International Journal of u- and e- Service, Science and Technology
Vol.8, No.2 (2015), pp.171-180
http://dx.doi.org/10.14257/ijunesst.2015.8.2.16

A Social Network based Bandwidth Sharing Model for P2P Streaming Service

Li Songjiang, Liu Yanheng and Wang Aimin

College of Computer Science and Technology, Jilin University, Changchun 130012

lsj84@outlook.com, lyh_lb_lk@yahoo.com.cn, wangam@jlu.edu.cn

Abstract

The combination of social network and P2P streaming service created a new kind of peers, these peers only share their bandwidth to social-connected peers and they do not need to watch the streaming video. Because of the particularity of such peers, ordinary P2P modelsdo not apply to this type of peers. Therefore, we propose a social network-based bandwidth sharing model. By creating download and upload neighbor list, our model separates the transmission of data blocks. Peer requests data blocks from peers in download neighbor list by the data block missing degree of peers in upload neighbor list. Meanwhile we redefine the buffer of social-connected peers, make the social-connected peers better collaborate with the sharing peers. Therefore the social-connected peers will not receive duplicate data blocks. Experiment data shows our sharing model could maximize the sharing ratio of sharing peers, and improve the playback quality of social-connected peers. When the number of sharing peers reaches a certain proportion, the playback delay of P2P streaming service with sharing peers is better than the playback delay of normal P2P streaming service.

Keyword: *P2P streaming, social network, bandwidth share, social-connected peer*

1. Introduction

In recent years, P2P streaming service has been greatly developed [1]. With the increment of network bandwidth and the development of video coding technology, P2P streaming services are able to provide video content at a bit rate over the 500 Kbps for hundreds of thousands of users. However, current streaming media technology has a lot of unsolved problems. P2Pstreaming service has the problem of long video start-up delays and playback between peers can't be synchronized. Paper [2] shows the start-up delay of existing p2p streaming systems is from 10s to 120s, and the maximum difference of playback between peers can be 140s. These problems lead to extremely bad user experience of Streaming system.

P2P streaming service quality depends on the ratio of upload bandwidth and downloads bandwidthin the whole P2P network. The larger the ratio, the better the P2P streaming service quality. Meanwhile the stabilization of peers affects the P2P streaming service quality [3]. But the bandwidth ratio is based on the peers' bandwidth of the whole P2P network, it is hardly to change. And the stabilization of peers cannot be controlled.If there are extra peers which are stable and have willing to share bandwidth, the quality of P2P streamingwill be improved. Social network peers are just this kind of peers.

Some studies [4, 5] have proposed adopting the concept of social network in P2P network. Paper [6] designed the architecture of online social network combined with P2P file-sharing. Based on the trace data of Facebook and BitTorrent, paper [6] proposed P2P system based on social networks, namely Social-P2P. Social-P2P groups common-multi-interest peers into a cluster and further connects socially close nodes within a cluster. Sharing files among socially close friends achieves highly efficient and

ISSN: 2005-4246 IJUNESST

trustworthy file sharing. Paper [7] proposed an effective resource allocation algorithm based on social relation for video streaming services over P2P network, it can provide an effective trade-off between the total trust value for future usage and the current video streaming quality. Paper [8] proposed some modifications to the neighborhood creation and chunk scheduling algorithm of a mesh-based P2P overlay, in order to favor peers belonging to a social network and to grant them better performance.

In [9] Hales and Arteconi and by Lin, *et al*., in [10] encourage the contribution of low-level resources using incentives generated at higher (social) layer to provide more cooperation among peers. The same approach is followed by Antoniadis and La Grand in [11]. TRIBLER, the social-basedP2P system proposed by Pouwelse, *et al*., in [12], uses the OSN relationships as the base layer of a P2P system, not only for content discovery or recommendation, but also to improve download performance. The improvement is achieved thanks to the cooperative downloading implemented by the users that join the same OSN groups, where members who trust each other cooperate.

