Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090351

Transcoding in Two-class P2P Streaming Systems

Shuai Zeng[1,*], *Lemin Li*[2] *and Dan Liao*[2]

[1] Chongqing University of Posts and Telecommunications, Chongqing, P. R. China
[2] University of Electronic Science and Technology of China, Chengdu, P. R. China

Received: 21 Aug. 2014, Revised: 21 Nov. 2014, Accepted: 22 Nov. 2014
Published online: 1 May 2015

Abstract: We study the issue of P2P streaming system with transcoding and expose the fundamental characteristics and mathematical theory of the system in the two-class network structure. We find out and prove that, in a two-class P2P streaming system, to provide peers receiving data above some given flow rate, there is a lower bound of source server load. We give an algorithm to achieve this minimal server load. We also compare the minimal server load by our algorithm with typical traditional design without transcoding in various situations in our simulation experiments, and analyze how much benefit system can get from the utilization of transcoding technique. The results show that, if transcoding technique is utilized appropriately, better performance of two-class P2P streaming system can be achieved.

Keywords: streaming system, P2P network, transcoding, heterogeneous network, content distribution

1 Introduction

With the development of P2P live streaming technology, the approach of multimedia streaming systems have evolved from unicast, tree, and multi-tree to mesh [1,2,3,4]. Peers become more and more complex. From bandwidth, memory to CPU, more and more resources of peers have been exploited for relieving the source server load and utilizing the systems resource more effectively. These clients, such as PC, TV, tablet, PDA, cellphone, and so on, have various screen sizes, color depth and video qualities. Specially, they may have different video coding with heterogeneous hardware and software, and they may have different bandwidth with heterogeneous network [3,4,5,6,7,8,9]. In traditional P2P multimedia streaming systems, the peers, which receive the same program with different video coding algorithm, cannot share their upload capacity in single overlay network. However more networks or video sources may need more resources.

In recent years, there are some literatures [6,7,8,9,10,11,12] that focus on transcoding technique utilized in P2P streaming systems. [6] proposes a multimedia streaming architecture in which transcoding services coordinate to transform the streaming data into different formats in P2P systems. [7] proposes a system named PAT (Peer-Assisted Transcoding) to enable effective online transcoding and seek to reduce the bandwidth consumption and computing overhead in P2P network. In [8], the transcoding technique is used in some total new network environments. The paper discusses issues that are relevant to enabling P2P streaming in networked consumer electronics, NAT/firewall traversal, and codec inflexibility. [10] also discusses the video transcoding in P2P network of IPTV system. [9] proposes a P2P transcoding method for heterogeneity mobile streaming. The paper seeks to increase the flexibility of coding data, which is based on diverse display size, computing power, memory, and media capabilities in devices. [11] presents a P2P streaming system named CloudStream, which is a cloud-based video proxy that can deliver streaming videos by transcoding the original video in real time to a scalable codec. And [12] proposes a collaborative strategy that leverages the peering architecture of P2P networks and makes the computational resources of peers sharable and collaborative. These researches announce that, in the some situations, compared with traditional systems, P2P streaming systems with transcoding have better performances. Nevertheless, existent studies just focus on network protocols design and video coding algorithm, which lacks of mathematically investigate and deeply understand their systems. Furthermore, there exists no relative research that focuses on what network environment the transcoding systems suits and how much benefit in quantity the new technique taken to the systems.

* Corresponding author e-mail: maillinger@gmail.com

At the same time, the research on P2P network structure pay more attention to the two-class model [13, 14, 15] [17], which means peers in the system are broadly classified into two classes, with each class having approximately the same upload capacity. These studies are reasonable as there are roughly two classes of peers, which are super and ordinary ones, in P2P streaming systems. In this paper, we are interested in the basic network fluid model for P2P streaming systems with transcoding in two-class P2P streaming systems. Our paper seeks to expose some fundamental characteristics and limitations of two-class P2P streaming systems with transcoding. There are some literatures [13, 14, 15, 16, 17] that discusses and analyzes the issues of P2P streaming systems capacity by mathematic fluid model in two-class P2P streaming systems without transcoding. [13] develops a basic stochastic model and fluid theory for the P2P streaming systems and discusses the theory in two-class systems. [14][14] derives and proves the performance bounds for minimum server load and maximum streaming rate in two-class P2P streaming systems. And in [17], the authors develop a fluid model for two-class P2P streaming systems with network coding and mathematically analyze the performance of this kind of systems.

In this paper, our analysis and results are based on both previous research and the features of two-class P2P streaming systems with transcoding. And, without sacrificing realistic assumptions of systems scale, we mainly investigate the minimal demand of video source server upload capacity, which also calls minimal server load, in two-class P2P streaming systems with transcoding. Furthermore, basing on our mathematical analysis and simulation experiment, we compare the performance of transcoding system with no transcoding one in two-class P2P streaming systems, and seek to answer the following questions.

