload to the *k* peers that recently provided it with the best downloading rate, even though it may have received requests from more than *k* peers. Unless otherwise mentioned we did not attempt to tailor this policy of streaming applications.

We also considered an alternate overlay organization where the peers formed a *directed mesh*. Under that organization, chunks will always move in one direction and a peer receiving chunks from a *parent peer* would not send back chunks to that parent. We allowed peers to have multiple parents and let peers dynamically replace poorly performing parents to improve performance. Each parent peer will typically upload to *k* peers in a first come first server approach, even though it may receive requests from more than *k* peers.

TABLE II. IMPACT OF DIFFERENT CONSTANT STREAMING RATES ON THE STREAMING QUALITY

Streaming Rate (Mbps)	Mesh	Directed Mesh
3	98.82	99.65
4	99.73	99.63
5	87.02	90.15

Unless otherwise mentioned, the constant number of concurrent peer transfers is set to 4 (default number used by BT clients).

4. Experimental Setup

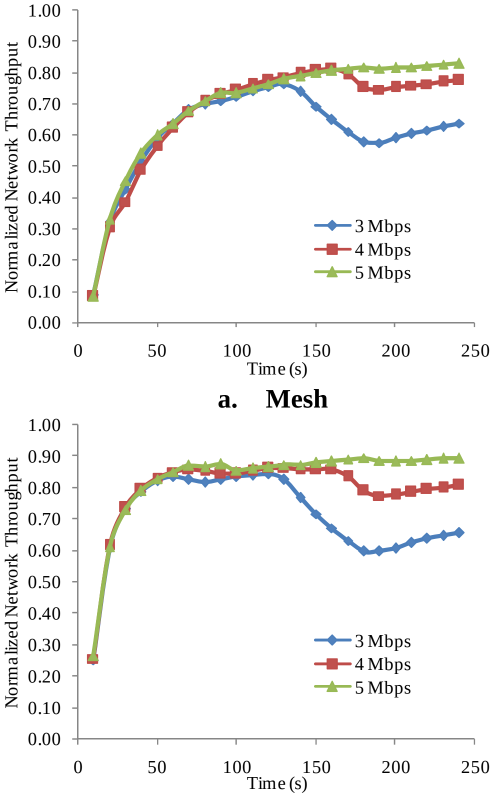
To evaluate the performance of our P2P streaming system, we use the Java based discrete-event General P2P Simulator (GPS) [YA05].

We modeled the network transmission and queuing delays, but assumed that the network propagation delays could be neglected since they are relevant only for small sized control packets while the multimedia streaming time is dominated by the chunk exchange traffic. To keep our model simple, we ignored the complexity of the dynamics of TCP connections. We assumed the idealized performance of TCP, and assumed that connections traversing a link shared its bandwidth equally.

Like previous simulation studies [SP07a, BHP06] we assumed that bandwidth bottlenecks only occurred at the edge and did not model shared bottleneck links in the interior of the system. We believe alternate techniques utilizing the physical topology, to improve the system performance are complementary to this work.

We assumed that each multimedia streaming session consisted of a single source streaming the multimedia data to all peers. Table I summarizes the main parameters of our model. We calculated the size of the sliding window using the formula $w = db/c$ where *d* is the playback delay, *b* is the video consumption rate and *c* is the chunk size [SP07a], assuming a playback delay of 90 seconds.

Finally, we assume that for a session the streaming rate is constant. The source splits the stream in a sequence of fixed size chunks and may apply redundant encoding such as multiple description coding to achieve better resilience to chunk loss. Constant streaming rate media is used widely in online streaming and the chunks here have the same relevance.



a. Mesh

b. Directed Mesh

Fig. 5.1.1. Normalized network throughput at different streaming rates.

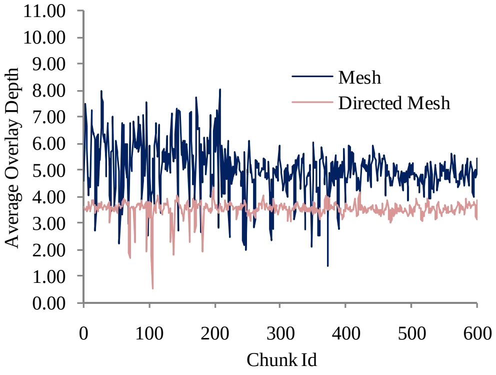


Fig. 5.1.2. Average overlay depth for system for streaming rate of 5Mbps.

