A Construction of Peer-to-Peer Streaming System Based on Flexible Locality-Aware Overlay Networks

Chih-Han Lai[1], Yu-Wei Chan[2], and Yeh-Ching Chung[1]

[1] Department of Computer Science, National Tsing Hua University
Hsinchu, Taiwan 30013, R.O.C
chl@sslab.cs.nthu.edu.tw, ychung@cs.nthu.edu.tw
[2] Department of Information Management, ChungChou Institute of Technology
Yuanlin, Taiwan 510, R.O.C
ywchan@dragon.ccut.edu.tw

Abstract. In the peer-to-peer multicast system, participants as peers are organized to construct overlay topology over physical infrastructures. In this manner, peers can easily disseminate data and gather from others by running multicast application. However, the negative impacts such as non-guaranteed transmission efficiency, heterogeneity of peers, dynamic of peers, which were related to the topology of overlay and directly affect the performance metrics, for example, the delivery efficiency and perceived quality. In this paper, we propose flexible locality-aware overlay to get better performance metrics. In the system, a peer can simply establish a streaming session and also as a source without the need of dedicated servers. The overlay is constructed with 2-layered structure to match the underlying topology and shorten the delivery paths. From the simulation results, our system has been demonstrated it had better transmission efficiency, shorter delivery delay, and higher reliability compared with those systems which have been developed.

1 Introduction

The success of peer-to-peer technology motivates the advance of peer-to-peer multicast [2] [4]. When applying streaming applications over peer-to-peer overlay network, the peer-to-peer streaming systems [5] [6] [8] [14] [16] [18] employ the neighbors of peers in an overlay as the streaming suppliers. These suppliers are chose by the topology of overlay, and directly affect the performance metrics, such as delivery efficiency and perceived quality. Due to the negative impacts such as non-guaranteed communication efficiency, limited upload capacity, dynamic of suppliers, etc.., these metrics may not been satisfied. As a result, how to form an overlay to properly combat these impacts is thus the challenge issues. A well-designed overlay for peer-to-peer streaming can keep stable suppliers, shorten transmission delays, and also balance the load of peers.

In this paper, we propose a flexible 2-layered locality-aware overlay by using the group concept to construct a peer-to-peer streaming system. By exploiting the surrounding neighbors of peers with low communication delay, the delivery efficiency

S. Wu, L.T. Yang, and T.L. Xu (Eds.): GPC 2008, LNCS 5036, pp. 296 – 307, 2008.

and perceived quality can be enhanced in our system. In the proposed 2-layered overlay, peers are clustered into locality groups based on the communication delay. These locality groups form the top layer of the overlay and interconnected as a tree rooted by the streaming source. In each locality group, peers form an overlay mesh for streaming. These overlay meshes form the bottom layer of the overlay. In order to construct the 2-layered overlay efficiently, some schemes are proposed to let peers of the system locate themselves into proper groups well are as follows:

1. The peer locating scheme: it is proposed to aid peers group locating.
2. The membership management scheme: it is used to help peers with organizing the membership of peers in locality groups.
3. The split and merge schemes: they are designed to let the overlay adjust itself with the dynamics of peers.
4. The backup group probing scheme: it is used to enhance the performance of the constructed peer-to-peer streaming system.

Applying the group concepts to the constructed system will enhance the delivery efficiency and perceived quality. For example, peers can not only obtain streaming suppliers easily from others which are in the same locality group, but also shorten the delivery latency from suppliers of other groups. Since the number of peers in a locality group has upper and lower-bounded limitation, the overlay mesh helps peers gather sufficient bandwidth and retain perceived quality more easily. In a streaming session, data disseminated from a streaming source to every end-host through locality groups which has been connected. By the locality groups, the communication latency of two peers in the same locality group will be decreased. Since the delivery paths of the source-to-end are composed of the delivery links of peers, the shorter delay of every links will result in shorter delay totally.

