A flexible locality-aware peer-to-peer streaming system

Yu-Wei Chan
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, Republic of China and Department of Information Management, Chung Chou Institute of Technology, Yuanlin, Taiwan, Republic of China, and
Chih-Han Lai and Yeh-Ching Chung
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, Republic of China

Abstract

Purpose – Peer-to-peer (P2P) streaming quickly emerges as an important application over the internet. A lot of systems have been implemented to support peer-to-peer media streaming. However, some problems still exist. These problems include non-guaranteed communication efficiency, limited upload capacity and dynamics of suppliers which are all related to the overlay topology design. The purpose of this paper is to propose a novel overlay construction framework for peer-to-peer streaming.

Design/methodology/approach – To exploit the bandwidth resource of neighboring peers with low communication delay, application of the grouping method was proposed to construct a flexible two-layered locality-aware overlay network. In the proposed overlay, peers are clustered into locality groups according to the communication delays of peers. These locality groups are interconnected with each other to form the top layer of the overlay. In each locality group, peers form an overlay mesh for transmitting stream to other peers of the same group. These overlay meshes form the bottom layer of the overlay.

Findings – Through simulations, the performance was compared in terms of communication efficiency, source-to-end delivery efficiency and reliability of the delivery paths of the proposed solution currently. Simulation results show that the proposed method can achieve the construction of a scalable, efficient and stable peer-to-peer streaming environment.

Originality/value – The new contributions in this paper are a novel framework which includes the adaptability, maintenance and optimization schemes to adjust the size of overlay dynamically according to the dynamics of peers; and considering the importance of locality of peers in the system.

Keywords Internet, Communication technologies, Telecommunication network management

Paper type Research paper

1. Introduction

The success of peer-to-peer technology motivates the advance of peer-to-peer multicast (Banerjee *et al.*, 2002; Castro *et al.*, 2002; Lao *et al.*, 2004; Ratnasamy *et al.*, 2001b). In the peer-to-peer multicast system, the participating nodes cooperate with each other in a peer-to-peer manner that there exist some uncertainty factors such as dynamics of peers, diverse locality of peers and heterogeneity of peers' capability. Currently, in the proposed peer-to-peer streaming systems (Hefeeda *et al.*, 2005; Kostic *et al.*, 2003; Liao *et al.*, 2006; Tran *et al.*, 2004; Xiang *et al.*, 2004; Zhang *et al.*, 2005), some peers in an overlay are selected as the streaming suppliers by the topology and the maintenance schemes which would affect the common performance metrics, such as data-stream delivery efficiency and perceived quality of streaming. A proper overlay for peer-to-peer streaming can keep stable suppliers, shorten transmission delays and balance the load of peers. Therefore, how to form an overlay to overcome the uncertainty factors like non-guaranteed communication efficiency, limited upload capacity, dynamic of suppliers and so on is thus becomes a challenging issue.

International Journal of Pervasive Computing and Communications
Vol. 6 No. 1, 2010
pp. 104-124

1742-7371
DOI 10.1108/17427370911033309

Several approaches have been proposed in the literatures to tackle those described difficulties above. In Banerjee *et al.* (2002), Lao *et al.* (2004) and Tran *et al.* (2004), the transmission path length between any source and end peer is bounded by using the overlay tree. But the delivery performance vibrates due to the overhead of overlay maintenance. Scribe (Castro *et al.*, 2002) and CAN (Ratnasamy *et al.*, 2001a) overcome the performance vibration by using the structured peer-to-peer approaches. However, they employ the single-supplier model which cannot guarantee the perceived quality of high-bit rate due to the limitations of upload bandwidth of peers. CollectCast (Hefeeda *et al.*, 2005), Bullet (Kostic *et al.*, 2003), AnySee (Liao *et al.*, 2006) and the multimedia distribution services (Xiang *et al.*, 2004) all refer the mOverlay (Zhang *et al.*, 2004) mechanism to support data streaming by collecting the resources of multiple suppliers in the sense that the perceived quality of the multicast sessions of high-bit rate can be maintained. Due to the optimization schemes adopted CollectCast, AnySee and Xiang *et al.* (2004) can effectively shorten the transmission latency. However, the streaming delivery path of CollectCast and AnySee would fail (or disconnect) frequently. Moreover, the limited upload bandwidth of the intermediate peers in the delivery paths would be the bottlenecks if they help to forward multiple streams. For Xiang *et al.* (2004), structure of the employed mOverlay would result in uneven resource utilization of peers due to the lack of flexibility in terms of neighboring peers. This phenomenon would become more severe when most peers are clustered in small regions of network.

In this paper, we propose a flexible two-layered locality-aware overlay network using the group concepts to construct a peer-to-peer streaming system. By exploiting the surrounding neighbors of peers with low communication delay, the data stream delivery efficiency and perceived quality can be constantly satisfied in our system. In the proposed two-layered overlay, peers are clustered into locality groups based on the communication delays of peers. These locality groups form the top layer of the overlay and are interconnected with each other as a tree rooted by the streaming source. In each locality group, peers form an overlay mesh for transmitting the data stream to other peers of the same group. These overlay meshes form the bottom layer of the constructed overlay. To keep the information of streaming sessions with the correspondingly constructed overlays and help to construct or maintain the overlay, we design an indexing server in our system. In addition, in order to construct the two-layered overlay efficiently, some schemes are proposed to let peers of the system locate themselves into proper groups well. These mechanisms are listed as follows:

(1) *The peer locating scheme*: this scheme is proposed to help the incoming streaming session participants for locating themselves into proper locality groups.

(2) *The membership management scheme*: this scheme is used to help peers with organizing the membership of peers in locality groups.

(3) *The split and merge schemes*: these schemes are designed to let the overlay adjust itself with the dynamics of peers.

(4) *The backup group probing scheme*: this scheme 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 received quality. For example, peers can not only obtain the streaming suppliers easily from other peers which are in the same locality group, but shorten the delivery latency from suppliers of other groups. Since the number of peers in a locality group is upper and lower bounded, the formulation of an overlay mesh in each locality group which helps peers to collect enough bandwidth for which that a streaming session can keep the received quality. In a streaming session, the streaming data are disseminated from a streaming source to each end-host along with the connected locality groups. With the locality properties of locality groups, the communication latency of a delivery link of two peers in the same locality group is short. Since a source-to-end delivery path is composed of the delivery links of peers, the short communication latency of delivery links result in totally short delay. Moreover, if we compare our system with the unbalanced power-law type peer-to-peer overlays (Adamic *et al.*, 2001; Ripeanu *et al.*, 2002), our proposed methods significantly improve the resource utilization of peers.

To evaluate the proposed flexible, locality-aware overlay network, we have implemented our overlay with proposed scheme on the simulator with varied physical topologies, streaming data rates and availabilities of peers. The streaming data delivery latency is measured base on the source-to-end delivery paths and the availabilities of streaming paths which are the performance metrics. We mainly compare our system with the other peer-to-peer (P2P) streaming system – AnySee (Liao *et al.*, 2006), which is one of the most popular peer-to-peer streaming systems currently. The simulation results show that our work can achieve better source-to-end delivery latency when the scale of online participants is exceeded the predefined threshold. The acceptable communication delay among suppliers can retain the received quality of streaming session while AnySee cannot. Besides, the reliability of source-to-end delivery paths is higher than AnySee.

