Low Latency Overlay Construction for Peer-to-peer Live Streaming *

Xinggong Zhang, Yan Pang, Zongming Guo
Institute of Computer Science & Technology
Peking University, Beijing, China
{zhangxg, pangyan, guozongming}@icst.pku.edu.cn

Abstract

Peer-to-peer live streaming is becoming popular and a large number of multimedia content is streamed to millions of users through it. Some major problems associated with the applications are long playback latency and fragile to network churn. In this paper, we propose a novel approach to construct a pyramid overlay. Compared with the conventional mesh-based and tree-based overlay, the pyramid overlay can achieve better result both in real-time playback and robust topology. Through limiting peers' connection bandwidth with upper layers and make it a priority to disseminate data to more different peers, more peers can share the upload bandwidth from the upper layers and the peer number is increased with level increment. Thus, our approach organizes the peers into a pyramid topology and reduces the total number of layer. At the same time, the participating peers build connections with multiple parent peers and child peers simultaneously and change data transmission path dynamically. Then, it is resilient to network churn. Experiments carried out over a simulated network of up to 500 peers illustrate the effectiveness of our approach.

1. Introduction

In recent years, peer-to-peer (P2P) live streaming becomes more and more popular worldwide [10, 9, 5]. A large amount of multimedia resources such as music, video program are streamed to millions of internet users through P2P network. P2P network offers an efficient approach to deliver multimedia content from one source to a large number of receivers by using the resources of participating peers. Because each peer contributes its bandwidth or disk resources, the P2P network can be easily scaled up to a large number of peers without additional requirements on the central server. Hence, P2P streaming is a competitive approach for multimedia content distribution.

*This work was supported by National Key Technology R&D Program of China under contract No.2006BAH02A10 and National High-tech R&D Program (863 Program)of China under contract No.2006AA01Z320.

Real-time playback is an essential requirement for P2P live streaming. The playback latency is the duration from the time a data segment is sent out from the root server to the time it is received by the destination peers. Recently, several studies have proposed the packet schedule scheme to address the problems[6] [11]. They focus on scheduling the data transmission in order to improve the overall latency and transmission efficiency. However, the approach has the following limitations: (1) the scheme targets to minimize the queuing delay at each peer, ignoring the number of relay hops; (2) the scheme depends on the structure of the underlying multicast trees. It is a heavy burden to maintain the structure in a dynamic network.

Robust overlay is another key requirement for P2P live streaming. Some previous works aim to construct a scalable and robust overlay, and a number of mesh-based P2P live streaming systems have been designed to meet these requirements, such as Coolstreaming [14] and Prime [7]. These systems focus on building a robust overlay with respect to network churn, and improving the bandwidth utilization among the participating peers. However they fail to achieve real-time playback. The segments are often relayed along long multi-hop paths and each hop introduces additional delay to the playback latency. Hence the solutions based on mesh-based approaches give us the impression that P2P live streaming system has a long playback latency.

In this paper, we propose a novel approach to organize peers into a pyramid topology, which accommodates more peers with fewer layers. Through reducing the average relay hops, the average relay delay can be significantly reduced. We firstly identify an important factor related to the playback latency: the number of relay hops along data transmission path. In our approach, it is a priority to disseminate data to more different peers. Through limiting the peers' connection bandwidth with the upper level, more peers can share the upload bandwidth of peers in the upper level. Compared with the tree-based overlay and mesh-based overlay, the pyramid can achieve low playback latency and resilience to network churn.

The rest of this paper is organized as follows: Section 2 gives a brief analysis on mesh-based and tree-based overlay. In section 3, we compare the pyramid overlay with a mesh-based overlay, and present its advantage over other overlay. Section 4 describes details of the dissemination-first approach and the procedure to construct a pyramid overlay; Section 5 presents the simulation methodology over a simulated network of up to 500 peers and the experimental results; Finally, a conclusion is given in Section 6.

2. Motivation

In this section, we firstly analyze a mesh-based overlay and calculate its playback latency; then we investigate the single tree overlay and the multi-tree overlay, and find their playback latency can be much less than the mesh-based overlay. For each overlay, we also analyze their drawbacks in delay and robustness.