1. How could transcoding technique be helpful in two-class P2P streaming systems? What situations are proper for using this technique?

2. What is the condition to make peers receiving data from system above some given flow rate? Can we find and achieve the minimal server load?

3. Compared with systems without transcoding, how much better the transcoding technique taken? What are the key parameters that make their difference performance?

The remainder of the paper is organized as follows. Section 2 describes and insights the basic features of two-class P2P streaming systems with transcoding and no transcoding. In Section 3, we compute and prove the minimal server load for some given flow rate of each peer, and we give an algorithm to achieve this minimal load in the proof. In section 4, we give our simulation experiment and compare system performance between transcoding and no transcoding. And. Finally, we conclude this paper in Section 5.

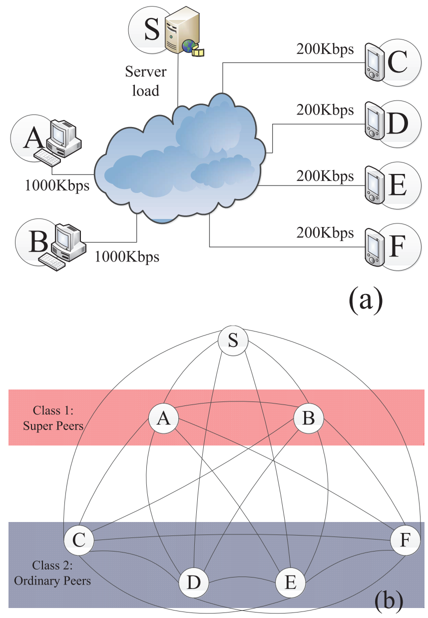

Fig. 1: An example of two-class P2P streaming system.

2 Two-class P2P Streaming Systems with Transcoding

Before we expose our theoretical analysis and model of two-class P2P streaming systems with transcoding, in this section, we first summarize some fundamental characteristics and overview design principles of the traditional design and transcoding cases.

As shown in Figure 1(a), this is a typical example of two-class P2P streaming system. There are six peers (A, B, C, D, E and F) and one source server (S) in the system. A and B are PCs with 1000 Kbps upload capacity respectively, which are seen as super peers, and C, D, E and F are cellphones with 200 Kbps upload capacity respectively, which are seen as ordinary peers. And, we further assume that the download bandwidth and computing capability is not the bottleneck of the system, which is a common setting of previous studies. There is a video program in S with some video coding rate. The goal of P2P streaming system design is to ensure peers receiving data above the video coding rate, and, at the same time, minimize the server load.

To illustrate clearly, the P2P network in Figure 1(a) is redrawn as an overlay complete network as Figure 1(b). In next subsection, based on the example of Figure 1, we describe how two-class P2P streaming system and traditional system work.

2.1 Traditional Designs without Transcoding

In a two-class P2P streaming system, the traditional designs without transcoding almost basically base on the model in [13][14][17], which is close to the reality situations of P2P streaming systems in the practice. In these systems, peers in the system are classified into two classes depending on their different hardware. As the example shown in Figure 2(a), A and B are super peers, which belong to class 1, while C, D, E and F are ordinary ones, which belong to class 2. In this two-class P2P streaming system, super peers and ordinary peers have their own video coding rates respectively. When a video program needs to send to all of these peers, two kinds of video coding data are transported by network independently. This means, as drawn by different colors in Figure 2(a), the whole P2P streaming system is divided into two subsystems. Every subsystem is an independent P2P streaming system, each of which has its different video coding rate, and no data exchange among them. We call this kind of systems No Transcoding Systems (OTS) in the rest of this paper.

Consider the situation of OTS in the setting of Figure 2(a). In this example, to support the video coding rates, super peer requires 800Kbps bandwidth and ordinary peer requires 300Kbps bandwidth. If the system needs all peers in class 1 get data no less than the rate 800Kbps and all peers in class 2 get data no less than the rate 300Kbps, as shown in Figure 2(a), the total upload capacity is at least 2800Kbps, and the source server load is at least 1200Kbps.

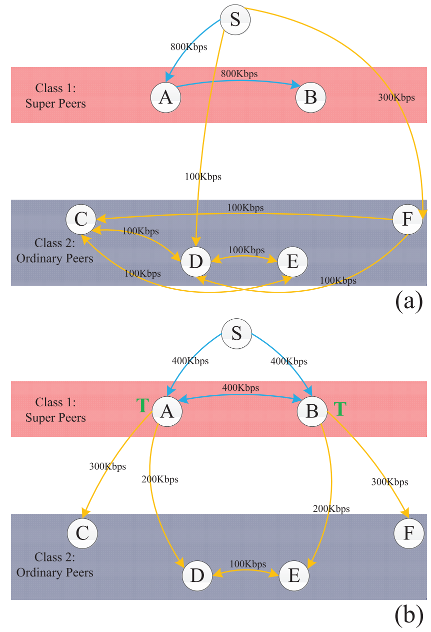

Fig. 2: Example of OTS and TS.

