Taxation-based Neighbor Selection for Layered Peer-to-Peer Streaming

Hongyun Yang[1, 2] and Ruimin Hu[1, 3]*

[1]*Computer School, Wuhan University, Hubei, 430079, China*
[2] *National Engineering Research Center for E-Learning, Central China Normal University, Hubei, 430072, China*
[3]*National Engineering Research Center for Multimedia Software, Wuhan University, Hubei, 430079, China*
yhycxh@gmail.com, hrm1964@163.com

Abstract

Peer-to-Peer (P2P) streaming is being considered as the most promising approach to deliver real-time video to large scale users over the Internet. Neighbor selection is one of the key components to construct overlay topology for P2P streaming systems. Currently the majority of QoS-aware neighbor selection approaches assume that the allocated bandwidth resources to individual peers are proportional to their incoming bandwidth rather than their outgoing bandwidth and don't suit for bandwidth resource scarce environments. In this paper, we incorporate taxation-based incentive mechanism into QoS aware neighbor selection method to computer the allocated number of neighbors of peers. The main contribution of this paper is: i) we incorporate linear taxation-model into SVC-based layered media delivery to determined the connection number of peers and propose a distributed bandwidth resource allocation policy; ii) when selecting neighbors, it considers existing peers' uplink capacity and source to peer's delay as a whole to handle long cumulate delay caused by bandwidth aggregation in mesh-based system. Simulation results demonstrate that under resource constraint scene, our proposed method can receive good performance compared with fixed random neighbor selection method and QoS-aware method based on bandwidth-latency ratio on the metric of the chunk loss rate, the average delivery delay, control overhead and PSNR.

Keywords: *Peer Selection, taxation, P2P Streaming, Resource Scarce Network Environment, layered media*

1. Introduction

Recently, Peer-to-Peer (P2P) streaming technology has emerged as the most promising approach to deliver multimedia contents to millions of users over the Internet at a low infrastructure cost. In P2P streaming systems, participating users, called peers or nodes contact with each other forming an overlay network and then deliver the content they have already received to other peers. So the system has two basic components, one is overlay topology construction and the other is data scheduling. According to the overlay network topology structure, P2P streaming systems can be broadly classified into two categories: tree-based and mesh-based [1]. In tree-based P2P streaming systems, such as ESM [2] and SplitStream [3], peers are well-organized into single-tree or multiple-tree structure and video data are being pushed from upper peer to all of its children. The major drawback of tree-based P2P systems is fragile to peers' churn. Once an upper peer crashed, all of its offspring in the tree will stop receiving data. The mesh-based (also called swarm-based) systems, such as Coolstreaming [4], by using gossip-based membership management mechanism, the peers self-organize themselves into a random overlay topology. Two peers that can communicate with each other are called neighbors in mesh-based systems. The source of the system divides the video stream into small chunks of data, which are separately and independently distributed on the overlay. Peers periodically exchange information about their data availability and pull the video data from their neighbors who have already received. The ways of establishing or terminating the relationship among peers is dynamic, which depends on the content availability and bandwidth availability on peers and makes mesh-based systems more robust to peers churn than tree-based systems. Therefore a majority of commercial P2P streaming systems such as PPLive [5], Sopcast [6] and UUSee [7] adopt mesh-based mode.

In mesh-based P2P systems, the connectivity between peers is established by the neighbor selection algorithm for each peer [8].Gossip-based fixed number neighbor selection approach is the first public known solution and is widely adopted by current many large-scale P2P streaming systems. The random neighbor selection method causes it hard to guarantee the quality of service of peers [8, 9], such as low delivery delay and high continuity of playback according to its neglecting the heterogeneity of peers. What's more, the fixed number of neighbors will increase the control overhead to low bandwidth peers for the maintenance of large neighborhoods. QoS-aware neighbor selection approaches with fixed or variable neighbor number have been proposed, such as [10, 11, 9, 8]. In these methods, the QoS characteristics, *e.g.*, the uplink bandwidth of peers or delay between peers are taken into account. Through selecting peers with high bandwidth and low delay, a new joining peer can obtain good performance. In variable number of neighborhood method, the number of neighbor of a peer is proportional to its download bandwidth. However, currently most of those existing neighbor selection methods build on the assumption that the total available upload bandwidth provided by participating peers exceeds the demand download bandwidth, that is resource-rich environment and the finite download bandwidth of peers is the peers' performance bottleneck. So satisfying the maximum received quality of service of each peer under their download bandwidth constraint is the main goal. However, as we have known, in mainstream Internet environments today, majority of hosts are behind asymmetric connection such as DSL and cable modern. The mismatch between upload and download bandwidth makes the available bandwidth resource of the system poor. In resource scarce environment, it is impossible to satisfy each peer's requirement proportional to their own download bandwidth.

In this paper, under the bandwidth-degree condition proposed by Nazanin and Reza [13], we introduce a tax-based incentive mechanism into neighbor selection scheme. The number of neighbors (*e.g.*, the connection number of parents) a peer received is assigned according to its own contribution (*e.g.*, uplink bandwidth) and the overall system resource condition. Furth more, we propose a new QoS aware neighbor selection method for heterogeneous P2P streaming with SVC (Scalable Video Coding) [14] encoded video. Considering the strong dependencies among the encoded video layers and the tradeoff between bandwidth and delay, we select parent neighbor peers according to the ratio of uplink capacity and source to peer's delay with a certain probability. The effectiveness of incorporating tax-based contribution-aware scheme in overlay construction of tree-based and mesh-based approach combining with MDC (Multiple Description Coding) has been investigated by Sung, *et al.*, [15] and by Magharei, *et al.*, [16]. And in [17], the authors focused on data scheduling and integrated the tax-based scheme into layer subscription and layer scheduling with SVC in P2P streaming. The current designs of MDC still incur much higher bandwidth overhead than that of SVC. Considering better scaling properties of mesh-based approach and higher video coding efficiency of SVC layered coding compared to MDC, we focus on designing neighbor selection policy for the mesh-based P2P streaming systems with SVC encoding video. Our main contributions as follows: Firstly, we used linear taxation model to direct bandwidth allocation in SVC encoded system to achieve efficiency and fairness in bandwidth resource scarce environments; Secondly, we proposed a distributed heuristic bandwidth resource allocation policy to monitor the overall system resources and adjust the connection among peers . Thirdly, we proposed to use the ratio of uplink capacity and source-to -peer delay as the selection criteria and combined with random selection to enhance the robust of the overlay. The extensive experiments demonstrate the proposed scheme effective in improving the quality of experience of end users in bandwidth-scarce, heterogeneous network operating environments.

The rest of this paper is organized as follows. Section 2 discusses some related work in this area. The proposed neighbor selection method is described in Section 3. Section 4 discusses our evaluation methods and metrics and then presents the evaluation results. Finally, Section 5 concludes our work and pointed out the further works.