In order to evaluate the proposed architecture, we have implemented the system with proposed scheme on the simulator with varied physical topologies, different streaming data rates, and availabilities of peers. The results of the system are compared with AnySee [8]. The simulation results show that our work can achieve better source-to-end delivery latency with different physical topologies and data rates. The perceived quality still retained high within acceptable delay while AnySee can not. Besides, the reliability of source-to-end delivery path is higher than AnySee.

The remainder of this paper is organized as follows. Session 2 reviews the related work. Session 3 describes our proposed streaming system and its schemes. Session 4 represents the simulation setup of our system. Session 5 proposes some experimental results. Session 6 concludes the paper.

2 Related Work

Many schemes have been proposed for efficient peer-to-peer streaming. The goal of these schemes is to assure that the delivery efficiency and perceived quality metrics can be constantly satisfied. They can be classified into tree-based peer-to-peer overlays [3, 6, 19, 21, 28] and mesh-based peer-to-peer overlays [9, 12, 14, 30, 34].

Most peer-to-peer multicast systems are based on tree-based overlays. CoopNet [11] is the pioneering peer-to-peer streaming system. A centralized approach is

employed to efficiently maintain the distribution tree, but may lead to the overload of the streaming source due to the huge connections. Scribe [4] was built upon the structured peer-to-peer overlay. It leverages the dedicated overlays with its native multicast routing schemes. In [13], the authors proposed some schemes based on the topology-awareness of underlying CAN [12] to improve the delivery efficiency. NICE [2] and Zigzag [14] adopt the hierarchical clustering and split/merge heuristics to minimize the transmission length. They were sensitive to node dynamics and needed to adjust the topology frequently that may cause worse streaming quality. Due to the streaming of high bit rate, the tree-based structure is not suitable properly because it does not take the heterogeneity of peers into account.

The mesh-based overlay is a novel model for peer-to-peer multicast since it takes the heterogeneity of upload of peers into account. Bullet [6] is a scalable and distributed algorithm used for constructing high-bandwidth streaming overlay. In Bullet, nodes can self-organize into an overlay tree to transmit the disjoint data sets and retrieve the missed parts simultaneously. Xiang et al. [16] builds a framework for media distribution service on top of mOverlay [19], a group-based locality-aware overlay. In [16], the proposed distributed heuristic replication strategies can leverage locality groups to efficiently disseminate media content. CollectCast [5] is the multi-supplier streaming service built on top of peer-to-peer lookup substrate. The specially constructed topology and selection algorithm are used to yield an active streaming sender set from a candidate peer set. DONet [18] is a data-driven overlay network for live media streaming. By employing a gossiping protocol, peers can periodically exchanges the availabilities of data blocks for retrieving yet unavailable data and supplying available data. However, the streaming quality of DONet can not be guaranteed. AnySee [8] is a peer-to-peer live streaming system built on top of Gnutella [1]. The location-aware topology matching (LTM) [9] scheme and the adaptive connection establishment (ACE) [17] scheme are proposed to optimize the connections of neighbor peers to tackle the power-law effects [2, 24]. In AnySee, by the usage of LTM and the proposed inter-overlay optimization scheme, a peer can retain efficient and available streaming paths on the mesh-based overlay.

3 System Overview

Fig. 1 shows the proposed 2-layered overlay structure. In Fig. 1, peers are clustered into groups with bounded size. The communication delays of peers in a locality group are below a pre-defined threshold. The top layer of the overlay consists of locality groups which are interconnected as a multicast tree rooted by the streaming source. Each locality group holds a derive level that represents the level in the multicast tree. The change of the derive level of a locality group indicates that the split or merge of the locality group. If the derive level is smaller, a peer joins this locality group would experience less relay time for gathering data from streaming source. In each locality group, peers form an overlay mesh for streaming and these overlay meshes form the bottom layer of the overlay. Due to the constructed structure, streaming data can be rapidly distributed. Thus, the efficiency of streaming delivery can be enhanced for the peers located in diverse locations.

