Lv ZH, Chen LJ, Wu J *et al.* PROSE: Proactive, selective CDN participation for P2P streaming. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(3): 540–552 May 2013. DOI 10.1007/s11390-013-1354-0

PROSE: Proactive, Selective CDN Participation for P2P Streaming

Zhi-Hui Lv[1] (吕智慧) *Member, IEEE*, Li-Jiang Chen[2] (陈李江), Jie Wu[1,*] (吴　杰), Da Deng[1] (邓　达)
Si-Jia Huang[1] (黄思嘉), and Yi Huang[1] (黄　翼)

[1]*School of Computer Science, Fudan University, Shanghai 20043, China*

[2]*HP China Labs, Beijing 100871, China*

E-mail: lzh@fudan.edu.cn; lijiang.chen@hp.com; {jwu, 11210240047, 082024070, 082024093}@fudan.edu.cn

Received May 11, 2012; revised January 23, 2013.

Abstract Many production peer-to-peer (P2P) streaming systems use content delivery networks (CDN) to protect the user's quality of experiences. Thus, how to efficiently utilize the capacity of CDN (e.g., which peers receive services from the CDN nodes) is a problem of practical significance. Existing solutions adopt a passive, on-demand approach, which is inefficient in utilizing CDN resources. In this paper, we propose PROSE, a simple, novel scheme to achieve proactive, selective CDN participation for P2P streaming. PROSE introduces novel concepts such as choke point expansion nodes/super nodes and leads to efficient, light-weighted, and distributed algorithms to identify and serve these nodes using CDN. Our experimental results show that PROSE achieves at least 10%~25% performance improvement and 2~4 times overhead reduction compared with existing general CDN-P2P-hybrid schemes.

Keywords content delivery network, peer-to-peer streaming, CDN-P2P-hybrid architecture, live streaming, video on demand streaming

1 Introduction

In recent years, a large number of commercial peer-to-peer (P2P) streaming systems, such as PPLive, UUSee, Joost, and Zattoo, have been deployed over the Internet at a large scale. By taking advantage of peer-to-peer technologies, they have greatly reduced the dependence on central content servers. However, as users emerge on the million-scale, P2P applications have started to adopt the mature and widely-deployed content delivery network (CDN) systems to increase their service capacity and reliability. Many P2P players, such as VeriSign, CacheLogic, Grid Networks, Internap, and Joost, have announced their own CDN-P2P services as well[1]. For example, VeriSign is launching a CDN-P2P combination project. Cachelogic has been working with carriers and already hawking their P2P-CDN offerings. Even pure-play P2P companies like Pando have started dabbling with P2P CDNs[2].

Although many P2P application providers have begun to rely on CDN services to improve the quality and reliability of their service, the cost of using CDN is high, especially, with a huge population of users. Relying on CDN to serve every request from a large-scale system is a costly and inefficient solution. Therefore, how to efficiently utilize CDN resources and capacity, in particular how to select which peers to connect to CDN, is an important problem to tackle.

In most existing CDN-aided P2P systems, CDN nodes act as passive content providers. They apply some simple approaches to select peers and serve when a large number of peers request the CDN at the same time. In Livesky[3], a CDN node serves a fixed sub-set of P2P nodes. When a large number of peers visit the CDN node concurrently, it becomes overloaded quickly. Akamai[4] uses a DNS-redirection method to direct a peer node to CDN or other peer nodes in the P2P network. This is a traditional CDN participating P2P solution, which only takes the network distance between peers and CDN nodes into consideration. It ignores the capabilities of peers and their strategic locations, which are important factors to consider for improving system performance and efficiently utilizing CDN capacity.

In this paper, we design a novel scheme: PROSE — Proactive, Selective CDN Participation for P2P Streaming. In our scheme, we introduce a novel concept of choke point expansion node which incorporates the location effect of peers in the system performance. We further identify super nodes according to the peers' upload capacities and their potential abilities to help other peers. By proactively selecting and serving these nodes, we can better improve the system performance

---

Regular Paper
Supported by the National Natural Science Foundation of China under Grant No. 60903164.

and optimize local utilization of CDN nodes at the same time. In order to select such nodes, a light-weighted, feedback-based algorithm is proposed. This algorithm uses CDN to collect statistical information and feedback from its interactive peer nodes and aggregates the information to evaluate the potential capacity of peers and their effects to the system. Both emulation and simulation experimental results show significant performance improvement of our scheme as compared with other general CDN-P2P systems.

Key contributions of this paper are as follows. First, we propose a novel scheme of proactive, selective CDN participation for P2P streaming. We identify choke point expansion nodes and super nodes of P2P streaming systems, and proactively inject CDN resource into them to better utilize CDN capacity and improve system performance. Second, we conduct both emulation and simulation experiments to verify the performance improvement and overhead reduction of our proactive and selective scheme. The results show that our scheme is superior to the general CDN-P2P-hybrid system and pure P2P streaming system, in terms of performance and reliability.

The rest of this paper is organized as follows. In Section 2, we introduce related work of current CDN-P2P-hybrid systems. In Section 3, we introduce PROSE: the basic architecture of our novel proactive, selective CDN participation P2P streaming system. We address our CDN-based algorithm to identify and serve choke point expansion nodes and super nodes in Section 4 and Section 5, respectively. Emulation and simulation experimental results are shown and discussed in Section 6. Conclusions and discussions are given in Section 7.

2 Related Work

2.1 CDN-P2P-Hybrid Systems

In a CDN and P2P hybrid network[2-22], streaming application clients fetch contents from either network (e.g., CDN edge network, P2P network), or both of them. There are two kinds of CDN and P2P hybrid overlay: PAC (Peer-Aided CDN)[4,9,13] and CAP (CDN-Aided P2P)[2,16-18]. Most CDN providers, such as ChinaCache[3], Akamai[4], integrate their CDN with P2P in the PAC manner. In this method, users are mainly served by CDN. P2P overlay is applied to improve user performance and to alleviate the stress of CDN. On the contrary, in the CAP method, most streaming contents are distributed to users through the P2P overlay network. CDN serves as a rescuer for some starving and urgent P2P peers. Most P2P application providers, such as PPLive, UUSee, integrate their P2P network with CDN in the CAP manner. Next, we introduce related work of PAC and CAP.

PAC Mechanism. Akamai, as the world biggest CDN provider, applies the PAC mechanism in its commercial systems. In the Akamai-PAC method, when one peer node's request is directed to the CDN edge server, the edge server can then choose how to handle this request: 1) delivering the content to the peer by itself, or 2) redirecting the request to a P2P network. Akamai uses traditional DNS-redirection approach to forward a peer's request to the nearby CDN node. CDN nodes in this system blindly serve requesting peers; they do not consider which peer they should select and better serve. ChinaCache, the largest CDN carrier in China, is also using a PAC-like approach in the Livesky CDN-P2P-hybrid architecture. The hybrid overlay routing mechanism relies on the CDN DNS redirection mechanism to reach an edge server. The edge server acts as a tracker of P2P overlay to help the requested peer find neighbors. Similar to the Akamai-PAC approach, in Livesky-PAC, an edge server only serves peers within its own region. The edge server does not identify or select peers to provide service. In a flash crowd scenario, if a large number of peers are redirected to an edge server, the edge server will be overloaded quickly.

CAP Mechanism. Currently, in order to improve service quality, many P2P application providers (e.g., PPLive, PPStream, UUSee), either use their own streaming servers or use third-party CDN to reinforce their P2P system. We summarize this method as CDN-aided P2P system (CAP). In this method, CDN is served as a backup system for rescuing. When a peer joins the overlay network, it will be redirected to the common P2P network first by a P2P tracker. If the peer cannot obtain enough content from the P2P network, it will ask the tracker for help and be redirected to a CDN to fetch the required content.