2. Related Works

Recently, several neighbor selection methods have been proposed in the literature. Those approaches could be classified into two categories: random selection method and QoS aware neighbor selection method [6, 22]. SCAMP [18] (a gossip-based membership management protocol) is the representatives of random neighbor selection method and has been used in DONet [4] and RPIME [13] systems. The random neighbor selection method has high ability of robust and resilient to peers' churn and balance the load. But this method neglects the QoS metrics, such as bandwidth and delay and is not suit for constructing the overlay of QoS-sensitive application in heterogeneous P2P steaming system. Therefore, some research communities proposed QoS aware neighbor selection methods, such as [8, 9, 11, 19-22] to construct tree-based or mesh-based overlay topology. In some tree-based QoS aware systems, such as NICE [22] and Zigzag [23], adopt the hierarchical clustering heuristics method to minimize the transmission delay. However tree-based structure is challenging in face of frequent peer churns and many recent P2P streaming systems adopt mesh-based streaming approaches [5-7]. For single layer mesh-based P2P streaming systems, Laizhong, *et al.*, [8] proposes bandwidth latency ratio (BLR) as a new selection metric and the probability that a new joining peer chooses a existing peer as its neighbor is directly proportional to existing peer's upload bandwidth and inversely proportional to the latency between them. Such approach leads to high delay from the source to end hosts. Dongni, *et al.*, [20] points out that aggregating the bandwidth of multiple parents to guarantee a certain streaming rate in mesh-based system comes with the cost of delay, proposes to use the ratio of potential parent's residual uplink capacity divided by the path delay from source to the new peer through the potential parent as the power of potential peer. By choosing the parents with large power in greedy manner, the newcomer gets an aggregate incoming streaming equaled to the streaming rate while achieving very low source-to-peer delay However, these works in single layer P2P streaming assume the total uplink bandwidth is larger than the total downstream bandwidth and each peer's download capacity is larger than the streaming rate, which means that each peer can receive the full service from their selected parents. It doesn't suit for high-definition (HD) video P2P applications, which requires large download bandwidth (1-5Mbit/s) [24]. Layered coding has emerged as a viable solution and can provide an adaptive support for delivering HD streaming content [25] to different downloading capacities on peers. For layered P2P streaming system, Xiao, *et al.*, [11] proposes a RTT (Round Trip Time)-based method in the OCals overlay construction protocol , where a new peer makes a selection for each layer based on comparisons with exact value(medium and minimum RTTs). Nguyen, *et al.*, [26] proposes a quality-based method in Chameleon where peers are classified into classes based on their download bandwidth capacity and each peer calculates the average quality level it has perceived so far periodically. When a peer selects a neighbor, it will choose the candidate whose average quality level is closet to its class. To overcome the effect of the join order and the population percentage of different peer classes to the system performance, subsequently Nguyen, *et al.*, [9] proposes a quality and context-aware method where a preemption rule is the core technique. Through allowing a higher uplink capacity peer to replace a lower capacity peer to be a neighbor of another peer with a certain probability, high capacity peers gear to good locations in the overlay, such as closer to the server than low capacity peers so as to maximize the use of high bandwidth capacity and available layers. However, those approaches have an implicit hypothesis that the download bandwidth capacity of peers determined the received performance of peers and the uplink bandwidth resource in the system is large than the requirements of different peers, which is not fit for resource constraint scenario. As we have known, the feasibility of P2P streaming primarily depends on the scalability of available outgoing bandwidth with the number of participating peers. Since the majority of hosts on the Internet is behind asymmetric connection *(e.g.*, DSL and Cable modem) and has different willingness to contribute, the download capacity is always larger than the upload capacity. So the bandwidth resources are often insufficient to maximize the delivered quality to individual peers [27]. In bandwidth scarce network environments, a fair scheme should allocate resources to individual peers proportional to their contributions (or outgoing bandwidth) rather than their demand (or incoming bandwidth) [16].

Incentive mechanism that provides service differentiation in overlay topology construction for P2P streaming system has been studied in recent years [28-30]. There have typical three categories incentive mechanisms in the literature [16, 30]: payment-based, reputation-based and instantaneous methods. In the payment-based methods, Golle, *et al.*, [31] propose a micro-payment mechanism, where each user can earn rewords if they upload to other users and the rewards can be used for future download. A micropayment mechanism may be a good solution that enables video broadcast users to cooperate. However, this often asks for a centralized broker for coordination, which can hinder the scalability of a peer-to-peer streaming systems [32]. Reputation-based approaches assign a peer a score according to its contribution, subsequently mapping the score to a global rank (or reputation), which determines the peer’s priority in receiving media service [33-35]. Hence reputation-based approaches depend on the history of a peer’s contribution to the P2P network [34]. Reputation-based methods are suitable for asynchronous systems such as VoD and file sharing applications where contribution and reward do not need to happen simultaneously and peers stay in the system long enough to build adequate reputation. In the context of live P2P streaming, measurement study [36] has shown that the peer behavior is very dynamic and the median session time of peers are very short (i.e. more than 50% of peers are in the system for less than 106s). It is essential to design an incentive mechanism that considers instantaneous contribution and demand. Instantaneous methods have been proposed [16, 37] and incorporated into peer selection. Pianese, *et al.*, [35] proposed an optimistic tit-for-tat peer selection policy based on pair-wise incentives. This is a direct reciprocity approaches and has been widely adopted in P2P file sharing applications for its simplicity and apparent fairness. However the key aspect that distinguishes live streaming applications from file download application is the stringent timeliness and real-time requirements Tit-for-tat model severely limits the amount the bandwidth the resource-poor peers can receive and caused these resource-poor peers would not participate in the system due to the poor quality. Researchers have found through measure that up to 80% of the peers on the Internet are resource-poor[38].

Therefore direct reciprocal mechanisms are not applicable for resource constraint scene in P2P streaming system. Taxation-based model has been introduce into P2P streaming system by Yang-Hua Chu, *et al.*, [38]. By leveraging resource-rich peers contribute more bandwidth to the system and subsidizing for the resource poor peers, the collective social welfare of the system can be maximized. Taxation scheme is an indirect reciprocal mechanism and relaxes strict contribution-based differentiation. The scheme can be used in a wide range of scenarios such as highly heterogeneous and asymmetric peers' bandwidth and bandwidth resource scarce environments. Sung, *et al.*, [15] incorporate the model into tree-based overlay construction and Hu, *et al.*, [17] focus on data scheduling and integrated the tax-based scheme into layer subscription and layer scheduling with SVC in P2P streaming. And the more related to our work, Magharei, *et al.*, [16] combine the mechanism with random neighbor selection policy to construct mesh topology. Similar to [16], we use the linear taxation model to constraint the entitled rate peers receive, subsequently map the rate to number of connections (*i.e.*, number of parents). However, there are three main differences between our work and [16]. Firstly, we don't make the assumption that the incoming bandwidth of each peer is large or equal to streaming bandwidth. Since High Definition video with video rate between 1-5Mbps become popular nowadays on the Internet, the incoming bandwidth of some Internet users may less than the streaming bandwidth. This implies that the incoming rate of peers is limited not only by the streaming rate, the entitled rate, but also by the download bandwidth a peer owns. Secondly, different from random neighbor selection method in [16], we use QoS-aware selection policy, which is similar to class-based selection mentioned in [26]. We use the uplink bandwidth and source-to-peer delay ratio as the selection metric. This difference is due to specific requirements for MDC and SVC encoded video. For MDC encoded video, the video signal is split into multiple substreams where each of the substreams is decodable in a stand-alone fashion. Therefore random peer selection can work well in spite of the different interest of heterogeneous peers. However SVC encodes video into correlated layers and the enhancement layer(s) would become useless for the receiver if the base layer is lost. Hence peers with different received capacity prefer to establish links with peers possessing the same capacity with similar interests in terms of the received bandwidth so as to expand the scope of data share. What' more, the stringent timeliness and real-time requirements of P2P streaming applications cause us to consider the data delivery latency. Thirdly, in [16], the authors use centralized approach for state collection and reporting through a bootstrap peer to determine the entitled rate a peer has. Considering the single point of failure and the scalability requirements in P2P streaming applications, in this paper we provide a distribute policy to determine the connection number a peer is eligible to receive.