2.2 Two-class P2P Streaming Systems with Transcoding

Based on [3,7,17], utilizing transcoding technique, P2P streaming systems can support multiple video coding rates and exchange their data in one overlay network. Especially, in a two-class P2P streaming system, as the same as OTS, there are two video coding data in the network. They have different video quality with different coding rate. In general, the high quality video data, which have high coding rate, is sent to the super peers (the peers in class 1) firstly. Super peers are responsible for transcoding this video data from high coding rate to low one. And send video data with low coding rate to the ordinary peers at last. We call this kind of systems Transcoding Systems (TS) in the rest of this paper.

Back to the example and consider the situation of TS in the setting of Figure 2(a). If the system needs all super peers get data no less than the rate 800Kbps and all ordinary peers getting data no less the rate 300Kbps. The original data could be pushed to A and B from S firstly, and A and B, which are marked ’T’ in the figure, transcode these data to the format and coding rate for the ordinary peers, Then, the transcoded data is pushed to C, D, E and F from A and B. So, calculating the result, as shown in Figure 2(b), we conclude that the necessity of total upload capacity is reduced to 2800Kbps and the necessity of source server load is reduced to 800Kbps.

It seems we can easily conclude that, compared with OTS, TS has better performances. But there are still some questions. First, is the server load of TS always better than OTS in any two-class P2P streaming systems? Second, how does transcoding technique affect the systems exactly? What is the essential relation between transcoding and no transcoding systems? Third, not intuitively, but mathematically, how many performances improvement of TS are compared with traditional systems? We discuss these questions in next section. Before that, for the sake of mathematical tractability, we make a few assumptions in TS. First, our study bases on the scenario of two-class P2P streaming system, which is described in many previous papers [13,14,15] [17]. Second, upload capacity is the only bottleneck of system. This assumption is according to most existing studies [13, 14,15,16,17] of P2P systems. Specially, in our study, it also means that peers can transcode their received data to other coding data, which has the same or lower video quality, and the transcoded data is always enough for uploading. This paper just discusses the network bandwidth model and seeks the basic fluid theory, and does not restrict the specific network protocols or video coding algorithm.

3 Model and Algorithm

3.1 Notations and Expressions

Before modeling and analyzing two-class P2P streaming systems with transcoding, introduce some necessary notations and expressions firstly.

Denote by n for total number of peers in the system. Let n_1 be the number of super peers and n_2 be the number of ordinary peers. Denote by $\mathbf{P}$ for the set of all peers. Let $\mathbf{P}_1$ be the set of super peers and $\mathbf{P}_2$ be the set of ordinary peers. Denote by s for the source server , p_{ij} for peer in class i. We have

$$\mathbf{P}_1 = \{p_{1j}\}, \quad for\ j = 1,...,n_1$$

$$\mathbf{P}_2 = \{p_{2j}\}, \quad for\ j = 1,...,n_2$$

$$\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2 = \{p_{ij}\}, \quad for\ i = 1,2;\ j = 1,...,n_i$$

$$n = n_1 + n_2 = |\mathbf{P}_1| + |\mathbf{P}_2| = |\mathbf{P}|$$

Furthermore, we adopt a network flow model and focus on the bufferless and instantaneous rate at which peers transmit bits. A super peer can playback the video whenever it receives fresh content bits at rate r_1, and an ordinary peer can playback the video whenever it receives fresh content bits at rate r_2. Denote by $\mathbf{R}$ for the set of required rate in the system, which is $\mathbf{R} = \{r_1,\ r_2\}$. Denote by u_s for the upload capacity of s and u_{ij} for the upload capacity of p_{ij}. Let $\mathrm{u}(\cdot)$ be the function of upload capacity summation. For example, $\mathrm{u}(\mathbf{P}) = \sum_{\mathbf{P}} u_{ij} = \sum_{j=1}^{n_1} u_{1j} + \sum_{j=1}^{n_2} u_{2j}$. Let $\overline{u}$ be the average upload capacity of all peers. Let $\overline{u_1}$ and $\overline{u_2}$ be the average upload capacity of peers in class 1 and class 2. We have

$$\overline{u} = \frac{\mathrm{u}(\mathbf{P})}{|\mathbf{P}|} = \frac{\mathrm{u}(\mathbf{P}_1) + \mathrm{u}(\mathbf{P}_2)}{|\mathbf{P}_1| + |\mathbf{P}_2|} = \frac{\sum_{j=1}^{n_1} u_{1j} + \sum_{j=1}^{n_2} u_{2j}}{n_1 + n_2}$$

$$\overline{u_1} = \frac{\mathrm{u}(\mathbf{P}_1)}{|\mathbf{P}_1|} = \frac{\sum_{j=1}^{n_1} u_{1j}}{n_1}$$

$$\overline{u_2} = \frac{\mathrm{u}(\mathbf{P}_2)}{|\mathbf{P}_2|} = \frac{\sum_{j=1}^{n_2} u_{2j}}{n_2}$$