However, these studies have focused on content discovery, resource allocation, and security and trust concerns with the social network peers. They ignored that peers in social networks have willingness to share resources, even if they have no benefit. This paper focus on how these sharing peers can work well in P2P network. Using the spare bandwidth of social network peers to transmit streaming media will increase the bandwidth ratio. So it will improve the playback quality of social-connected peers. We call the bandwidth share peer – sharing peer. Sharing peer is different from normal P2P peers. It only shares its upload bandwidth to its social-connected peers, and it does not playback the video. So sharing peer does not need to completely fill the buffer like normal peers do. If sharing peer fill buffer as normal peer did, it is inevitable that sharing peer downloads data blocks which are no social-connected peers need. It is a waste of sharing peer's download bandwidth, and it will lower the bandwidth ratio. Therefore the existing P2P models do not apply to sharing peer, so we propose a social network-based bandwidth sharing model or SNSM for short. SNSM makes full use of the particularity of sharing peer. Sharing peer only downloads the data blocks which social-connected peers need, and transmits the data blocks in the form of "push". SNSM can effectively reduce the download overhead of sharing peer, and increase the bandwidth ratio.

The remainder of the paper is organized as follow. Section 2 describes the SNSM in detail, including the transmit strategy of sharing peer and buffer split strategy of social-connected peer; Section 3 presents the experiment results and data analysis. Section 4 gives the conclusions of this paper.

2. Social Network Based Bandwidth Sharing Model

In social network, people always have strong willing to share resources with their friends, because the sharing situation will affect the social condition in their real life. But sharing peers do not want to share their bandwidth to peers without social relationship. This means that sharing peers can only connect their social-connected peers in the P2P network, and it is very different to the original P2P peers. To solve the problems above, we proposed social network based bandwidth sharing model.

A. Construct SNSM Topology Structure

In our model, peers are divided into two categories: normal peers and social-connected peers (SCP). Normal peers transmit data blocks based on the original P2P protocol. SCP has social connections with the sharing peers. The difference between normal peers and SCP is thatSCP are able to use the bandwidth of sharing peers. Figure 1 shows the relationship of normal peers, SCP and sharing peers.

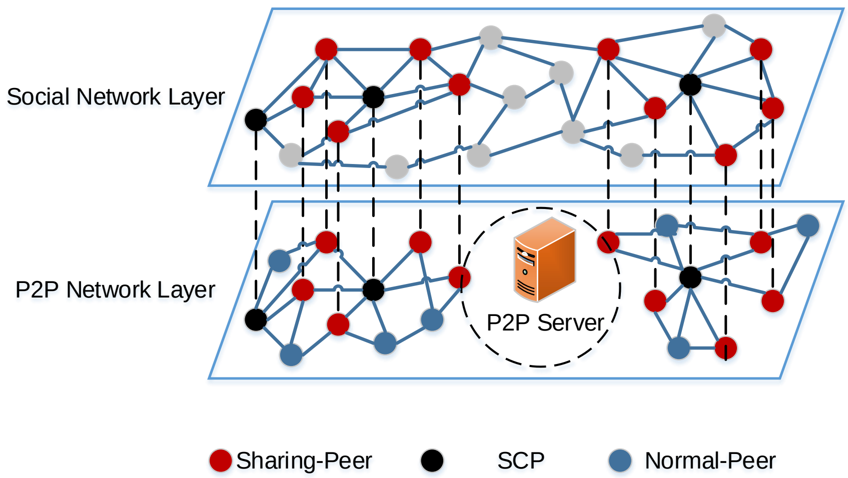

Figure 1. Peer Relationship of SNSM

The procedures of SCP connect with sharing peers is described down below: SCP search sharing peers through the social network. If the number of searched sharing peer is too large, then filter the sharing peers based on the RTT and the SCP number of sharing peers. Sending connect requests to sharing peers which have low RTT and small SCP number. If there is no social peer want to be sharing peers, then search the indirect social peers of SCP. The indirect social peers are as Figure 2 shown. First-degree-friend is social peers which have direct relationship with SCP. Second-degree-friend is social peers which have the same friend peer with the SCP, but there is no direct connection between second-degree-friend and SCP. If SCP still does not find peers have willing to share in indirect peers, then it will cancel the connect operation and turn into normal peer. After a certain interval, normal peer start the search procedure again. When sharing peer receive connect requests, it needs to judge whether the connection number reach the limit. If the connection number has not reached the limit, then sharing peer will accept the connection request, establish connection with the SCP. If the connection number has reached the limit, then sharing will reject the connection request.

After the SNSM topology structure is constructed, there are still some problems need to be solved. First, the number of SCP in P2P network is unknown. If sharing peer connect few peers, due to the network latency the effect of sharing peer will not be obvious. Second, if sharing peer only connects with SCP, sharing peer may not found the peers which contain the data blocks that sharing peer needs. In this situation, sharing peer can't service the SCP even if it has a lot of upload bandwidth.