CDN is widely deployed in current P2P streaming systems. In both PAC and CAP mechanisms, CDN serves peer nodes in a request-and-service manner. It blindly uses its capacity to serve as many requesting peers as possible. CDN does not know serving which peer will better improve the performance of the overall P2P system and how long it should serve a particular peer. This problem, referred to as CDN injection, is still an open and unsolved problem. It pertains to efficient utilization of CDN to improve the performance of P2P streaming systems.

2.2 Super Node Selection in P2P Networks

Some researchers have considered how to identify stable or superior nodes in a large-scale P2P environment. Wang *et al.*[23] presented a systematic study on the existence, importance, and application of stable nodes in peer-to-peer live video streaming. They showed that, although the number of stable nodes is not very large, their longer life spans ensure their significance in P2P streaming delivery. Liu *et al.*[24] focused on distilling superior peers in large-scale P2P streaming systems. They discovered some critical factors that may influence the longevity and bandwidth contribution ratio of peers based on runtime traces from UUSee. Mitra *et al.*[25] developed an analytical framework which explains the emergence of superpeer networks on the execution of commercial P2P protocols by incoming nodes. The results show that the number of superpeers produced in P2P depends on the protocol as well as the properties of the joining nodes. These papers identify stable and super nodes in a pure P2P environment; no CDN node is actively involved in the super nodes selection process. Wang *et al.*[23] used the global log data collection and statistical analysis method. Their method is more suitable for finding stable or super nodes over a longer period, not for the CDN real-time environment or for efficient identification of stable or super nodes based on local partial information.

3 PROSE: Proactive, Selective CDN Participation for P2P Streaming

In this paper, we present PROSE, a system that proactively and selectively utilizes the capacities of CDN nodes for P2P streaming.

As described in Fig.1, the basic architecture of PROSE has two layers: the CDN network layer and the P2P network layer. The system employs a CDN-aided P2P (CAP) scheme. CDN proactively joins the P2P network as a strong and stable node to provide content service. Different from previous CAP systems, a CDN node in our system is a passive content provider as well as a proactive content injector that aggressively injects content to selected peers to improve performance of the overall P2P streaming system.

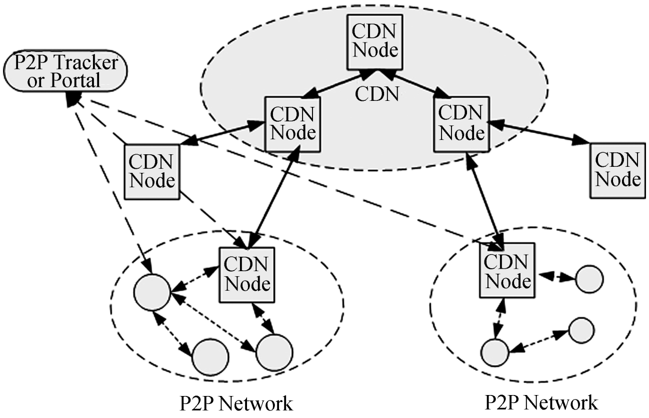

Fig.1. PROSE basic architecture.

3.1 Basic Architecture

In our system, CDN interacts with P2P nodes using P2P protocol. When one peer node enters the overlay network, it gets peer list from a P2P tracker or portal. It connects to peers in the list as its neighbors and obtains content from them. If a peer is struggling for starting up or cannot play smoothly, it will ask the tracker for help and be redirected to connect to a CDN node. When a CDN node receives incoming requests from peers, it decides which peers it should serve and how long to serve according to our P2P node selection algorithm and node replacement strategy. In our system, we ascribe a CDN node and a group of peers who are served by this CDN node to a region. Thus, the overall system is divided into different regions; each of them is led by a CDN node. Within each region, the CDN node can obtain statistical information from its serving peers to identify the system choke point and select potential P2P nodes to provide services.

3.2 Overview of P2P Node Selection Methods

The key mechanism of PROSE is how to select P2P nodes that directly receive content services from CDN nodes. We call the nodes that receive data directly from the CDN as the bridging P2P nodes. PROSE proactively selects bridging P2P nodes not only by their endpoint upload capacities, but also by their strategic locations, which can determine if they can alleviate choke points in the distribution of the P2P network. We refer to the first type of bridging P2P nodes as the super nodes, and the second type as the choke point expansion nodes.

A P2P node is considered as a super node, if it has better properties to serve other peers in the P2P network. Upload capacity, stability, and data availability are three most important properties to consider. In order to select good-property nodes, we propose an approximation algorithm, based on the statistical information collected by the CDN node. To measure a node's upload capacity in run-time is difficult, due to the vulnerability of network and dynamical run-time capacity utilization of a peer. With information collected by CDN, we are able to estimate the upload capacity of a peer in a dynamic P2P streaming environment. Stability is of interest because a more stable node can serve other peers longer and more reliably. A hidden factor that prevents us from identifying super nodes is data availability. If a node has large upload capacity but little data to serve other peers, it is difficult to identify this node using run-time statistical feedback measurements. However, to serve such a super node will potentially yield better performance gain.

A choke point expansion node is a node that expands the choke point of the P2P overlay network by obtaining services from a CDN node and further better serving other common peers. In order to find the choke point expansion nodes, we propose a feedback-based, distributed choke point discovery algorithm to identify topology choke point within local P2P regions. We test P2P nodes around the choke point to evaluate their magnification effectiveness to further select choke point expansion nodes.

We know that P2P streaming system is an overlay network whose topology is dynamically changing and contains many heterogeneous peer nodes. Choke point expansion nodes can dynamically improve the topology structure in multiple various local P2P regions, thus achieving the local optimization effect. When P2P network is in the long-term running process, more and more choke point expansion nodes and super nodes are gradually filtered out and take a greater responsibility. Therefore, it is important to proactively and efficiently find and utilize these two kinds of nodes, specifically served by CDN nodes.

4 Identification of Choke Point Expansion Nodes and Super Nodes

In this section, we introduce our novel algorithm to identify choke point expansion nodes and super nodes. As discussed earlier, most existing methods that integrate CDN with P2P systems use the passive schemes[4,9,13]. That is to say, CDN in those systems serves peers in a help-as-requested manner. They only serve peers who are in urgent and hungry states and request data from CDN. There are two drawbacks of the passive scheme: first, the assistance from CDN is activated only by the requests from peers. In other words, the impact of CDN mainly focuses on rescuing, instead of optimizing. Second, there is no way for CDN to further improve the system performance in a proactive manner. In our design, we provide a mechanism for CDN to discover topology choke point of its local region and proactively select choke point expansion nodes and super nodes to help other common peers.

There are a few challenges for our algorithm: 1) How to identify a topology choke point and choke point expansion nodes in a local region? 2) How to serve the choke point expansion nodes and evaluate the CDN's proactively injection effect of those peers? 3) How to identify super nodes? In order to solve these problems, CDN is applied to collect a small amount of critical statistical information from peers it serves.

4.1 Identifying Choke Point

We propose a distributed, feedback-based approach to identify choke point expansion peers within a region. Through proactively serving these choke point expansion peers, we can improve the performance of the entire P2P streaming system.

Before figuring out the choke point expansion nodes, we should be able to identify where the choke point is. Whenever we find a choke point in a P2P streaming network, we are able to locate choke point expansion nodes around it. Through providing services to peers in its local region, each CDN has a partial view of the total system. By collecting information from interacting peers, the CDN can identify choke point of its local region and further identify choke point expansion nodes. The idea is to collect blaming reports from insufficient peers. Whenever a peer connects to CDN to request data, it indicates that some of its neighbors may have problem of uploading sufficient data to it. By aggregating these blaming reports, we are able to identify a weakest spot of the local region. Around this weakest spot, there is a largest number of starving peers requesting CDN for help. We call this spot the choke point of the region. Take the situation described in Fig.2 for example, nodes 2, 6 and 8 are starving peers requesting data from CDN. They report their blamed neighbors, {4, 5}, {3, 4}, {4}, respectively, when they connect to CDN. By aggregating these reports, we find out that node 4 is reported most times from its neighbors. It indicates that node 4 is the weakest or key spot within this region. Therefore, we say there is a choke point around node 4.