The new contributions of comparing our research works with other existing systems are listed as follows:

(1) We propose a novel peer-to-peer streaming architecture which contains the adaptability, maintenance and optimization schemes to adjust the size of the overlay dynamically according to the dynamics of peers.

(2) We consider the importance of locality of peers in the system. With the designed schemes such as the peer locating scheme, the backup group probing scheme and the predefined threshold of transmission latency, we successfully reduce the source-to-end delivery latency and communication delay. Also, we increase the reliability of source-to-end delivery paths of our system.

The remainder of this paper is organized as follows. In section 2, some related works are given. In section 3, we propose our system model and present it more specifically. In section 4, we propose some experimental scenarios to evaluate our works via simulation. Section 5 gives the discussions of the feasibility of our system on different aspects. Section 6 gives the concluding remarks and the future directions.

2. Related works

Recently, many overlay schemes have been proposed in the literature for efficient peer-to-peer streaming. The goals of these schemes are to assure that the data streaming delivery efficiency and the received quality metrics can be constantly satisfied. They can be classified into tree-based (Banerjee *et al.*, 2002; Castro *et al.*, 2002; Padmanabhan *et al.*, 2002; Ratnasamy *et al.*, 2001b; Tran *et al.*, 2004) and mesh-based (Hefeeda *et al.*, 2005; Jannotti *et al.*, 2000; Liao *et al.*, 2006; Xiang *et al.*, 2004; Zhang *et al.*, 2005) structures.

A large fraction of peer-to-peer multicast systems are based on the tree-based overlays. In Lao *et al.* (2004), based on the conventional client–server/proxy model, the authors proposed a scalable and efficient hybrid overlay multicast architecture in which the multicast trees can be simply constructed for peers. CoopNet (Padmanabhan *et al.*, 2002) is the pioneering peer-to-peer streaming system. A centralized (streaming video source) approach is employed to efficiently maintain a distribution tree, but may lead to the overload of a streaming video source due to the huge service requests. Scribe (Castro *et al.*, 2002) and CAN multicast (Ratnasamy *et al.*, 2001b) are built upon the proposed Pastry (Rowstron and Druschel, 2001) and CAN (Ratnasamy *et al.*, 2001a) systems, respectively. They leverage the dedicated overlays with their native multicast routing schemes. The multicast distribution trees constructed exhibit simplicity and scalability. In Ratnasamy *et al.* (2002), the authors proposed schemes based on the topology-aware of underlying CAN to improve delivery efficiency in CAN multicast. NICE (Banerjee *et al.*, 2002) and Zigzag (Tran *et al.*, 2004) both adopt the hierarchical clustering and split/merge heuristics to minimize the transmission path length. They are sensitive to node dynamics and need to do topology adjustments frequently that may lead to worse streaming quality. With the growing requests of streaming of high bit-rate, the suitability of tree-based overlays would degrade in peer-to-peer environments since they do not take the heterogeneity of upload capacity of peers into account.

The mesh-based overlay is a novel model for peer-to-peer multicast since it takes the heterogeneity of upload capacity of peers into account. Bullet (Kostic *et al.*, 2003) system is a scalable and distributed algorithm used for construct a high bandwidth streaming overlay mesh. In the Bullet system, nodes can self-organize into an overlay tree to transmit the disjoint data sets and retrieve the missed parts simultaneously. Xiang *et al.* (2004) built a framework for multimedia distribution service on top of the mOverlay architecture which is a group-based and locality-aware peer-to-peer overlay. The proposed distributed heuristic replication strategies in Xiang *et al.* (2004) which can leverage the locality groups to efficiently disseminate the media content. The proposed CollectCast system is a multi-supplier streaming service built on top of the peer-to-peer lookup substrate (e.g. Pastry). The specially constructed virtual tree topology and a selection algorithm are used to yield an active streaming sender set from a candidate peer set. The rate redistribution and peer switch mechanisms help to retain the receive-side quality. DONet (Zhang *et al.*, 2005) system is a data-driven overlay network for live media streaming. By employing the gossip protocol, peers can periodically exchange the availabilities of data blocks for retrieving unavailable data and supplying available data. However, the streaming quality of the DONet system cannot be guaranteed. The AnySee system is a peer-to-peer live streaming system built on top of the Gnutella (Gnutella website, 2002) architecture. The location-aware topology matching (LTM) (Liu *et al.*, 2005) scheme and the adaptive connection establishment (Xiao *et al.*, 2005a) scheme were proposed to optimize the connections of neighbor peers so as to tackle the power-law effects (Adamic *et al.*, 2001; Ripeanu *et al.*, 2002) in Gnutella. In the AnySee system, by the usage of the LTM scheme and the proposed inter-overlay optimization scheme, a peer can maintain the efficient and available streaming paths on a mesh-based overlay.

Our system is a hybrid architecture which consists of the tree and the mesh structures. The tree structure is constructed by connecting the peers who maybe the super-peers or the stable peers whose resources are more stable and more sufficient. Besides, peers of the same overlay connect with each other to construct the mesh structure. In one of the meshes, peers connect with each other according to the predefined threshold of transmission latency. If the transmission latency of peers is smaller than the threshold, these peers form a group and connect with each other to construct the mesh by themselves. In addition, the tree-push/mesh-pull hybrid transmission mechanisms significantly improve the system performance in terms of communication latency and system reliability.

3. System model

Based on the concept of super-peer overlay network, we propose a two-layered system architecture shown in Figure 1. In Figure 1, peers are clustered into locality groups with bounded size. The communication delay of peers in a locality group is below the predefined threshold. The top layer of the overlay consists of locality groups which are interconnected as a multicast tree rooted at the streaming source. Each locality group holds a derive level that represents the level of a locality group in the multicast tree. The change of the derive level of a locality group indicates the split or merge of the locality group. If the derive level of a locality group is smaller, a peer who has joined this locality group would experience less relay time for collecting streaming data from the streaming source. In each locality group, peers form an overlay mesh for streaming and these overlay meshes form the bottom layer of this overlay. With the help of the constructed overlay mesh, the streaming data just obtained by certain peers in a locality group can be rapidly distributed to other group members. Thus, the efficiency of streaming delivery can be enhanced for the peers located in diverse network regions.