So, sharing peers in SNSM have two neighbor lists, download neighbor list and upload neighbor list. Download neighbor list could contain any peers in the P2P network, no matter these peers are SCP or not. Upload neighbor list only contains SCP. Sharing peer requests data blocks from peers in download neighbor list, and transmits data block to the peers in upload neighbor list. In this way, sharing peer can get any data blocks it want, and only share its upload bandwidth to SCP. Separating the neighbor list does not mean sharing peers could serve the SCP well. It needs a relevant transmission strategy.

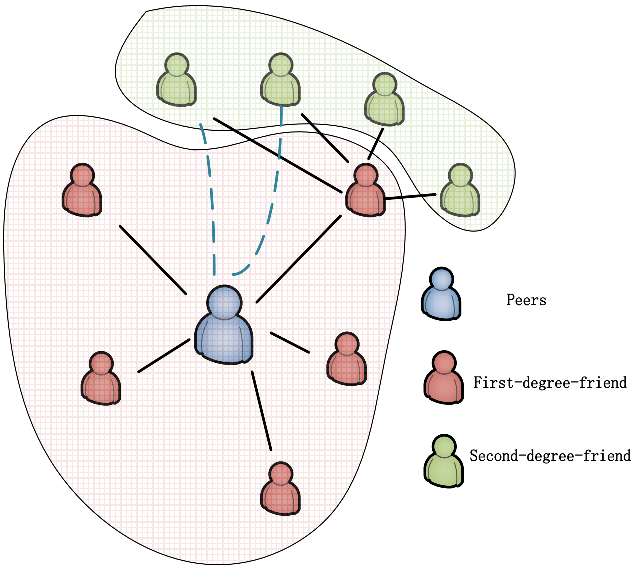

Figure 2. Relationship of Direct and Indirect Social Peers

B. Original P2P Transmission Strategy

To evaluate the peers' contribution to the P2P network, we proposed the concept of sharing ratio. Sharing ratio is the ratio of peer's upload data block number and download data block number in a certain time. Assuming the sharing ratio of peer is C, sharing peer's download data block number is D and upload data block number is U in T seconds. Then the equation of sharing ratio is$C = (U - D)/T$.When C is bigger than zero, the peers contribute bandwidth to the P2P network, when C is smaller than zero, and the peer consumed the bandwidth of P2P network.

In original P2P transmission strategy, assuming the sharing ratio of normal peers is C_n, upload data block number is U_n and download data block number is D_n in T seconds. Because each peer needs to smooth play the video, it needs data block number played in one second is V. Peer need to satisfy the situation that $D_n \geq V \times T$, so the peer can smooth play the video. So we can get the minimum sharing ratio of peer is$C_n = U_n/T - V$. Because V and T is constant value, the sharing ratio is based on the upload bandwidth. The transmission strategy of original P2P is to do its best effort to download and upload data blocks. Only if the download rate is higher than the streaming rate that normal peer can playback the streaming video. Sharing peer does not apply to this strategy although it can serve the social-connected peers well. Because sharing peer will consume more upload bandwidth of P2P network more than it needs. It will lower the bandwidth ratio of the whole P2P network. On the other hand, social-connected peers pull data blocks from sharing peer. Due to the network latency, the upload bandwidth of sharing peer may not be completely used.

C. Independent Transmission Strategy

We design the independent transmission strategy for the sharing peer. In the independent transmission strategy, sharing peer contains push queue and pull queue. Push queue contains data blocks that sharing peer has, and it will sort by the data block missing number of peers in upload neighbor list. Pull queue contains data blocks that sharing peer does not have and are needed by the peers in upload neighbor list. It is also sorted by the data block missing number. Sharing node pushes data blocks in push queue to the peers in upload neighbor list, and pulls data blocks in pull queue from peers in download neighbor list. The sort process is to make sure sharing peer gives priority to push or pull the data blocks which are most needed by social-connected peers.