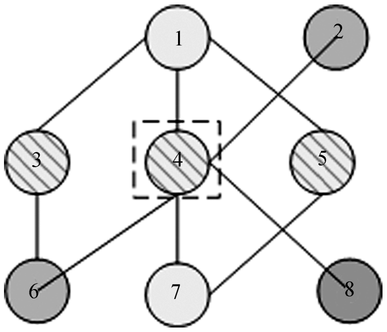

Fig.2. Example of identifying choke point expansion nodes.

Information Collection Methodology. When a peer i connects to a CDN and requests data, the CDN will ask peer i to provide some simple statistical information. Every peer node does not have to periodically report all kinds of information to its regional CDN, just the information when they request data blocks from CDN in some cases, as described in Table 1. Every report must include peer ID and time stamp. A common peer node is expected to report the following: 1) requested block-list and current total stored block-list; 2)

Table 1. Peer Node Report

Category	Content	
Peer basic information	Peer ID, time stamp	
Common peer node report	1. Requested block-list and current total stored block-list 2. Neighbor-list, the current uploading amount to every neighbor and blamed neighbors	
Choke point expansion node report	Decide to serve	3. Buffer-map 4. Playing point
	Decide whether continue serving	5. Current uploading amount to every neighbor 6. Buffer-map 7. Upload bandwidth usage 8. Playing point

neighbor-list, the current uploading amount to every neighbor and blamed neighbors. We cannot directly get the value of peer i's current uploading rate $U_i(t_0)$. We use NB_i to present peer i's neighbor list, and denote the uploading amount from i to its neighbor j at time t_0 by $U_{ij}(t_0)$. The value of $U_i(t_0)$ is computed as (1)[22]: the total uploading amount in a period of time Δt divided by the time Δt:

$$\begin{aligned} U_i(t_0) &= \frac{\mathrm{d}(U_i)}{\mathrm{d}t} \\ &= \lim_{\Delta t \to 0} \frac{\sum_{j \in NB_i} U_{ij}(t_0) - \sum_{j \in NB_i} U_{ij}(t_0 - \Delta t)}{\Delta t}. \end{aligned} \tag{1}$$

If one common peer node is selected as the choke point expansion node to be served by CDN, it will report buffer-map, and playing point. When the choke point expansion nodes receive data from CDN, they have a better upload ability to serve more peer nodes, which realizes some amplification and expansion effect. When CDN decides whether to continue serving this choke point expansion node, CDN will let it report: current uploading amount to every neighbor, buffer-map, upload bandwidth usage, and playing point. In Table 1, statistical information items 1∼4 are also needed when identifying super nodes.

In a practical system, instead of serving all peers, CDN is only connected by a fraction of peers who cannot play smoothly. Therefore, the amount of information it needs to collect is modest. On the other hand, uploading small pieces of information does not affect the performance of the requesting peers, because the demand of those peers is a high download rate.

Information Aggregation. CDN aggregates all the information after each short period of time t. We denote by $S_{\rm req}$ the set of all peers connected to CDN to request data within time t, and by $S_{\rm nb}$ the set of peers who are in the neighbor list of node i[22]. Thus,

$$S_{\rm nb} = \bigcup\nolimits_{i \in S_{\rm req}} L_i. \tag{2}$$

We assign a weight w_j to each $j \in S_{\rm nb}$ as follows:

$$w_j = \sum_{j \in S_{\rm nb}}^{i \in S_{\rm req}} \gamma_{ij}, \tag{3}$$

where,

$$\gamma_{ij} = \begin{cases} 0, & \text{if } j \notin L_i, \\ 1, & \text{if } f \in L_i. \end{cases} \tag{4}$$

We rank peers in $S_{\rm nb}$ according to their w_j and pick top k peers as choke point expansion peer candidates. We use CDN to serve the candidates respectively. After every time interval $t_{\rm in}$, we assess the gain of serving these peers to determine who should be served continually and who should be disconnected.

In order to evaluate the gain of those active help from CDN, we introduce magnification effectiveness, a metric that reflects the magnifying contribution of a data unit served by CDN. For example, CDN serves a data unit to a peer p at time t; and some time later, at time $t + k$, peer p serves it to m other peers. We say this data unit is magnified m times at time $t + k$. The magnification effectiveness is m/k. From this example, we know that larger magnification effectiveness leads to faster data spreading.

There are two steps to calculate the magnification effectiveness of each peer.

First, we estimate the total magnifying rate of each peer who is being served by CDN. Based on peer report and (1), at the beginning time t_0, we calculate the uploading rate $U_i(t_0)$ of each peer, which is used as reference uploading capacity; then at every time interval $t_{\rm in}$ ($t_{\rm in} = t - t_0$), we calculate each peer's uploading rate $U_i(t)$. The uploading magnifying rate $U_m(t)$ of each peer i is calculated as follows:

$$U_m(t_{\rm in}) = \frac{U_i(t) - U_i(t_0)}{t - t_0}. \tag{5}$$

Second, we evaluate the average magnifying contribution of data units. Because of the CDN support strategy, it is possible that peers do not receive the same amount of data from the CDN. For fair comparison, we calculate the average magnifying contribution of each data unit in a peer as the magnification effectiveness. We denote the total data of a peer served by CDN within time interval $t_{\rm in}$ as $D_i(t)$. The magnification effectiveness is:

$$M_i(t) = \frac{U_m(t)}{D_i(t)} = \frac{U_i(t) - U_i(t_0)}{D_i(t) \times (t - t_0)}. \quad (6)$$

Therefore, CDN chooses to serve data to peers who have higher magnification effectiveness. By this way, we can speed up the data spread in the system. In fact, these peers are known as choke point expansion nodes who have great potential to better alleviate the choke points.

4.2 Indentifying Super Nodes

The choke point expansion nodes help to expand choke points in the existing P2P topology due to existing rate allocation. PROSE also evolves the P2P topology by identifying and serving P2P super nodes with strong serving-others capacities, thus constructing an effective P2P streaming network with support of both choke point expansion nodes and super nodes. PROSE identifies super nodes according to their content delivery capacities. To better estimate the delivery capacity of a peer node, we take three key properties into consideration: upload capacity, stability, and data availability. A peer with higher upload capacity and stability can serve more peers for a longer period. These two properties are intuitive factors for super nodes selection. However, a hidden factor prevents us from identifying super node is the data availability of a peer, especially a fresh peer of the system. If a peer has large upload capacity but little data to serve other peers, it is difficult to identify this node using run-time statistical feedback measurement. However, we are able to foresee that identifying and serving such a super node as early as possible, will give a better potential gain for improving performance. With statistical information collected by CDN, we are able to identify super nodes in dynamic environments.

Upload Capacity. To measure upload capacity of a node in run-time is difficult, due to the vulnerability of network and dynamical run-time capacity utilization of a peer. As Table 1 describes, with the information collected by CDN, we can estimate the upload capacity of a peer in a dynamic network environment according to the following formulas:

$$U_i^C(t) = \beta \times U_i^C(t-1) + (1-\beta) \times U_i^t, \quad (7)$$

$$\beta = \frac{U_i^{t-1}(t) + U_i^{t-2}(t)}{2 \times U_i^{\rm Avg}(t)}, \quad (8)$$

where $U_i^C(t)$ and $U_i^C(t-1)$ are the approximated upload capacity of peer i at time t and time $t-1$, respectively. $U_i^t(t)$, $U_i^{t-1}(t)$ and $U_i^{t-2}(t)$ are upload rate at time t, $t-1$ and $t-2$ (the last three sampling times), respectively, which are computed as in (1). $U_i^{\rm Avg}(t)$ is the average upload rate of peer i from $t-2$ to t. Therefore, β is a historical factor coefficient.