3. The Contribution and QoS-aware Neighbor Selection Approach

In this section, we will illustrate our scheme from three aspects: how many peers should a new peer select as parents? Which peers should be selected as parents? And when should a new parent be selected? [12].

In P2P streaming applications, bandwidth is the bottleneck resource. And there are two key problems to design in neighbor selection policy. One is the number and the other is object of selected neighbors. In this section, we firstly describe the assumption in Section 3.1 and then considering the complete knowledge of the network topology and user pool, we present the bandwidth distribution policies in resource scarce environment and combining with the bandwidth-degree condition [13], we transferred the entitled bandwidth a peer received to its incoming degree and uplink capacity to its outgoing degree in Section 3.2. and proposed our distributed variable QoS-aware neighbor selection algorithm in Section 3.3 Since the quality of peer is impacted by the join order of peers of different peer classes and the percentage in population, we proposed a preemption rule, and the high capacity peers replace a lower capacity peers near the source server in Section 3.4.

3.1. System model and Assumptions

In this paper, we consider a P2P streaming session with one streaming source and multiple participating receivers. A subset of the receivers retrieve the media contents directly from the source ,while the other retrieve the media contents from one or more receivers. When a new peer joins the session, it contacts the track server to get a list of active peers in the session as candidate upstream neighbors. Then according to neighbor selection algorithm, the new peer picks one or more peers from the candidate list to request connection setting. This constructs the initial mesh overlay topology for the streaming session. We model the overlay network as a directed graph G= (V, E), where V is the set of vertices, representing the participating peers and E=V*V is the set of direct arcs, representing the overlay links. Let S be the streaming source, T be the set of receivers and N is the total number of receivers in the session. Then we have $V = S \cup T$ .For any peer p_j in G, the forward capacity F_j and receive capacity R_j respectively represent the upper bound bandwidth that peer p_j can contribute to and receive from the P2P system, which are static host characteristics that can be easily determined by the software agent running on the peer host. For any arc $< p_j, p_h >$ in G, the latency of the arc is the underlay unicast path latency from peer j to peer h in the physical network. We let L_{jh} represent the latency of $< p_j, p_h >$. Then we have $L_{jh} = L_{hj}$. The latency between peers can be obtained by measurement (e.g. the round trip time). We use $P_k=\{p_1,p_2,..,p_m\}$ to represent the candidate set a new peer p_k returned from the tracker server.

We make some assumptions about the system behavior. Firstly, we assume that the congestion won't be happened at the core of the network and mostly happen at the access links on the Internet, which means peers have finite upload and download bandwidth and the core network isn't the performance bottleneck. Secondly, we assume that peers are altruistic, which means that every peer is willing to contribute its bandwidth to upload data to other peers and there is no free-rider in the system. Thirdly, all peers are strategic, i.e., only if they see clear benefits from cooperation they will contribute more resource.

3.2. Taxation-based Bandwidth Distribution Policies

In P2P network, the connectivity between peers is established by neighbor selection method. There are three problems in neighbor selection design, including how many peers should a new peer select as parents? Which peers should be selected as parents? And when should a new parent is selected? Section 3.2 and 3.3 will answer these problems.

Bandwidth-degree condition has been proposed by Nazanin and Reza [13] to maximize the utilization of both incoming and outgoing access link bandwidth of all peers. By introducing the concept of bandwidth-per-flow(bwpf), which implies that all connections in the overlay should have roughly the same bandwidth, the heterogeneous and asymmetric incoming and outgoing access link bandwidths of individual peers (and the source) can be translated to their incoming and outgoing degrees, respectively. In this section, inspired by [13], we first propose the contribution aware bandwidth allocation scheme and than determine the number of neighbors according to bandwidth-degree condition.

To support the contribution awareness, each participating peer adopts a generic cost function [38] to determine the incoming bandwidth:

$$r_d = \frac{1}{t} f_d + \frac{t-1}{t} \sum_{j=1}^{N} \frac{f_j}{N} \quad (1)$$

Where r_d is the bandwidth peer p_d is entitled to receive, t is the tax rate, f_d is the actual bandwidth peer pd contributes to the system, and N is the number of participating peers. As shown in (1), r_d is the sum of two terms. The first term is the minimum bandwidth a peer entitled to receive due to its own contribution (f_d). The second term represents the average residual bandwidth shared among participating peers. The residual bandwidth resources rely on the session group size and the amount of aggregate available resources in the system. Here, the concept of tax rate different from the traditional literature in linear taxation where tax rate is no more than 1. In (1), the tax rate (t) is at least 1 to balance the supply and demand for resource in the system. If t = 1, the tax scheme become a simple tit-for-tat policy. If t>1, a peer contributes more than it receives ($f_i > r_i$), the leftover bandwidth can be shared evenly among all the participators.

Linear taxation has been widely studied in the optimal income taxation literature. And a fixed and linear taxation scheme in P2P streaming systems was proved effective and robust under a variety of peer environments [38], especially the peer bandwidth heterogeneous scene. In this paper the tax rate is a configured parameter and known to each participating peers. For our later simulation, we pick the tax rate of 2.

In practice, there are two issues that affect the entitled rate a peer received. One is the maximum stream bandwidth, e.g. the rate the source sends video data. When the aggregate incoming bandwidth of a peer reaches the maximum stream bandwidth, it doesn't need extra incoming bandwidth. This suggests that the incoming bandwidth of peers is limited by r_s, where an r_s represents the maximum stream bandwidth. Peers whose incoming bandwidth is large than or equal to r_s is in saturated state [16]. The other is the download capacity R_j, which is the upper bound bandwidth a peer can received. This implies that the entitled incoming bandwidth should not large than the upper bound bandwidth. Therefore the entitled incoming bandwidth of a peer should be restricted. We can revise (1) to satisfy these two issues as follows:

$$r_d = min\{(\frac{1}{t} f_d + \frac{t-1}{t} \sum_{j=1}^{N} \frac{f_j}{N}), r_s, R_d\} \quad (2)$$

As bandwidth-degree condition, the entitled incoming degree (also called the entitled number of neighbors a peer received) can be computed as formula (3). At the same time, In order to avoid over-estimating the amount of allocated resources to each peer, we always use the floor of the resulting value from (2).

$$inDeg_d = \left\lfloor \frac{r_d}{bwpf} \right\rfloor \quad (3)$$

Where bwpf is per-flow-bandwidth, which is a configurable parameter.

