Quality-aware Membership Management for Layered Peer-to-Peer Streaming

Anh Tuan Nguyen, Frank Eliassen, and Michael Welzl
Department of Informatics, University of Oslo, Oslo, Norway
{*anhtn,frank,michawe*}@*ifi.uio.no*

***Abstract*—With the standardization of SVC, the scalable extension of H.264/AVC, layered peer-to-peer streaming has attracted more and more attention as it offers adaptability to network fluctuations and heterogeneous end users. Although overlay construction is important for system performance, not much effort has been spent on unstructured overlay construction for layered peer-to-peer streaming. Related work concentrates on layered streaming algorithms, and assumes that a list of peers for data exchange, called neighbors, is provided by traditional membership management protocols, e.g., SCAMP. Our previous studies have demonstrated that a *random* overlay is not good enough for layered peer-to-peer streaming. In this paper, we propose a new membership management protocol, based on peer sampling services. The protocol is *quality-aware* as it constructs the overlay so that (1) high capacity peers will be located at good positions in the overlay, e.g., close to the server, and (2) peers having similar capacity are likely to connect to each other. Both features are necessary to maximize bandwidth utilization of peers and to mitigate layer bottlenecks. With implementation of the protocol in PeerSim, we evaluate important graph properties of the overlay, constructed by the proposed protocol, to understand how it evolves during the streaming session with peer churn. Evaluation results show that the overlay is (1) *scalable*: it is stable with different sizes, from hundreds up to 10000 peers; and (2) *robust*: the good features are maintained or recovered fast under a high peer churn rate, and it only becomes disconnected when more than 86% of the peers are removed from the network.**

I. Introduction

Layered streaming has attracted much attention because it can adapt to bandwidth variations and user heterogeneity. Adaptability is very much required when video streams are transferred across best-effort IP networks. Since SVC (Scalable Video Coding – the extension of H.264/AVC standard) was standardized in 2007 [1], there have been an increasing number of studies on layered streaming. Layered streaming is different from traditional (single layer) streaming in two important points. First, users can receive reduced quality, rather than suffer playback skips, when their bandwidth drops. Second, different users can receive different quality levels, rather than the same quality, according to their available bandwidth. With the reality of peer-to-peer (P2P) streaming [2], the use of layered coding in P2P streaming is beneficial to provide adaptive streaming to a large number of users with low cost servers. However, layered P2P streaming also poses unique challenges, of which one challenging problem is the overlay construction that needs to be designed carefully to mitigate content and bandwidth bottlenecks.

P2P overlays can be structured or unstructured (mesh-based). Structured overlays ease the data delivery. However, since they are easily affected by peer dynamics, an additional protocol is usually run to restore/reshape the overlay structure. On the other hand, unstructured overlays make the system more robust to network fluctuations without the need of a global mechanism to maintain the overlay. Therefore, unstructured overlays are more suitable for P2P streaming in dynamic environments, e.g., the Internet [3].

In an unstructured P2P overlay, each peer maintains a list of other peers, called neighbors, for data exchange. This list represents the local view that a peer has of the total system, which is critical to the video quality the peer receives. As peers can join and leave the system at any time, and bandwidth fluctuates while streaming, a membership management protocol is necessary to update the view by removing and/or adding neighbors accordingly. Owing to its important role, there have been many studies on membership management in single-layer P2P streaming. However, they are insufficient when applied to layered P2P streaming because of the following reason. In single-layer P2P streaming, peers receive the same video stream regardless of their bandwidth capacity. In layered P2P streaming, the video stream is encoded into quality layers, and peers aim to receive the maximum number of layers according to their available bandwidth. Therefore, while the average playback skip rate is the main performance metric in single-layer P2P streaming, the ratio of the experienced quality level and the expected quality level determined by the bandwidth capacity, called quality satisfaction, is also an important metric in layered P2P streaming. We have demonstrated that a *random* overlay is not good enough for layered P2P streaming [4], [5].