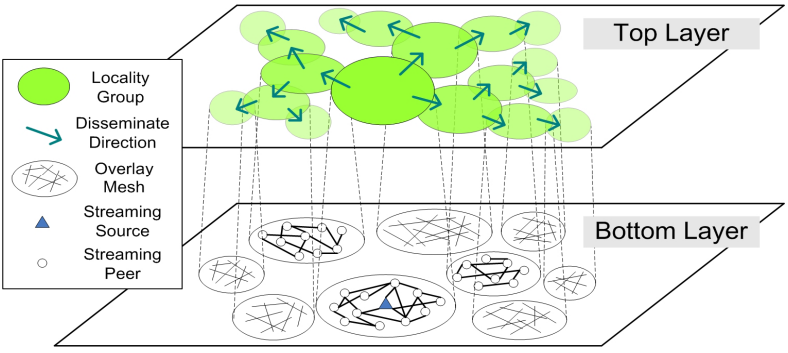

Fig. 1. Proposed flexible locality-group based peer-to-peer overlay network architecture

In this paper, we propose some schemes which have been constructed to make the system more efficiently. An *indexing server* is used to keep the information of streaming sessions with the correspondingly constructed overlay. The new peers join the proper locality group of the overlay by using the *peer locating scheme*. Streaming data from the streaming source are disseminated along with the multicast tree by continuous requests and relays. The clustered peers in a locality group are managed by the *membership management scheme*. To keep sufficient and stable suppliers, the *split/merge scheme* for overlay maintenance would be performed on locality groups. The scheme makes the overlay flexible and scalable because of the ability to grow or shrink the number of groups in an overlay. For those peers that cannot satisfy the performance metrics, the *backup peer probing scheme* is used to improve the satisfaction of peers. In the following, we will describe these schemes in detail.

3.1 The Locality Group

A locality group consists of a set of peers. In this paper, we assume that peers in a locality group are classified into two disjoint subsets, *candidate* and *separate* subsets. For peers in the candidate subset, network delays among peers are less than or equal to a predefined value according to the rate of a streaming session. In this paper, the predefined value was set between $l/2$ and l based on [16, 19], where l is the tolerable delivery latency. The delays of peers between the candidate subset and the separate subset are greater than the predefined value. The size of a locality group is bounded by $[k, (3k - 1)]$ according to [2, 14], where $k \geq 1$. If the size of a locality group is equal to $3k - 1$, it represents that the locality group is full. When a peer joins the full group, it will cause the locality group spilt into 2 groups. If the size of a group except for the streaming source is less than k due to some peers leave, the locality group will be merged with other groups resulting a size under $3k$. If no such locality group available, the merge will be delayed until such a locality group is available; or be aborted when the size of the locality group is greater than or equal to k again.

In the system, each peer will join the *default group* initially. Certain peers may act as gateway-like peers by joining another locality group which is the *source group* to handle the relays among groups. They gather streaming data from the source group and disseminate them to members which are in default group. The derive level of the source

group equals to the derive level of the default group minus 1. In this situation, peers may play different roles in each joined locality group. A peer is called a *contributor* in a locality group if it contributes its upload bandwidth and helps to forward the stored streaming data. A contributor is called a *maintainer* in a locality group if it is responsible for overlay maintenance and membership management. A peer is called a *free-rider* if it is neither a contributor nor a maintainer in a locality group.

3.2 The Indexing Server

The indexing server records the essential information of published sessions and corresponding overlays as metadata. End users can obtain a list of metadata of sessions from the indexing server. Four operations which are query, add, update, and remove, are provided to access the indexing server for overlay construction and maintenance. The metadata format stored in the indexing server is divided into two parts which are named as SSPR and LGR. The SSPR represents the specification of an established streaming session. It consists of two fields, session ID and rate. The session ID field is used to recognize each streaming session. The rate field is used to specify the streaming data rate of this session. The LGR stores the information of locality groups in the corresponding overlay. It consists of three fields, group ID, derive level, and maintainer which used to record the ID, the derive level, and the maintainer of a locality group.