5. Results

We examine the performance of our two overlay organizations, namely, *mesh* using a tit-for-tat incentive policy and *directed mesh* letting peers dynamically replace poorly performing parents. We ran all our experiments using the same simulator and the same workloads in order to enable a fair comparison. We first investigate the ideal case where network connections never fail and then consider the impact of failed connections on the performances of both overlay organizations.

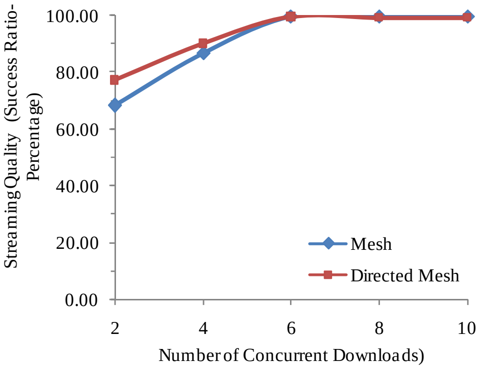


Fig. 5.1.3. Impact of increasing peer connectivity for streaming rate of 5 Mbps.

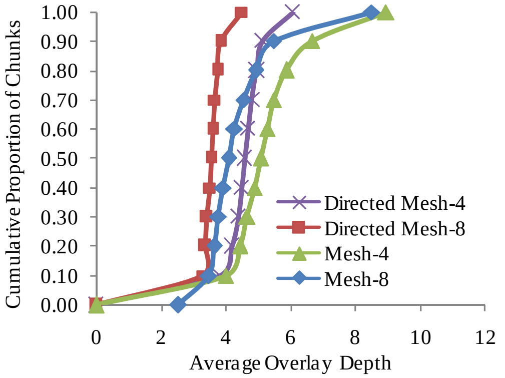


Fig. 5.1.4. Comparing average overlay depths for different peer connectivities.

5.1. Streaming Rate

Table II displays the *average streaming quality* (in terms of *success ratio*-percentage) obtained by peers in the system for streaming rates varying between 3 and 5 Mbps. The *success ratio* is defined as the fraction of chunks that arrive before their scheduled playback deadline. As we can see, both overlay organizations achieve very good success ratios for 3 and 4 Mbps and less satisfactory ratios for 5Mbps.

Fig. 5.1.1 shows how the normalized network throughput evolves over time at different streaming rates. We first observe that both overlay organizations see their network throughputs raise to a maximum then stabilizes at a value depending on the particular streaming rate. We also notice that the directed mesh organization reaches it maximum network throughput two to three times faster than the unrestricted mesh organization.

We will now focus on the performance of the two overlay organizations in a bandwidth limited environment, that is, for a streaming rate of 5MBps.

As Fig. 5.1.2 shows, the *average overlay depth*, that is, the average hop count for each chunk, for the directed mesh is 3-4 peers while that for the unrestricted mesh is approximately 4-5 peers. As a result, the directed mesh organization lets chunks reach more quickly to more peers.

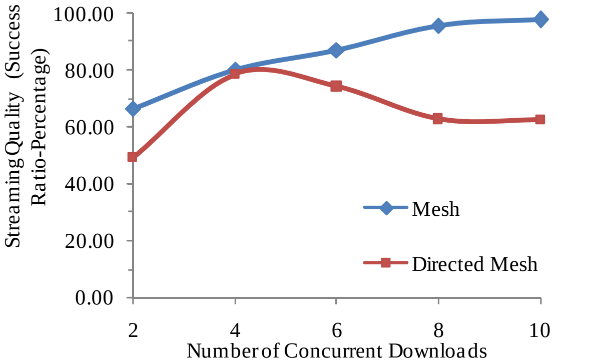


Fig. 5.2.1. Effect of increasing peer connectivity in the presence of connection failures for a streaming rate of 5Mbps.