As a streaming session starts, the corresponding overlay would be constructed by the streaming source according to the specifications of the session (in this paper, a streaming session is regarded as a constant bit rate session). An indexing server is used to keep the information of streaming sessions with the correspondingly constructed overlay. The incoming session participants act as peers to join a proper locality group in 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 collaborate for streaming. To keep sufficient and stable suppliers for streaming, the split/merge schemes for overlay maintenance would 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. The ability to grow or shrink the number of locality groups in an overlay makes the proposed overlay flexible and scalable. For those peers that cannot satisfy the performance metrics, the backup peer probing scheme is used to enhance the performance of the constructed peer-to-peer streaming system. In the following sections, we will sequentially describe the locality

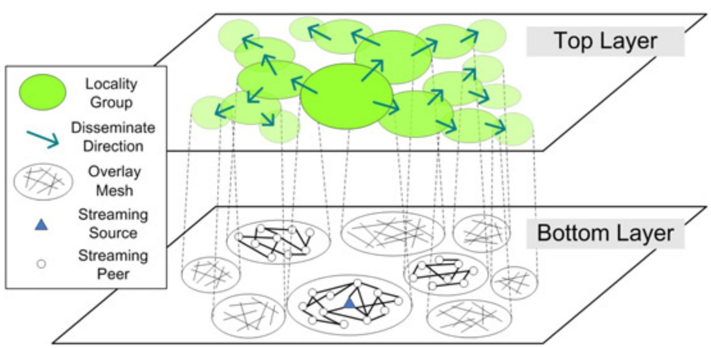

Figure 1.
System architecture

group, the indexing server, the peer locating scheme, the membership management scheme, the spit/merge schemes and the backup peer probing scheme in details.

3.1 The locality group

A locality group consists of a set of peers. In a locality group, peers are classified into two disjoint subsets, candidate and separate subsets. For peers in the candidate subset, the network delays among them are less than or equal to the predefined value according to the rate of a streaming session. In this paper, the predefined value is set between $l/2$ and l based on these research works (Xiang *et al.*, 2004; Zhang *et al.*, 2004), where l is the tolerable delivery latency of the streaming rate of a session. The network delays of peers between the candidate subset and the separate subset are greater than the predefined value. We mainly refer these research works (Banerjee *et al.*, 2002; Tran *et al.*, 2004) to set the size (number of peers) of a locality group which is bounded by $[k, (3k - 1)]$, except for the locality group where the streaming source belongs is bounded by $[1, (3k - 1)]$, where $k \geq 1$. If the size of a locality group is equal to $3k - 1$, we say that the locality group is full. When a peer joins a full locality group, it will cause the locality group spilt into two smaller locality groups. If the size of a locality group except for the streaming source belongs is less than k due to some peers leave, the locality group will be merged with other groups such that the size of the merged locality group is larger than $3k$. If no such locality group is available, the merge processes will not performed until such a locality group is available or when the size of the locality group is greater than or equal to k again.

In our overlay, each peer will join one locality group which is called default group for streaming initially. Certain peers may act as the gateway-like peers by joining another locality groups which are called source group to help for relaying among groups. They collect streaming data from the source group and disseminate these data to members who are in the default group. The value of derive level of the source group is the value of 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 when certain degree of availability is satisfied (e.g. data cache or stable supplies). A contributor is called a maintainer in a locality group if it is responsible for overlay maintenance and membership management of peers and is also a member in another locality group (source group). 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 the published sessions and the corresponding overlays which can be considered as the metadata of the streaming data. End users can obtain a list of metadata of favored sessions from the indexing server for participation. Four operations such as query, add, update and remove operations are provided to access the indexing server for overlay construction and maintenance. The metadata format stored in the indexing server is shown in Table I.

Table I. The metadata format for each published streaming session

SSPR		LGR		
Session ID (unique)	Rate (bps)	Group ID (unique)	Derive level	Maintainer (IP/ports)

In Table I, the metadata of a published session contains one streaming session property record (SSPR) and multiple locality group records (LGR). The SSPR represents the specification of an established streaming session. It consists of two fields, session ID and rate. The session ID (unique) field is used to recognize each streaming session recorded in the indexing server. The rate field is used to specify the (constant) streaming data rate of this session. Relying on the value of rate, our overlay can be properly constructed to achieve perceived quality. The LGR stores the information of locality groups in the corresponding overlay of a streaming session. It consists of three fields which are group ID, derive level and maintainer, respectively.

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 of the corresponding two-layered overlay. 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. Moreover, the indexing server sets the value of derive level field 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 as a free-rider 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, p_i gets one entry from the indexing server and inserts this entry with measured network delay of p_i and maintainer in the entry to its group cache. The entries of LGR stored in the group cache are in increasing order according to the measured network delay and the derive level of a locality group.

Step 2. For the first m entries in the group cache of p_i, the maintainer in each entry sends all entries in its group cache 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 and maintainers in the entries to its group cache. This step is performed n times, where n is the system defined 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. S_2 is the exclusion of S_1 in the group cache. If there is an LGR entry whose derive level is the smallest one among those who are not full locality groups in S_1, 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 the dynamic landmark for positioning in our overlay. Some works related to the landmark issues in peer-to-peer system are described in Francis *et al.* (2001), Hotz (1994) and Ng and Zhang (2002). 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. With this scheme, a peer has the higher chance to be located to the candidate subset of a locality group with shorter delivery latency. If some peers cannot be located to a candidate subset of a locality group, this scheme accommodates them into a proper locality group to reduce the frequent corresponding adjustments in the overlay. By this way, our overlay can be constructed with the balance between the stability of topology and the delivery efficiency of peers.

3.4 The membership management scheme

The membership management scheme is used to organize the peers in a locality group. By this scheme, the peers in a locality group can collaborate for streaming data. Based on the super-peer network, the maintainer of a locality group acts as a super-peer to handle the join and leave operations of peers, monitor the status of peers, manage contributors and broadcast the information of contributors and the LGR entry of joined source group.

In our system, a member cache is used to store the information of members in a locality group. For each joined locality group, a peer maintains the corresponding member cache. The information stored in the member cache consists of four fields which are 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 subsets (candidate or separate) of members. During the join and leave procedures of peers, the information of the members is recorded in the member cache of the maintainer. 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 maintainer does not receive the keep-alive messages from a member in a period of time, it will drop the information of that member. If a peer is available to be a contributor, it informs the maintainer of the default group. To manage a contributor in the member cache, the maintainer will set the type field of the corresponding entry as the contributor with a contributor rank. The contributor rank is a unique stamp (e.g. the global timestamp when receiving informing messages) generated by the maintainer. When a contributor lacks of the streaming data in its data cache, it will inform the maintainer. The maintainer will set the contributor as a free-rider by setting the type field of the corresponding entry in the member cache. 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 the failure of the maintainer.

With the proposed membership management, the members of a locality group would exploit their member caches to hold the contributors located in the candidate subsets prior to streaming. A set of contributors act as the streaming suppliers are called active contributors. Corresponding to the rate of a session, a peer obtains the streaming data from the data caches of active contributors based on the collected bandwidth. If the status of an active contributor is changed (e.g. leave, lower bandwidth and so on), a peer would seek one non-active contributor in its member cache to replace this active contributor. In a locality group, each contributor is a potential maintainer. When the members realize the absence of the maintainer, each contributor checks its rank in the member cache. The contributor with the lowest rank would become the new maintainer and join to the source group of the absent maintainer according to the received LGR entry. If the maintainer of the source group also fails, the new maintainer will use the source group ID to query the indexing server for the corresponding LGR entries. If such corresponding LGR entries cannot be found, the new maintainer will employ the peer locating scheme to search a new source group to join.