From (2), we can see that to compute the entitled incoming bandwidth, peers have to periodically approximate the total resource utilized (*i.e.*, $\sum_i f_i$) and the number N of peers. What' more, f_i is the actual contribution bandwidth of i at a given instant, it may vary during the session. Hence it is necessary to collect these system-wide parameters periodically. In [13], the authors present a simple centralized approach for state collection and report through a bootstrap node. However, considering the effect of single point of failure and the scalable request in P2P streaming system, we use a distributed bandwidth allocation method. When a new peer joins the session, it calculates its entitled incoming bandwidth and entitled incoming degree based on (4) and (3). Then the peer adopts the QoS aware neighbor selection algorithm described in Section 3.3 to determine its parents. Since using (4) to compute reduces the original connection number of peers, to effectively use the residual system bandwidth and make peers who have not received maximum incoming bandwidth achieve their entitled degree, we need to periodically search new neighbor through gossip-based peer discovery [12, 39]. Section 3.3 describes the process.

$$r_d = min(\frac{1}{t} f_d, r_s, R_d) \quad (4)$$

3.3. A Distributed Neighbor Selection Algorithm

There are two basic QoS criteria for upstream neighbor selections: relative delay and available bandwidth between two peers. Relative delay between any two peers can be estimated in a scalable fashion such as using network coordinates-based solutions, e.g. Vivaldi [40]. However, estimating available bandwidth between two peers requires end-to-end measurement, such as sending a series of probing packets; it will increase joining time of new peers. What's more, the dynamics of bandwidth variations require each peer to periodically estimating the available bandwidth from all other peers through measurement and further increase the probability of interference among different measurements with the number of peers joining the overlay. So it limits the scalability of the system.

Some researches propose to use delay between peers as the main selection criteria. However there isn't any relativity between delay and bandwidth. While these parent peers selection strategies minimize associated network load, they may not provide sufficient bandwidth to individual peers because delay is often not a good indicator for available bandwidth between two peers [12]. At the same time, in mesh-based P2P streaming systems, bandwidth aggregation of multiple parents guarantee the robustness and a certain streaming rate while coming with the cost of source-to-end delay [20]. Hence we should find a tradeoff between minimizing overall delay and maximizing delivered bandwidth to each peer.

Peers' outgoing bandwidth is an upper bound for available bandwidth and only affected by peers' access link capacity. It is the static host characteristics that can be easily determined by the software agent running on the host and enables the receiver to roughly classify different parents. So in the paper, we use uplink bandwidth instead available bandwidth as one estimate criteria. Other criterion is source-to-peer delay. It can be defined as follow (5).

$$D_j^s = \min_{i:i \in C_j}(D_i^s + L_{ij}) \quad (5)$$

Where D_i^s represents the delay of peer p_j along the overlay paths, which is the minimum of the delays of peer j's parents plus the connection delay (L_{ij}) between p_j and its parents. C_j is the candidate neighbor list of peer p_j.

In this paper, we identify potential good parents for a receiver using the relative utility. The utility of a candidate peer p_i for a receiver p_j is a function of their relative network distance from the source and the outgoing bandwidth of the candidate peer. The utility function can be describes as following:

$$u(p_i, p_j) = f(F_i, D_j^s) \quad (6)$$

Here we consider a simple utility function below. The metric a new joining peer p_j chooses a peer p_i as its neighbor is directly proportional to peer p_i outgoing bandwidth and inversely proportional to the latency between the source and new peer along the overlay path crossing peer p_i.

$$u(p_i, p_j) = \frac{F_i}{D_j^s} \qquad (7)$$

After determined the number of neighbor (in Section 3.2) and the selected object, another design parameter we should determine is a probability of selecting a peer at random, which can enhance the connectivity and robustness of overlay [41, 9]. We use α to represent the probability and the value chosen by experiments. However, how to choose a good α in general cases is an important issue, and we leave it as our future work.

We use k_p to present the incoming degree of each peer. $K_d = \max(5, inDeg_d)$.Here k_p is no less than 3 to insure the connectivity of peers of low bandwidth. We select αk_p peers according to their utility and select $(1-\alpha)k_p$ peers randomly. We are now ready to present the complete neighbor selection method. There have three events trigger the neighbor selection process.

3.3.1. Initial Phrase: When a new peer joins the system, it connects the well-known rendezvous point with necessary information such as its uplink bandwidth capacity and its IP address. The rendezvous point maintains a global peer view that contains up to N records where each record represents an active peer's information: 1) IP address, 2) source-to-peer delay, 3) number of received layers, 4) timestamp when the record was last generated by a peer, 5) Fi and Ri. The rendezvous peer initiates the peer discovery process and randomly pick an active peers list to the new peer and log the source to this new peer's delay, which is calculated as. $T_{diff} = T_{recv} - T_{sent}$.Here T_{sent} is the time the new peer sent joining request and T_{recv} represents the time that the rendezvous point received the request. Random selection tends to balance the overhead among all the peers in the system. After received the active peers list, the new peer launch the peer selection process described as Algorithm 1.

```
Algorithm 1 Initial Neighbor Selection-Request
AL: The active peer list in rendezvous peer;
pk: the request joining peer;
CL: Candidate list return from the rendezvous peer;
PNL: the parent neighbor list;
m : the number of candidate peers;
TL: temporary peer list;
CL<-P2PSubstrateLookUp(AL,random);
for pi in CL do
    CalculateUtility(u(pi,pk));
end for
TL<-Sort all peers in CL according to u(pi,pk) in descending order;
//first selected (1-a)kp  QoS aware peers from TL
K=0;
Index=0;
for pi in TL do
   index++;
```

```
     SendConnectionRequest(pi, pk);
     if(accept(pi)) then
        Insert(PNL,pi);
        k=k+1;
     end if
     if (k==(1-a)kp) then
        break;
     end if
  end for
  //if two few neighbors can accept the new joining peer
  if(k<(1-a)kp && index>=m) then
       Wait();
       RequestToRendezvousPeer();//request for new candidate list
  end if
  if(index<m && (m-index)>akp) then
       SelectedRandomPeer(TL, index,m-1,akp)  //Then randomly selected
                           akp peers
  end if
```

3.3.2. Improvement in Performance: Peers periodically improve their video quality. In our bandwidth resource allocation, we use formula (4) to calculate the initial incoming bandwidth of peers. There are $(1-\frac{1}{t})\sum_i \frac{f_i}{N}$ bandwidths resources left which will be further distribute. Peers locate other peers using gossip-like message distribution mechanism. For example, peer p_j exchange neighbor list periodically with its neighbors and during each update period peer p_j will select a new peer from its neighbor's neighbor list and establish connection. The algorithm describes as following.

```
  Algorithm 2 Updating Neighbor Selection-Request
  PNL: the parent neighbor list;
  PPNL: the neighbor's neighbor list;
  TL: temporary peer list;
  While the session is not over do
    PPNL= ExchangeNeighborListMap(pk,PNL); //get the neighbor's neighbor
                             list
    for pi in PPNL do
       CalculateUtility(u(pi,pk));
    end for
    TL<-Sort all peers in CL according to u(pi,pk) in   descending order;
    for pi in TL do
       SendConnectionRequest(pi, pk);
       if(accept(pi)) then
          Insert(PNL,pi);
          break;
       end if
  end for
end while
```

3.3.3. Replacing a poorly-performance parent*:* When available bandwidth from an existing parent is significantly reduced for a long time or a parent leaves the session, the receiver can select another peer after a threshold. The process is just like B). We omit here.