Stability. Stability $S_i(t)$ is considered because a more stable node can serve other peers longer and more reliably. The longer life time a peer has in a session, the higher probability it will keep staying in the session in future, thus it is considered as a more stable peer. Besides, we also prefer a stable peer who joins the system earlier because potentially an earlier peer has more time to serve others. Therefore, to evaluate the stability property of a peer, we consider both its life time and the time it joins the session. We compute the stability property of a peer i at time t as follows:

$$S_i(t) = \frac{t - t_0}{L} \times \frac{L - t_0}{L} = \frac{(t - t_0) \times (L - t_0)}{L^2}. \quad (9)$$

In above formula, L is the entire life time of the session, t_0 is the time peer i joins the session. $L - t_0$ computes the remaining time of the session that peer i can stay. The earlier peer i joins, the longer the remaining time will be.

Data Availability. Data availability $A_i(t)$ is calculated by the buffer ration of peer i at time t as:

$$A_i(t) = \sum\nolimits_{i=t_{\rm bg}}^{t_{\rm end}} Blk_i \Big/ \sum\nolimits_{i=t_{\rm bg}}^{t_{\rm end}} Buf_i. \quad (10)$$

Here $t_{\rm bg}$ and $t_{\rm end}$ are the beginning and ending time of the current window, Blk_i is the size of node i's current total stored block in its buffer, and Buf_i is node i's buffer size. This information can be retrieved from the peer report as described in Table 1.

Combined Model. We combine the above three metrics to compute the overall approximation capacity $P_i(t)$ of a peer i using the following model:

$$P_i(t) = \exp[\eta_1 \times S_i(t) + \eta_2 \times U_i^C(t) - \eta_3 \times A_i(t)]. \quad (11)$$

Here η_1, η_2, and η_3 are smoothing factors between 0 and 1. Note that higher $P_i(t)$ indicates higher possibility that the peer i is to be selected as a super node.

5 Serving Choke Point Expansion Nodes and Super Nodes

Since the overall capacity of CDN is not unlimited, we should consider how many peers the CDN can serve aggressively and how long they serve each peer. The intuitive consideration is that, for those peers with high magnification effectiveness, we should serve them long enough, so that they can maximally assist other peers, including choke point expansion nodes and super nodes. For those peers without magnification effectiveness, we should limit the service time to them. Therefore, an adaptive peer replacement strategy is needed for CDN to better replace its service list.

In order to better utilize the capacity of the CDN and prevent one from overusing it, we take the upload capacity of the CDN, $U^C_{\rm CDN}$, into consideration. In our design, we first guarantee enough capacity for the requesting peers who cannot play well (in the passive scheme). The capacity of CDN used for proactive service should satisfy the following constraints:

$$\begin{aligned}
&U^C_{\rm CDN} = \sum\nolimits_{i\in S_{\rm req}} U_{{\rm CDN},i}(t) + \sum\nolimits_{j\in S_{\rm agg}} U_{{\rm CDN},j}(t),\\
&\sum\nolimits_{i\in S_{\rm req}} U_{{\rm CDN},i}(t) = (1-\alpha(t))\times U^C_{\rm CDN},\\
&\sum\nolimits_{j\in S_{\rm agg}} U_{{\rm CDN},j}(t) = \alpha(t)\times U^C_{\rm CDN}. \qquad (12)
\end{aligned}$$

Here $U_{{\rm CDN},i}(t)$ is the uploading rate from the CDN to peer i at time t, $S_{\rm req}$ is the set of peers who cannot play well and need data urgently. $S_{\rm agg}$ is the set of choke point expansion nodes and super nodes that are served by the proactive CDN scheme. $a(t)$ is an adjustable coefficient at time t. $a(t)$ is also a key parameter that determines whether we can efficiently utilize the CDN capacity. According to Theorem 1[26], r is the streaming rate (in bps) and the maximum achievable streaming rate $r_{\max}$ is given by:

Theorem 1.

$$r_{\max} = \min\Big\{u_s, \frac{u_s + \sum\nolimits_{i=1}^{n} u_i}{n}\Big\}, \qquad (13)$$

where u_s (in bps) is the total upload rate of all the CDN servers, u_i is peer node i's upload rate, $i = 1, 2, 3, \dots, n$.

Peer nodes join and depart according to the Poisson distribution, so we have $\rho_i = \lambda_i/\mu_i$. Recall that $P_i(t)$ denotes the number of active type-i peers at time t[26]; here, $P_1(t)$ and $P_2(t)$ are the choke point expansion/super node number and common peer node number respectively. The appearing probability of P_1 and P_2 is $f_1(P_1)\times f_2(P_2)$. u_1 and u_2 are the choke point expansion/super node and common peer node's average upload rate respectively. η is the average upload bandwidth utilization rate of P_1 and P_2. In this situation, according to Theorem 1[26], the CDN server's upload bandwidth need is:

$$u_s = (P_1+P_2)\times r - (P_1\times u_1 + P_2\times u_2)\times\eta. \qquad (14)$$

Now let us consider the total expectation value of CDN upload bandwidth listed as below[22]:

$$\begin{aligned}
\overline{u_s} = &\sum[(P_1(t)+P_2(t))r - (P_1(t)u_1 + P_2(t)u_2)]\times\\
&f_1(P_1)\times f_2(P_2)\\
= &\sum_{j=0}^{\infty}\sum_{i=0}^{\infty}\frac{e^{-\rho_1}\times\rho_1^i}{i!}\times\frac{e^{-\rho_2}\times\rho_2^j}{j!}\times[(i+j)\times r-\\
&(iu_1+ju_2)\times\eta]\\
= &\sum_{j=0}^{\infty}\sum_{i=0}^{\infty}\frac{e^{-\rho_1}\times\rho_1^i}{i!}\times\frac{e^{-\rho_2}\times\rho_2^j}{j!}\times[(i\times(r-u_1\eta)+\\
&j\times(r-u_2\eta))]\\
= &\sum_{j=0}^{\infty}\sum_{i=1}^{\infty}\frac{e^{-\rho_1}\times\rho_1^i}{(i-1)!}\times\frac{e^{-\rho_2}\times\rho_2^j}{j!}\times(r-u_1\times\eta)+\\
&\sum_{j=1}^{\infty}\sum_{i=0}^{\infty}\frac{e^{-\rho_1}\times\rho_1^i}{i!}\times\frac{e^{-\rho_2}\times\rho_2^j}{(j-1)!}\times(r-u_2\times\eta)\\
= &\sum_{i=1}^{\infty}\frac{e^{-\rho_1}\times\rho_1^i}{(i-1)!}\times\sum_{j=0}^{\infty}\frac{e^{-\rho_2}\times\rho_2^j}{j!}\times(r-u_1\times\eta)+\\
&\sum_{i=0}^{\infty}\frac{e^{-\rho_1}\times\rho_1^i}{i!}\times\sum_{j-1}^{\infty}\frac{e^{-\rho_2}\times\rho_2^j}{(j-1)!}\times(r-u_2\times\eta)\\
= &\rho_1\times(r-u_1\times\eta)+\rho_2\times(r-u_2\times\eta)\\
&\Big(\text{because}: xe^x = x\sum_{i=0}^{\infty}\frac{x^i}{i!} = \sum_{i=0}^{\infty}\frac{x^{i+1}}{i!}\Big)\\
= &(\rho_1+\rho_2)\times r - (\rho_1\times u_1+\rho_2\times u_2)\times\eta. \qquad (15)
\end{aligned}$$

We compute $a(t)$ according to:

$$\alpha(t) = \overline{u}_s = (\rho_1+\rho_2)\times r - (\rho_1\times u_1+\rho_2\times u_2)\times\eta. \quad (16)$$

Therefore, at any given time t, the capacity of CDN can be used in the proactive CDN scheme $C_{\rm agg}(t)$ as follows:

$$\begin{aligned}
C_{\rm agg}(t) &= \sum\nolimits_{j\in S_{\rm agg}} U_{\rm CDN,j}(t)\\
&= U^C_{\rm CDN}((\rho_1+\rho_2)\times r - (\rho_1u_1+\rho_2u_2)\times\eta).
\end{aligned} \qquad (17)$$

The initial value of $\alpha(t)$ is 0.2. Through computing, if $\alpha(t) < 0$, then $\alpha(t) = 0.2$. If $\alpha(t) > 1$, then $\alpha(t) = 0.8$.