Notice the definition of universal streaming (US) in [13]. Similarly, in a two-class P2P streaming system, when all peers in $\mathbf{P}_1$ receiving video data no less than r_1 and all peers in $\mathbf{P}_2$ receiving video data no less than r_2, we say that the system provides universal streaming in two-class system (UST) or the system runs on UST. Denote by $u_{s\min}$ for the minimal server load of s to ensure system running on UST. Notations introduced in this paper are summarized in Table 1.

Table 1: Main notations used in this paper.

s	Source server.
p_{ij}	Peer j in the class j.
n	Total number of peers in the system.
n_i	Total number of peers in the class i.
$\mathbf{P}$	Set of all peers.
$\mathbf{P}_i$	Set of peers in the class i.
r_i	Required rate in the class i.
$\mathbf{R}$	Set of required rate in the system.
u_s	Source server upload capacity.
u_{ij}	Upload capacity of p_{ij}.
$\mathrm{u}(\cdot)$	Function of upload capacity Summation.
$\overline{u}$	Average upload capacity of all peers.
$\overline{u_i}$	Average upload capacity of peers in the class i.
$u_{s\min}$	Minimal server load for system running on UST.
u'_s	Rest of s available upload capacity.
u'_{gh}	Rest of p_{gh} available upload capacity.
u'_{all}	Total rest of available upload capacity.
s^o_{ij}	The flow rate of video substream from s to p_{ij}.
s^{gh}_{ij}	The flow rate of video substream from p_{gh} to p_{ij}.
tr_{ij}	The total rate of p_{ij} receiving video data.

3.2 Minimal Server Load

We give the proof of $u_{s\min}$ in this subsection. The proof is divided into two parts. We give a lower bound for $u_{s\min}$ in part one and prove UST can be supported with this bound in part two. Notice that part two is also an algorithm to achieve the minimal server load. We call this algorithm minimal server load algorithm (MSL) in the rest of this paper. Let $u_{s\min}$ denote minimal server load for a two-class P2P streaming system with transcoding providing UST, then

$$u_{s\min} = \max_i \left(r_i + \sum_{k=0}^{i-1} n_k (r_k - \overline{u_k}) \right), \tag{1}$$
$$n_0 = r_0 = \overline{u_0} = r_3 = 0, \quad i = 1,2,3.$$

Proof:

Part one:

Notice that for the whole system $\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2$, it is obviously $u_{s\min} \geq r_1$ and $u_{s\min} + \sum_{i=1}^{2} n_i (\overline{u_i} - r_i) \geq 0$. And, for the subsystem $\mathbf{P}_2$, $u_{s\min} + n_1 (\overline{u_1} - r_1) \geq r_2$, where the peers in class 1 at least costs $n_1 (r_1 - \overline{u_1})$ bandwidth.

Therefore

$$\begin{cases} u_{s\min} \geq r_1 \\ u_{s\min} \geq r_2 + n_1 (r_1 - \overline{u_1}) \\ u_{s\min} \geq \sum_{i=1}^{2} n_i (r_i - \overline{u_i}) \end{cases} \tag{2}$$

For convenience, let $n_0 = r_0 = \overline{u_0} = r_3 = 0$. Then, combining these three inequalities gives

$$u_{s\min} \geq \max_i \left(r_i + \sum_{k=0}^{i-1} n_k (r_k - \overline{u_k}) \right), \tag{3}$$
$$n_0 = r_0 = \overline{u_0} = r_3 = 0, \quad i = 1,2,3$$

It remains to show that if

$$u_s = \max_i \left(r_i + \sum_{k=0}^{i-1} n_k (r_k - \overline{u_k}) \right), \tag{4}$$
$$n_0 = r_0 = \overline{u_0} = r_3 = 0, \quad i = 1,2,3$$

then UST can be supported.

Part two:

Let $\mathbf{P}_0 = \{s\}$. Consider the subsystem $\mathbf{P}_0 \bigcup \mathbf{P}_1$ firstly.

When $r_1 \geq n_1 (r_1 - \overline{u_1})$, i.e. $r_1 \leq \frac{n_1 \overline{u_1}}{n_1 - 1}$.

