A Scalable Live Streaming Media Distribution Service for Peer-to-Peer Network

Jiujun Cheng [a], Chungang Yan [a], Kuifei Yu [b], Jian Ma [b]
[a] *College of Electronics & Infromation Engineering, TongJi University, No. 4800 Cao An Road, 201804, Shanghai, China*
[b] *NOKIA Research Center, 100013, Beijing , China*
chengjj@mail.tongji.edu.cn

Abstract

Live streaming media distribution in a large scale P2P network and the impact on the real-time caused by the dynamicity of peers are still not well resolved. In order to address this problem, a scalable live streaming media distribution service in P2P network is presented. Our approach is that the P2P network combining tree with clusters is designed, and some super-peers are reserved in all clusters. In addition, we adopt the maximum accepted connection number of each super-peer and its average disposing delay to select the optimal father peer, and the limitation of packet discard between super-peers is given. The experiment results show that the distribution service assures the improvement of real-time and scalability in the large-scale P2P network.

1. Introduction

There has been much work [1][2][3][4][5][6] in recent years on the topic of content distribution of live streaming media. In the case of the live streaming, Overcast [1] organizes dedicated servers into a source-rooted multicast tree using bandwidth estimation measurements to optimize bandwidth usage across the tree. SplitStream [2] distributes the forwarding load over all participants using multiple multicast trees. CoopNet [3] addresses a hybrid system for streaming media, which utilizes multiple application-level trees with striping and Multiple Description Encoding (MDC) [4]. In CoopNet, a centralized server is used to stream media. The content is divided into multiple sub-streams using multiple descriptions coding (MDC) and each sub-stream is delivered to the requesting client via a different peer. SpreadIt [5] which builds an application level multicast tree over the set of clients, in SpreadIt, it uses only a single distribution tree and hence is vulnerable to disruptions due to node departures.

Our work on distributing live streaming media is also similar to SpreadIt. A key distinction of it is that we focus on the P2P network combining tree with cluster, and some super-peers in each cluster are reserved for peer’s arrival and departures. Especially we adopt the maximum accepted connection number of each super-peer and its average disposing delay to select the optimal father peer. Through arranging the tree structure rationally, we can make the tree depth be shorter as more as possible, so that the communication time delay between super-peers can be shorten, and the real-time performance can be improved.

The remainder of the paper is organized as follows. In section 2, key components of new live streaming media distribution P2P network is presented. In section 3, the distributing process on live streaming media is described. The limitation of packet

---

This work is supported by Program for Young Excellent Talents in Tongji University and the Tongji-NOKIA project of “CODIO” and “Virtual Operator Grid”, National Natural Science Foundation (No.90718012, No.60534060), National Basic Research Program of China (2007CB316502) and Shanghai Key Basic Research Project(07JC14016).

discard during the distribution on live streaming media and the performance analysis and simulation results are respectively depicted in section 4 and 5. Finally, conclusion is made in section 6.

2. Key components of new live streaming media distribution P2P network

2.1. The live streaming media distribution P2P network

In our P2P network, it includes four kinds of peers, they are respectively the media server, the index server, super-peer (principal peer and subordinate peer), and ordinary peer. Figure 1 illustrates the live streaming media distribution P2P network.