CDN can get the average value ρ_1 and ρ_2, η through collecting statistical data from the requesting peer nodes at each time interval $t_{\rm in}$.

$$\eta = \frac{(\rho_1+\rho_2)\times r - U^u_{\rm CDN}}{\rho_1\times u_1+\rho_2\times u_2}. \qquad (18)$$

$U^u_{\rm CDN}$ is the total actual bandwidth consumption value of all CDN nodes. Based on (2)∼(4), (11), CDN first selects the top K ($K = C_{\rm agg}(t)/r$) choke point expansion nodes and super nodes to directly serve with the streaming rate r. Both nodes enter the ranks of K according to a first-come-first-served order. Afterwards, CDN evaluates these K nodes according to their magnification effectiveness and updates the top k nodes periodically.

6 Experiments and Analysis

We have implemented a P2P live streaming prototype system based on our novel scheme. First, we construct the emulation experiments using both well-known commercial and experimental environments, Amazon EC2 and Emulab. We deploy our source server and CDN servers in Amazon EC2 and run our P2P system in Emulab. In order to simulate the real environment, we use three types of peers with maximum upload capacity of 384 Kbps, 512 Kbps and 1 024 Kbps, respectively. Since the scale of Emulab is limited, to better evaluate the performance of our algorithm, we further carry out a larger-scale simulation experiment using the well-known simulator in [28]. The detailed parameter settings of our experiments are shown in Table 2.

To evaluate our proposed algorithms, we compare our PROSE algorithm with existing NATIVE-CDN-P2P algorithm, similar to Livesky, as proposed in [3, 9]. In the NATIVE-CDN-P2P scene, each arriving peer node selects a certain number of nodes as its neighbors from the tracker. At the same time, some startup or emergent peer nodes will be redirected to one CDN node which has the minimum delay to that peer node. A CDN node serves a fixed subset of P2P peers in a first-in-first-out (FIFO) and best effort manner. Therefore, all peer nodes will be separated into different regions. Each region is led by a CDN node. In our experiments, PROSE is based on our proposed algorithm: identifying and serving choke point expansion nodes and super nodes. As mentioned, the choke point expansion nodes can expand choke points in the existing topology according to existing rate allocation. PROSE also improves the P2P topology by identifying and serving P2P super nodes with strong delivery capacities.

In the NATIVE-CDN-P2P experiment, every CDN node will use its maximum bandwidth to serve every requesting peer node until overload. In the implementation scheme of PROSE, based on (12)~(18), every CDN node divides their upload bandwidth into two parts, the passive serving part for startup or urgent nodes, and the proactive serving part — $\alpha(t)$ for choke point expansion nodes and super nodes.

In our experiments, we use the following metrics to evaluate performance, which together represent the quality of service experienced by end users.

- *Startup latency*: the time from when an end user joins the session of the video stream to when it has received enough data blocks to start playback;
- *Playback delay*: the time taken for a data block to be generated by the source to when it has been played by all end users;
- *Total CDN upload bandwidth utilization rate*: the proportion of current used upload bandwidth of CDN to the total bandwidth provided by CDN;
- *Average download rate*: the average amount of data downloaded by each peer per second, including content data and message overhead;
- *Average overhead*: the average data amount of reporting and exchanging messages among peer-and-peer and peer-and-CDN (except content data) each peer per second, such as peer nodes buffer-map exchanging, peer node reporting information to CDN.

6.1 Emulation Experiments and Analysis

We first carry out emulation experiments with two CDN nodes in Amazon EC2 and 100 peer nodes in Emulab, as described in Table 2.

Fig.3 shows the cumulative distribution function (CDF) of the startup latency of the NATIVE-CDN-P2P system and our PROSE-CDN-P2P system with 100 nodes. In the PROSE-CDN-P2P system, average startup delay is 9.37 seconds, while in the NATIVE-CDN-P2P system, average startup delay is 11.46 seconds.

Table 2. Experimental Settings

Experiment Methods	Emulation	Simulation
Number of peers	100 at Emulab	1 000
Streaming rate	512 Kbps	512 Kbps
Peer upload capacity	1 Mbps/15%, 512 Kbps/15%, 384 Kbps/70%	1 Mbps/20%, 512 Kbps/30%, 384 Kbps/50%
Peer download capacity	2.5 Mbps	2.5 Mbps
Source upload capacity	1 source server, 3 840 Kbps	1 source server, 3 840 Kbps
CDN upload capacity	2 CDN servers at EC2 each 10 Mbps	4 CDN servers, each 10 Mbps
Piece size	60 KB	60 KB
Sub-piece size	1 KB	1 KB
Session length	300 seconds	300 seconds
Buffer time (size)	35 seconds	35 seconds
Average neighbors	15	15

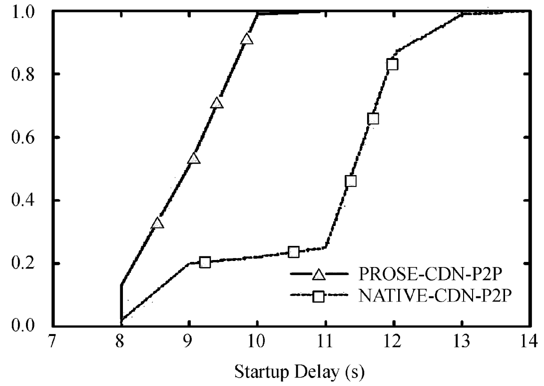

Fig.3. 100-node startup latency comparison result.

Fig.4 illustrates the CDN nodes total utilization rate. We observe that, PROSE CDN utilization is 0.95, while Native CDN utilization is only 0.63. Fig.5 shows the average download rate CDF of these two systems. As the live stream playing rate is 512 Kbps, we observe that almost all peer nodes can play well in both systems. But the download rate of the NATIVE-CDN-P2P system is larger than that of the PROSE-CDN-P2P system. This is because the NATIVE-CDN-P2P system has more message exchanging overhead than the PROSE-CDN-P2P system.

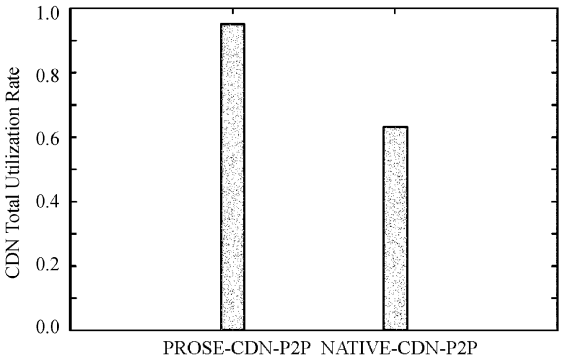

Fig.4. 100-node CDN total utilization rate comparison result.