For each peer in the overlay, its playback latency is the sum of relay delays at each hop along the transmission path. Each relay delay mainly consists of two parts: the buffer delay (q_i) and the transmission delay(t_i). The buffer delay is measured by the time that a peer keeps the data before it sends out it, while the transmission delay is measured by the time from the moment a segment is sent out to the moment this data segment arrives at next peer. For example, there are N relay peers along the transmission path to the destination peer, its playback latency is:

$$T_N = \sum_{0 \le i \le N} (t_i + q_i) \tag{1}$$

According to Equation (1), we could see that there are three methods to reduce the playback latency: reducing the transmission delay and buffer delay at each peer, and the total number of relay hops. Some previous works introduce a method to reduce the delay at each peer. However, they ignore the relay hops which is a main reason for latency. On the other hand, if we can reduce the numbers of relay, some peers will be removed from the transmission path and the delay will be eliminated. Based on the above consideration, our dissemination-first approach is designed to reduce the relay hops and the peer number along the transmission path.

2.1. Mesh-based Overlay

In a mesh-based overlay, the data transmission path is established dynamically and the topology is built based on data availability [14] or network resilience [7]. The participating peers build connections with multiple parents peers and child peers simultaneously, and frequently inquire their data. If they found some interesting data in a peer, they would build a new transmission path with it. Once they receive all the new segments and there is no more interesting data in the neighbor peers, the transmission path will be destroyed. Then the topology of mesh-based overlay always keeps changing and irregular.

In above irregular topology, some peers would exhaust all upload bandwidth of the root server or near-root peers, but the others may be far away from the root server and have much more relay hops[12]. There is no coordination among peers. Each peer tries to find more interesting peers and requests data as much as possible. The result is that only a few of peers are closed to root server and have small relay hops, the rest peers have long relay hops.

We present an optimal mesh-based overlay, in which all upload bandwidths have been used up by the peers in upper level. Once a new peer wants to join into the overlay, it searches the peers with minimum hops and available upload bandwidth, and attaches to them. As shown in Fig.1, only the bottom peers have rest bandwidth.

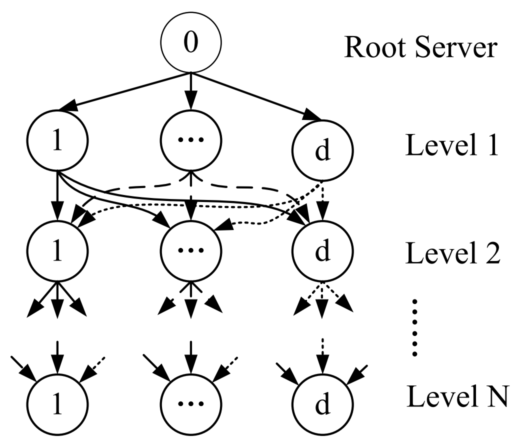

Figure 1. An optimal mesh-based overlay

Suppose, in Fig.1 there are P peers. Each peer has the same number of incoming and outgoing connections(degrees), denoted as d. t denotes the time to relay one data segment along one connection. In the scenario to stream d segments from the root server, each peer in level 1 builds one connection with the root server and requests the different segments. After t time, there are d different segments in level 1 and the time to receive all data for the peers in level i is as Equation (2). Only for the peers in level 1, it takes dt time to receive all data. So the playback latency T_i for the peers in level i is:

$$T_i = \begin{cases} d \cdot t & \text{if} \quad i = 1 \\ i \cdot t & \text{if} \quad i > 1 \end{cases} \tag{2}$$

For there are not spare upload bandwidth in each layer, each layer has d peers and the total layer numbers of overlay are

$(P-1)/d$. The average playback latency of overlay is:

$$\overline{T} = \sum_{0 \le i \le \frac{P-1}{d}} T_i \cdot d/P \tag{3}$$

In the scenario that the peer numbers P are greater than 1, the average playback latency approximately equals to:

$$\overline{T} \approx 0.5t + \frac{(P-1)t}{2d} \tag{4}$$

Thus, the latency algorithmic complexity of mesh-based overlay is $O(P)$. For an overlay with millions of users, the average playback latency is increased as a linear function of peer numbers. This is the reason that the previous P2P streaming systems have long playback latency.

2.2. Tree-based Overlay

For the same number of peers, the tree structure overlay has the fewest layer number. Some tree-based systems, such as Splitstream[4], NICE[2], Zigzag[13] organize participating peers into single tree or multi-tree topology. A peers can join multiple trees in the multi-tree overlay. It is placed as an internal node in only one tree and as leaf node in the rest trees.