3.5 The overlay maintenance schemes

To keep sufficient and stable suppliers for streaming data and ensure the moderate loading of a maintainer, the split and the 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 our overlay, 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 split scheme would be triggered to split this locality group into two smaller locality groups. The following is the procedure of the split scheme.

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

Step 2. Then, 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 acknowledging that the maintainer m_i is the new group ID g_j where the group ID field, the derive level field and the maintainer field of the LGR entry are set to g_j, the derive level of $m_i + 1$ and c_j, respectively.

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

(1) The non-active contributors in separate subset ranked from high to low.

(2) The active contributors in separate subset ranked from high to low.

(3) The free-riders in separate subset.

(4) The non-active contributors in candidate subset ranked from low to high.

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 its 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 stays in the original locality group if it is not in the split list and does not gather the major part of streaming bandwidth from the contributors in the split list and c_j. A member migrates from the original locality group to the new locality group if it is not a contributor in the split list or it gathers the major part of streaming bandwidth from the contributors in the split list and c_j. A member joins the original locality group and the new locality group to relay data streams if it is a contributor in the split list and does not gather the major part of streaming bandwidth from the contributors in the split list and c_j. When a member joins locality groups, g_i and g_j will be the source group and the default group of this member, respectively. 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 of the default group of the maintainer should be modified correspondingly. The maintainer would update the field of derive level of the corresponding LGR entry in the indexing server and inform the changes to its members. If some of its members are maintainers, they would change their derive level as well. By iterative updating and informing processes, the changes of derive levels on the related locality groups would be reflected.

In the split scheme, the criteria listed in step 3 help to split out the members of separate set in a locality group. The delivery performance in a locality group can thus be retained or even improved after the split scheme has been performed. Based on step 5, the major part of the active contributors of a member can be kept, thus eliminates some overheads by the split scheme.

3.5.2 The merge scheme. To maintain sufficient and available 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. There are three cases which may happen:

Case 1. If the size of the group g_s is less than $3k$ after performing the merge process with the group g_i, all members in g_i would join the group g and the maintainer m_i of g_i would act as the contributor in g_s. The corresponding LGR entry of g_i would be removed from the indexing server. For those peers that are free-riders in g_i and maintainers in their default locality group, they need to change the value of derive level to the value of derive level of g_s plus 1.

Case 2. If case 1 is not satisfied, the maintainer m_i will probe the LGR entries in S_1 of its group cache to find a locality group g_j whose derive level is lower than or equal to the derive level of g_j and its size is less than $3k$ after the merge with g_i. If such a group g_j is found, all members in g_i would join g_j and the maintainer m_i would act as a contributor in g_j. 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 and maintainers in their default locality group, they need to change their derive levels to the derive level of g_j plus 1.

Case 3. If such a g_j is not available, the maintainer m_i will probe the LGR entries S_1 in its group cache to find a locality group g_k whose size is less than $3k$ after the merge with g_i. If such a group g_k is found, all the members in g_k would join g_i and the maintainer m_k of g_k, would act as a contributor in g_i. The corresponding LGR entry of g_k would be removed from the indexing server by m_k. For those peers that are free-riders in g_k and maintainers in their default locality group, they need to change their derive levels to the derive level of g_i plus 1.

If none of cases specified above is satisfied, the merge process will repeated from cases 1 to 3 until one of the cases is satisfied or the size of g_i is greater than or equal to k.

3.6 The backup group probing scheme

When a peer is in the separate subset of a locality group, the received streaming quality of this peer cannot be constantly satisfied when it gathers the streaming bandwidths. Besides, the streaming delivery performance would also be affected when this peer acts as a contributor. To tackle those negative effects by adjusting the locality groups of peers, the backup group probing scheme is proposed to optimize our overlay based on the size of the locality group. With this scheme, a peer in the separate subset of a locality group can be moved to the candidate subset of another locality group. The following steps are the procedure of the backup group probing 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. If the number of members in the separate subset is less than k, the maintainer would choose other members from the candidate subset based on the same time order to make the total members selected to k.

Step 2. If a member p_a is 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 $3k - 1$. We have the following two cases:

(1) If p_a only joins g_i and such a group g_j is found, p_a will relocate itself from the group g_i to the group g_j.

(2) If p_a acts as the maintainer of its default group and the derive level of g_j which is the source group is less than or equal to g_i, p_a will relocate itself from the group g_i to the group g_j. Moreover, p_a will change its derive level accordingly and broadcast the change information to let its members.

Step 3. If a member selected p_a 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 $3k - 1$. We have the following three cases:

(1) If p_a only joins the group g_i and such a group g_j is found, p_a will relocate itself from the group g_i to the group g_j.

(2) If p_a acts as the maintainer in its default group and the derive level of g_j is less than or equal to g_i, p_a will relocate itself from g_i to g_j. The peer p_a will change its derive level accordingly and broadcast the change information to let its members.

(3) If p_a acts as a contributor in g_i and joins a source group, and such a group g_j is found, p_a will relocate itself from the group g_i to the group g_j.

By the effort of case 2 in steps 2 and 3, the source-to-end streaming delivery paths can be shortened by the optimization to the suppliers of contributors. The streaming delivery efficiency on each peer can thus be improved gradually. The size checking of locality groups in step 1 stabilizes the topology of our overlay. Without the checking, the flash crowd problem may encounters by the huge migration of peers among locality groups that results in the frequent splits and merges.

4. Simulation

In this section, we evaluate the proposed work by several series of simulation taken on our system and the AnySee system. Based on the different aspects, we take the measurements on the peers in both systems to compare their performance by analyzing the behavior of the corresponding overlay. In Section 4.1 we present the simulation environment. The simulation results are shown in Section 4.2.

4.1 The settings of simulation environment

In our simulation, we generate two types of topologies. They are physical and logical topologies, respectively. 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 formulate the peer-to-peer overlay upon the physical topology. BRITE (Medina *et al.*, 2001) and Inet (Winick and Jamin, 2002) are the topology generation tools based on AS and router model. We adopt the hierarchical top-down model with GLP model (Bu and Towsley, 2002) on AS/router layer in BRITE and the pure router model in the Inet generator generates the graphs of physical topology of 5,000 nodes which vary to yield different average network delays. The detailed parameters we applied are shown in Tables II and III.

Table II shows the system parameters of simulation environment. The simulation environment is constructed by two generators which are BRITE and Inet-3.0 generator, respectively. There are totally 5,000 nodes adopted in the simulation. Table III shows the settings of the topology generators adopted in simulation. They include some system parameters running on the generated topologies. The values of the BRITE specific parameters are also shown in Table III. In the generated physical topologies, a set of stub (single degree) nodes represent the routers on the internet. They are directly attached by the end hosts. In both models we simulated, a number of end hosts are attached to randomly selected stub nodes via their access links with identical 2 Mbps bandwidth and 1 ms delay.

For comparing between our works with the AnySee system, we have implemented both their protocols on the same simulator. The protocols employ the TCP connections for the communications of end hosts. The communication latency (round-trip time (RTT)) is calculated by TCP-related formula based on the shortest path of end hosts in the physical topology.