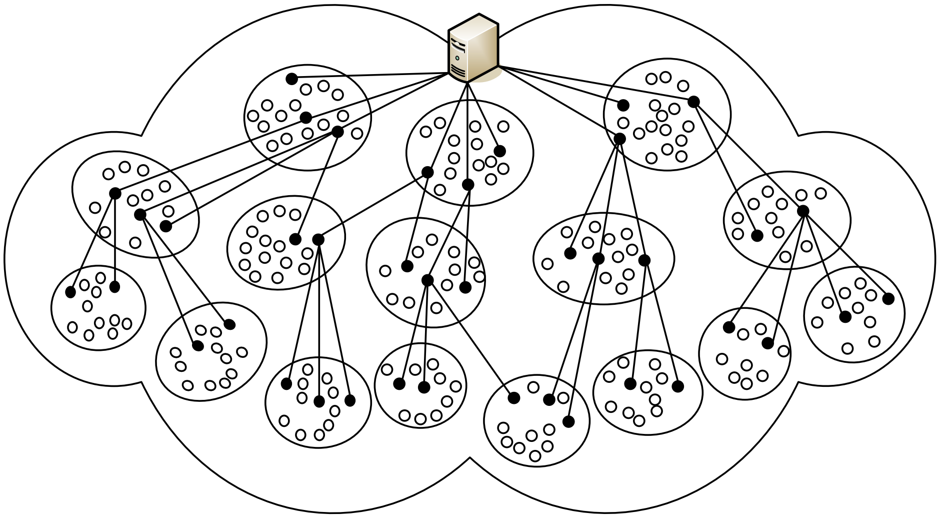

Figure 1. The framework of live streaming media distribution P2P network

In this simple example, there exist sixteen clusters, and they are respectively C_1, C_2,…,C_8, C_9,…C_{15},C_{16}. Several different super-peers in each cluster are illustrated, such as peers P_1, P_2, P_3 in C_1, and P_{13}, P_{14} in C_5. In addition, a certain number of super-peers in each cluster are picked as principal peers which are member peers in the whole tree, and they are used to connect with the super-peers in other clusters, such as P_1, P_2 in C_1, and P_7, P_9 in C_3; Some super-peers are reserved as subordinate peers, such as P_3 in C_1, and P_5 in C_2, they are leaf peers in the tree. When the principal peers departed from the cluster, the subordinate peers in the same cluster or principal peers in other cluster would be selected as the new principal peers.

In each cluster, the subordinate peers are ranged in an increasing by degrees according to the average disposing delay themselves, and the average disposing delay is gained by the fixed queue system M/G/1 [7]. For example, in C_1, it supposes that there are n peers, where, P_1, P_2, P_3 are super-peers, and the others are ordinary peers, all peers in C_1 can be denoted by set S, that is $S = \{< P_1, P_2, P_3>, P_i ,\dots,P_k,\dots,P_n\}$, where, $< P_1,P_2,P_3>$ denote super-peers, and they are in turn principal peers and the subordinate peers which are ranged in an increasing by degrees according to the average disposing delay themselves. In set S, they are ordinary peers except super-peers, and they can gain the stripes from principal peers, and also gain the stripes from other ordinary peers in the same cluster by P2P protocol, such as *BT* [8].

Each super-peer must keep the correlative information of all other super-peers in the same cluster, that is, the first element in the set. For example, $< P_1, P_2, P_3>$,$< P_4, P_6, P_5>$ denote respectively the relation amongst peers in C_1 and C_2. In addition, each super-peer must keep the correlative information of his father peer and his sub-peers, for instance, the father peer and the sub-peers of P_{14} are respectively P_6 and P_{27}, P_{28}, P_{29}.

2.2. The live streaming media index server

In order to make the connectivity of each super-peer be saturated, shorten the depth of the tree, and further improve the real-time of the whole P2P network, here, we adopt the index server to maintain the information of all super-peers in the whole tree. The information includes primarily as follows: ① the father peer, the brother peers and the sub-peers; ② the maximum connectivity and the current connectivity of each peer; ③ the average disposing delay t_i of each peer; ④ the average delay T_i between each sub-peer and its father peer.

The index server need maintain two lists: the adjacency list and the contrary adjacency list. Where, the adjacency list depicts the relationship between each super-peer and its sub-peers and its brother peers in the tree; the contrary adjacency list describes the relationship between each super-peer and its father peer in the tree. The index server will renew its adjacency list and contrary adjacency list to find super-peer's joining and departure in the tree, which maintains well the dynamic of the tree.