3.3 The Peer Locating Scheme

To establish a peer-to-peer streaming session, the streaming source acts as the maintainer of the initial locality group. It first publishes the properties of streaming session by inserting values of the rate field of SSPR and the maintainer field of LGR to the indexing server. After receiving the information, the indexing server then constructs the metadata of the session by assigning values to the session ID field of SSPR and the group ID field of LGR and setting the value of the field of derive level of LGR to be zero. Finally, the group ID is sent back to the streaming source.

When an end host p_i decides to participate a published streaming session s_j, it will call the peer locating scheme to join a locality group according to the LGR records of the session. The peer locating scheme is performed as follows:

Step 1. If no entry of LGR of s_j is stored in the group cache of p_i, then p_i gets one entry from the indexing server and inserts this entry with measured network delay of p_i.

Step 2. For the first m entries in the group cache of p_i, the maintainer in each entry sends all entries to p_i, where m is the system defined *probe number*. After received all entries from maintainers, p_i inserts these entries with measured network delays of p_i. This step is performed n times, where n is the *group probing threshold*.

Step 3. In the group cache of p_i let S_1 be a set of LGR entries whose network delays are under the predefined value according to the rate of s_j. If there is an LGR entry whose derive level is the smallest one, the locality group in this entry is the one for p_i to join. If two or more locality groups satisfy the condition, the one with the smallest network delay will be selected. If no LGR entry can be selected in S_1, the selection with the same policy is applied to S_2.

Step 4. If all locality groups of LGR entries in the group cache are full, if S_1 is not empty, the locality group of the entry with the smallest derive level will be selected. Otherwise, the locality group of the entry with the smallest derive level in S_2 will be selected.

In the peer locating scheme, the group cache of each peer is used to store the LGR entries with measured network delay. The maintainers act as *dynamic landmark* for positioning in the overlay. The indexing server randomly selects an LGR entry as a bootstrap for the peer locating scheme to distribute the probe requests of peers among all locality groups. If some peers can not be located to a candidate subset of a locality group, this scheme accommodates them into proper group to reduce the times of adjustments.

3.4 The Membership Management Scheme

The membership management scheme is used to organize the membership in a locality group. Based on structure of the super-peer network, the maintainer of a locality group in the system acts as the super-peer to handle the join and leave operations of peers, monitor the status of peers, manage contributors, and broadcast the information of contributors.

In this system, a *member cache* is used to store the information of members in a locality group. For each joined group, a peer maintains the corresponding member cache. The information stored in the member cache consists of four fields, *type*, *network address*, *contributor rank*, and *subset*. The type field specifies the role of a member. The network address field is used to record the network address of a member. The contributor rank field is used to record the rank among all contributors. The rank is used to recover the failure of the maintainer and for the split scheme. The subset field specifies the subset (candidate or separate) of a member belongs. For monitoring the status of peers, a maintainer receives the "keep alive" messages from its members constantly to assure that they are alive. If a peer is available to be a contributor, it informs the maintainer of the default group. When a contributor lacks of the streaming data in its data cache, it will inform the maintainer. The maintainer will set the contributor as the free-rider. Based on the management of contributors, a maintainer periodically updates the information of contributors to each member. Besides, the LGR entries of the source group of the maintainer would be broadcasted periodically to organize contributors and recover failures of the maintainer.

3.5 The Overlay Maintenance Scheme

To keep sufficient and stable suppliers for streaming and ensure the loading of a maintainer, the split and merge schemes will be performed on locality groups if the number of peers in a locality group is over its bounded size or less than a threshold, respectively. In this system, a maintainer periodically checks the size of its locality group and performs the split/merge schemes if needed.

3.5.1 The Split Scheme

When the size of a locality group is larger than $3k - 1$, the following procedure is performed to split this locality group into two locality groups.

Step 1. The maintainer m_i of a locality group g_i chooses the contributor c_j with the lowest rank in its member cache as the maintainer of a new locality group.