Consider a video stream of rate r_1. Divide this video stream into n_1 substreams, with the jth substream having rate

$$s^o_{1j} = \frac{u_{1j} r_1}{\mathrm{u}(\mathbf{P}_1)} = \frac{u_{1j} r_1}{n_1 \overline{u_1}}, \quad for\ j = 1, \ldots, n_1$$

Notice that $\sum_{j=1}^{n_1} s^o_{1j} = r_1 \leq u_s$. So the source server can copy the jth substream to the p_{1j} respectively. Furthermore, because

$$(n_1 - 1) s^o_{1j} = \frac{(n_1 - 1) u_{1j}}{n_1 \overline{u_1}} \cdot r_1 \leq \frac{(n_1 - 1) u_{1j}}{n_1 \overline{u_1}} \cdot \frac{n_1 \overline{u_1}}{(n_1 - 1)} = u_{1j}$$

p_{1j} can copy its stream to each of the other $n_1 - 1$ peers in the class 1. Thus each super peer receives a substream from the source server and also receives $n_1 - 1$ additional substreams from the other $n_1 - 1$ super peers. The total rate at which p_{1j} receives is

$$tr_{1j} = s^o_{1j} + \sum_{k:k \neq j} s^o_{1k} = \sum_{j=1}^{n_1} s^o_{1j} = r_1$$

Hence, when $r_1 \geq n_1 (r_1 - \overline{u_1})$, the rate r_1 can be supported in the class 1.

When $r_1 < n_1 (r_1 - \overline{u_1})$.

In this situation, divide the video stream into $n_1 + 1$ substreams, with jth substream having rate

$$\begin{cases} s^o_{1j} = \frac{u_{1j}}{n_1 - 1}, & for\ j = 1, \ldots, n_1 \\ s^o_{1 n_1 + 1} = r_1 - \frac{\mathrm{u}(\mathbf{P}_1)}{n_1 - 1} = r_1 - \frac{n_1 \overline{u_1}}{n_1 - 1} \end{cases}$$

And the source server copy two substreams to each peer j in the class 1: the jth substream s^o_{1j} and the substream $s^o_{1 n_1 + 1}$. The source server can do this because

$$\sum_{j=1}^{n_1} s^o_{1j} + n_1 s^o_{1 n_1 + 1} = \frac{n_1 \overline{u_1}}{n_1 - 1} + n_1 \left(r_1 - \frac{n_1 \overline{u_1}}{n_1 - 1} \right) = n_1 (r_1 - \overline{u_1}) \leq u_s$$

Furthermore, because

$$(n_1 - 1) s^o_{1j} = (n_1 - 1) \cdot \frac{u_{1j}}{n_1 - 1} = u_{1j}, \quad for\ j = 1, \ldots, n_1$$

p_{1j} can copy its stream s^o_{1j} to each of the other $n_1 - 1$ peers in the class 1. Thus each super peer receives two substreams from the source server and also receives $n_1 - 1$ additional substreams from the other $n_1 - 1$ super peers. The total rate at which p_{1j} receives is

$$tr_{1j} = s^o_{1 n_1 + 1} + s^o_{1j} + \sum_{k:k \neq j} s^o_{1k} = s^o_{1 n_1 + 1} + \sum_{j=1}^{n_1} s^o_{1j} = \left(r_1 - \frac{n_1 \overline{u_1}}{n_1 - 1} \right) + \frac{n_1 \overline{u_1}}{n_1 - 1} = r_1$$

Hence, when $r_1 < n_1 (r_1 - \overline{u_1})$, the rate r_1 can be supported in the class 1 too.

Then, by above algorithm, whether $r_1 \geq n_1 (r_1 - \overline{u_1})$ or $r_1 < n_1 (r_1 - \overline{u_1})$, each super peer can get video data at rate r_1, and the total rest of available upload capacity is

$$u'_{\mathrm{all}} = u'_s + \sum_{h=1}^{n_1} u'_{1h} = u_s + n_1 (\overline{u_1} - r_1) \geq r_2$$

Next, we consider the whole system $\bigcup_{k=0}^{2} \mathbf{P}_k$.

When $r_2 \geq n_2 (r_2 - \overline{u_2})$, i.e. $r_2 \leq \frac{n_2 \overline{u_2}}{n_2 - 1}$.

Consider a video stream of rate r_2, which can get from source server and super peers. Divide this video stream into n_2 substreams, with the jth substream having rate

$$s^o_{2j} = \frac{u_{2j}}{\mathrm{u}(\mathbf{P}_2)} \cdot \frac{u'_s}{u'_{\mathrm{all}}} \cdot r_2 = \frac{u_{2j}}{n_2 \overline{u_2}} \cdot \frac{u'_s}{u_s + n_1 (\overline{u_1} - r_1)} \cdot r_2, \quad for\ j = 1, \ldots, n_2$$

And divide this video stream into n_i substreams, with the jth substream having rate

$$s^{1h}_{2j} = \frac{u_{2j}}{\mathrm{u}(\mathbf{P}_2)} \cdot \frac{u'_{1h}}{u'_{\mathrm{all}}} \cdot r_2 = \frac{u_{2j}}{n_2 \overline{u_2}} \cdot \frac{u'_{1h}}{u_s + n_1 (\overline{u_1} - r_1)} \cdot r_2, \quad for\ j = 1, \ldots, n_2$$