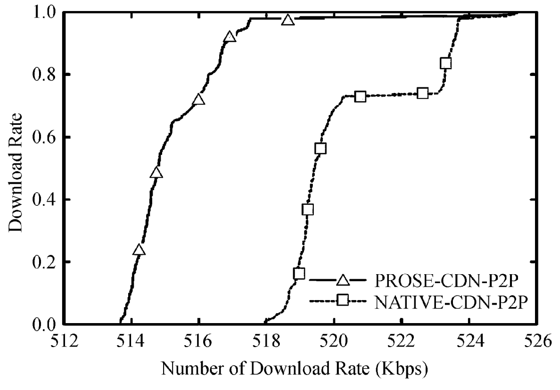

Fig.5. 100-node average download rate comparison result.

As Fig.6 shows, PROSE-CDN-P2P average overhead is 428.08 bytes/second per peer, while NATIVE-CDN-P2P average overhead is 1 080.23 bytes/second per peer. This is because the CDN proactive serving method decreases the message exchanging among peer nodes. Therefore, our PROSE algorithm achieves more than half reduction of the message overhead, as PROSE can maximize CDN usage and decrease peers message exchanging.

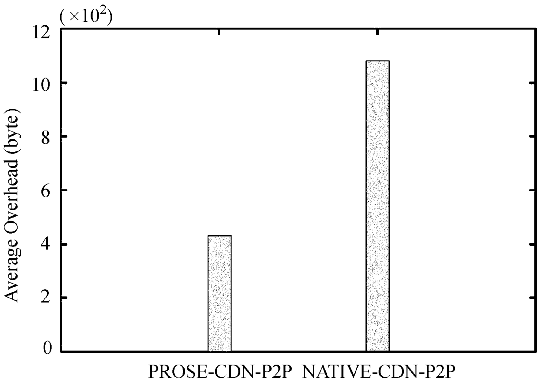

Fig.6. 100-node average overhead comparison result.

Fig.7 shows the 100 nodes' average playback delay of the NATIVE-CDN-P2P system and our PROSE-CDN-P2P system at 300 seconds of session time. We notice that the playback delay has reduced from approximately 4 700 ms to 2 300 ms. In PROSE, CDN directly, selectively helps the choke point expansion nodes and super nodes, making the playback delay greatly reduced.

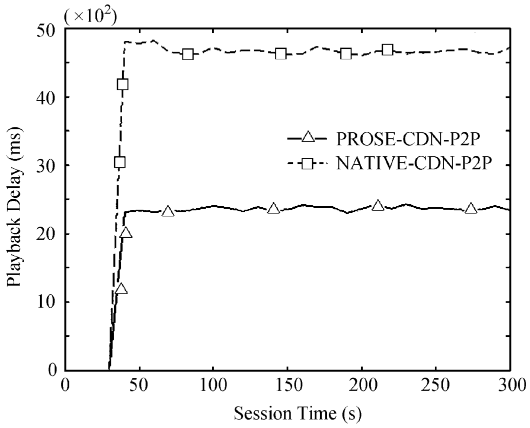

Fig.7. 100-node playback delay comparison result.

6.2 Simulation Experiments and Analysis

In the following simulation experiment, we increase the node scale to 1 000. We use a well-known P2P streaming simulator developed by Zhang *et al.*[28] This simulator only supports a general P2P pull-based live streaming scheme. We refine the simulator and add a configurable CDN layer into this simulator. In our experiment, we can easily configure multiple CDN nodes as strong nodes to simulate a NATIVE-CDN-P2P system and a PROSE-CDN-P2P system, respectively. A CDN node will have much greater upload capacity (10 Mbps) and maintain a larger number of neighbors. We have four CDN nodes in 1 000-node experiment. They provide all data blocks of the video stream. When peers join the system, they will connect to existing P2P peers through a tracker. If the data from P2P neighbors of a peer cannot satisfy the streaming rate, this peer will be redirected to a CDN node with minimum delay to acquire additional data. CDN will use the passive serving part for startup or urgent nodes, and use the proactive serving part for choke point expansion nodes and super nodes.

Fig.8 shows the CDF of the startup latency of the two systems on a 1 000-node scale. In the PROSE-CDN-P2P system, the average startup delay is in 13.4 seconds, while in the NATIVE-CDN-P2P system, the average startup delay is 13.57 seconds. This is because after the choke point expansion nodes and super nodes directly obtaining the help of the CDN, they played better amplification affect.

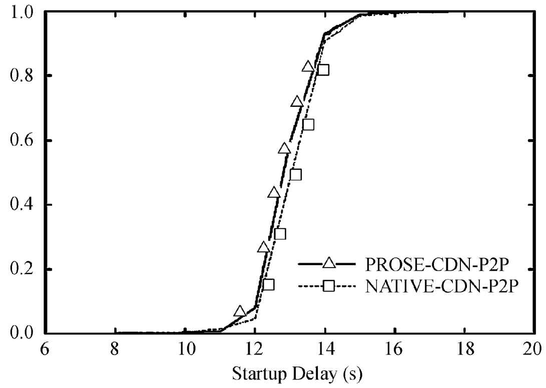

Fig.8. 1 000-node startup delay comparison result.

At the same time, PROSE CDN utilization rate is 0.94, while Native CDN utilization rate is 0.86, as seen in Fig.9. We can see that when CDN is divided into two parts: the passive serving part and the proactive serving part and the proactive serving part can make CDN more fully utilized, thus increasing CDN usage.

We see from Fig.10 that PROSE average overhead is only 380.56 bytes/second per peer, while Native average overhead is 1 370.86 bytes/second per peer. Therefore, our PROSE system reduces the overhead by more than 72%. This is because the CDN directly helps the choke point expansion nodes and super nodes, thus reducing the interaction between the CDN and the common peer nodes, also reducing the interaction among the common peer nodes.

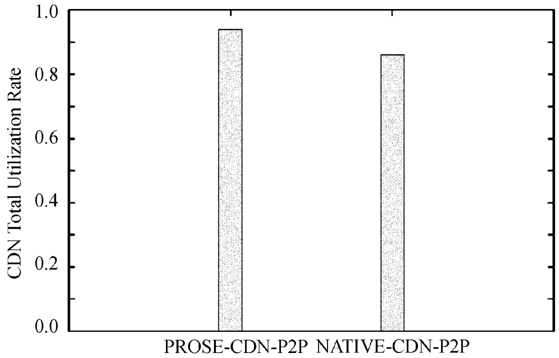

Fig.9. 1 000-node CDN total utilization rate comparison result.

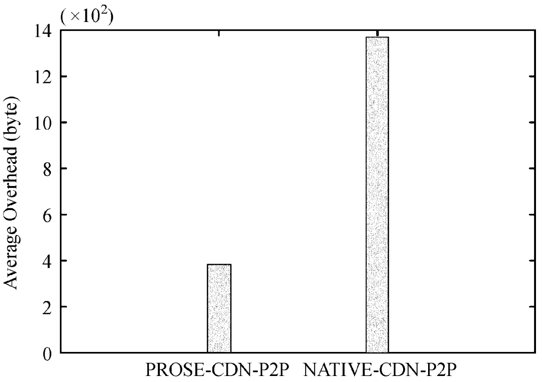

Fig.10. 1 000-node average overhead comparison result.

Fig.11 shows the 1 000 nodes' playback delay of the NATIVE-CDN-P2P system and our PROSE-CDN-P2P system at 300 seconds of session time. With our passive-proactive-hybrid CDN serving approach, the playback delay has been reduced from around 6 800 ms to 5 800 ms. This is because that the CDN node in PROSE system is a proactive content injector that aggressively injects content to the selected choke point

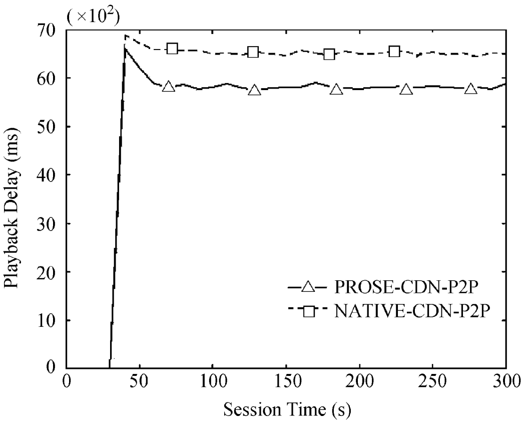

Fig.11. 1 000-node playback delay comparison result.

expansion nodes and super nodes to improve the overall P2P streaming system performance.