The tree topology is fragile in a dynamic network and need a large amount of overhead to maintain the structure. The topology is static. The join procedure is complex and time consuming. The new peer searches from the root to the bottom layers in order to find a parent peer which has spare outgoing bandwidth. Then it attaches to the parent peer and keeps its position unchanging unless the parent peer departs. When a peer leaves the overlay, all of its child peers would depart from the overlay and rejoin the system. Therefore, the tree structure is fragile and complex.

For a single tree overlay, its average playback latency is as Equation (5).

$$\overline{T} \approx t \cdot log_d[P \cdot (d-1) + 1] - \frac{td}{d-1} \tag{5}$$

For a multi-tree structure overlay, it comprehends the feature of single tree and mesh[8]. Each peers connects with multiple parents peers and child peers. It constructs a structured mesh by organizing peers into multiple tree. Peers connect with all trees and receive one stream from each tree. If a peer leaves a tree, it only affects the stream in one tree and its child peers can receive the stream data from other trees. But in a multiple tree overlay, the playback latency for a peer is the last data which it receives. Therefore, the playback latency for a multi-tree overlay is:

$$\overline{T} \approx t \cdot log_d[P \cdot (d-1) + 1] - t \tag{6}$$

Thus, in a tree-based overlay, its latency algorithmic complexity is $O(log_d P)$. Compared with the mesh-based overlay, its playback latency is reduced. On the other hand, the tree topology maintenance workload is heavy. It needs a lot of bandwidth and time to recover the system from churn and exchange status messages among peers. It is a heavy workload for a large scale overlay.

3. Pyramid Overlay

To solve above problems, we target to construct an overlay with the minimal layers and the robust topology.

The pyramid topology is an extension of the mesh topology. It consists of overlapping meshes with successively increasing peers. The peers in each level form a mesh and their incoming bandwidth is less than their outgoing bandwidth. In the topology, the peer number of each layer is increased from top level to bottom level. At the same time, each peer keeps connection with multiple parents and child peers, and changes the data transmission path dynamically. To construct a overlay as Fig.2, we limit peers' incoming degree to $d/2$.

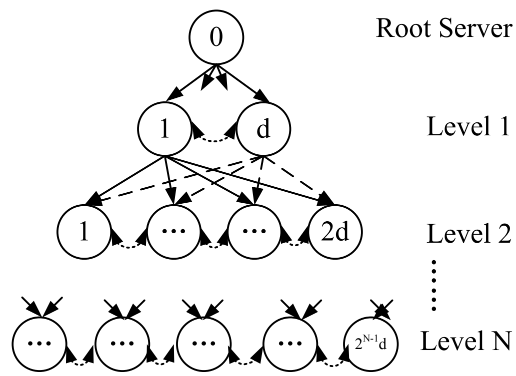

Figure 2. An example of pyramid topology

For an overlay with P peers, the spare outgoing degree is $(P+1) * d/2$. It means the overlay can accommodate $P+1$ new peer in the bottom level. The root server is at level 0 and accommodates d peers in level 1. From level 1, each peer has d outgoing bandwidth and $d/2$ incoming bandwidth. So the peer number for level i is:

$$P_i = \sum_{0 \le j \le i-1} P_j + 1 = d \cdot 2^{i-1} \tag{7}$$

To compare with mesh-based overlay, we set the same parameter in the dissemination-first overlay. The delay for level 1 is dt, and at the t time, there are d different segments in level 1. Because each peer has $d/2$ incoming degree, the delay in level 2 is $2t$ time. For other layers, the delay is t.

So the playback latency for level i is:

$$T_i = \begin{cases} d \cdot t & \text{if} \quad i = 1 \\ (i+1) \cdot t & \text{if} \quad i > 1 \end{cases} \tag{8}$$

N denotes the bottom level. From (7) and (8), the total latency for the overlay is:

$$T = \sum_{0 \leq i \leq N} T_i \cdot P_i = d \cdot t \cdot (d + \sum_{2 \leq i \leq N} i \cdot 2^{i-1}) \tag{9}$$

For the overlay, the peers number P is:

$$P = \sum_{0 \leq i \leq N} P_i = 2^N d + 1 - d \tag{10}$$

From the (9) and (10), and if P is greater than d, the mean playback latency of the overlay is:

$$\overline{T} = T/P = t \cdot \Big(1 + log_2\Big(\frac{P-1}{d} + 1\Big)\Big) \tag{11}$$

Hence, its algorithmic complexity of playback latency is $O(log(P))$. For a large scale P2P system, the average playback latency of pyramid overlay is increased as a logarithm function of peer numbers. Compared with the mesh-based overlay, average playback latency have a significant reduction in the pyramid overlay.