We simulated our system by running an experimental application framework on each end host depicted in Figure 2. In the application framework, the implemented protocol formulates the two-layered overlay. This overlay is employed as an application substrate to obtain the streaming suppliers and streams to other peers in a session. By the government of the Status Monitoring and Adaptation Scheme, the assign/adjust rate trial component yields the active contributors from the contributors in the joined locality group. Based on this framework, the behavior of our overlay can be observed.

In this simulation environment of the AnySee system, we have constructed the underlying mesh-based overlay with a varied size. A single session is initialized in AnySee by a streaming source and a set of other peers which maintain their streaming paths. We

Topology #	Generator	Average link delay of stub nodes	Total node numbers	Size of node plane	Node placement
1	BRITE	20.8998 ms	5,000 (50 ASes, 100 routers in each AS)	HS: 1,500 LS: 100	Heavy tailed
2		33.246 ms			Random
3	Inet-3.0	29.1676 ms	5,000	5,000	N/A
4		58.3299 ms		10,000	

Table II. The system parameters of simulation environment

Topology generator	BRITE (hierarchical (AS/router) top-down, GLP)		Inet-3.0		
Link bandwidth distribution (heavy tailed)	AS: 10-1 Gbps, router: 1 Gbps-100 Mbps		10 Gbps-100 Mbps		
BRITE specific parameters					
Growth type	Preferential connection	Edge connection model	P	β	m
Incremental	ON	Smallest degree of non-leaf model	0.4695	0.6447	1

Table III. The settings of the topology generators adopted in simulation

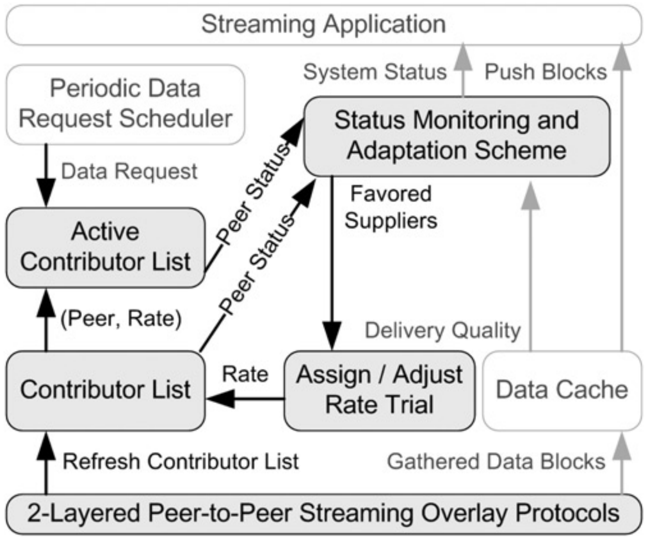

Figure 2.
Proposed experimental architecture

observe the varieties of those paths to evaluate the efficiency of the system. In all simulations, the first joining peer in the overlay acts as the streaming source and never fails.

On the implementation of protocols, the rank of contributors is decided by the receiving time (global in simulator) of claim messages. The communication of peers is handled by two TCP connections for messaging/streaming data. The transmission unit of data stream is the data block with fixed size based on the streaming session rate. A message of single-block size which contains the sending time as the stamp would be replied by the maintainer for network delay measurement. The delay is the difference of the stamp value and the receiving time. The predefined value l is set to be the tolerable delay. The default parameters used in our simulation are listed in Tables IV, and V.

4.2 Performance evaluation

We evaluate the system performance based on two major metrics. The first one is the average maximum delivery latency of a data block from streaming source to each participant. Following the streaming delivery path, we calculate the RTT value of sending a data block by pairs of sender and receiver and take the summation of those RTT values as the source-to-end delivery delay. The second one is the average communication delays among the participant and its upstream peers. With regard to this metric, we adopt the physical link delay of end hosts as the metric. A simulation process begins when the first peer has appeared and ended at 180 s after the last peer had appeared. The measurements are taken when every ten peers sequentially appear in the system. As the system time elapsed, the changes of peering hosts lead varying

Table IV.
The simulation parameter settings of our system and AnySee system

Our proposed system		The AnySee system	
The seek number of active contributors	5	Number of active streaming paths	5
The parameter k	17	Min. number of neighbors of a peer	5
The probe number	3	The parameter $\mu_D(S)$	25 (s)
The group probing threshold	20	Number of participants of a session	First 500 peers

delivery paths and delays. We track each following simulation series based on the size of overlay to observe the performance variations during the whole simulation period.

4.2.1 The comparison results under different physical topologies. In this section, we compare our proposed overlay with AnySee based on four different topologies. Figures 3 and 4 illustrate the average source-to-end delays and communication delays under the situation where the different amount of peers had joined to a session. Since the locality groups in our overlay would interconnect together as a multicast tree.

From Figure 3, we can see that when the relay hops/groups increase by more peers participant, the source-to-end delivery latency will increase. On the contrary, a peer in AnySee must actively and individually probe the available streaming paths. Relying on the optimization of neighboring peers in terms of communication latency, participants in AnySee can find streaming paths with similar delivery latencies. We can observe that as the average delay of nodes increases, our system scales better. This is because that a peer cannot easily locate itself such that a locality group has more members in the candidate subset and tends to cover the broader network regions.

Figure 4 shows that our proposed system works better than Anysee system under all predefined topologies. From this figure, we can see that the short link delays with active contributors induce better streaming quality. In addition, we notice that the schemes of AnySee which are used to probe the shortest paths among the neighboring peers will cause frequent connections and disconnections by the dynamics of peers. Thus, the schemes would result in lower stability of probed streaming paths shown in Figure 5.

Default common parameter settings	
Total numbers of joining peers	2,000
Time interval of peers join	Exponential distribution
Streaming rate/data block size	1.6 Mbps/40 kB (range: 1.2M/30-2M/50)
Upload bandwidth of contributor	Uniform distribution (1-1.5 times of session rate)
Default physical topology	Generated topology #2