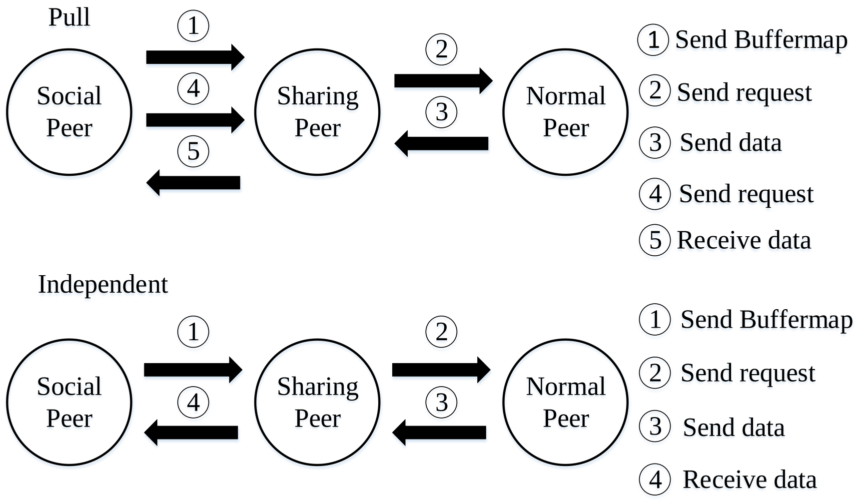

Figure 3. Data Block Request Process in Different Transmission Strategy

In the independent transmission strategy, sharing peer decides which data block is to be pulled or pushed. It is more effective than pull transmission strategy. As Figure 3 shown, social-connected peer needs 5 communications to get data blocks in pull strategy, and needs only 4 communications to get data blocks in independent transmission strategy. The independent transmission strategy is described in detail below.

Step1: Initialize the push queue and pull queue in sharing peer.

Step2: Traverse the upload neighbor list, and get the missing data blocks of each social-connected peer. If missing data block exists in push queue, go to step 3, else then go to step 4. After all the peers have been traversed, go to step 5.

Step3: Increase the missing number of this data block in push queue. Go to step 2.

Step4: If sharing peer has the missing data block, then insert the data block in to the push queue. Else then insert the data block in to the pull queue, if pull queue already contains the data block, then increase the request number of this data block. Go to step 2.

Step5: Sort the push queue in descending order based on the missing number of each data block. Push the data blocks in push queue to the social-connected peers in upload neighbor list.

Step6: Sort the pull queue in descending order based on the request number of each data block. Request data blocks in pull queue from the peers in download neighbor list.

Figure 4 illustrates the process of sharing peer generates push queue and pull queue based on three social-connected peers data block missing situation. Sharing peer has data blocks “1, 2, 3, 5”. Based on the neighbors’ missing data blocks situation, sharing peer generates a push queue which has data blocks ordered as “2, 1, 3, 5”. Meanwhile, sharing peer also generates a pull queue to get data blocks from other peers.

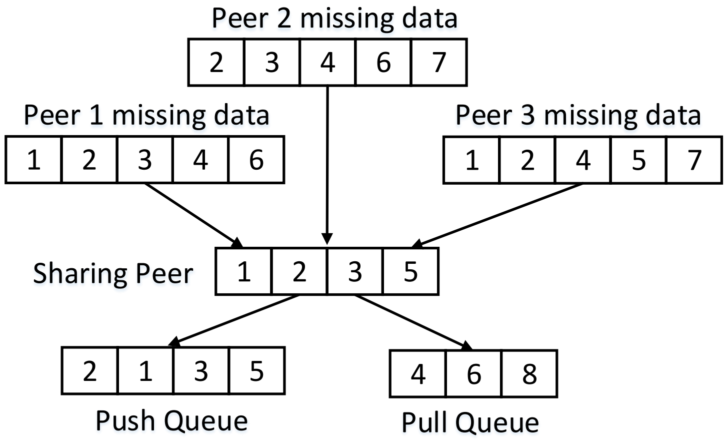

Figure 4. Sharing Peer Generates Push and Pull Queue

Assuming the sharing ratio of sharing peers is C_s, the download data block number isD_s and upload data block number is U_s in T seconds. The sharing ratio of sharing peer is $C_s = (U_s - D_s)/T$. $D_s = \sum D_i$ and $U_s = \sum n_i D_i$, n_i is the the transmit times of data block$i$. Then we can get sharing ratio is $C_s = (\sum n_i D_i - \sum D_i)/T$. The simplification of the equation is $C_s = \sum (n_i - 1) D_i / T$. Sharing peers choose data block based on the missing situation, so each data block that sharing peers get will at least transmit one time, the n_i will always bigger than 1. So the sharing peers' sharing ratio in independent transmission strategy will always bigger than 0. The sharing peers will always contribute to the P2P network.