3. The distributing process on live streaming media

3.1. Joining process on new super-peer

The process which the new super-peers join the tree is described as follows. First, the new super-peer sends a *HTTP GET* message to the Web server to ask for the information on the file of live streaming media, and the message includes the index server's *IP* address, port, the media codec format, and the *Normal Playing Time (NPT)* of the file, and so on. The Web server adopts description information to respond the new super-peer. Here, the description information includes the index server's *IP* address and port.

According to the information returned by the Web server, the new super-peer connects with the Web server and send *SETUP* message to it, where, the *SETUP* message includes the new peer's port for receiving the media, the way of transferring data, and *IP* address. According to the information provided by the new super-peer and the list which it own maintains, the index server seeks the cluster which the new super-peer belongs to, and assigns it the *ID* of the cluster.

If the index server found that no cluster belongs to the new super-peer, then it would create a new cluster for the new super-peer. At the same time, the index server returns the information on unsaturated super-peers to the new super-peer, and the information includes *IP* address, port, the average disposing delay, and the depth which the super-peer situates in the tree, the *ID* which the super-peer belongs to, and *NPT* on the stripes in the cache of peer.

After the new super-peer has received the *SETUP* Response message, it would send the Trace route message with its port and *IP* address to the unsaturated super-peers, and the unsaturated super-peers return it the Trace route response message which includes the average delay $T_{interval}$ between super-peers. The new super-peer may select the optimal father peer with T, and the value T is as follows:

$$T = \min[\sum_{i=1}^{n}(t_i + T_i) + T_{\text{interval}}] \quad (i = 1 \ldots n) \qquad (1)$$

where, n is the number of super-peers between the media server and the unsaturated peer; i is the scalar quantity of each super-peer between the media server and the unsaturated super-peer; T_i is gained by the Trace route message, and t_i is the average disposing delay which is got by the M/G/1 queue model. In figure 1, assume that P_{10} is an unsaturated peer, then, n is 2, and t_1 and T_1 denote respectively P_2's average disposing delay and the average delay between P_2 and the media server. Similarly, t_2 and T_2 denote respectively P_{10}'s average disposing delay and the average delay between P_{10} and P_2. $T_{interval}$ is the average delay between the new super-peer and P_{10}. After the new super-peer selects the optimal father peer, the stripes are transmitted from its father peer to it by *RTP* and *RTCP* protocol.

3.2. Departure process on the super-peer in the tree

In P2P network, due to the dynamic of peer, some principal super-peer may unexpectedly leave the tree, so it is very important that how to find the suitable father peer to ensure his descendant peer be real-time, and how to make other super-peers in the tree be saturated in order to shorten the depth of the tree.

Next, we describe the process that the descendant peers P_{27}, P_{28}, P_{29} in figure 1 how to find the new father peer when their father super-peer P_{14} has departed from the tree. The descendant peers P_{27}, P_{28}, P_{29} in advance keep each other's information, so, when they find that their father peer has left the tree, one of them will send the message on their father peer's departure to the index server. As soon as the index server receives the message sent by a certain descendant peer, it will seek the adjacency list and the contrary adjacency list which are maintained by it, and find that P_{14} and its descendant peers are disappeared, at the same time, it will find that P_6's descendant peer is P_{14} and P_{15} in the adjacency list. This indicates that P_{15} is P_{14}'s brother peer.

In this way, the index server will find that the cluster which P_{13} belongs to is the same as that of P_{14}, and seek the saturation of P_{14}'s brother peer and the super-peer in the same cluster, then, it will send *IP* address, port, and the average disposing delay of the unsaturated peers to P_{27}, P_{28}, P_{29}. According to the average delay T_i between the descendant peer and the unsaturated peer detected by Trace route message, and the average disposing delay t_i of the descendant peer, it would select the optimal father peer with $T_i + t_i$.