Table V. The common parameter settings of our system and AnySee system

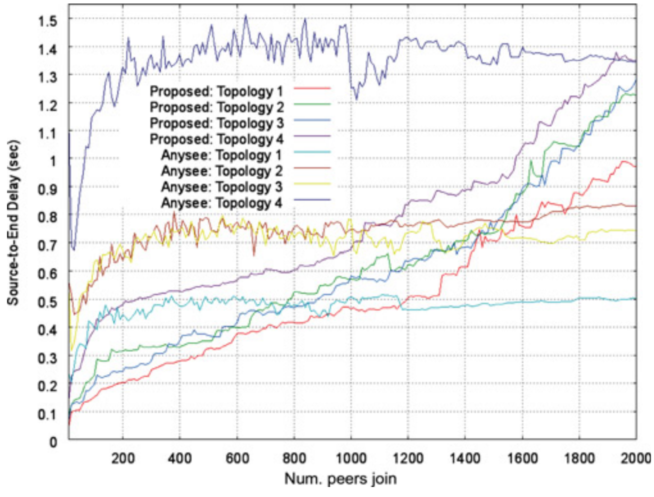

Figure 3. Source-to-end delay under different topologies

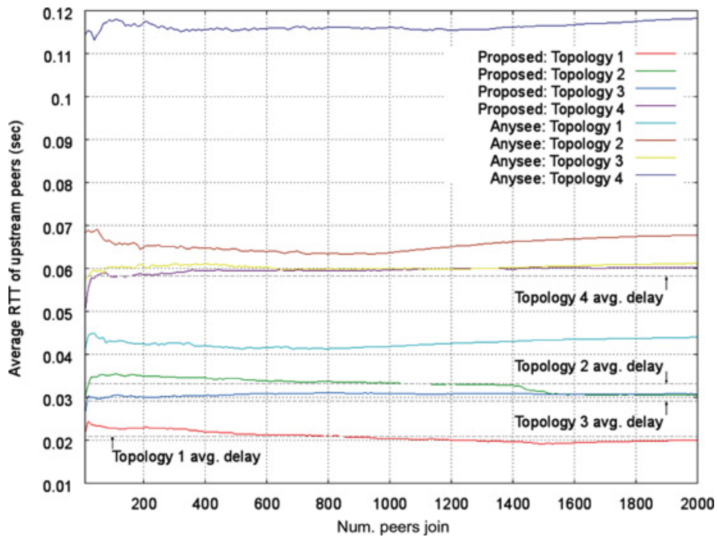

Figure 4.
Communication delay under different topologies

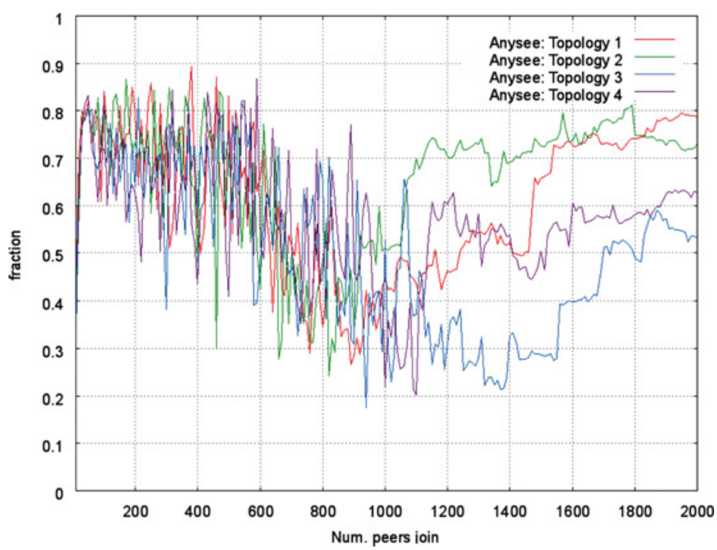

Figure 5.
The impact of the available streaming paths of AnySee under different topologies

4.2.2 The comparison results under different streaming data rates. In this section, we try to verify the effectiveness between our system with AnySee system under different streaming rates. Figure 6 shows that our works perform better than AnySee with larger rates. The result of Figure 7 specifies the similar behavior of performance of Figure 4. When the rate of a session becomes higher, peers would interconnect to other peers in a broader network region because suitable locality groups for each peer do not appear. Otherwise, the communication delay decreases because of the relocations of peers to suitable locality groups.

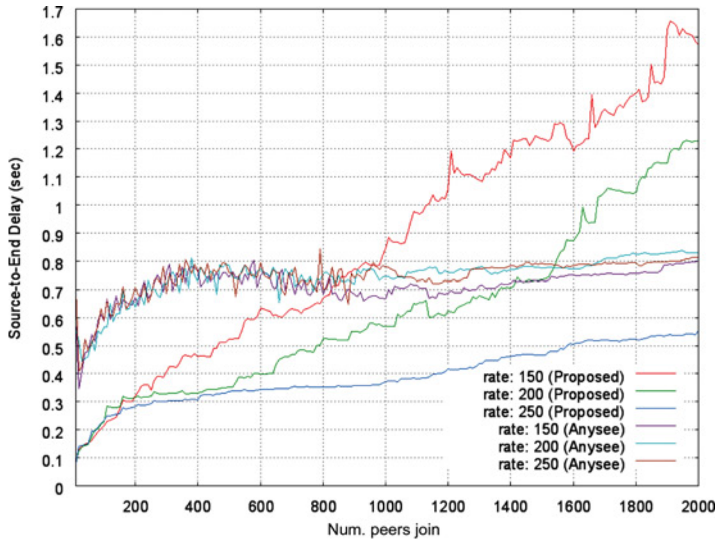

Figure 6.
Source-to-end delay under different streaming data rates

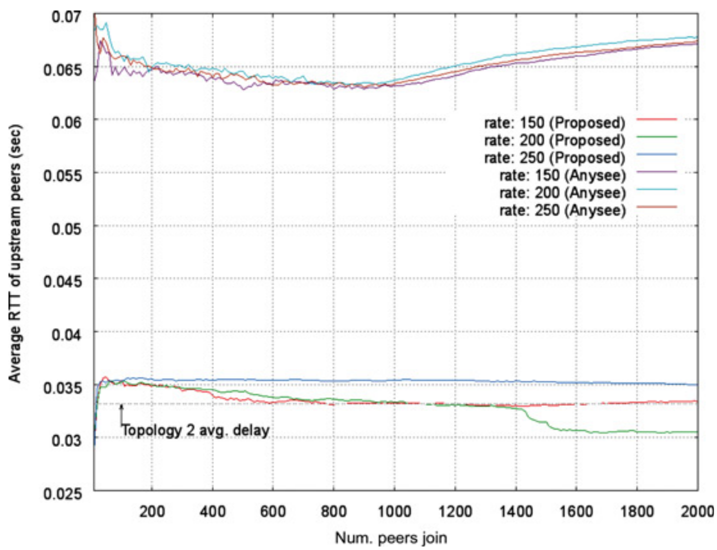

Figure 7.
Communication delay under different streaming data rates

4.2.3 The comparison results under the situation of failure of peers. In this section, we investigate the behavior of our system and AnySee system considering the situation of peers' failure. The steps for simulation are described as follows.

Step 1. Each participant as a peer keeps an assigned availability (a generated random number in (0, 1)).

Step 2. We schedule the failure trials every 7 s throughout a stream session. Upon each trial, a peer in an overlay is selected randomly and it is failed according to its availability.

Step 3. 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.

From Figure 8, we can see that as the population is less than 1,000 in our system, the source-to-end delivery delays decrease as the mean availability of peers' decreases. This phenomenon reflects the flexibility of our system which can adjust the topology to shorten the delivery latency while AnySee system cannot. This is because that it constantly keeps a degree of delay about delivery shown in Figure 9. When the number of peers exceeds 1,000, we cannot find a regular trend. The AnySee system suffers from the similar problem shown in Figure 5 with the dynamics of peers in an overlay.