Not much effort has been spent on constructing unstructured overlays for layered P2P streaming. Early work concentrates on streaming algorithms, and assumes that a list of neighbors is already available or provided by a traditional membership management protocol, such as SCAMP [6]. More recent efforts on unstructured overlay construction for layered P2P streaming include a hybrid (structured and unstructured) overlay [7] and SCAMP-based protocols [5], [8]. However, how these overlays evolve over time under peer churn, and how scalable and robust they are under peer churn is still unclear.

In this paper, we present a new membership management protocol, based on peer sampling services [9], for layered P2P streaming. An important feature of the protocol is *quality-awareness* as it (1) gears high capacity peers to good locations in the overlay, e.g., close to the server; and (2) creates clusters of peers having similar capacity. The first action maximizes data flows in the overlay because high capacity peers can quickly deliver the data they have received. The second action helps mitigate the layer bottleneck problem, which happens when high capacity peers can not receive a high quality stream

This work is partially funded by the Research Council of Norway, grant number 176756/S10.

because they are surrounded by low capacity peers. Altogether, quality-aware overlays are expected to improve the overall system performance. With our implementation in PeerSim [10], we evaluate important graph properties of the overlay during streaming with peer churn. Evaluation results on large overlays, up to 10000 peers, show that the protocol is *scalable*, *robust* to peer dynamics, and *quality-aware*.

The rest of this paper is organized as follows. Section II discusses related work. Section III presents the proposed membership management protocol. Section IV analyzes simulation results. Section V concludes the paper.

II. Related Work

There has been a substantial amount of research on overlay construction in single-layer P2P streaming. Some studies focus on structured overlays, e.g., tree-based overlays [11], while others spend efforts on unstructured overlays [2]. Magharei *et. al.* [3] and Seibert *et. al.* [12] present comparisons of the P2P streaming approaches and demonstrate that mesh-based approaches consistently exhibit a superior performance over tree-based approaches in dynamic environments, while, in stable environments, tree-based systems are better in terms of delivery time. To take advantage of both approaches, hybrid overlays are also proposed, e.g., [13].

In layered P2P streaming, [14], [15] present multiple-tree based systems using multiple description coding to provide differentiated services. The general idea is that each video description is delivered in one tree, and peers can receive more than one description by being a node of more than one tree. Early work in unstructured layered P2P streaming, however, mainly focuses on peer coordination and streaming algorithms, rather than overlay construction [16], [17].

Another notable work is from Liang *et. al.* [18]. The authors propose RandPeer, a decoupled membership management for QoS sensitive P2P applications. Although it is claimed that the service is suitable for P2P streaming, its applicability to layered P2P streaming is limited because of the following reason. RandPeer uses binary trees for membership organization. It uses ID prefixes to cluster peers with similar QoS. Consequently, peers only connect to those who have the same prefix. This mechanism creates separate clusters in the overlay. However, in layered P2P streaming, it is necessary for peers with a different QoS to be connected to each other.

As one of the first efforts on constructing mesh-based overlays for layered P2P streaming, Zhao *et al.* proposed LION [7], a layered overlay multicast system. LION progressively organizes peers into layered meshes. Within each mesh, one quality layer is delivered. Each peer can subscribe to a proper number of meshes to maximize its throughput by fully utilizing its available bandwidth. Although each video layer is delivered in a mesh which is unstructured, the whole overlay structure of LION is quite well-organized and maintained by a distributed heuristic algorithm. LION aims to support small-scale applications in stable environments.

SCAMP [6] is a well-known gossip-based membership management protocol. There has been work tailoring SCAMP for layered P2P streaming. Xiao *et. al.* propose OCals [8], which constructs the overlay in two stages. The first stage, SCAMP-based, is to probe existing nodes to find a certain number of logical partners, which are interested in the same set of layers. In the second stage, it will select neighbors for each layer based on the RTT (Round Trip Time). In our previous work [4], [5], we have also proposed a SCAMP-based protocol for layered P2P streaming. The protocol classifies peers into classes based on their bandwidth capacity. It uses both the class and the current quality of peers to choose neighbors.