In the above experiments, we let 1 000 peer nodes join the streaming system according to the Poisson distribution at the beginning. In the following experiment, we construct a dynamic jitter scenario and compare NATIVE-CDN-P2P and PROSE-CDN-P2P schemes. We first let 600 peer nodes join the system for 150 seconds, and then the other 400 nodes join the session simultaneously between the 160∼170 second interval to simulate a dynamic jitter.

Figs. 12∼15 compare the experimental results. In this simulation, the average startup delay of PROSE is 17.66 seconds, compared with 19.24 seconds from the NATIVE-CDN-P2P system; PROSE-CDN-P2P utilization rate is 0.95, while NATIVE-CDN-P2P utilization rate is 0.85. PROSE average overhead is 790.87 bytes/second per peer, while NATIVE-CDN-P2P average overhead is 2 978.4 bytes/second per peer.

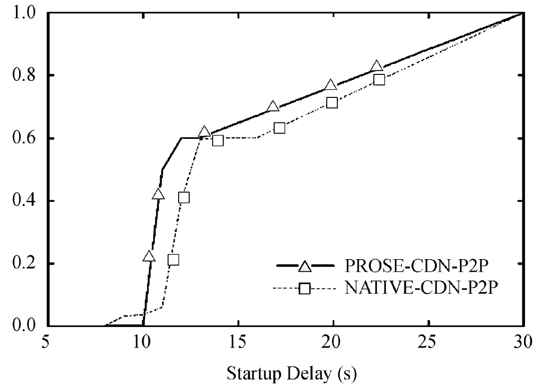

Fig.12. 1 000-node startup latency comparison result with dynamic jitter.

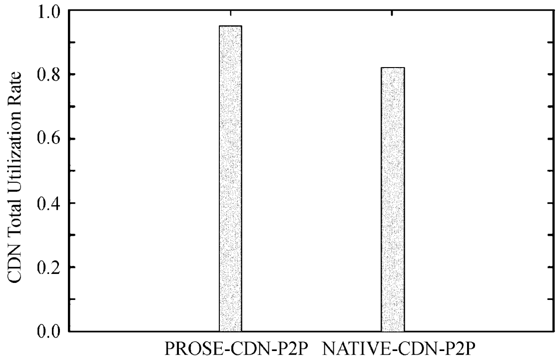

Fig.13. 1 000-node CDN total utilization rate comparison result in the presence of dynamic jitter.

The PROSE-CDN-P2P scheme also has improved the live streaming playback delay performance to a little extent, as Fig.15 shows. When the dynamic jitter happens between the 160∼170 second interval, the NATIVE-CDN-P2P system's playback delay increases to 30 000 ms, while PROSE's playback delay only increases 25 000 ms. After around 200 seconds, the playback delay of PROSE decreased faster. This is because both choke point expansion nodes and super nodes amplification effect makes the system return to normal from dynamic jitter faster.

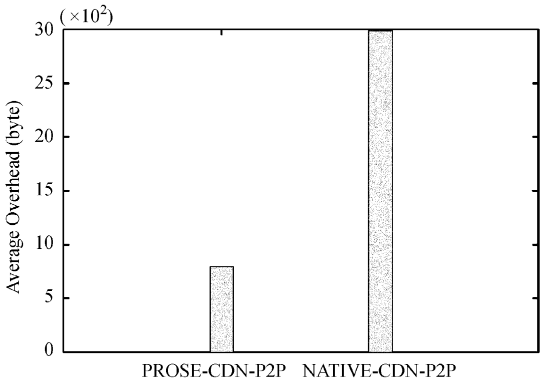

Fig.14. 1 000-node average overhead comparison result with dynamic jitter.

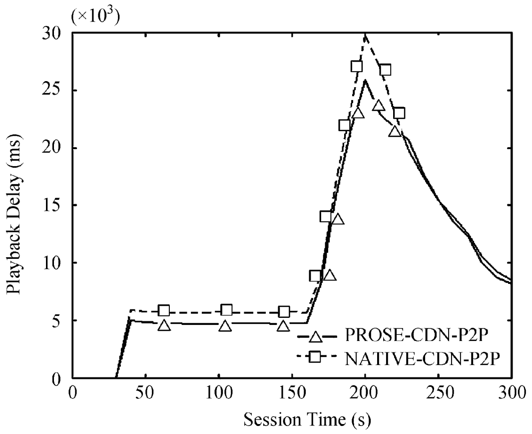

Fig.15. 1 000-node playback delay comparison result with dynamic jitter.

7 Conclusions and Future Work

In recent years, a large number of commercial P2P streaming systems have been widely deployed in the Internet. In order to increase P2P reliability, many P2P application providers have begun to rely on CDN services. However, the cost of using CDN is expensive, especially, with a huge population of users visiting it. Therefore, how to efficiently utilize CDN capacity, in particular selecting which peers to connect to CDN, is a significant open problem.

Previous studies do not solve the problem of which P2P nodes should receive service from the CDN. Based on the key problem of CDN passive and inefficient serving P2P, we designed PROSE, a novel proactive, selective CDN participation scheme for P2P streaming. We have developed a prototype system and conducted emulation and simulation experiments to validate our scheme. There are several improvement directions for our future work. First, how CDN nodes can identify super nodes better, particularly in the early stage when nodes joining. Second, in every CDN-covered region, when every CDN node gathers local information from some peer nodes, how CDN nodes utilize CDN multiple nodes natural collaboration mechanism to exchange local information and get more global information. Making use of this collaboration mechanism will help find more choke point expansion nodes and super nodes. Third, CDN nodes just use requesting peer nodes' report information, rather than large-scale monitoring information, to identify choke point expansion nodes and super nodes. In this situation, how to further improve the recognition accuracy and efficiency, especially while updating the high-quality node list? Fourth, cloud computing era calls for a more open content service mode, which requires content services become available on-demand and be utilized in an open and loosely-coupled fashion. Although the research community has put much effort into CDN-P2P-hybrid technology, many hybrid architectures belong to the tightly-coupled hybrid model. To the best of our knowledge, the loosely-coupled hybrid model is less explored. We have researched the Web services and representational state transfer (REST) based CDN, P2P and VoD (video on demand) loosely-coupled hybrid models in [19-20, 22]. As our next step, we will look into cloud-oriented loosely-coupled CDN-P2P-hybrid models. Finally, we plan to carry out larger-scale experiments through improving our prototype system and working on real deployment in our ongoing CNGI (China Next Generation Internet) project — National Higher Education Conference Video Resources Sharing Project, which now has more than 50TB academic video contents and hundreds of thousands of users.

Acknowledgement We would like to give thanks to the researchers of Laboratory of Networked Systems (LANS) at Yale University led by Proefssor Yang Richard Yang for cooperation research work. We also thank Dr. Nie Wei at the College of Information Engineering, Shenzhen University, for valuable suggestions and English writing correction.

References

[1] Huang C, Wang A, Li J, Ross K W. Understanding hybrid CDN-P2P: Why limelight needs its own red swoosh. In *Proc. the 18th NOSSDAV*, May 2008, pp.75-80.

[2] Malik O. Grid joins the P2P CDN party. http://gigaom.com/2007/10/30/grid-networks/, May 2012.

[3] Yin H, Liu X N, Zhan T Y *et al.* Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with LiveSky. In *Proc. the 17th ACM Int. Conf. Multimedia*, Oct. 2009, pp.25-34.