Notice that

$$\sum_{j=1}^{n_2} s^o_{2j} = \frac{\sum_{j=1}^{n_2} u_{2j}}{\mathrm{u}(\mathbf{P}_2)} \cdot \frac{u'_s}{u'_{\mathrm{all}}} \cdot r_2 = u'_s \cdot \frac{r_2}{u'_{\mathrm{all}}}$$

As $u'_{\mathrm{all}} \geq r_2$, we have $\sum_{j=1}^{n_2} s^o_{2j} \leq u'_s$. So the source server can copy s^o_{2j} to the $p_{2j} (j = 1, \ldots, n_2)$ respectively. And, as

$$\sum_{j=1}^{n_2} s^{1h}_{2j} = \frac{\sum_{j=1}^{n_2} u_{2j}}{\mathrm{u}(\mathbf{P}_2)} \cdot \frac{u'_{1h}}{u'_{\mathrm{all}}} \cdot r_2 = u'_{1h} \cdot \frac{r_2}{u'_{\mathrm{all}}} \leq u'_{1h}$$

$p_{1h} (h = 1, \ldots, n_1)$ can copy s^{1h}_{2j} to $p_{2j} (j = 1, \ldots, n_2)$ respectively. So $p_{2j} (j = 1, \ldots, n_2)$ gets

$$s^o_{2j} + \sum_{h=1}^{n_1} s^{1h}_{2j} = \frac{u_{2j} r_2}{\mathrm{u}(\mathbf{P}_2)}$$

Furthermore, because

$$(n_2 - 1) \cdot \left(s^o_{2j} + \sum_{h=1}^{n_1} s^{1h}_{2j} \right) = \frac{(n_2 - 1) u_{2j}}{\mathrm{u}(\mathbf{P}_2)} \cdot r_2$$
$$= \frac{(n_2 - 1) u_{2j}}{n_2 \overline{u_2}} \cdot r_2 \leq \frac{(n_2 - 1) u_{2j}}{n_2 \overline{u_2}} \cdot \frac{n_2 \overline{u_2}}{n_2 - 1} = u_{2j}$$

$p_{2j} (j = 1, \ldots, n_2)$ can copy $s^o_{2j} + \sum_{h=1}^{n_1} s^{1h}_{2j}$ to each of the other $n_2 - 1$ peers in the class 2. Thus each ordinary peer receives a substream from each super peer and the source server, and also receives n_2-1 additional substreams from the other n_2-1 ordinary peers. The total rate at which p_{2j} receives is

$$tr_{2j} = \sum_{j=1}^{n_2}\left(s_{2j}^{o} + \sum_{h=1}^{n_1} s_{2j}^{1h}\right) = \frac{\sum_{j=1}^{n_2} u_{2j}}{\mathrm{u}\left(\mathbf{P}_2\right)}\cdot r_2 = r_2$$

Hence, when $r_2 \geq n_2\left(r_2-\overline{u_2}\right)$, the rate r_2 can be supported in the class 2.

When $r_2 < n_2 r_2 - n_2\overline{u_2}$.

Consider a video stream of rate r_2, which can get from source server and super peers. Divide this video stream into n_2+1 substreams, with the jth substream having rate

$$\begin{cases} s_{2j}^{o} = \frac{u_{2j}}{n_2-1}\cdot\frac{u'_s}{u'_{\text{all}}} = \frac{u_{2j}}{n_2-1}\cdot\frac{u'_s}{u_s+n_1\left(\overline{u_1}-r_1\right)}, \\ \quad for\ j=1,\ldots,n_2 \\ s_{2n_2+1}^{o} = \left(r_2-\frac{\mathrm{u}(\mathbf{P}_2)}{n_2-1}\right)\cdot\left(\frac{u'_s}{u'_{\text{all}}}\right) \\ = \left(r_2-\frac{n_2\overline{u_2}}{n_2-1}\right)\cdot\left(\frac{u'_s}{u_s+n_1\left(\overline{u_1}-r_1\right)}\right) \end{cases}$$

And divide this video stream into n_2+1 substreams, with the jth substream having rate

$$\begin{cases} s_{2j}^{1h} == \frac{u_{2j}}{n_2-1}\cdot\frac{u'_{1h}}{u'_{\text{all}}} = \frac{u_{2j}}{n_2-1}\cdot\frac{u'_{1h}}{u_s+n_1\left(\overline{u_1}-r_1\right)}, \\ \quad for\ j=1,\ldots,n_2 \\ s_{2n_2+1}^{1h} = \left(r_2-\frac{\mathrm{u}(\mathbf{P}_2)}{n_2-1}\right)\cdot\left(\frac{u'_{1h}}{u'_{\text{all}}}\right) \\ = \left(r_2-\frac{n_2\overline{u_2}}{n_2-1}\right)\cdot\left(\frac{u'_{1h}}{u_s+n_1\left(\overline{u_1}-r_1\right)}\right) \end{cases}$$