Step 2. The contributor c_j claims itself as the maintainer m_j of a new locality group g_j by adding an LGR entry to the indexing server and acknowledges m_i the new group ID g_j.

Step 3. To decide what members should be located in the new locality group, m_i uses the following criteria to select k candidates. m_i will first select those members that fit the following criterion 1. If the number of members selected is less than k, then it will select those members that fit criterion 2, and so on, until k members are selected.

Step 4. The maintainer m_i creates a *split list* that stores the information of these k candidates, broadcasts the split list along with the LGR entry of g_j to all members in g_i, and alters the status of the contributors in the split list and c_j to free-rider in its member cache.

Step 5. When a member received the split list, it refers Table 1 to locate itself to proper group(s). When m_j changes its source group later by the split scheme, this member should follow this change as well.

Step 6. If the derive level of the source group of a maintainer changes, the derive level should be modified correspondingly. The maintainer would update the field of derive level of the LGR entry and inform this change to its members.

Table 1. Guidance of m_i when received the split list

Condition of m_i (C_1: gather streaming bandwidth from the contributors in the split list)	Decision
not in the split list and C_1 is not met	stays in g_i
not a contributor in the split list or C_1 is met	migrates from g_i to g_j
a contributor in the split list and C_1 is not met	joins g_i and g_j to relay data streams

3.5.2 The Merge Scheme

To keep moderate resources in each locality group, a locality group would perform the merge scheme when the size of the locality group is under the predefined threshold k. Assume that the size of a locality group g_i is under the predefined threshold k. The maintainer m_i of g_i first queries the maintainer, m_s of its source group g_s to obtain the size of g_s. The procedures of the scheme are that if the size of g_s is less than $3k$ after merging with g_i, all members in g_i would join g_s and m_i would act as a contributor in g_s. The corresponding LGR entry of g_i would be removed from the indexing server by m_i. For those peers that are free-riders in g_i, they need to change their derive levels.

3.6 The Backup Group Probing Scheme

When a peer is in the separate subset of a locality group, the perceived streaming quality of this peer cannot be constantly satisfied. As long as this peer acts as a

contributor, it cuts down the streaming delivery performance. To tackle those negative effects, the backup group probing scheme is proposed to optimize our overlay based on the size of the locality group. The following is the procedure of the scheme.

Step 1. A maintainer of a locality group g_i periodically checks whether its size exceeds $2k$. If yes, it selects k members from the separate subset based on the time order they joined g_i for backup group probing.

Step 2. If a member p_a selected is in the candidate subset, p_a will try to find a locality group g_j in S_1 of its group cache such that the measured network delay of p_a and the maintainer of g_j is less than or equal to $l/2$ and the size of g_j is less than $3k$.

Step 3. If a member p_a selected is in the separate subset, p_a will try to find a locality group g_j in S_1 of its group cache such that the measured network delay of p_a and the maintainer of g_j is less than or equal to l and the size of g_j is less than $3k$.

4 Simulation Setup

In this section, we present the simulation setup for the evaluation. In our simulation, we generate two types of topologies, physical and logical. The physical topology represents the real network topology based on the Internet characteristics. The logical topology is composed of a number of hosts which act as peers to form the peer-to-peer overlay upon the physical topology. We adopt the Hierarchical Top-down model with GLP model [3] on AS/router layer on BRITE [10] and the pure router model on Inet-3.0 [15] to generate 5000 nodes graphs of physical topology with varied settings to yield different network delays. The detail parameters we applied on BRITE and Inet-3.0 are described in [7].

We simulate our system by running an experimental application framework on each end host. In the framework, the implemented protocol formulates the 2-layered overlay. The way we simulate the AnySee [8] system is to construct the underlying mesh-based (Gnutella-type) overlay. We observe that the dynamics of streaming paths of AnySee and evaluate its efficiency. In all simulations, we assume that the first joining peer in an overlay will act as the streaming source and will never fail. The details of the parameters we used are described in [7].

5 Performance Evaluations

In this section, we evaluate our proposed work and AnySee. Based on different aspects, we take the measurements to compare the performance of these both systems by analyzing the behavior of the corresponding overlays.