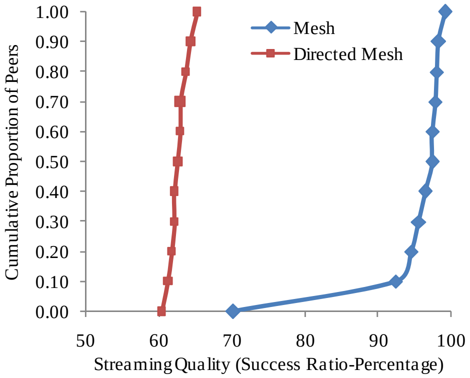


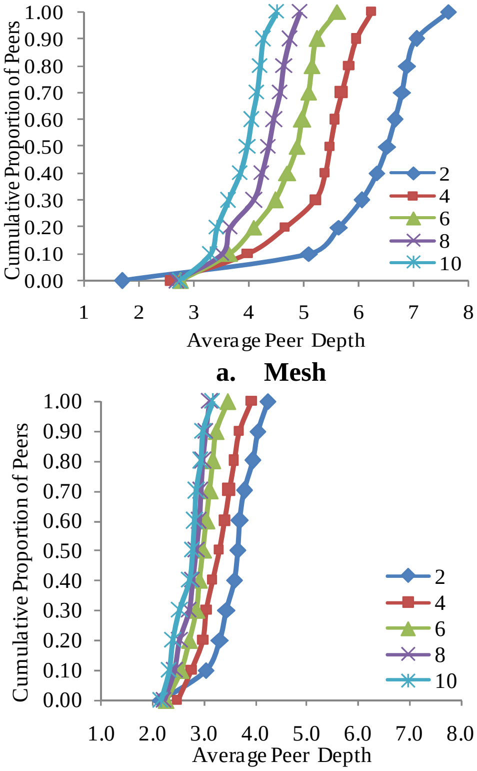
Fig. 5.2.2. Comparing overlay organization in terms of observed streaming quality for peer connectivity 8 and a streaming rate of 5Mbps in the presence of connection failures.

Consider now how the *peer connectivity*, that is, the number of concurrent upload transfers, affects the streaming quality. As Fig. 5.1.3 shows, the streaming quality improves as the number of concurrent downloads increases from 2 to 6 then becomes very close to 100 percent. As we can see in Fig. 5.1.4, one factor explaining this improvement is the lower average overlay depths experienced by chunks when the peer connectivity increases.

5.2 Impact of Connection Failures

Change in peer participation before the end of the session can be voluntary or resulting from network failures. For multimedia streaming systems, the outage is highly undesired because it generally leads to discontinuity in playback for the remaining peers. In this sub-section we study how resilient these overlay organizations can be to undetected connection failures. To evaluate the tolerance of our system to connection failures, we repeated our experiments assuming that ten percent of the connections were not working at any given time.

As Fig. 5.2.1 indicates, the streaming quality of the unrestricted mesh can approach 100 percent when the peer connectivity exceeds 8 concurrent downloads per peer. The same is not true for the directed mesh, whose performance actually deteriorates when the peer connectivity exceeds 4 concurrent downloads per peer.



a. Mesh

b. Directed Mesh

Fig. 5.2.3 Effect of peer connectivity on streaming quality in the presence of connection failures.

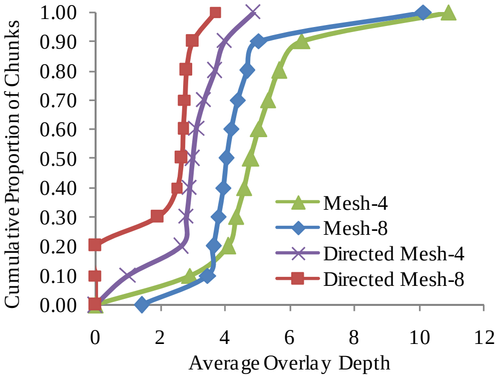
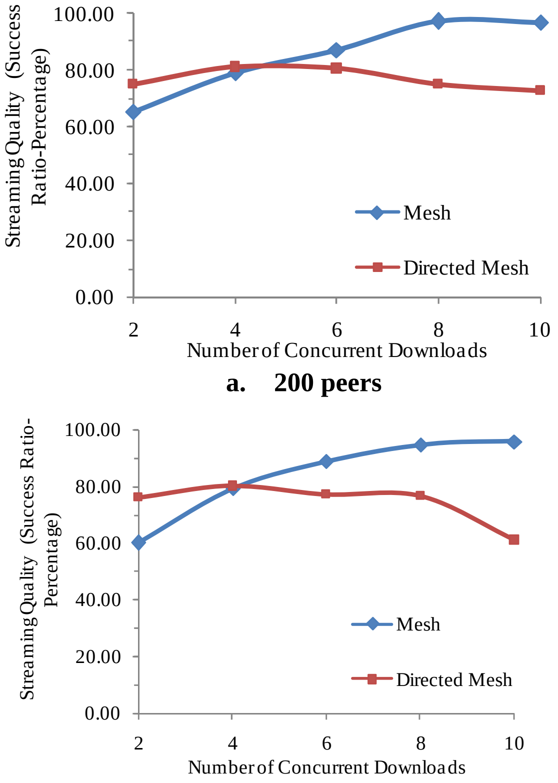