Notice that

$$\sum_{j=1}^{n_2} s_{2j}^{o} + n_2 s_{2n_2+1}^{o} = \frac{n_2\left(r_2-\overline{u_2}\right)}{u_s+n_1\left(\overline{u_1}-r_1\right)}\cdot u'_s$$

As (4), we have $u_s \geq \sum_{k=1}^{2} n_k\left(r_k-\overline{u_k}\right)$, i.e.

$$u_s+n_1\left(\overline{u_1}-r_1\right) \geq n_2\left(r_2-\overline{u_2}\right)$$

So $\sum_{j=1}^{n_2} s_{2j}^{o} + n_2 s_{2n_2+1}^{o} \leq u'_s$, which means the source server can copy $s_{2j}^{o}+s_{2n_2+1}^{o}$ to the $p_{2j}(j=1,\ldots,n_2)$ respectively. And, as

$$\sum_{j=1}^{n_2} s_{2j}^{1h} + n_2 s_{2n_2+1}^{1h} = \frac{n_2\left(r_2-\overline{u_2}\right)}{u_s+n_1\left(\overline{u_1}-r_1\right)}\cdot u'_{1h} \leq u'_{1h}$$

$p_{1h}(h=1,\ldots,n_1)$ can copy $s_{2j}^{1h}+s_{2n_2+1}^{1h}$ to $p_{2j}(j=1,\ldots,n_2)$ respectively. So $p_{2j}(j=1,\ldots,n_2)$ gets

$$\left(s_{2j}^{o}+\sum_{h=1}^{n_1} s_{2j}^{1h}\right)+\left(s_{2n_2+1}^{o}+\sum_{h=1}^{n_1} s_{2n_2+1}^{1h}\right)$$

Furthermore, because

$$\left(n_2-1\right)\cdot\left(s_{2j}^{o}+\sum_{h=1}^{n_1} s_{2j}^{1h}\right) = \left(n_2-1\right)\cdot\left(\frac{u_{2j}}{n_2-1}\cdot\frac{u'_s}{u'_{\text{all}}}+\frac{u_{2j}}{n_2-1}\cdot\frac{\sum_{h=1}^{n_1} u'_{1h}}{u'_{\text{all}}}\right) = u_{2j}$$

$p_{2j}(j=1,\ldots,n_2)$ can copy $s_{2j}^{o}+\sum_{h=1}^{n_1} s_{2j}^{1h}$ to each of the other n_2-1 peers in the class 2. Thus each ordinary peer receives two substreams from each super peer and the source server, and also receives n_2-1 additional substreams from the other n_2-1 ordinary peers. The total rate at which p_{2j} receives is

$$\begin{aligned} tr_{2j} &= \sum_{j=1}^{n_2}\left(s_{2j}^{o}+\sum_{h=1}^{n_1} s_{2j}^{1h}\right)+s_{2n_2+1}^{o}+\sum_{h=1}^{n_1} s_{2n_2+1}^{1h} \\ &= \sum_{j=1}^{n_2}\left(\frac{u_{2j}}{n_2-1}\cdot\frac{u'_s}{u'_{\text{all}}}+\frac{u_{2j}}{n_2-1}\cdot\frac{\sum_{h=1}^{n_1} u'_{1h}}{u'_{\text{all}}}\right) \\ &+ \left(r_2-\frac{\mathrm{u}(\mathbf{P}_2)}{n_2-1}\right)\cdot\left(\frac{u'_s}{u'_{\text{all}}}\right) \\ &+ \left(r_2-\frac{\mathrm{u}(\mathbf{P}_2)}{n_2-1}\right)\cdot\left(\frac{\sum_{h=1}^{n_1} u'_{1h}}{u'_{\text{all}}}\right) = r_2 \end{aligned}$$

Hence, when $r_2 < n_2\left(r_2-\overline{u_2}\right)$, the rate r_2 can be supported in the class 2 too.

To sum up, when u_s satisfies (4), by the MSL algorithm, the system can provide video data at rate r_1 for each super peer and at rate r_2 for each ordinary peer, which means the whole system can run on UST. Considering (3), we have

$$\begin{aligned} & u_{s\min} = \max_i\left(r_i+\sum_{k=0}^{i-1} n_k\left(r_k-\overline{u_k}\right)\right), \\ & n_0 = r_0 = \overline{u_0} = r_3 = 0, \quad i=1,2,3 \end{aligned}$$