We evaluate the performance based on two major parts. Firstly, we evaluate the average of maximum delivery latency of a data block from the streaming source to each participant. The related queuing delays and processing delays are ignored. Secondly, we evaluate the average communication delays between participants and its upstream peers.

5.1 Results for Different Physical Topologies

Here we compare the proposed overlay with AnySee based on four different topologies. Fig. 2 and 3 depict the measured source-to-end delays and the average

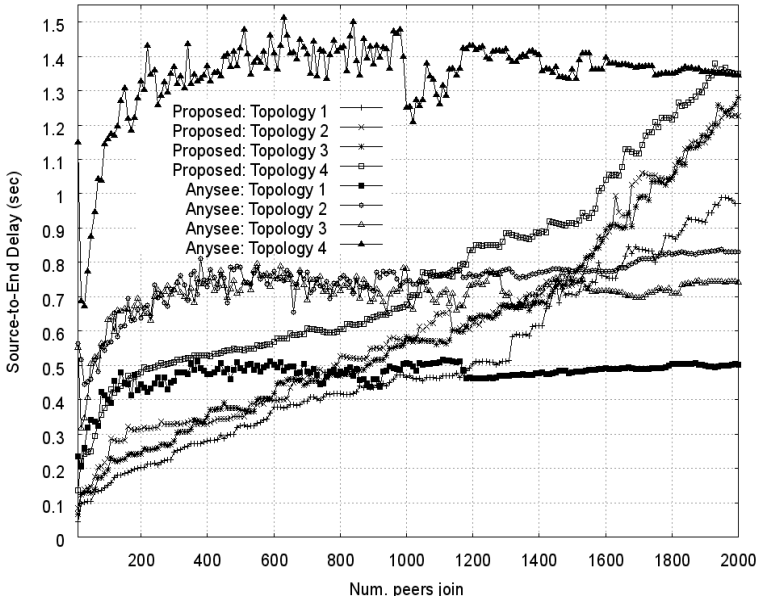

Fig. 2. Source-to-end delay under different topologies

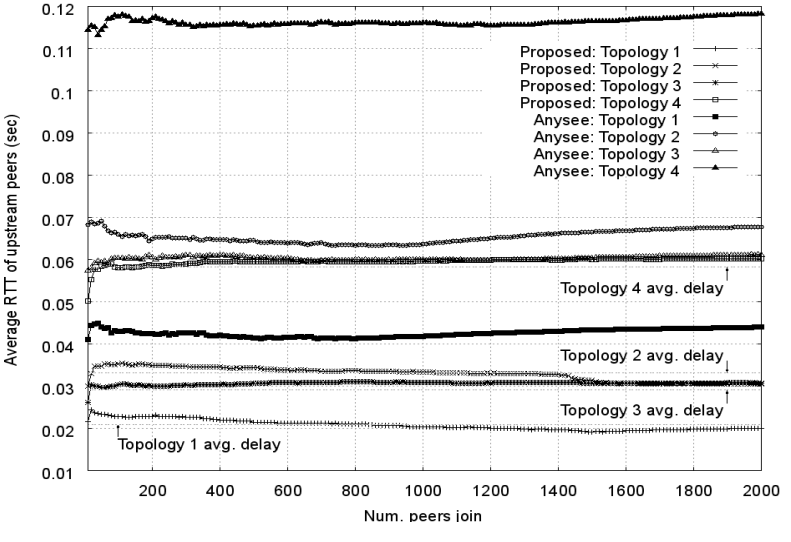

Fig. 3. Communication delay under different topologies

communication delays with increasing overlay size. In Fig. 2, the results show that the delivery latency increases because of the growing number of relay hops/groups with the increasing participants. In contrast, a peer in AnySee must actively examine the available streaming paths. According to the Fig. 2, we can realize that when the average delay of nodes increases (from Topology 1 to Topology 4), our system scales better. Also, from the Fig. 3, we can show that our system works better than Anysee that is shorter link delays and better streaming quality.