Fig. 5.2.4. Compared average overlay depth for different peer connectivities in the presence of connection failures.

Fig. 5.2.2 illustrates this phenomenon. When the peer connectivity is equal to 8, all the peers in the directed mesh experience very poor streaming qualities while more than 90 percent of the peers in the unrestricted mesh achieve streaming qualities exceeding 90 percent. This is a truly remarkable result when we consider it is achieved while 10 percent of the connections were not working.



a. 200 peers

b. 400 peers

Fig. 5.3 Effect of increasing peer connectivity for varying system sizes in the presence of connection failures.

Fig. 5.2.3 shows the CDF for *average peer depth*, that is, average hop count for peers. This explains why the performance of the unrestricted mesh improves when the peer connectivity is increased. In presence of failed connections and low peer connectivity, peers cannot always find among their neighbors all the chunks they need by the time they need them. As the peer connectivity increases, the probability of finding these chunks among their peers increases. As a result, the average hop count decreases when the peer connectivity increases and more chunks are delivered on time to more peers. Additionally, the tit-for-tat policy is able to select reciprocal neighbors and avoid performance degradation.

The poor performance of directed meshes in the presence of connection failures is further illustrated by Fig. 5.2.4. Unlike unrestricted meshes, directed meshes do not attempt to police the swarm by penalizing underperforming peers. As a result, chunks continue to be sent to peers that are unable or unwilling to forward them. The bandwidth utilization of the swarm decreases and the chunk distribution process slows down.

This vulnerability could be further exploited by *freeloaders*, that is, peers that prefer not to forward to other peers the chunks they receive.

5.3 System Size

We repeated our experiments increasing the number of peers from 100 to 400. Fig. 5.3 summarizes our results. As the system size increases the streaming quality remains virtually unaffected by the change of system sizes. In both cases, we observe that the best streaming quality is achieved by the unrestricted mesh and a peer connectivity of 8 to 10 peers.

6. Conclusions

One of the major differences among current P2P solutions for multimedia streaming is the way they organize the exchange of multimedia data among their peers. We examined the performance of two of these organizations, namely, an unrestricted *mesh* using a tit-for-tat incentive policy and a *directed mesh* letting peers dynamically replace poorly performing parents. We found out that both organizations performed fairly well under ideal circumstances where all peers were cooperating and connections never failed. We also noted that the directed mesh was able to use more quickly the network bandwidth than the unrestricted mesh and explained this observation by the lower average number of hops a chunk had to go through under the directed mesh organization.

We then considered how the same two organizations would perform in the presence of connection failures. We found out that the only organization capable of providing a high quality of service was the unrestricted mesh organization, provided that the peer connectivity level allowed at least eight concurrent downloads. The same was not true for the directed mesh whose performance actually decreased when we allowed more than four concurrent downloads. We observed that these results held for various system sizes and concluded that the tit-for-tat policy of the unrestricted mesh organization made it more fault-tolerant than the directed mesh organization.

Future work includes observing our system in an actual network. This will shed light on network dynamics and allow us to demonstrate its applicability for a wide deployment.

References

[Aka07] Akamai white papers. *http://www.akamai.com/html/perspectives/whitepapers.html.*

[BB07] S. Birrer and F. Bustamante, *A comparison of resilient overlay multicast approaches*, In IEEE JSAC Journal, 25(9), Dec. 2007.