D. Buffer Division Strategy

In P2P streaming system, buffer is used to storage data blocks for smooth play. The buffer's situation can be expressed as 0-1 array. 0 represents peer does not have the data block and 1 represents peer has the data block. Each item in the buffer array has a data block number. Use the buffer array, peer can clearly know which data blocks it needs, and from which peer it can get the missing data block.

In social network-based bandwidth sharing model, social-connected peers get data blocks from normal peers and sharing peers at the same time. Because of social-connected peer does not know which data blocks will be sent by sharing peers, it may pulls the same data blocks from normal peers. That is a waste of upload bandwidth. To solve this problem, we propose the buffer division strategy. Buffer division strategy divides the buffer of social-connected peers in to three zones: buffer-up zone, dead-line zone and buffer-aid zone.

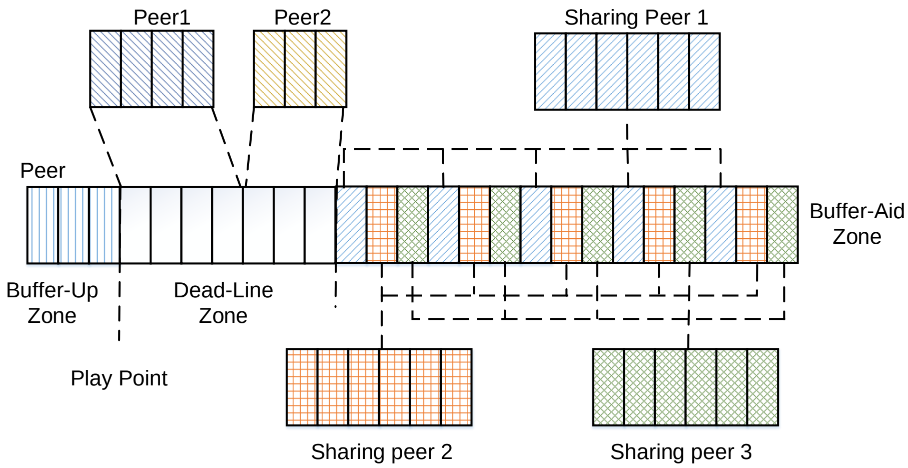

Figure 5. Buffer Division Strategy

As Figure 5 shown, buffer-up zone is the zone which contains the data blocks has been played. Dead-line zone is the zone begin at the play point and last a few seconds, it is used to receive the data blocks social-connected peers pull from other peers. Buffer-aid zone begins at the end of dead-line zone and contains the rest of the buffer; it is used to receive the data blocks sent by sharing peers. Meanwhile, buffer-aid zone will divides into several sub-zones based on the connected sharing peer number, each sub-zone will be sent to sharing peer, sharing peer shares its bandwidth based on the sub-zone. Buffer-aid zone will be converted to the dead-line zone according to the video bit-rate. No matter whether buffer-aid zone has been completely fill or not, the convert process will continue, to make sure the dead-line zone maintain constant length. The missing data blocks in the buffer-aid zone will be got from other peers in the pull way. In this way, buffer division strategy ensures that the social-connected can get all the data blocks even if the sharing peer is invalid.

About the proportion of each kind of buffer zone, we allocate Buffer-Up zone at 20%, Dead-Line Zone at 40% and Buffer-Aid zone at 40%. This kind of ratio is considered about the failure of sharing peers. Even if the sharing peers are all invalid, 40% of buffer is long enough to ensure the smooth play.

3. Results and Discussion

In our simulation, OMNET++ [13] is used for performance verification. Based on the DENACAST [14], we develop social network module and sharing peer module. It is assumed that P2P network exists one streaming server, one tracker server, 200 peers, peer bandwidth is 8 Mbps, and video streaming rate is 512 Kbps. We test our model in six scenarios; each scenario is based on the proportion of sharing peer. The proportions are 0%, 5%, 10%, 15%, 20%, 25%. The proportion 0% can be seen as normal P2P streaming system without social-connected peers. And we set social-connected peer can connected 3 sharing peers at most.