3.3 The limitation of packet discard among super peers in the P2P network during the distribution on live streaming media

During the distribution on live streaming media, successive packet discard between peers can also affect the real-time of live streaming media and the visual effect of Video, therefore, in order to assure *QoS* of live streaming media, it must rationally design the maxium packet discard number which is allowed between super-peers. Here, we mainly discuss the relationship among the maxium packet discard number and the buffer time of sender and receiver and the time delay between super-peers. Here, the time when the father super peer sends data is *S*, and the time that the sub-super-peer received data is *T*, and the time delay between the father super peer and the sub-super-peer is T_{E-E}, the average buffer time of the sub-peer is t_i, the time interval of date packet arriving between super-peers is $T_{interval}$, the time when data packets happen to replay is R, the time when the sub-super-peer detects the packet discard is D, the playing time of live streaming media is P, and then, the *i*th data packet has:

$$T_i = S_i + T_{E-E} \quad (2)$$

$$P_i = T_{E-E} + t_i \quad (3)$$

The time when the *(i+n)*th data packet arrivals T_{i+n} is:

$$T_{i+n} = T_i + nT_{\text{int}erval} \quad (4)$$

During the live streaming media playing, the maxium n data packet discard can be allowed when it happens to replay. When the sub-peer found that n data packets had been discarded at T_{i+n}, it would send the replaying information to his father, then, the time when the data packets replay R_{i+1} is:

$$R_{i+1} = T_{i+n} + T_{E-E} \quad (5)$$

It could satisfy with the real-time only if the sub-peer received the *(i+1)*th data packet at P_{i+1}, so it must be satisfied:

$$P_{i+1} > R_{i+1} + T_{E-E} \quad (6)$$

That is:

$$P_{i+1} > T_i + nT_{\text{int}erval} + 2T_{E-E} \quad (7)$$

and:

$$P_{i+1} = T_i + T_{\text{int}erval} + t_i \quad (8)$$

and then:

$$t_i > (n-1)T_{\text{int}erval} + 2T_{E-E} \quad (9)$$

Therefore, when the average disposing time t_i of the sub-peer can satisfy with (9), the data packet can be recovered when the sub-peer successively discarded *n* data packets, and the streaming media can play lively; When the successive packets discard exceed *n*, QoS of the live streaming media can not assured. Here, t_i is relative with $T_{\text{int}erval}$ and T_{E-E}, thus, if $T_{\text{int}erval}$ and T_{E-E} are reduced possibly, it could improve the real-time of live streaming media playing in the P2P network.

4. Performance analysis and experiment results

In this section, we will try to simulate the advantage of the real-time on the new P2P network which combines tree with clusters, which compares with live streaming media over a P2P network presented by *Hrishikesh Deshpande[9]*. Here, we adopt *NS-2.29 [10]* developed by Berkeley to test, and simulate respectively the real-time on a number of live streaming media are distributed and on the descendant peers how to find the new father peer after the father peer left abruptly from the tree in the two above mentioned P2P network.

4.1. The real-time which a number of live streaming media are distributed in the two different P2P network

Assume that there are 340 peers in the two different P2P network; each peer's average disposing delay ti is *100~500ms*, and it obeys uniform distribution; the average delay T_i between two peers in the tree is *10~100ms*, and it also obeys uniform distribution. We consider one scenario as follows: the stripe is *256Kb*, and the link band-width is *100~500Kbps*.

The topologies on the two different P2P network are respectively illustrated by figure 2 and figure 3.

Figure 2 describes the topology of the P2P network presented by Hrishikesh Deshpande. In this structure, it supposes that each peer has 4 descendant peers, so 340 peers can be divided into 4 layers, and each layer has respectively 4, 16, 64, and 256 peers.