5.2 Results for Peers Failure

In the section, we investigate the behavior of two overlays by considering the failure of peers. We schedule failure "trials" in every 7 seconds throughout a stream session. Upon each trial, a peer in an overlay is selected randomly. If a randomly generated number between 0 and 1 is greater than the availability of this peer, it would fail. Otherwise, this peer keeps joining and the session continues normally until the next trial. In our simulations, the mean availability of participants is varied from 0.6 to 1.0.

We compare the proposed work with AnySee. The results are shown in Fig. 4 and 5. Fig. 4 points out the population are less than 1000, the source-to-end delivery delay decreases as the mean availability of peer decreases. This phenomenon reflects the flexibility of our system which can adjust the topology to shorten the delivery latency while AnySee cannot. It is shown in Fig. 5.

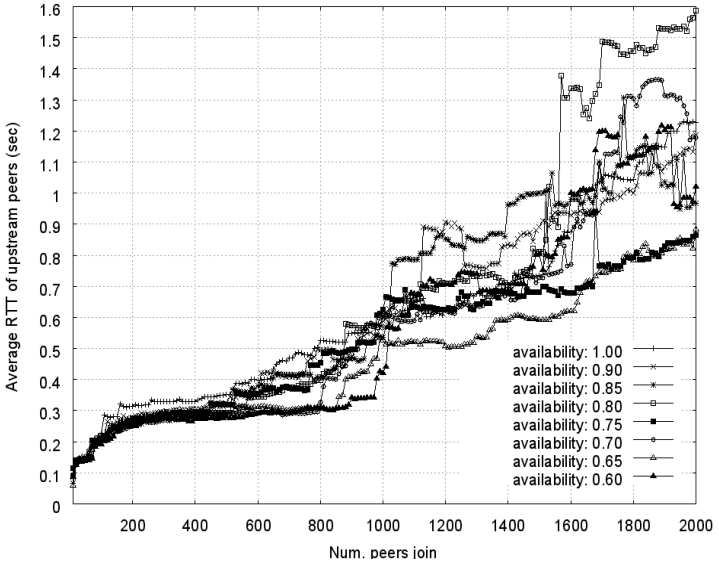

Fig. 4. Source-to-end delay of our system with peer failures

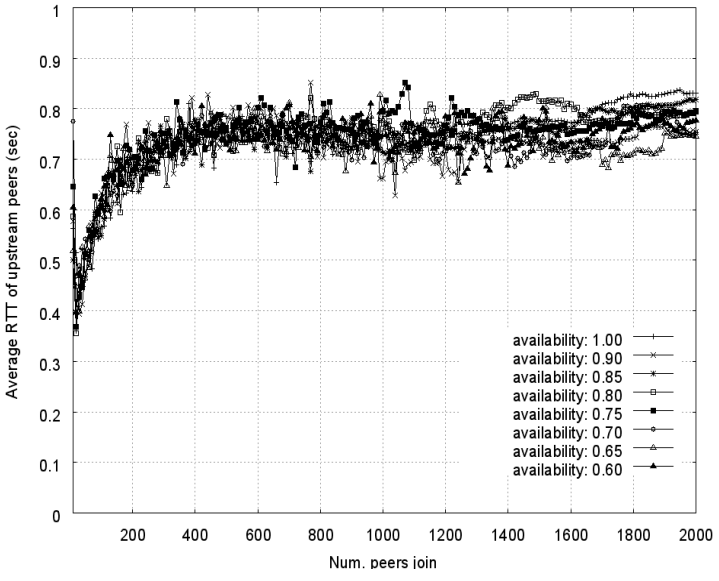

Fig. 5. Source-to-end delay of AnySee with peer failures

6 Conclusions

In this paper, we have presented a peer-to-peer streaming system based on a flexible 2-layered locality-aware overlay network. In our system, a peer can simply establish a streaming session and as a streaming source without the help by dedicated streaming servers. Based on the flexibility and locality-awareness in our overlay, session participants as peers would benefit from sufficient, stable, and efficient suppliers in the joined locality groups for streaming. Compared with AnySee, the simulation results show the proposed overlay exhibits a degree of source-to-end delivery efficiency, and lower communication latencies of streaming suppliers. Moreover, our system also retains higher reliability on streaming delivery paths. Those results demonstrate the scalability, efficiency and stability of our system, in which the data stream delivery efficiency and the perceived quality can be constantly satisfied.