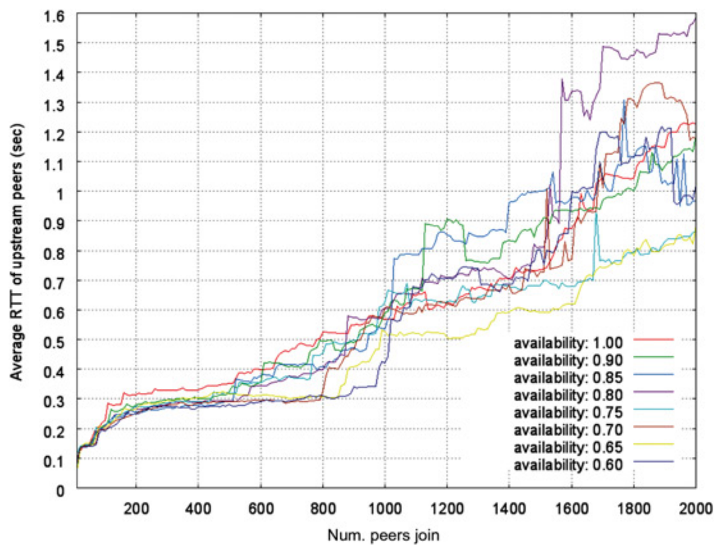

Figure 8.
Source-to-end delay of our system under peers' failure situation

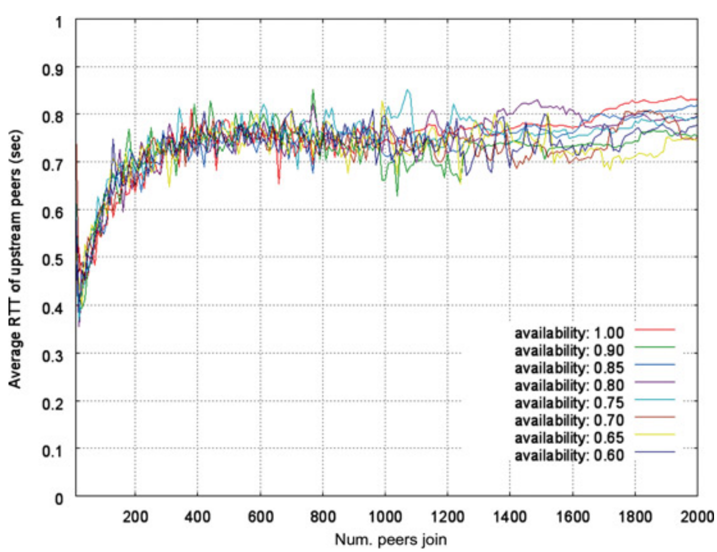

Figure 9.
Source-to-end delay of AnySee under peers' failure situation

Figure 10 illustrates the stability of the retained streaming paths by each participant which becomes worse as the mean availability of peers decreases.

5. Discussion

In the above sections, the simulation results bring us some issues related to peer-to-peer streaming which roughly demonstrates the effectiveness of our system. In this section, we will give some discussions about the feasibility of our system on different aspects according to these described issues. The first one issue is the rate of streaming session. When the session's rate increases, the load of upload bandwidth of peers would become heavier. Since the limited capacity of bandwidth and policy on each end host, relaying the streaming data becomes problematic. Thus, a dedicated multicast architecture for peer-to-peer streaming is considered to be more viable. Considering the high rate session, the provision of streaming data with constant acceptable performance may be difficult due to the limited upload bandwidth of end hosts.

Comparing with the tree-based multicast architecture, a participant can gather resources from multiple suppliers to easily achieve the acceptable performance in our system. On the other hand, when the rate of streaming session is relatively low, the amount of forwarded data streams would be minor to end users so as to release those limitations specified above. On this condition, the benefit of this paradigm can be exploited to shorten the streaming delivery paths by the escaped factor of overlay size as the simulation results show.

A benefit of typical multicast architecture can be demonstrated that the participants only concern about their directly connected suppliers, not the delivery paths from the multicast source. The peer-to-peer streaming system applies this kind of architecture to keep the consistency of streaming delivery paths. Since each peer is responsible for transmitting the data streams for its suppliers and customers, this will produce long latency due to the predecessors on those paths. Although some optimization schemes of these systems such as AnySee, CollectCast or the properties of these structured overlay networks effectively decrease the delivery latency and relay hops from the

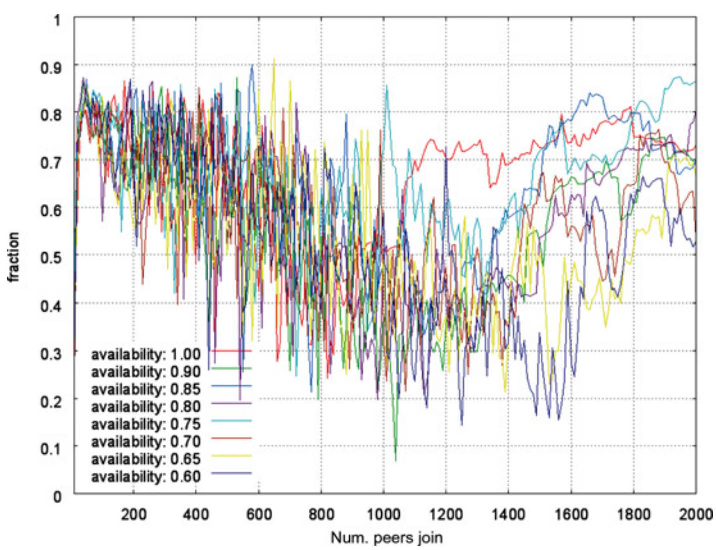

Figure 10.
The impact of the available streaming paths of AnySee under peers' failure situation

streaming source, they have frequent recovery times of failure paths due to the dynamics of peers. The situation is shown in Figure 10. From Figure 10, we can see that the dynamics of peers cause less stability of the streaming paths.

Finally, we assume that if the source-to-end delivery delay is an important factor to evaluate the efficiency of a streaming system. Our system exhibits the flexibility in a small-scale system by the compact streaming delivery paths. Comparing with the other systems which rely on the intermediate peers for communicating, our works scale better in terms of the numbers of relay hops.

5.1 Conclusions and future works

In this paper, we have presented a peer-to-peer streaming system based on a flexible two-layered and locality-aware overlay network. By exploiting the surrounding neighbors of peers with low communication delays, our overlay is constructed to match the underlying network topology. Based on the group concept, the resources of peers in our overlay can be evenly utilized in their locality groups. An indexing server of our system records the metadata of streaming sessions with the correspondingly constructed overlays and helps to construct and maintain the overlay structure. In our system, a peer can simply establish a streaming session and become the streaming source without the help of dedicated streaming servers. Based on the properties of flexibility and locality awareness of our system, the session participants as peers would benefit from sufficient, stable and efficient suppliers in the joined locality groups for streaming. Comparing with AnySee, the simulation results showed that our system had successfully reduced the source-to-end delivery latency and lowered the communication latency between peers with their streaming suppliers. Furthermore, our system also had higher reliability of the streaming paths. Besides, the simulation results have demonstrated the scalability, efficiency and stability of our system, in which the data stream delivery efficiency and the perceived quality can be constantly satisfied.

With respect to the future works, we will investigate and apply more efficient and reliable transmission schemes such as network coding and layered coding to achieve high throughput, scalable and robust peer-to-peer streaming environments. Besides, the incentive mechanisms will be investigated to construct more reliable, fair and robust resource exchange methods among peers by considering the heterogeneity of peers in terms of their upload and download capacities.