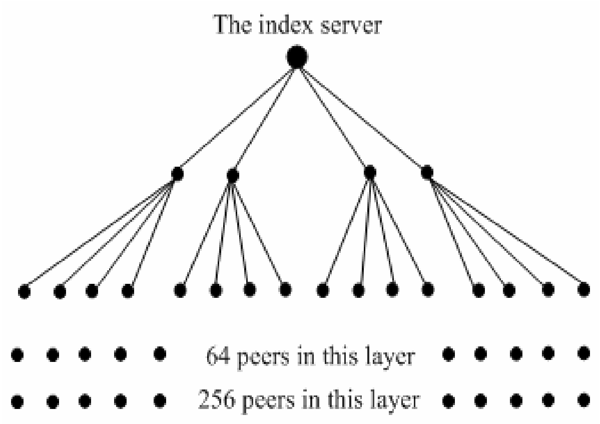

Figure 2. The topology represented by Hrishikesh Deshpande

Figure 3 depicts the topology on the new P2P network which combines tree with clusters, in this structure, there are six clusters, where, the first and the second cluster have respectively 2 super-peers and 54 ordinary peers, and the remainder four clusters have respectively 4 super-peers and 53 ordinary peers.

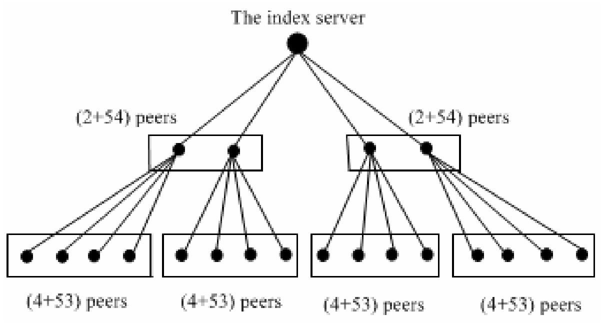

Figure 3. The topology on the new P2P network

For the scenario, the simulation results are illustrated by figure 4, it illustrated the real-time that the stripe is *256Kb* and the link band-width is *100~500Kbps*. Here, we adopt the number of peers which have received the stripes in different interval to denote the real-time in the two different P2P network.

In figure 4, the time that the former four peers have received media data is equal, which indicates that the media data can be only transmitted in the tree during a period of time. After the former 4 peers in the new P2P network have finished downloading, next, the stripes are transmitted not only in the other peers of the tree, but also in the ordinary peers of cluster where P2P protocol is used. Therefore, at the same moment, the number of peers which have received the stripes in the new P2P network is obviously more than that in the P2P network presented by Hrishikesh Deshpande.

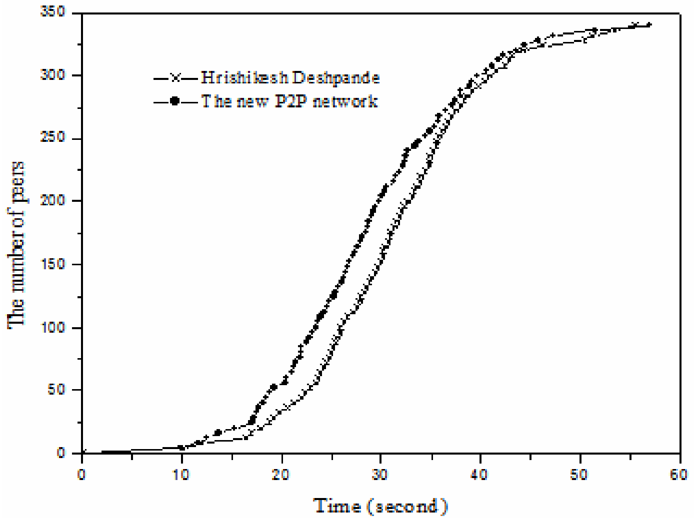

Figure 4. The relationship between time and the number of peers: link bind-width is*0.1~0.5M*bps and media data is*256Kb* in the two different P2P network

4.2. The real-time which the descendant peers find the new father peer in the two different P2P network

We assume that the structures of the two P2P network are identical and illustrated by the figure 5. In figure 5, when P_i departed abruptly from the tree, the approach that its descendant peers P_j, P_k find it having left is the same in the two different networks, thus, the costs that the finding process are identical.

Next, what the descendant peers P_j, P_k do is starting to find the new father peer. The idea that the descendant peers find the new father peer presented by Hrishikesh Deshpande is: the descendant peers send the message to the source peer S, and S returns P_t to P_j, P_k, either of P_j, P_k starts to connect with P_t, here, assume that the peer is P_j. At the same time, P_k connects with P_j, then, it becomes P_j's descendant peer.