4. Construction of Pyramid Overlay

We propose a dissemination-first approach to organizes peers into a pyramid overlay. In the approach it is a priority to disseminate data to more different peers. Each peer is assigned a level and builds multiple connections with neighbor peers. We limit the peer's connection degree with upper level in order to accommodate more peers with increasing level. The peers build a data transmission path dynamically according to the data availability and distribute the different data to the child peers. Through the dissemination-first approach, we can build a robust overlay with less levels.

In our approach, there are four categories of peers: the root server, the index server, the common peer, and the bottom peer. Among them, the root sever initially generates the streaming data, and the index server manages all the peers' arrival and departure. In our approach, we use a centralized node management algorithm like BitTorrent [3]. Before new peers attach to a parent peer, all of they belong to the bottom peers which are on the bottom level of the overlay. The common peer's incoming degree with upper layer is limited to be the half of its outgoing degree, while the bottom peer does not have any limitations on its incoming degree, and can exchange data with any peer.

To construct the overlay, there are two procedures: the joining procedure, the parent selection procedure.

In the joining procedure, the newly joining peer contacts with the index server and announces its arrival. Then the index server searches for the peer which has spare outgoing degrees. If there is not suitable peer in the current level, we would move to the next level to continue the search until we find one. Thus, the new peer gets its level, and all peers in the upper level become the candidate neighbors peers of the new peer.

In the parent selection procedure, the new peer builds up connections with its neighbor peers and selects the parent peers from them. Firstly it sends out a data request message and listens to their replies. When it received replies from them, the new peer checks its current number of incoming connections with the neighbors. If the connections are less than the incoming limitations, it selects the peer to be its parent peer and build a new data transmission path with it.

5. Experiment

We have evaluated the performance of the proposed approach over a network simulated in ns-2. [1] Based on the assumption that the backbone links are sufficiently provided and the bottleneck of the network is the access links, we used a star topology with asymmetric access links. The downlink capacity is 8 times larger than the uplink speed, and the latency of each link ranges from 1-50ms randomly. All connections are congestion controlled using RAP and the number of peers participating in the overlay varies between 10 and 500. The bit rate of streaming media is 1Mbps and its length is 1 hour. The segment unit is 1 second. There is no peer in the system at the beginning, and all the peers join it randomly within the first 10s. After all peers have joined the system, the root server begins to generate steaming data.

5.1. Simulations Result

Fig. 3 illustrates the distribution of data arriving time for the two overlays. As we expected, the percentage population of the proposed approach, indicated by the left vertical axis, increases quickly as the time passes. Almost 70 percent peers have received the data in 20 seconds from the time 2470s to the time 2490s. Compared with the mesh-based overlay, the peers in our approach have almost the same playback latency. It is because the levels of the newly joining peers are reasonably arranged and each peer has equal chances to get data from upper levels.

We also compared the two overlays with the different numbers of participating peers. The result in Fig. 4 shows the advantage of the proposed approach for the large scale peers. As the number of the peers increases, the average playback latency in our approach increases slowly, and it is nearly a logarithm curve. In contrast, the average playback