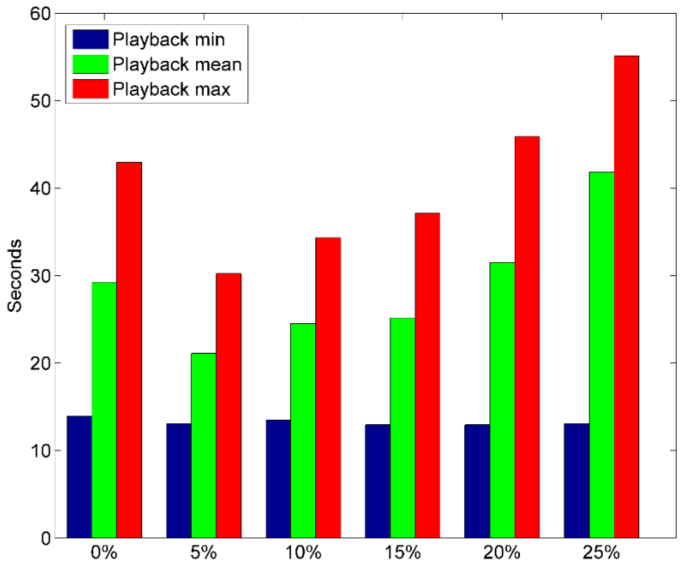

Figure 6. Playback Delay under Different Proportions

Playback delay is an important performance index to measure the quality of P2P streaming service. In the simulation, we record the playback delay of P2P streaming service, so we get results as Figure 6 illustrate. When the proportion between 5% and 15%, streaming service with sharing peers have better playback delay than streaming service without sharing peers, whether in average playback delay, minimum playback delay or maximum playback delay. When the proportion between 20% and 25%, streaming service with sharing peers have worse average playback delay than streaming service without sharing peers have, but it has better playback delay than streaming service without sharing peers at minimum playback delay and maximum playback delay. It is because that sharing peers share their upload bandwidth, the bandwidth ratio will increase, so the average playback delay has become better. But with the sharing peers' proportion increases, the bandwidth overhead of sharing peers also increases. It will affect the performance of peers without sharing peers connected.

The hop count also affects the playback delay, sharing peer as the middle peer of normal peer and social-connected peer, it will increase the hop count. As Figure 7 shown, the hop count increases when the proportion increases. When the proportion is small, in some ways the push process of sharing peer lower the hop count. When the proportion is large, sharing peers consume more upload bandwidth of normal peers, and the hop count is in inverse proportion to the upload bandwidth. Therefore the playback delay becomes worse when the proportion increase.