[BHP06] A. Bharambe, C. Herley and V. Padmanabhan, Analyzing and improving a BitTorrent network's performance mechanisms, *Proc. IEEE INFOCOM Conf.* Apr. 2006.

[BL+03] S. Banerjee, S. Lee, B. Bhattacharjee and A. Srinivasan, Resilient multicast using overlays, *Proc. ACM SIGMETRICS Conf.*, June 2003.

[CD+03] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron and A. Singh, SplitStream: high-bandwidth multicast in a cooperative environment, *Proc. ACM 19th SOSP Symp.*, Oct. 2003.

[Coh03] B. Cohen, Incentives build robustness in BitTorrent, *Proc. P2PEcon Workshop*, June 2003.

[CRZ00] Y. Chu, S. Rao and H. Zhang, A case for end system multicast, *Proc. ACM SIGMETRICS Conf.*, June 2000.

[Fas07] FastTrack protocol specifications, *http://en.wikipedia.org/wiki/FastTrack*

[Gnu07] Gnutella protocol specifications *http://www.gnutella.com*.

[HH+03] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, PROMISE: peer-to-peer media streaming using CollectCast, *Proc. ACM Conf. on Multimedia*, Nov. 2003.

[JD+03] X. Jiang, Y. Dong, D. Xu, and B. Bhargava, Gnustream: a p2p media streaming prototype, *Proc. IEEE ICME. Conf.*, July 2003.

[Li04] J. Li, *PeerStreaming: a practical receiver-driven peer-to-peer media streaming system*, MSR-TR-2004-101.

[LN06] J. Liang and K. Nahrstedt, *DagStream: locality aware and failure resilient peer-to-peer streaming*, *Proc. SPIE MMCN Conf.*, Jan. 2006.

[Nap07] Napster protocol, *http://www.napster.com*.

[NR07] N. Magharei and R. Rejaie, PRIME: Peer-to-Peer receiver-driven mesh-based streaming, *Proc. IEEE INFOCOM Conf.*, May 2007.

[PK+05] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy and A. Mohr, Chainsaw: eliminating trees from overlay multicast, *Proc. IPTPS Workshop*, Feb 2005.

[PW+02] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai, Distributing streaming media content using cooperative networking, *Proc. ACM NOSSDAV Workshop*, May 2002.

[SBK02] S. Banerjee, B. Bhattacharjee and C. Kommareddy, Scalable application layer multicast, *Proc. ACM SIGCOMM Conf.*, Aug. 2002.

[SP07a] P. Shah and J.-F. Pâris. Peer-To-Peer multimedia-streaming using BitTorrent, *Proc. IEEE IPCCC Conf.*, Apr. 2007.

[SP07b] P. Shah and J.-F. Pâris, Incorporating trust in the BitTorrent protocol, *Proc. IEEE/SCS SPECTS Symp.*, June 2007.

[THD03] D. Tran, K. Hua and T. Do, Z.igzag: an efficient peer-to-peer scheme for media streaming, *Proc. IEEE INFOCOM Conf*, Mar. 2003.

[TK07] S. Tewari, and L. Kleinrock, Analytical model for BitTorrent- based live video streaming, *Proc. IEEE NIME Workshop*, Jan. 2007.

[VIF06] A. Vlavianos, M. Iliofotou and M. Faloutsos, BiToS: enhancing BitTorrent for supporting streaming applications, *Proc. IEEE GI Symp.*, Apr. 2006.

[VFC06] V. Venkataraman, P. Francis and J. Calandrino, Chunkyspread: multi-tree unstructured peer-to-peer multicast, *Proc. IPTPS Workshop*, Feb. 2006.

[XH+02] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava, On peer-to-peer media streaming, *Proc. IEEE ICDCS Conf.*, July 2002.

[YA05] W. Yang and N. Abu-Ghazaleh, GPS: a general Peer-to-Peer simulator and its use for modeling BT, *Proc. MASCOTS Symp.*, Sep. 2005.

[ZLL05] X. Zhang, J. Liu, and B. Li, On large-scale peer-to-peer live video distribution: CoolStreaming and its preliminary experimental results, *Proc. IEEE MMSP Workshop*, Oct. 2005.