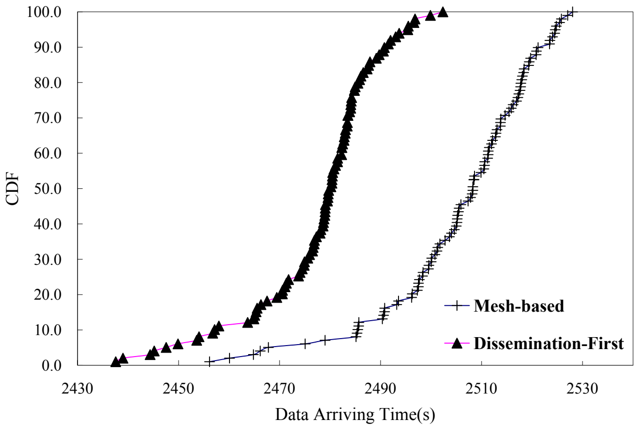

Figure 3. Distribution of data arriving time

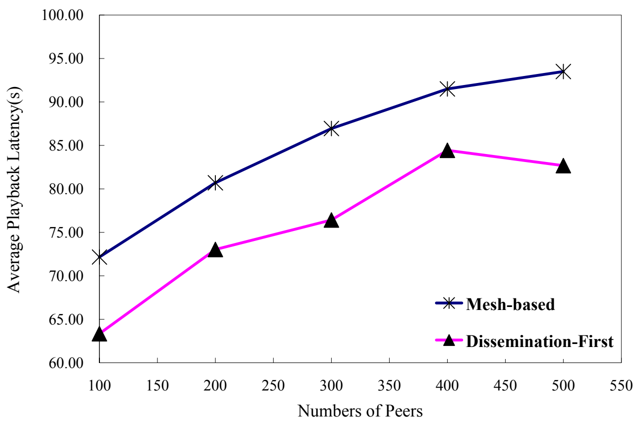

Figure 4. Average playback latency

latency increases linearly in mesh-based overlay. Some peers exhausts all upload bandwidth of the root server or near-root peers, so that the overall latency is longer than our approach.

6. Conclusion

In this paper, we presented a novel approach to construct a pyramid overlay and minimize the playback latency in P2P streaming. The approach organizes peers into a pyramid overlay through restricting the peer's incoming bandwidth, thus reduces the overlay layers. Comparing with the mesh-based overlay, the relay hops is reduced and the average playback latency is restricted in a limited duration. Simulation results validate the effect of pyramid overlay and show the scalability of our system. Our future work is to deploy the experiments on the planet-lab test bed and optimize it in a churn network.

References

[1] *The Network simulator: ns2*. Sources and Documentation from http://www.isi.edu/nanam/ns.

[2] S. Banerjee, B. Bhattachrjee, and C. Kommareddy. Scalable application layer multicast. In *Proceedings of ACM SIGCOMM*, volume 31, August.

[3] A. Bharambe, C. Herley, and V. Padmanabhan. Analyzing and improving a bittorrent networks performance mechanisms. In *Proceedings of the INFOCOM*, pages 1–12, April 2006.

[4] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Nandi. Splitstream: High-bandwidth multicast in cooperative enviroments. In *Proceedings of the SOSP*, pages 298–313, 2003.

[5] M. Hefeeda, A. Habib, B. Bptev, and D. Xu. Promise:peer-to-peer media streaming using colletcast. In *Proceedings of the ACM MM*, pages 45–54, Nov. 2003.

[6] J. Li, C. Yeo, and B. Lee. Peer-to-peer streaming scheduling to improve real-time latency. In *Proceedings of the ICME*, pages 36–39, July 2007.

[7] N. Magharei and R. Rejaie. Prime:peer-to-peer receiver-driven mesh-based streaming. In *Proceedings of the INFOCOM*, pages 1415–1423, May 2007.

[8] N. Magharei, R. Rejaie, and Y. Guo. Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In *Proceedings of the INFOCOM*, pages 1424–1432, May 2007.

[9] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proceedings of the NOSSDAV*, pages 177–186, May 2002.

[10] S. Ratnasamy, M. Handley, and R. Karp. Application-level multicast using content-addressable networks. In *Proceedings of the NGC*, 2001.

[11] E. Setton, J. Noh, and B. Girod. Low latency video streaming over peer-to-peer networks. In *Proceedings of the ICME*, pages 569–572, July 2006.

[12] T. Small, B. Liang, and B. Li. Scaling laws and tradeoffs of peer-to-peer live multimedia streaming. In *Proceedings of ACM Multimedia*, pages 539–548, October 2006.

[13] A. A. Tran, K. A. Hua, and T. T. Do. A peer-to-peer architecture for media streaming. *IEEE J. Select. Areas Commun.*, 22:121–133, Jan 2004.

[14] X. Zhang, J. Liu, B. Li, and T. S. P. Yum. Coolstreaming: A data-driven overlay network for live media streaming. In *Proceedings of the INFOCOM*, pages 2102–2111, 2005.