[4] Afergan M M, Leighton T F, Parikh J G. Hybrid content delivery network (CDN) and peer-to-peer (P2P) network. United States Patent Application 20080155061.26, Jun. 2008, http://www.freepatentsonline.com/y2008/0155061.html, Jan. 2013.

[5] Xu D Y, Kulkarni S S, Rosenberg C, Chai H Y. Analysis of a CDNP2P hybrid architecture for cost-effective streaming media distribution. *Multimedia Systems*, 2006, 11(4): 383-399.

[6] Wu J, Lu Z H, Liu B S, Zhang S Y. PeerCDN: A novel P2P network assisted streaming content delivery network scheme. In *Proc. the 8th Int. Conf. Computer and Information Technology*, July 2008, pp.601-606.

[7] Chen Z J, Yin H, Lin C, Liu X N, Chen Y. Towards a trustworthy and controllable peer-server-peer media streaming: An analytical study and an industrial perspective. In *Proc. GLOBECOM 2007*, Nov. 2007, pp.2086-2090.

[8] Liu X N, Yin H, Lin C, Liu Y, Chen Z J, Xiao X. Performance analysis and industrial practice of peer-assisted content distribution network for large-scale live video streaming. In *Proc. the 22nd AINA*, March 2008, pp.568-574.

[9] Yin H, Liu X N, Zhan T Y, Sekar V, Qiu F, Lin C, Zhang H, Li B. LiveSky: Enhancing CDN with P2P. *ACM Transaction on Multimedia Computing, Communications, and Applications*, 2010, 6(3): 16-34.

[10] Huang C, Wang A, Li J, Ross K W. Measuring and evaluating large-scale CDNs. Technical Report, MSR-TR-2008-106, Microsoft Research, Oct. 2008.

[11] Jiang H, Li J, Li Z C *et al.* Efficient large-scale content distribution with combination of CDN and P2P networks. *International Journal of Hybrid Information Technology*, 2009, 2(2): 13-24.

[12] Ha D H, Silverton T, Fourmaux O. A novel hybrid CDN-P2P mechanism for effective real-time media streaming, http://www-rp.lip6.fr/∼fourmaux/Stages/HA.ACM_Rapport.pdf, Jan. 2013.

[13] Yang H C, Hsieh M Y, Yu H F, Tseng L M. A replication-aware CDN-P2P architecture based on two-step server selection and network coding. In *Proc. PCM*, Dec. 2008, pp.738-747.

[14] Su A J, Choffnes D R, Kuzmanovic A, Bustamante F E. Drafting behind Akamai (travelocity-based detouring). In *Proc. the 2006 SIGCOMM*, Sept. 2006, pp.435-446.

[15] Mansy A, Ammar M. Analysis of adaptive streaming for hybrid CDN/P2P live video systems. In *Proc. ICNP 2011*, Oct. 2011, pp.276-285.

[16] Lv Z H, Wu J, Chen L J, Huang S J, Huang Y. CPH-VoD: A novel CDNP2P-hybrid architecture based VoD scheme. In *Proc. the 11th International Conference on Web Information Systems Engineering*, Dec. 2010, pp.578-586.

[17] Gannes L. Cisco to distribute grid networks' P2P client. http://newteevee.com/2008/03/16/cisco-to-distribute-grid-networks-p2p-client/, May 2012.

[18] Choffnes D R, Bustamante F E. Taming the torrent: A practical approach to reducing cross-ISP traffic in peer-to-peer systems. In *Proc. SIGCOMM 2008*, August 2008, pp.363-374.

[19] Lv Z H, Wu J, Xiao C, Fu W M, Zhong Y P. WS-CDSP: A novel web services-based content delivery service peering scheme. In *Proc. IEEE SCC*, Sept. 2009, pp.348-355.

[20] Lv Z H, Wu J, Fu W M. Towards a novel web services standard-supported CDN-P2P loosely-coupled hybrid and management model. In *Proc. IEEE International Conference on Services Computing*, July 2010, pp.297-304.

[21] Lv Z H, Gao X H, Huang S J *et al.* Scalable and reliable live streaming service through coordinating CDN and P2P. In *Proc. the 17th ICPADS*, Dec. 2011, pp.581-588.
[22] Lv Z H, Wu J, Huang Y *et al.* CPDID: A novel CDN-P2P dynamic interactive delivery scheme for live streaming. In *Proc. the 18th ICPADS*, Dec. 2012, pp.299-306.
[23] Wang F, Liu J C, Xiong Y. Stable peers: Existence, importance, and application in peer-to-peer live video streaming. In *Proc. the 27th IEEE INFOCOM*, April 2008, pp.1364-1372.
[24] Liu Z M, Wu C, Li B C, Zhao S Q. Distilling superior peers in large-scale P2P streaming systems. In *Proc. the 28th IEEE INFOCOM*, April 009, pp.82-90.
[25] Mitra B, Dubey A K, Ghose S *et al.* How do superpeer networks emerge? In *Proc. IEEE INFOCOM*, March 2010, pp.1514-1522.
[26] Kumar R, Liu Y, Ross K. Stochastic fluid theory for P2P streaming systems. In *Proc. IEEE INFOCOM*, April 2007, pp.919-927.
[27] Wang F, Xiong Y Q, Liu J C. MTreebone: A hybrid tree/mesh overlay for application-layer live video multicast. In *Proc. IEEE ICDCS*, June 2007, pp.49-58.
[28] Zhang M. Peer-to-peer streaming simulator, http://media.cs.tsinghua.edu.cn/~zhangm/download/p2pstrmsim_allinone.tar.gz, May 2012.
[29] Karagiannis T, Rodriguez P, Papagiannaki D. Should Internet service providers fear peer-assisted content distribution? In *Proc. the 5th Internet Measurement Conference*, October 2005, Article No. 6.
[30] Bonald T, Massouli L, Mathieu F, Perino D, Twigg A. Epidemic live streaming: Optimal performance trade-offs. In *Proc. SIGMETRICS 2008*, June 2008, pp.325-336.
[31] Huang C, Li J, Ross K W. Can internet video-on-demand be profitable? In *Proc. SIGCOMM*, August 2007, pp.133-144.
[32] Huang Y, Fu T Z J, Chiu D M *et al.* Challenges, design and analysis of a large-scale P2P-VoD system. In *Proc. SIGCOMM 2008*, August 2008, pp.375-388.

Zhi-Hui Lv received the Ph.D. degree in computer application technology from Fudan University, Shanghai, in 2004, and now is an associate professor in the School of Computer Science, Fudan University. He is a member of the IEEE. His research interests are network multimedia technology, content delivery network, P2P streaming technology, and service computing technology.

Li-Jiang Chen is a research scientist at HP China Labs, Beijing. He received his Ph.D. degree in computer science from Peking University in 2011. His research interests include distributed systems and P2P networks, Web data mining and content recommendation, information retrieval, content delivery networks. Currently, He is working on Ad/coupon recommendation and context-aware information analytics projects.

Jie Wu is a professor at School of Computer Science, Fudan University. He received his Ph.D. degree in computer application technology from Fudan University in 2008. His research interests are service computing, cloud computing, and P2P streaming technology. Dr. Wu is a member of WSSG (Web Service Study Group) and SOA-SG (SOA Study Group) of ISO/IEC JTC1 SC7 on behalf of China.

Da Deng is a master student at School of Computer Science, Fudan University. His major is computer application technology. His research interests are cloud computing and P2P streaming technology.

Si-Jia Huang got the M.S. degree in computer application technology from the School of Computer Science, Fudan University in 2011. Her research interests are CDN and P2P streaming technology.

Yi Huang got the M.S. degree in computer application technology from the School of Computer Science, Fudan University in 2011. His research interests are VoD and P2P streaming technology.