For each candidate, after receiving a request, it will respond according to its upload capacity. If the current number of children is below its maximum number of children, it will accept the request. Otherwise a local preemption policy (Section 3.4) will be used. The response algorithm is describes below.

```
Algorithm 3 Neighbor Selection-Response
m: the current number of child peers;
Pk: the peer sending connection request;
Ph:the peer receiving connection request from pk
CNL: the child neighbor list;
ReceiveConnectionRequest(pk);
If(m<outDegh) then
   SendReply(ph, pk ,ACCEPT);
   Insert(CNL,pk);
else
   LL=SelectPeersWithLowerBandwidth(CNL, pk);
   x=sizeOfList(LL);
   if(x=0) then
       SendReply(ph, pk,REJECT);
   else
     SelectToPreemptionRandom(pk, LL);//local preempt the peer with low
                    bandwidth randomly
     SendReply(ph, pk, ACCEPT);
   end if
end if
```

3.4. Local Preemption Policy

In real network environment, the upload bandwidth of end host is finite, which limit the number of children peers can support. The maximum number of children a peer can service can be calculated as formula (8).

$$outDeg_d = \left\lfloor \frac{f_d}{bwpf} \right\rfloor \quad (8)$$

When a parent peer receive a request for connection and the current number of child peers is less then the outgoing degree (maximum number of child peers), the parent peer accept the request and connection can be established. However if the outgoing degree is fully utilized, then the parent peer compares the upload bandwidth of the request peer with the upload bandwidth of its existing child peers. If there is no child peers' upload bandwidth less than the request peer, the parent peer reject the connection request. Otherwise it randomly selects one from the existing child peers with upload bandwidth lower than the request peer and terminates the connection with that selected peer. The preempted peer has to launch a new peer selection process.

4. Performance Evaluations

In this section we carried our simulation to evaluate the performance of our schemes (Cont-aware QoS-based neighbor selection methods) with two approaches, namely RNS, Cont-Agnostic QoS-based neighbor selection methods. In the RNS scheme, newly arrived peers randomly select fixed number of neighbors. Because it is simple, it has been adopted by a number of streaming protocols and systems [4, 42]. In the Cont-Agnostic QoS-based scheme, peers select fixed number of peers according to their upload bandwidth and end-to-end delay ratio.

4.1. Simulation Setup

P2PTV-sim [43] is an open source chunk-level discrete event-driven simulator written in standard C++ developed within the Napa-Wine project[44].and it is capable of simulating a large scale mesh-based P2P live streaming with single channel. On the basis of P2PTV-sim, we make some extension and conduct a series of extensive simulations to study the impacts of our neighbor selection algorithm.

In the simulation we consider scenarios comprising N = 2000 peers, if not otherwise indicated. The peers can be classified into three types: DSL peers, Cable peers and Ethernet peers. As shown in Table 1, The bandwidth distribution follows an extensive bandwidth measurement from a large scale real-world streaming in [20] .The default uplink bandwidth of source node is 5Mbps. We employ real-world end-to-end latency matrix (2500*2500) measured on the Internet provided by Meridian project [47]. The average end-to-end delay is 39ms. We simply map each node pair in our simulation to each pair in the latency matrix randomly. We use offer/select signaling mechanisms [45] to trade chunks in a quick and efficient way.

Table 1. Bandwidth Ration Distribution Setup

Peer Type	Uplink Bandwidth	Downlink Bandwidth	Percentage
DSL	400Kbps	1Mbps	45%
Cable	800Kbps	2Mbps	40%
Ethernet	1500Kbps	5Mbps	15%

In the simulator, for the evaluation the received video quality, we use H.264 temporal scalable video in our current experiments. This is mainly because H.264 temporal scalable video does not lose any video coding efficiency compared with H.264 single-layer video. Furthermore, we encoded the videos in H.264/SVC with the JSVM software Version 9.18 and integrate SVEF [46] (a scalable video-streaming evaluation framework for H.264/SVC) into P2PTV-sim to evaluate the quality of experience. The 300 frames of the encoded sequence are concatenated multiple times in order to create a 40 second video clip that is used afterwards in our simulations. We adopt one-chunk to one-frame mapping and design a content-aware chunk scheduling method. The total chunk number is 1200. Since the actual design of the chunk scheduling is out of the scope of this paper, readers are suggested to refer our further work [47] for more detail.

The video rate of the encoder r_s is a pre-configure parameter that we vary to enforce different values of the system resource index [48] (RI) computed as (9). Table 2 show the resource index under different video rate.

$$RI = \frac{R_s + \sum_{i=1}^{N} f_i}{r_s * N} \qquad (9)$$

Table 2. System Resource Index Change with the Video Rate

Video Rate(Kbps)	RI	Video Rate(Kbps)	RI	Video Rate(Kbps)	RI
400	1.853	700	1.059	1000	0.741
500	1.482	800	0.926	1100	0.674
600	1.235	900	0.823	1200	0.618

In RNS algorithm, the fixed number of neighbors is set 20. And in cont-agnostic QoS method and our proposed algorithm, the random factor α is set 0.1. The update period of connection among peers is set 10s.

4.2. Performance Metrics

We define the following evaluation metrics that we would use in the analysis:

Chunk Delivery Latency: defined as the delay between the time when one chunk is sent out from the source node and when it is finally arrived at a peer after one or several hop(s). Here we calculate the average delivery latency of all chunks to estimate the real-time performance of different algorithms.

Chunk Loss Ratio: defined as the number of chunks that have never been received by peers over the total number of chunks [49], which includes losses and chunks arrived after the playout deadline. In this paper, we only consider the latter to estimate the playback continuity.

Control Overhead: defined as the ratio of the volume of control messages divided the volume of total distributed chunks in the session. The control messages mainly include exchanging buffer map signals, such as offer, select message and messages that are used to periodically update the relationship between peers.