References

1. Gnutella Website, http://www.gnutella.com
2. Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application layer multicast. In: Proceedings of conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM 2002, Pittsburgh, PA, USA (August 2002)
3. Bu, T., Towsley, D.: On distinguishing between Internet power law topology generators. In: Proc. INFOCOM 2002, New York City, NY, USA (June 2002)

4. Castro, M., Druschel, P., Kermarrec, A.-M., Rowstron, A.I.T.: Scribe: a large-scale and decentralized application-level multicast infrastructure. IEEE J. Select. Areas in Comm. 20(8) (October 2002)
5. Hefeeda, M., Habib, A., Xu, D., Bhargava, B., Botev, B.: CollectCast: A peer-to-peer service for media streaming. ACM/Springer Multimedia Systems Journal 11(1) (November 2005)
6. Kostic, D., Rodriguez, A., Albrecht, J., Vahdat, A.: Bullet: high bandwidth data dissemination using an overlay mesh. In: Proc. ACM SOSP 2003, NY, USA (October 2003)
7. Lai, C.H., Chung, Y.C.: A Construction of Peer-to-Peer Streaming System Based on Flexible Locality-Aware Overlay Networks, M.S. Thesis, National Tsing-Hua University, Hsin-Chu, Taiwan (2007)
8. Liao, X., Jin, H., Liu, Y., Ni, L.M., Deng, D.: AnySee: Peer-to-peer live streaming. In: Proc. INFOCOM 2006, Barcelona, Catalunya, Spain (April 2006)
9. Liu, Y., Xiao, L., Liu, X., Ni, L.M., Zhang, X.: Location awareness in unstructured peer-to-peer systems. IEEE Transactions on Parallel and Distributed Systems 16(2) (February 2005)
10. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: an approach to universal topology generation. In: Proc. MASCOTS 2001, Cincinnati, OH (August 2001)
11. Padmanabhan, V.N., Wang, H.J., Chou, P.A., Sripanidkulchai, K.: Distributing streaming media content using cooperative networking. In: Proc. NOSSDAV 2002, Miami, FL, USA (May 2002)
12. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. In: Proc. ACM SIGCOMM 2001, San Diego, CA, USA (August 2001)
13. Ratnasamy, S., Handley, M., Karp, R., Shenker, S.: Topologically-aware overlay construction and server selection. In: Proc. INFOCOM 2002, New York, NY (June 2002)
14. Tran, D.A., Hua, K.A., Do, T.T.: A peer-to-peer architecture for media streaming. IEEE J. Select. Areas in Comm. 22(1) (January 2004)
15. Winick, J., Jamin, S.: Inet-3.0: Internet topology generator, Technical Report, CSE-TR-456-02, Department of EECS, University of Michigan (2002)
16. Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z., Zhang, Y.-Q.: Peer-to-peer based multimedia distribution service. IEEE Transactions on Multimedia 6(2) (April 2004)
17. Xiao, L., Liu, Y., Ni, L.M.: Improving unstructured peer-to-peer systems by adaptive connection establishment. IEEE Transactions on Computers 54(9) (September 2005)
18. Zhang, X., Liu, J.-C., Li, B., Yum, T.-S.P.: Coolstreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In: Proc. INFOCOM 2005, Miami, FL, USA (March 2005)
19. Zhang, X.Y., Zhang, Q., Zhang, Z., Song, G., Zhu, W.: A construction of locality-aware overlay network: mOverlay and its performance. IEEE J. Select. Areas in Comm. 22(1) (January)