Jelasity *et. al.* [9] propose a framework to implement peer sampling services based on gossiping. A peer chooses one of its neighbors to exchange views; then, both peers know more peers in the system for updating their view. The authors demonstrate that such peer sampling services construct more scalable and robust overlays than SCAMP-based protocols. In addition, since explicit attempts are made towards the construction of an overlay in SCAMP, peer sampling services are less expensive in terms of bandwidth cost as fewer requests are sent in the network. However, the current peer sampling services are not quality-aware as they do not take peer capacity into account when choosing neighbors.

The idea of locating higher capacity peers closer to the source than lower capacity peers is similar to the idea of gradient overlays, recently presented in [19], [20]. There are several main differences between these studies and ours. The work of Sacha *et. al.* [19] does not target P2P streaming applications. It elects super peers with highest utility to discover globally similar neighbors, while lower utility peers have mostly random neighbors. The election method is unaffordable for P2P streaming with a strict timing requirement. Payberah *et. al.* present gradienTv [20] using gradient overlays for live P2P streaming. GradienTv is a multiple tree based system for single-layer P2P streaming, whereas our protocol is made for unstructured layered P2P streaming. The media source splits the media into a number of stripes, and it constructs an overlay tree for each stripe. GradienTv constructs the gradient overlay by using two neighbor lists – random view and similar view – and tries to maintain the tree structure for all peers, while, in our protocol, peers locate themselves in different locations by exchanging their *local* view. We reserve comparing gradienTv with our protocol for future work. However, we believe that our simple, yet effective, approach is more suitable in highly dynamic environments than gradienTv.

III. Protocol Design

This section describes in detail our quality-aware membership management protocol for layered P2P streaming. The protocol is initially based on the generic peer sampling service [9]. Therefore, the first part of this section gives a brief overview of the peer sampling service, while the second part presents the proposed protocol.

A. Peer Sampling Protocol: An Overview

A peer maintains a local view including up to S (the maximum view size – defined by the system) neighbors. At the beginning, when a peer P joins the system, it contacts a rendezvous peer (a well-known peer) to receive the IP address of a peer Q in the system to start with. Q will send its view to

P, and P will consider this view as its initial view. After the above join process, periodically, P selects one neighbor N to exchange its view with. This selection is called *PeerSelection*. After the exchange, P and N have new candidates for updating their view: some neighbors may be removed from, or some candidates may be added to the neighbor list. This step is called *SetView*. It is known that this simple gossip-based protocol is scalable and creates overlays with self-organizing and self-healing (self-*) properties [9].

B. A Quality-aware P2P Membership Management Protocol

To inherit the advantageous features of the peer sampling protocol, our protocol is based on the view exchange mechanism, but with enhancements for quality-awareness. The idea is to differentiate peers in the view exchange by taking peer capacity into account. In particular, by setting different priorities for different neighbors in PeerSelection and SetView of each peer, it is expected that the resulting overlay would have high priority peers at good locations, while low priority peers are located in the remaining part of the overlay. To achieve a certain level of randomness, which is important to the self-* properties [9], peers having the same priority are chosen randomly in PeerSelection and SetView.

In layered P2P streaming, the video stream is encoded into a number of quality layers. The number of quality layers a peer receives depends on its bandwidth capacity. It has been shown that (1) peers with similar capacity should connect to each other to maximize bandwidth and layer utilization [4] (F1), and (2) high capacity peers should be closer to the streaming source than low capacity peers to avoid layer bottlenecks [5] (F2). To satisfy these two requirements, our new membership management protocol works as follows:

- ▷ When a new peer P contacts a rendezvous peer R to join the system, R will send properties of the stream, e.g. the number of quality layers (L) and bit rates, back to P. It also assigns a class identifier C_i $(0 < i \leq L)$ to P based on P's bandwidth capacity and the video properties. Peers of class C_i have higher bandwidth capacity and can receive more quality layers than peers of class C_j if $i > j$. The server is assigned to the class C_{L+1} (or any value larger than the highest identifier used for peers).
- ▷ Neighbors in P's view are arranged based on their class identifier. Those who have the same identifier as P are located at the head of the view, followed by those with higher identifiers in increasing order, and finally those with lower identifiers in decreasing order. The arrangement is illustrated in Figure 1.
- ▷ PeerSelection: a peer Q will be selected from the head of the view. If more than one peer of the same class exist, one is chosen at random.
- ▷ SetView: after a view exchange, new candidates are inserted into the current view of P according to the above arrangement. The view is likely to have more than S peers. Then, S peers are selected from the head of the current view to make the new view. This view update is also carried out at Q's side.

The arrangement of the neighbor list and the ordered selection in PeerSelection and SetView are simple but able to achieve

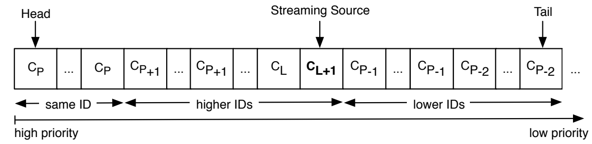


Fig. 1. The arrangement of neighbors in the neighbor list of peer P. C_P is the class identifier of P.

the two preferred features for the overlay. It can be seen from Figure 1 that, starting from the head of the list:

- ▷ F1: P prefers neighbors who belong to the same class.
- ▷ F2: If there are not enough such neighbors, P chooses those with higher class identifiers. Since the streaming source has the highest class identifier, the higher P's class identifier is, the more likely it is that the source will be added to the new view because it is closer to the head of the list. On the other hand, the source is placed towards the tail of the list if P's class identifier is low. Consequently, low capacity peers do not have good chances to connect directly to the source.

However, initial experiments have shown that, if we just take the S first neighbors from the head of the list, peers of the same class are gradually connected to each other and do not have connections with peers of other classes. Consequently, separate clusters are created in the overlay. To keep the overlay connected, each peer should have at most K $(K < S)$ neighbors of the same class, and keep at least $S - K$ connections with other classes, which are also chosen based on the priority in the list. These $S - K$ connections are particularly helpful for lower classes as they have links to higher classes to receive data. The ratio of K to S is called *clustering ratio* as it determines how well peers of the same class are connected.

IV. SIMULATION

PeerSim [10] is chosen for implementing the proposed membership management protocol because it provides a simple network model and good support for investigating graph properties of the overlay in both static and dynamic scenarios. We start this section by introducing metrics to evaluate the protocol. Then, simulation results are presented and analyzed.

A. Evaluation Metrics

To separately evaluate our mechanism, we do not use a streaming protocol on top of the overlay. Rather, we assume that a *good* overlay will lead to a high performance streaming system. The important question is: *what is a good overlay?* The graph of an overlay is generated by considering peers as nodes and neighbor relationships as edges. By analyzing the graph, we understand how the protocol behaves and what the overlay looks like over time. From experiments with Chameleon [4], we are particularly interested in:

- ▷ Connectivity: is the graph well connected during streaming? The most basic requirement for the protocol is that no peers are disconnected from the system, i.e., there is always at least one path from the source to any 'alive' peer at any time even with peer churn.
- ▷ Clustering: the clustering coefficient of a group of peers is defined as the ratio of links existing among the peers

of the group over the total number of links of those peers. If we consider that peers of the same class are in one group, the clustering coefficient of that group shows how well peers of the class are connected. If the cluster coefficient of a group is 1, the group is disconnected from the network (called separated cluster). On one hand, a high clustering coefficient is good in terms of bandwidth utilization. On the other hand, the higher the clustering coefficient, the higher the probability of the class being disconnected from the network, especially under high churn rate. This is a tradeoff between connectivity and clustering.