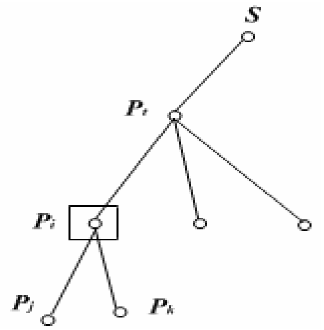

Figure 5. The structure on the new father peer being found

In the new P2P network, when P_j, P_k find that their father peer has been left, either of them send the message to the index server. After the index server has renewed the adjacency list and the contrary adjacency list, it will send the message on P_i's unsaturated brother peers and the subordinate peers which situate with P_i in the same cluster to P_j, P_k. Then, P_j, P_k will start to create new connection with these peers.

In order to validate the real-time on finding the new father peer in the two different P2P networks, the simulation starts from P_j, P_k having found that P_i had been left, and terminates when P_j, P_k have fully receiving *256 KB*. We assume that the link band-width and the delay between two peers are respectively *100~500Kbps* and *10~100ms*, and they obey uniform distribution. In addition, the stripe which each peer reserves and transmits is *256KB/16*.

In figure 6, we can make conclusions as follows. From zero to a period of time, the media data which the two different mechanisms on finding the new father peer have finished are both zero, it indicates that the descendant peers are finding the new father peer, but have not received the media data.

About 0.3 second later, the media data starts to be transmitted in the new P2P network, it denotes that P_j, P_k have found the new father peer and starts to receive the media data; About 0.5 seconds later, the media data starts to be transmitted in the P2P network presented by Hrishikesh Deshpande, it indicates that the time when peers find its father peer in the new P2P network is faster than that in the P2P network presented by Hrishikesh Deshpande.

In addition, in the P2P network presented by Hrishikesh Deshpande, one starts to connect with its father peer and receive the media data, but the other hasn't connected its father peer, because the exchange message between the peer and the father peer has been blocked, the connection process has been postponed. Thus, the number of media data received in the P2P network presented by Hrishikesh Deshpande is always smaller than that in the new P2P network. When P_j, P_k in the two different P2P networks have fully received *256KB* media data, the interval is *1.5s*. Here, the media data distribution adopts reserving and transmitting technology, thus, in figure 6, the data on the two curves keep invariable for a moment.

From the simulation results, it is not difficult finding that the new P2P network has improved the real-time on the live streaming media.

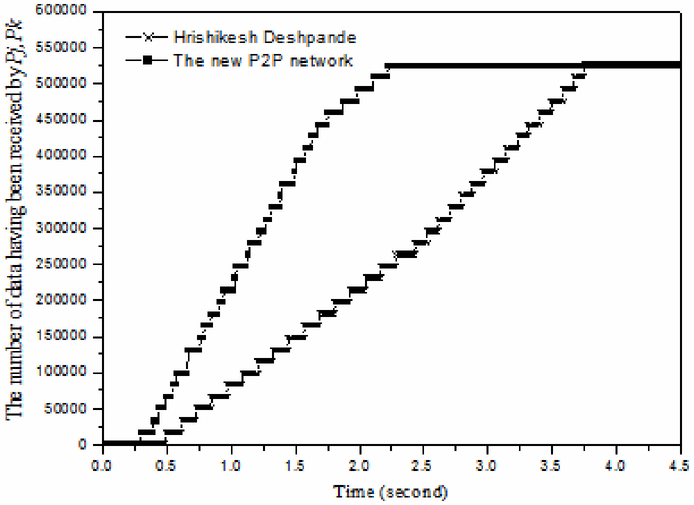

Figure 6. The relationship between time and the number of media data received by *Pj, Pk* before and after having found the new father peer in the two different P2P network

5. Conclusion

In this paper, we propose a live streaming media distribution service in P2P network, which combines tree with cluster. According to the geographical locations of peers, the whole P2P network is divided into multi-clusters, and some super-peers in each cluster are picked as the members in the structure of the tree. The live streaming media runs through the tree into the whole P2P network. In addition, we reserve some super-peers in each cluster as spare ones in order to remedy the topology of the tree when the principal peer is disabled, it can reduce the effect by the dynamic of peers.