PSNR (Peak Single-to-Noise Ratio): is a widely adopted objective video quality index and used as a measure of the distortion introduced by the P2P streaming system[50] to reflect the quality perceived by the users (QoE). We use luminance component in the YUV colourspace to compute PSNR, denoted as Y-PSNR. The formula is below: It provides some kind of trade-off between chunk losses and encoding artifacts[41].To access QoE at the receiver, P2P-tv performance should not be evaluated by considering networking indices, such as the chunk delivery delay and chunk loss probability traditionally taken into account by previous work. The effect of loss pattern on video quality integrity must be taken into account since the same average chunk loss probability might induce very different effects on the quality of the reconstructed video.

$$Y-PSNR = 10\log_{10}\left(\frac{\max_I^2}{MSE}\right) \tag{10}$$

Where

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\| I(i,j) - K(i,j) \|^2 \tag{11}$$

I(i,j) and K(i,j) are the value of the luminance component for pixel (i,j) in the original video and the encoded. Reconstructed video respectively; $\max_I = 2^{Nb}-1$ is the absolute maximum value for luminance component. Nb is the bits each component is coded. Here we obtain the average Y-PSNR by averaging over all peers the PSNR per peer. Since that the computation of the PSNR cannot be done in correspondence of a missing frame. We assume that the receiver uses the last correctly decoded frame as reference to compute the PSNR.

4.3. Simulation Results

To estimate the system's network performance and user Quality of Experience, we mainly simulate a stable environment. For the whole session only persists about 40s, so that only a marginal percentage of peer is expected to leave or join the system. So the effect of peer churning is neglected at first. And we will explicitly assess its impact in our future works. When all the nodes join in an initialization period, they persist in the lifetime of the streaming. All results are averaged over at least ten independent simulation runs.

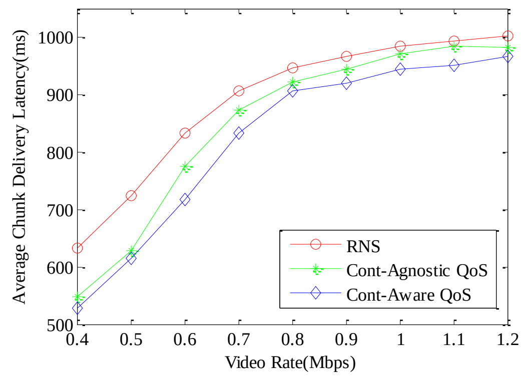

Figure 1. Average Chunk Delivery Latency as a Function of the Video Rate

Figure 1 shows the average chunk delivery latency as a function of the target video bitrate with a certain playout delay (5s) of the three schemes we study. We observe that for all of the three methods, with the increscent video bitrate, the average chunk delivery latency increase. That's because as the video coding rate increases, the chunk size increases as well and therefore the diffusion of a given chunk takes longer. Cont-aware QoS scheme outperforms the other two schemes as the video rate grows. This can be attributed to the nature of cont-aware QoS scheme which picks up peers with lower source to end delay and high bandwidth and thus more branching occur near the source.

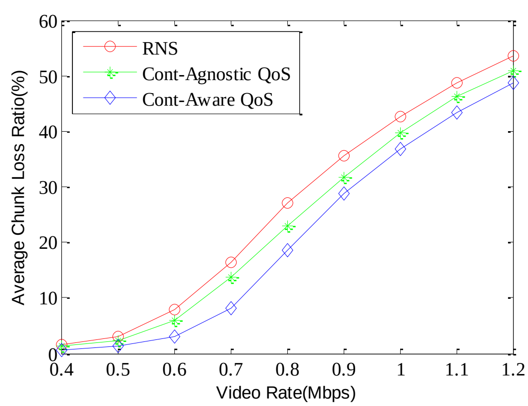

Figure 2. Average Chunk Loss Ratio as a Function of the Video Bitrate with N=1000, Mc=1200

Figure 2 shows the average chunk loss ratio as a function of the target video bitrate with a certain playout delay (5s). When a chunk is not received within its playout time, it is deemed “lost”. When the video bitrate is low and the system is underloaded, *e.g.*, video bitrate is no more than 0.6Mbps; the chunk loss rate is less then 6%. However when the video rate grows, the system’s total requirement increases while the total supply unchanged, that’s the total load increases. As a result, the chunk delivery delay becomes longer which causes the number of postponing chunk increment at a given target playout delay. So the number of lost chunks depends on the media bitrate. The loss ratio in our proposed scheme is lower than the two others.

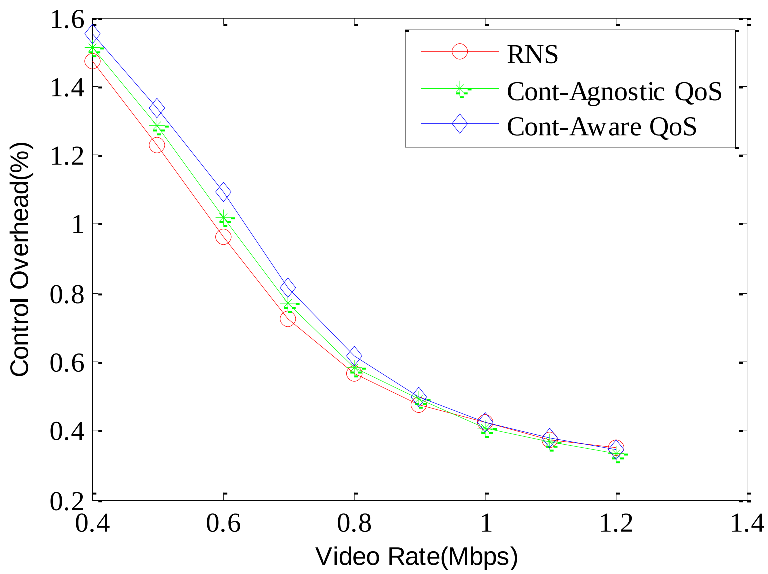

Figure 3. Control Overhead as a Function of the Video Bitrate with N=1000, Mc=1200. (Control overhead = Control traffic volume/Video traffic volume at each peer)

Figure 3 depicts the normalized control traffic as a function of the video rate. Not surprising the overhead decreases with an increase of the video rate, but as compared to video traffic, the control traffic is essentially minor, even with 0.4Mbps video rate (less than 2% of the total traffic).From the Figure 3, we observe when the video rate is low, the control overhead of our proposed scheme is slightly larger than the others for it has to update neighbor relationship periodically so as to reach the entitled degree (the number of parent peers’ connection). However, with the video rate increments, the proportion of control message comparing with data volume decreases.

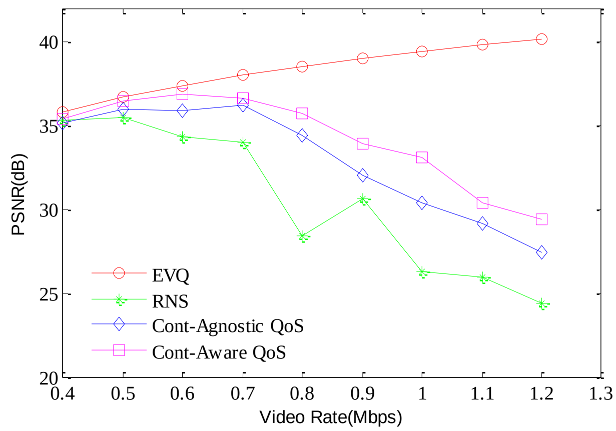

Figure 4. Y-PSNR as a function of the video bitrate with N=1000,Mc=1200 and playout delay=5s

Figure 4 provides some insight into how chunk losses affect Y-PSNR values of reconstructed video. In the figure, we use Encoded Video Quality (EVQ) at the source as reference. The effects introduced by the distribution system on the perceived quality can be grasped by comparing the received PSNR with the PSNR at the source with no chunk loss. In figure 4, we observe when the system is in underloaded environment, with the incensement of video rate, the video quality of the three schemes grows adjacent to EVQ. However when the bitrate is 0.7Mbps, the system resource index is about 1 and the system is in critical region, the video quality begins to decrease. In random scheme, under overloaded environment, the video quality fluctuates due to the character of randomness. The quality descends more smoothly in Qos aware schemes. And our proposed can achieve a higher quality; the Y-PSNR is higher from 0.45dB to 1.97dB. Note that the PSNR scale is logarithmic in dB, so that a difference of 1.97dB corresponds to a very large improvement of the QoE.