References

Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A. (2001), "Search in power-law networks", *Physical Review E*, Vol. 64 No. 4, article no. 046135, pp. 1-8.

Banerjee, S., Bhattacharjee, B. and Kommareddy, C. (2002), "Scalable application layer multicast", *Proceedings of the ACM SIGCOMM'02, Pittsburgh, PA, August*.

Bu, T. and Towsley, D. (2002), "On distinguishing between internet power law topology generators", *Proceedings of the INFOCOM'02, New York, NY, June*.

Castro, M., Druschel, P., Kermarrec, A.-M. and Rowstron, A.I.T. (2002), "Scribe: a large-scale and decentralized application-level multicast infrastructure", *IEEE Journal on Selected Areas in Communication*, Vol. 20 No. 8, pp. 1489-99.

Francis, P., Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y. and Zhang, L. (2001), "IDMaps: a global internet host distance estimation service", *IEEE/ACM Transactions on Networking*, Vol. 9 No. 5, pp. 525-40.

Gnutella website, (2002), available at: www.gnutella.com

Hefeeda, M., Habib, A., Xu, D., Bhargava, B. and Botev, B. (2005), "CollectCast: a peer-to-peer service for media streaming", *ACM/Springer Multimedia Systems Journal*, Vol. 11 No. 1, pp. 68-81.

Hotz, S.M. (1994), "Routing information organization to support scalable interdomain routing with heterogeneous path requirements", PhD paper (draft), University of Southern California, Los Angeles, CA.

Jannotti, J., Gifford, D., Johnson, K., Kaashoek, M. and O'Toole, J. (2000), "Overcast: reliable multicasting with an overlay network", *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation*, *San Diego, CA*.

Kostic, D., Rodriguez, A., Albrecht, J. and Vahdat, A. (2003), "Bullet: high bandwidth data dissemination using an overlay mesh", *Proceedings of the ACM SOSP'03*, *New York, NY*.

Lao, L., Cui, J.-H. and Gerla, M. (2004), "A scalable hybrid overlay multicast architecture for large-scale applications", Technical Report, Computer Science Department, University of California, Los Angeles, CA.

Liao, X., Jin, H., Liu, Y., Ni, L.M. and Deng, D. (2006), "AnySee: peer-to-peer live streaming", *Proceedings of the INFOCOM'06*, *Barcelona, Catalunya*, *April*, pp. 163-74.

Liu, Y., Xiao, L., Liu, X., Ni, L.M. and Zhang, X. (2005), "Location awareness in unstructured peer-to-peer systems", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16 No. 2.

Medina, A., Lakhina, A., Matta, I. and Byers, J. (2001), "BRITE: an approach to universal topology generation", *Proceedings of the MASCOTS 2001*, *Cincinnati, OH*, *August*.

Ng, T.S.E. and Zhang, H. (2002), "Predicting internet network distance with coordinate-based approaches", *Proceedings of the INFOCOM'02*, *New York, NY*, *June*.

Padmanabhan, V.N., Wang, H.J., Chou, P.A. and Sripanidkulchai, K. (2002), "Distributing streaming media content using cooperative networking", *Proceedings of the NOSSDAV'02*, *Miami, FL*, *May*.

Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S. (2001a), "A scalable content addressable network", *Proceedings of the ACM SIGCOMM'01*, *San Diego, CA*, *August*.

Ratnasamy, S., Handley, M., Karp, R. and Shenker, S. (2001b), "Application-level multicast using content-addressable networks", *Proceedings of the NGC*, *London*, *November*.

Ratnasamy, S., Handley, M., Karp, R. and Shenker, S. (2002), "Topologically-aware overlay construction and server selection", *Proceedings of the INFOCOM'02*, *New York, NY*, June.

Ripeanu, M., Iamnitchi, A. and Foster, I. (2002), "Mapping the Gnutella network", *IEEE Internet Computing*, Vol. 6 No. 1, pp. 50-7.

Rowstron, A. and Druschel, P. (2001), "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems", *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, *Heidelberg*, *November*.

Tran, D.A., Hua, K.A. and Do, T.T. (2004), "A peer-to-peer architecture for media streaming", *IEEE Journal on Selected Areas in Communication*, Vol. 22 No. 1, pp. 121-33.

Winick, J. and Jamin, S. (2002), "Inet-3.0: internet topology generator", *Technical Report, CSE-TR-456-02*, Department of EECS, University of Michigan, Ann Arbor, MI.

Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y.-Q. (2004), "Peer-to-peer based multimedia distribution service", *IEEE Transactions on Multimedia*, Vol. 6 No. 2, pp. 343-55.

Xiao, L., Liu, Y. and Ni, L.M. (2005a), "Improving unstructured peer-to-peer systems by adaptive connection establishment", *IEEE Transactions on Computers*, Vol. 54 No. 9, pp. 1091-103.

Zhang, X., Liu, J.-C., Li, B. and Yum, T.-S.P. (2005), "Coolstreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming", *Proceedings of the INFOCOM'05*, *Miami, FL, March*.

Zhang, X.Y., Zhang, Q., Zhang, Z., Song, G. and Zhu, W. (2004.), "A construction of locality-aware overlay network: mOverlay and its performance", *IEEE Journal on Selected Areas in Communication*, Vol. 22 No. 1, pp. 18-28.

Further reading

Banerjee, S., Lee, S., Bhattacharjee, B. and Srinivasan, A. (2003), "Resilient multicast using overlays", *Proceedings of the ACM SIGMETRICS'02, Marina Del Rey, CA, June*.

Chu, Y.-H., Rao, S.G. and Zhang, H. (2000), "A case for end system multicast", *Proceedings of the ACM SIGMETRICS'00, Santa Clara, CA*.

Liebeherr, J., Nahas, M. and Si, W. (2002), "Application-layer multicasting with Delanury triangulation overlays", *IEEE Journal on Selected Areas in Communication*, Vol. 20 No. 8, pp. 1472-88.

Rejaie, R., Yu, H., Handley, M. and Estrin, D. (2000), "Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet", *Proceedings of the INFOCOM 2000, Tei Aviv, March*.

Sen, S., Rexford, J. and Towsley, D. (1999), "Proxy prefix caching for multimedia streams", *Proceedings of the INFOCOM'99, New York, NY, March*.

Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S. and Willinger, W. (2000), "Network topology generators: degree-based vs. structural", *Proceedings of the ACM SIGCOMM*, 2002, *Pittsburgh, PA, August*.

Xiao, L., Zhuang, Z. and Liu, Y. (2005b), "Dynamic layer management in superpeer architectures", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16 No. 10, pp. 1078-91.

Zegura, E.W., Ammar, M.H., Fei, Z. and Bhattacharjee, S. (2000), "Application-layer anycasting: a server selection architecture and use in a replicated Web service", *IEEE/ACM Transactions on Networking*, Vol. 8 No. 4, pp. 455-66.

Corresponding author

Yeh-Ching Chung can be contacted at: ychung@cs.nthu.edu.tw