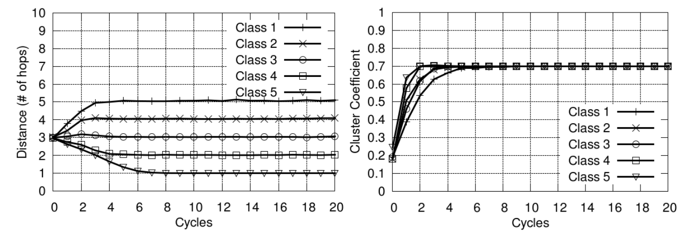
▷ Average path length: is the average of the shortest path lengths (number of hops) between any two peers. In this work, the average length between peers and the source is more relevant. Therefore, we will evaluate the average path length of different peer classes to the source. In this context, we call the average path length between peers and the source the *distance* to the source. It is desired that high capacity peers have smaller distances to the source than low capacity peers.

B. Simulation Results

We face difficulties in choosing related membership management protocols to compare with ours because none of them looks into the graph properties of the overlay during the streaming session to see how they affect the performance. Previous protocols are evaluated indirectly through the overall streaming performance, which is impacted by other components, e.g., peer coordination and streaming algorithms. Convinced that understanding the topology of the overlay and how it evolves over time is important to improve the overall performance of a streaming system, in this paper, we separately evaluate the overlay construction by concentrating on its graph properties. Rather than comparing with related work in layered P2P streaming, we use CYCLON [21], a traditional membership management protocol, as a baseline. The purpose is to emphasize differences and trade-offs (if existing) between quality-aware and non quality-aware protocols.

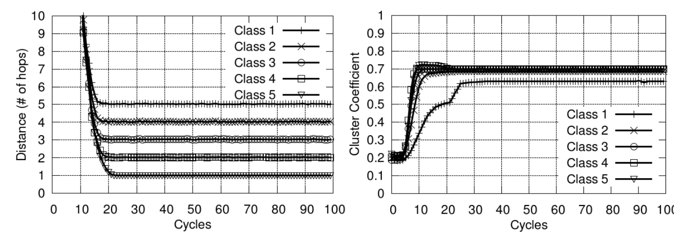
A layered stream with 5 quality layers is used. Corresponding to the stream, there are 5 peer classes in the system. The number of peers in our experiments is set to 10000 in static scenarios. This is also the maximum number of peers in dynamic networks. Each peer is randomly assigned to one (and only one) class. The view size S is set to 50 as in [4], and the clustering ratio is set to 0.7, which means that each peer has at most 70% neighbors of the same class, and the other 30% are neighbors from other classes. Different values of S and the clustering ratio are also experimented and presented in the technical report [22]. The cycle-based engine of PeerSim is used. In each cycle, the protocol is run at every peer. After each cycle, properties of the graph are plotted. To evaluate the effect of randomness in our experiments, each experiment is run several times with different random seeds in PeerSim. Since the standard deviations of the metrics are very low, which means that the randomness does not cause instability to the overlay, we only plot the average values of the metrics.

1) Static networks: In this scenario, the overlay has a fixed number of peers (10000), and starts with a certain topology. We would like to observe how the protocol behaves and constructs the overlay. Two initial topologies used are one *ring* topology and one *random* topology. Figure 2 and Figure 3 show the clustering coefficient and the distance of the peer classes for the two cases. Since CYCLON does not take peer capacity into account when choosing neighbors, the metrics have approximate values for different classes (not shown in the figures for clearness).



(a) Distance (b) Clustering Factor

Fig. 2. The overlay evolves from a random topology.



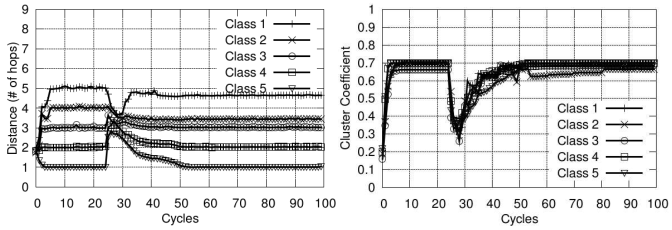
(a) Distance (b) Clustering Factor

Fig. 3. The overlay evolves from a ring topology.