We can use the maximum accepted connection number of each super-peer and its average delay, and design purposefully the structure of the whole tree, and shorten the depth of the tree, in order to reduce communication delay between super-peers, and consequently improve the real-time in the whole P2P network. The simulation results demonstrate that the new live streaming media distribution service assures the real-time in the large-scale P2P network, and at the same time, it can also improves the scalability of the P2P network.

6. References

[1]J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole. Overcast: Reliable multicasting with an overlay network. In OSDI 2000, San Diego, CA, 2000.

[2]M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. singh. Splitstream: High-bandwidth content distribution in cooperative environments. In Proceedings of IPTPS03, Berkeley, USA, Feb. 2003.

[3]V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In NOSSDAN, Miami Beach, FL, USA, May 2002.

[4]V. K. Goyal. Multiple description coding: Compression meet the network. IEEE Signal Processing Magazine, 18(5):74-93, Sept. 2001.

[5]H. Deshpande, M. Bawa, and H. Garcia-Molina. Streaming live media over a peer-to-peer network, Apr. 2001. Stanford university, CA, USA.

[6]M. Zhang, Y. Xiong, Q. Zhang, and S. Yang, "On the Optimal Scheduling for Media Streaming in Data-driven Overlay Networks", in IEEE Globecom 2006.

[7]Choi B D, Kim B, M/G/1 Queueing system with fixed feedback policy, the ANZIAM Journal, October 2002, 44(2): 283-297.

[8] Bram Cohen, Incentives Build Robustness in BitTorrent. Bitconjurer.org/BitTorrent, May 22, 2003.

[9] Hrishikesh Deshpande,Mayank Bawa,Hector Garcia-Molina.Streaming Live Media over a Peer-to-Peer Network, Stanford Technical Report 2001-30.

[10] NS2,URL http://www.isi.edu/nsnam/ns.

Authors

Jiujun Cheng was born in 1976, obtained his PhD degree from Beijing University of Posts and Telecommunications in 2006. He is presently a lecturer of Tongji University. He received the support from Young Excellent Talents in Tongji University. He is author of more than 10 conference and journal papers, and 2 books. His research interests span the area of P2P Overlay and Social networks with a focus on IN/Internet interworking, and wireless sensor network.

Chungang Yan was born in 1963, She received Her PhD at Tongji University in 2006. She is presently a professor of tongji University. She has published over 30 papers in different journals and conferences, including Int. J. of Studies in Informatics and Control, and Science World. She received the support from Natural Science Foundation of China and National Basic Research program of China. She specializes in research of trusty theory on software process, concurrent model and algorithm, and Petri Net theory.

Kuifei Yu was born in 1981, obtained his M.S. degree from Beijing University of Posts and Telecommunications in 2007. Then he joined Nokia Research Center, Beijing. He has published several papers and have several patents granted. His research interests include social networking services, mobile P2P and Grid computing.

Jian Ma was born in 1959, he received His PhD at Helsinki University of Technology in 1994. He is presently a Principal Member of Research Staff Nokia Research Center Beijing. He is also guest professor in Computer Science in BUPT, CCCE-GUCAS, ICT-CAS and Tsinghua University. He is author and co-author of more than 160 conference and journal papers, and 4 books and book chapters. He has a dozen of patents granted and tens of patent applications. He has also been serving numerous co-chairs and members of organization committees or program committees in several leading conferences like ICC, AINA, PIMRC, ICDCS, WWW, ChinaCom, etc. His current research interests are mobile enabled wireless sensor networks (mWSN) and its applications and services, ubiquitous reality search, grid services, ad-hoc networking and p2p networking, and etc.