5. Conclusion and Future Works

In this paper, we have explicitly considered bandwidth resource scarce environments and proposed a contribution and QoS-aware neighbor selection approach in live streaming mesh, formed by peers with heterogeneous bandwidths. By incorporating linear taxation model into bandwidth resource allocation, the incoming degree of peers is proportional to peers' uplink capacity and the average residual resources of systems. And peers with high bandwidth and low source-to-peer delay will be selected as parents in preference. We conducted extensive simulations to illustrate the network performance and video quality at the receivers. Evaluations show that the contribution and QoS aware neighbor selection method achieves high efficiency and high compliance without incurring significant overhead.

As a future work, we plan to pursue this work along the following directions: 1) we would further research into the impact of different tax rate and extend the fixed linear taxation-model to feedback-based method so as to adjust the tax rate according to the system's variable resources.2) flash crowds are common in live streaming, which would arise acute churn when a large amount of users joining the session when a popular program is about to begin and leaving immediately after the program finish. So it is essential to investigate how flash crowds impact our scheme and find out ways to deal with this problem.

Acknowledgements

This paper is supported in part by the major national science and technology special projects in China （Grant No. 2010ZX03004-003-03） and The National Basic Research Program of China (973 Program) (Grant No. 2009CB320906).

References

[1] L.Yong, "A Survey on Peer-to-Peer Video Streaming Systems", Journal Peer-to-Peer Networking and Applications, vol. 1, no. 1, **(2008)**.
[2] Y. Chu, S. G. Rao and S. Seshan, "A Case for End System Multicast", Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, **(2000)** June 17-21, Santa Clara, CA, USA.
[3] C. Miguel, P. Druschel and K. Anne-Marie, "Splitstream: High-Bandwidth Multicast in Cooperative Environments", Proceedings of the nineteenth ACM symposium on Operating systems principles, **(2003)** October 19-22, Bolton Landing, New York, USA.
[4] Z. Xinyan, L. Jiangchuan and L. Bo, "Coolstreaming/Donet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming", Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies, **(2005)**.
[5] PPLive, www.pplive.com.
[6] SopCast, www.sopcast.org.
[7] UUSee, www.uusee.com.
[8] C. Laizhong, J. Yong and W. Jianping, "Employing Qos Driven Neighbor Selection for Heterogeneous Peer-to-Peer Streaming", 2011 IEEE International Conference on Communications (ICC), **(2011)** June 5-9, Kyoto, Japan.
[9] N. A. Tuan, L. Baochun and F. Eliassen, "Quality- and Context-Aware Neighbor Selection for Layered Peer-to-Peer Streaming", 2010 IEEE International Conference on Communications (ICC), **(2010)** May 23-27, Cape Town, South Africa.
[10] J. Ghoshal, B. Ramamurthy and L. Xu, "Variable Neighbor Selection in Live Peer-to-Peer Multimedia Streaming Networks", TR-NUL-CSE-2007-021, UNL Department of Computer Science and Engineering, **(2007)**.
[11] X. Xin, S. Yuanchun, Z. Baopeng and Y. Guo, "Ocals: A Novel Overlay Construction Approach for Layered Streaming", 2008 IEEE International Conference on Communications (ICC '08), **(2008)** May 19-23, Beijing, China.
[12] R. Rejaie and S. Stafford, "A Framework for Architecting Peer-to-Peer Receiver-Driven Overlays", Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video, **(2004)** June 18, Cork, Ireland.
[13] N. Magharei and R. Rejaie, "Prime: Peer-to-Peer Receiver-Driven Mesh-Based Streaming", J.IEEE/ACM Transactions on Networking (TON), vol. 17, no. 4, **(2009)**.
[14] H. Schwarz, D. Marpe and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/Avc Standard", J. IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, **(2007)**.
[15] Y. W. E. Sung, M. A. Bishop and S. G. Rao, "Enabling Contribution Awareness in an Overlay Broadcasting System", J.IEEE Transactions on Multimedia, vol. 9, no. 8, **(2007)**.
[16] N. Magharei, R. Rejaie and Yang Guo, "Incorporating Contribution-Awareness into Mesh-Based Peer-to-Peer Streaming Systems", J.Peer-to-Peer Netw. Appl., **(2010)**.
[17] H. Hu, Y. Guo and Y. Liu, "Peer-to-Peer Streaming of Layered Video: Efficiency, Fairness and Incentive", J. IEEE Transactions on Circuits and Systems for Video Technology, vol. 99, **(2011)**.
[18] A. J. Ganesh, "Scamp: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication", Networked Group Communication, Springer Berlin / Heidelberg, **(2001)**, pp. 44-55.
[19] J. Chakareski, "Topology Construction and Resource Allocation in P2P Live Streaming", J. Intelligent Multimedia Communication: Techniques and Applications, vol. 280/2010, **(2010)**.
[20] R. Dongni, Y. T. H. Li and S. H. G. Chan, "On Reducing Mesh Delay for Peer-to-Peer Live Streaming", The 27th IEEE Conference on Computer Communications (Infocom 2008), **(2008)** April 15-17, Phoenix, AZ, USA.
[21] C. Yang-hua, S. G. Rao and S. Seshan, "A Case for End System Multicast", J.IEEE Journal on Selected Areas in Communications, vol. 20, no. 8, **(2002)**.
[22] S. Banerjee, B. Bhattacharjee and C. Kommareddy, "Scalable Application Layer Multicast", Proceedings of ACM Sigcomm 2002, **(2002)** August 19-23, Pittsburgh, Pennsylvania.