Firstly, the overlay is well connected in all cases (not shown in the figures). Secondly, it can be seen from Figure 2 and Figure 3 that the overlay can converge to a stable overlay regardless of its initial topology with desired properties: the peer classes have high clustering coefficients (approximate to the clustering ratio), and the higher the class identifier, the smaller the distance to the source. In particular, at the beginning, the peer classes have similar distances and clustering coefficients. However, when the quality-aware protocol is invoked, the overlay quickly changes itself to differentiate peer classes. In Figure 2, it takes only about 6 cycles, which means that each peer only exchanges its view 6 times, to achieve stability from a random topology. The ring topology requires more time (about 25 cycles) because each peer has only two neighbors at the beginning (Figure 3). However, after converging, the overlay is still able to achieve similarly good values for the metrics as in the case of the random topology.

2) Dynamic networks: We now turn our attention to how the protocol behaves under peer churn. In the first experiment, we have a stable overlay with 200 peers until the 24^{th} cycle. From the 25^{th} cycle, in each cycle, 200 peers join the system until the size of the overlay reaches 10000. Figure 4 shows the clustering coefficient and the distance of the peer classes as peers join.

It can be seen from Figure 4 that, at the 25^{th} cycle, the clustering factor and the distance change rapidly. The



(a) Distance (b) Clustering Factor

Fig. 4. The overlay evolves as peers join.

fluctuation is caused by doubling the network size from 200 to 400 peers (flash crowd scenario). However, after that, the overlay converges to a stable state quickly (in about 25 cycles), even when new peers keep joining the system.

Finally, to evaluate the robustness to leaving peers, we remove peers randomly and calculate how many separate clusters are created. We carry out 50 experiments. From a stable overlay, we remove 50%, 51%, ..., 99% of the total number of peers at random. Figure 5 shows the number of separate clusters created in the experiments, with the results obtained from similar experiments on CYCLON, a more random-based protocol, to compare their robustness.

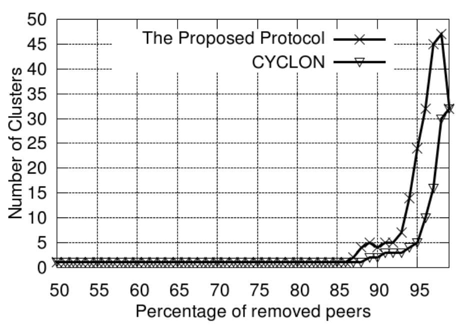


Fig. 5. Number of clusters generated when peers leave the system

Figure 5 shows that the proposed protocol is robust to peer leaves as the first disconnection occurs only when up to 86% of the total number of peers are removed. However, the proposed protocol is less robust than the random-based protocol, whose first disconnection occurs only when at least 88% of peers are removed. In addition, the number of separate clusters (disconnected sub-graphs) created by CYCLON is smaller than by our protocol. The reason is that the clustering coefficient of peer classes in the proposed protocol is higher than that of the peer classes in CYCLON. This can be considered as a tradeoff between bandwidth utilization and robustness. However, with the robustness up to 86% of the total number of peers leaving the system, we believe that the proposed protocol is robust enough for P2P streaming.

V. Conclusion

We propose a new membership management protocol for layered P2P streaming. Compared to traditional protocols, the new feature of our protocol is quality-awareness. In addition to scalability and robustness, the overlay has good features for layered P2P streaming: (1) peers having similar capacity are likely to connect to each other, and (2) higher capacity peers are closer to the source than lower capacity peers. Simulation results demonstrate the advantageous features of the protocol. Currently, peer capacity is defined as peer bandwidth. However, the capacity can include other parameters, e.g., RTT or physical distance between peers. As future work, we are building up a more practical protocol by taking underlying network parameters into account. Another direction is to evaluate the whole streaming system by adding a layered P2P streaming protocol, e.g. Chameleon [4], on top of the overlay.