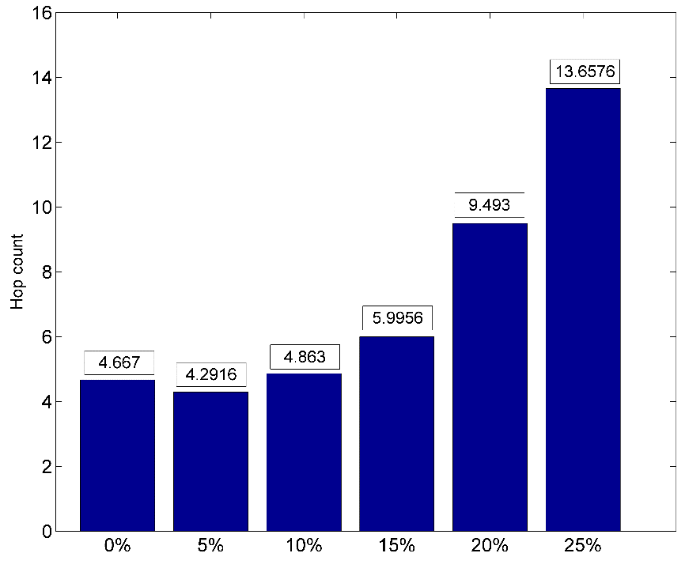

Figure 7. Hop Count Under Different Proportions

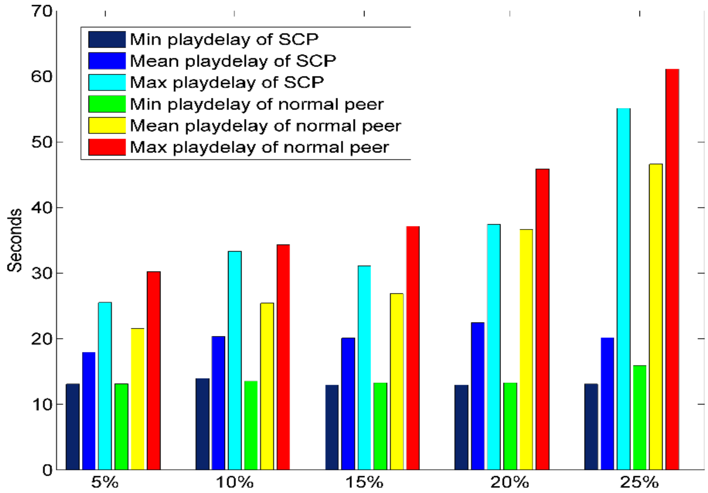

Figure 8. Playback Delay of Peers Under Different Proportions

On the other hand, as Figure 8 shown, in either case, SCP have better playback delay than normal peers in any cases, it proves that the exist of sharing peers could improve the playback quality of social-connected peers. The SCP with the bandwidth of sharing peers always have a stable neighbors, and the sharing peers' "push" operation is more effective than the normal peers' "pull" operation.

4. Conclusion

In this paper, we propose social network-based bandwidth sharing model, this model uses the upload bandwidth of sharing peers in social network to improve the quality of the P2P streaming service. We describe independent transmission strategy and buffer division strategy in social network-based bandwidth sharing model in detail. Through these strategies the sharing peers can effectively share their bandwidth to the social-connected peers, and the social-connected peers can receive data blocks in an orderly and efficient manner. Experimental results illustrate that sharing peers can lower the playback delay and improve the playback quality of social-connected peers. But as the sharing peers proportion increases, the effect of sharing peers is weakened. The focus of our next study is to better organize these large-scale sharing peers in P2P streaming service.

References

[1]. S. Zhijie, L. Jun, R. Zimmermann and A. V. Vasilakos, “Peer-to-peer media streaming: Insights and new developments”, Proceedings of the IEEE, vol. 99, **(2011),** pp. 2089–2109.
[2]. X. Hei, C. Liang, J. Liang, Y. Liu and K. W. Ross, “A Measurement Study of a Large-Scale P2P IPTV System”, Multimedia, IEEE Transactions, vol. 9, no. 8, **(2007)**.
[3]. F. Wang, J. Liu, Y. Xiong and S. Fraser, “Stable Peers: Existence, Importance, and Application in Peer-to-Peer Live Video Streaming”, INFOCOM.27th Conference on Computer Communications. IEEE, (**2008**) April 13-18, Phoenix, AZ.
[4]. J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen and H. Sips, “Tribler: A social based peer-to-peer system”, Concurrency and Computation: Practice and Experience, vol. 20, **(2008),** pp. 127–138.
[5]. S. Buchegger, D. Schiöberg, L. H. Vu and A. Datta, “PeerSoN: P2P Social Networking — Early Experiences and Insights,” In SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, **(2009)**, pp. 46-52.
[6]. Z. Li and H. Y. Shen, “Social-P2P: Social Network-based P2P File Sharing System”, Network Protocols (ICNP), 20th IEEE International Conference, **(2012)** October 30-November 2, Austin, TX.
[7]. D. Ho and H. Song, “Resource Allocation Algorithm based on Social Relation for Video Streaming Services over P2P Network”, Networks (ICON), 18th IEEE International Conference, **(2012)** December 12-14, Singapore.
[8]. M. L. Merani, “How Helpful Can Social Network Friends Be in Peer-to-Peer Video Distribution?”, Parallel and Distributed Systems (ICPADS), IEEE 17th International Conference, **(2011)** December 7-9, Tainan.
[9]. D. Hales and S. Arteconi, “Friends for Free: Self-Organizing Artificial Social Networks for Trust and Cooperation”, Algorithmic Aspects of Large and Complex Networks, vol. 05361, **(2006).**
[10]. W. S. Lin, H. V. Zhao and K. J. R. Liu, “Incentive Cooperation Strategies for Peer-to- Peer Live Multimedia Streaming Social Networks”, Multimedia, IEEE Transactions, vol. 11, no. 3, **(2009)**.
[11]. P. Antoniadis and B. Le Grand, “Self-organized Virtual Communities: Bridging the Gap between Web-based Communities and P2P Systems”, International Journal of Web Based Communities, vol. 5, no. 2, **(2009)**.
[12]. J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen and H. J. Sips, “Tribler: A social-based Peer-to-Peer System”, Recent Advances in Peer-to-Peer Systems (IPTPS 2006), **(2006)**.
[13]. http://www.omnetpp.org
[14]. http://www.denacast.org