[23] D. A. Tran, K. A. Hua, T. Do, "Zigzag: An Efficient Peer-to-Peer Scheme for Media Streaming", The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, (Infocom 2003), **(2003)** March 30-April 3,San Francisco, CA, US.
[24] A. Couto da Silva, E. Leonardi and M. Mellia, "Chunk Distribution in Mesh-Based Large-Scale P2P Streaming Systems: A Fluid Approach", J. IEEE Transactions on Parallel and Distributed Systems, vol. 99, **(2010)**.
[25] L. Zhengye and S. Yanming, "Layered Video for Incentives in P2P Live Streaming", ACM Sigcomm'07, **(2007)** August 27-31, Kyoto, Japan.
[26] N. A. Tuan, L. Baochun and F. Eliassen, "Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding", **(2010)** March 14-19, San Diego, CA, USA.
[27] L. Bo, X. Susu and Q. Yang, "Inside the New Coolstreaming: Principles, Measurements and Performance Implications", The 27th IEEE Conference on Computer Communications (Infocom 2008), **(2008)** April 15-17, Phoenix, AZ, USA.
[28] S. Shijun and L. Sanli, "P2P-based Video-on-Demand Systems: A Survey", J. Chinese Journal of Computers, vol. 33, no. 4, **(2010)**.
[29] Q. Fenlin, L. Ju, "Key Technologies in P2P Media Streaming", J.Chinese Journal of Electronics, vol. 39, no. 4, **(2011)**.
[30] X. Su, S. K. Dhaliwal, "Incentive Mechanisms in P2P Media Streaming Systems", J. IEEE Internet Computing, vol. 14, no. 5, **(2010)**.
[31] P. Golle, K. Leyton-Brown and I. Mironov, "Incentives for Sharing in Peer-to-Peer Networks", J. Electronic Commerce, **(2001)**.
[32] L. Jiangchuan, S. G. Rao and L. Bo, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast", J. Proceedings of the IEEE, vol. 96, no. 1, **(2008)**.
[33] T. Guang and S. A. Jarvis, "A Payment-Based Incentive and Service Differentiation Scheme for Peer-to-Peer Streaming Broadcast", J. IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 7, **(2008)**.
[34] A. Habib and J. Chuang, "Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming", J. IEEE Trans on Multimedia, vol. 8, no. 3, **(2006)**.
[35] F. Pianese, D. Perino and J. Keller, "Pulse: An Adaptive, Incentive-Based, Unstructured P2P Live Streaming System", J. IEEE Transactions on Multimedia, vol. 9, no. 8, **(2007)**.
[36] S. Agarwal, J. P. Singh, A. Mavlankar, P. Baccichet and B. Girod, "Performance and Quality-of-Service Analysis of a Live P2P Video Multicast Session on the Internet", 16th International Workshop on Quality of Service (IWQoS 2008), **(2008)** June 2-4, Netherlands.
[37] M. K. H. Yeung, Y. -K. Kwok, "On Game Theoretic Peer Selection for Resilient Peer-to-Peer Media Streaming", J. IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 10, **(2009)**.
[38] Y. Chu, J. Chuang and H. Zhang, "A Case for Taxation in Peer-to-Peer Streaming Broadcast", Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems, Portland, **(2004)** August 30-September 3, Oregon, USA.
[39] W. P. Yiu, J. Xing and S. H. G. Chan, "Challenges and Approaches in Large-Scale P2P Media Streaming", J. IEEE Multimedia, vol. 14, no. 2, **(2007)**.
[40] R. Cox F. Dabek, F. Kaashoek and R. Morris, "Vivaldi: A Decentralized Network Coordinate System", ACM SIGCOMM 2004, **(2004)** August 30-September 3, Portland, Oregon, USA.
[41] R. Fortuna, E. Leonardi and M. Mellia, "Qoe in Pull Based P2P-Tv Systems: Overlay Topology Design Tradeoffs", 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), **(2010)** August 25-27, Netherlands.
[42] P. Vinay, K. Kumar and K. Tamilmani, "Chainsaw: Eliminating Trees from Overlay Multicast", **(2005)** February 24-25, Ithaca, NY, USA.
[43] P2PTVSim, P2Ptvsim Simulator, http://napa-wine.eu/cgi-bin/twiki/view/Public/GRAPES.
[44] NAPA-WINE, European Commission through Napa-Wine Project (Network-Aware P2P-Tv Application over Wise Network), Ict Call 1 Fp7-Ict-2007-1, http://www.napa-wine.eu/cgi-bin/twiki/view/Public.
[45] A. Carta, M. Mellia, M. Meo and S. Traverso, "Efficient Uplink Bandwidth Utilization in P2P-Tv Streaming Systems", IEEE Global Communications Conference (IEEE GLOBECOM 2010), **(2010)** December 6-10, Miami, Florida, US.
[46] A. Detti and C. Pisa, "Svef: Scalable Video-Streaming Evaluation Framework", http://svef.netgroup.uniroma2.it/.
[47] Y. Hongyun, "An Content-Aware Chunk Scheduling Algorithm in Mesh-Based Layered P2P Streaming System", To be published, **(2012)**.
[48] R. Kumar, Y. Liu and K. W. Ross, "Stochastic Fluid Theory for P2P Streaming Systems", 26th IEEE International Conference on Computer Communications (INFOCOM 2007), **(2007)** May 6-12, Anchorage, Alaska, USA.

[49] A. Couto da Silva, E. Leonardi and M. Mellia, "Exploiting Heterogeneity in P2P Video Streaming", J. IEEE Transactions on Computers, vol. 1, no. 1, **(2010)**.
[50] C. Kiraly, L. Abeni and R. Lo Cigno, "Effects of P2P Streaming on Video Quality", Proceedings of the IEEE International Conference on Communications (ICC 2010), **(2010)** September 20-22, Cape Town, South Africa.

Authors

Hongyun Yang, received her PhD from Wuhan University in 2012.Now she is a post doctor at National Engineering Research Center for E-Learning, Central China Normal University. She was a student Member of IEEE, ACM and CCF. Her research interests include cloud computing, P2P streaming and education.

Professor Ruimin Hu, born in 1964, received his PhD from HuaZhong University of Science and Technology in 1994. He became Member of IEEE in 2009 and Senior Member in 2010. He was approved to receive the State Council special allowance. He is Vice chairman of academic committee of Wuhan University, member of the eighth review group for National Natural Science Foundation Committee, head of the Chinese delegation for ISO JCT1/DCMP, member of China Information Technology Standardization Committee (TC28), member of China Security System Standardization Committee (TC100), executive chairman of audio group is the part of MIIT digital video and audio codec expert group, member of the expert group on technology support program of the Ministry of Public Security. He also served as chief scientist and dean of the CSST Research Institute. CSST is the NYSE listed company, the stock code is CSR. His primary research activity is in the domain of Audio and Video Signal Processing, Multimedia Network Communication, Security and Surveillance Technology and Digital Multimedia Content Management and Protection. From 1996 to 2000, Professor Ruimin Hu served as associate dean of Department of Information Engineering, Wuhan Technical University of Surveying and Mapping. From 2000 to 2011, Professor Ruimin Hu served as associate dean of School of Computer, Wuhan University. From 2005 to 2011, Professor Ruimin Hu also served as director of National Engineering Research Center for Multimedia Software. Professor Ruimin Hu has published 139 papers in IEEE Trans., DCC, ICASSP, China's telecommunications and other important international and domestic academic journals and conferences, 15 published papers are indexed by SCI; He applied for 134 national invention patents, 64 are authorized. He presided over multiple demonstration projects of digital security industry and cooperated with 18 NYSE listed companies, produced more than one billion of economic benefits.