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[2] X. Zhang, J. Liu, B. Li, and T.-S. Yum, "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," in *Proc. of IEEE INFOCOM 2005*, March 2005.

[3] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live p2p streaming approaches," in *Proc. of IEEE INFOCOM 2007*, May 2007.

[4] A. T. Nguyen, B. Li, and F. Eliassen, "Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding," in *IEEE INFOCOM*, 2010.

[5] ——, "Quality- and Conext-aware Neighbor Selection for Layered Peer-to-Peer Streaming," in *Proc. of IEEE ICC 2010*, May 2010.

[6] A. Ganesh, A.-M. Kermarrec, and L. Massoulie, "Peer-to-peer membership management for gossip-based protocols," *IEEE Trans. on Computers*, vol. 52, no. 2, pp. 139–149, Feb. 2003.

[7] J. Zhao, F. Yang, Q. Zhang, Z. Zhang, and F. Zhang, "LION: Layered Overlay Multicast With Network Coding," *IEEE Trans. on Multimedia*, vol. 8, no. 5, pp. 1021–1032, October 2006.

[8] X. Xiao, Y. Shi, B. Zhang, and Y. Gao, "OCals: A Novel Overlay Construction Approach for Layered Streaming," in *Proc. of ICC*, 2008.

[9] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-based Implementations," in *Proc. of the 5th ACM/IFIP/USENIX International Conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2004, pp. 79–98.

[10] PeerSim: A Peer-to-Peer Simulator. [Online]. Available: http://peersim.sourceforge.net/

[11] D. A. Tran, K. A. Hua, and T. Do, "ZIGZAG: an efficient peer-to-peer scheme for media streaming," in *Proc. of INFOCOM 2003*, April 2003.

[12] J. Seibert, D. Zage, S. Fahmy, and C. Nita-Rotaru, "Experimental Comparison of Peer-to-Peer Streaming Overlays: An Application Perspective," in *LCN 2008*, Oct. 2008, pp. 20–27.

[13] F. Wang, Y. Xiong, and J. Liu, "mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live Video Multicast," in *IEEE ICDCS 2007*, June 2007.

[14] J. D. Mol, D. H. P. Epema, and H. J. Sips, "The Orchard Algorithm: Building Multicast Trees for P2P Video Multicasting Without Free-Riding," *IEEE Trans. on Multimedia*, vol. 9, no. 8, Dec. 2007.

[15] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: high-bandwidth multicast in cooperative environments," in *Proc. of the 19th ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003, pp. 298–313.

[16] R. Rejaie and A. Ortega, "PALS: peer-to-peer adaptive layered streaming," in *Proc. of NOSSDAV 2003*, June 2003.

[17] Y. Cui and K. Nahrstedt, "Layered Peer-to-Peer Streaming," in *Proc. of NOSSDAV 2003*, June 2003.

[18] J. Liang and K. Nahrstedt, "RandPeer: Membership Management for QoS Sensitive Peer-to-Peer Applications," in *IEEE INFOCOM*, 2006.

[19] J. Sacha, B. Biskupski, D. Dahlem, R. Cunningham, R. Meier, J. Dowling, and M. Haahr, "Decentralising a Service-Oriented Architecture," *Peer-to-Peer Networking and Applications*, vol. 3, pp. 323–350, 2010.

[20] A. Payberah, J. Dowling, F. Rahimian, and S. Haridi, "*gradienTv*: Market-Based P2P Live Media Streaming on the Gradient Overlay," in *Distributed Applications and Interoperable Systems*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6115.

[21] S. Voulgaris, D. Gavidia, and M. V. Steen, "CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays," *Journal of Network and Systems Management*, vol. 13, 2005.

[22] A. T. Nguyen, "Quality-aware Membership Management for Layered Peer-to-Peer Streaming," *Technical Report, Department of Informatics, University of Oslo*, 2010.