4 Simulation Experiment and Performance Analysis

In the simulation experiment, we implement P2P streaming system with transcoding and traditional no transcoding design described in the paper in C++. The network environment in the simulation follows Georgia Tech's Internet Topology Generator [18]. To conduct rigorous quantitative analysis of the systems under wide range of working conditions, we implement our experiments to emulate the characteristics of realistic systems with different parameters and a large number of test times. The practical algorithm in the simulation of OTS is based on [13,14,17], and TS is based on MSL. We mainly investigate the server load of TS using MSL algorithm, and compare the performance with OTS in

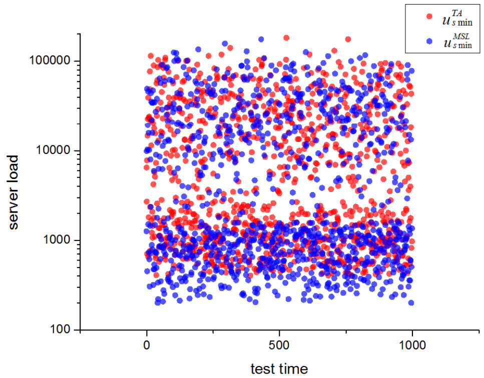

Fig. 3: Random test. (unit: Kbps)

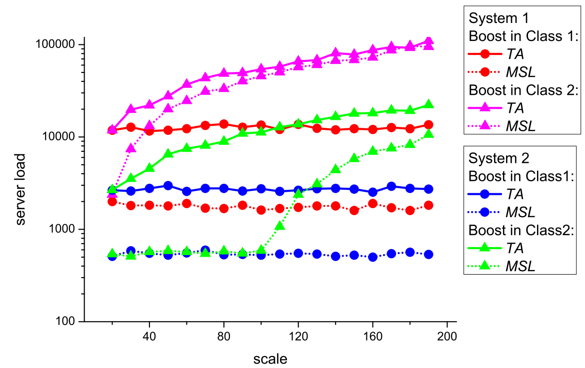

Fig. 4: $u_{s\min}^{TA}$ and $u_{s\min}^{MSL}$ change in different scales. (unit: Kbps)

various settings. For brevity, denote by $u_{s\min}^{TA}$ for the minimal server load in OTS simulation system using traditional algorithm (TA), and by $u_{s\min}^{MSL}$ for TS using MSL.

First, we take an overlook for the $u_{s\min}^{TA}$ and $u_{s\min}^{MSL}$ in complete random settings. The experiment tests two-class P2P streaming systems with all random settings, which randomly chooses n_i from 5 to 100, r_i from 200Kbps to 1600Kbps, and $\overline{u_i}$ from 100Kbps to 2000Kbps. We test OTS and TS 1000 times respectively. The results are shown in Figure 3. We can see that, with the random settings, $u_{s\min}^{TA}$ and $u_{s\min}^{MSL}$ spread all over the figure, but, notice of the lower part, $u_{s\min}^{TA}$ is hardly less than 400Kbps, while $u_{s\min}^{MSL}$ descends to 200Kbps in some situations. TS shows some advantages in this random test.

Table 2: Initial setting of systems.

Class	Type	Upload	Number	Rate
Super	LAN	2M	20	1600Kbps
Ordinary	ADSL	300K	20	800Kbps
Super	WIFI	1M	20	500Kbps
Ordinary	CDMA	100K	20	200Kbps

For more details, next, we simulate and test two practical two-class systems: the system 1 bases on LAN and ADSL, and the system 2 bases on WIFI and CDMA. The initial setting of the systems is on Table 2. In this test, we seek to know how server load changes by the different system scale. The peer number in any class is 20 initially. Then we add peer number in only one class every time, and measure the corresponding minimal server load. For example, as the green short dot curve in Figure 4, first, we test the System 2 and measure minimal server load in the initial setting as Table 2, after that, ordinary peer number is added to 30, 40, and so on, while other parameters are held constant, and we measure the minimal server load orderly. In the experiment, we compare the minimal server load in OTS and TS from the boost of both super and ordinary peers. The results are shown in Figure 4. The performance of server load in TS is better than the one in OTS, which follows the previous analysis.

5 Conclusion

In this paper, we have mathematically studied the performance of two-class transcoding P2P streaming systems. We have derived the performance bounds of minimal server load for this kind of systems, and given an algorithm to achieve the minimal server load. Besides mathematical investigation, we have also done some simulation experiments to compare the minimal server load of transcoding system with traditional situation, and to know how much benefit new design takes in various situations. The results have shown the advantage of transcoding system by our algorithm.

Acknowledgement

This work is partially supported by The National Basic Research Program of China (973 Program) (2012CB315803, 2012CB315806), The National Natural Science Foundation of China (61402065, 61379159), and Natural Science Foundation Project of Chongqing (cstc2012jjA40060). The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] J. Li, P. A. Chou, and C. Zhang, Mutualcast: an efficient mechanism for content distribution in a p2p network, Proc. ACM SIGCOMM Conference, (2005).

[2] N. Magharei, R. Rejaie, and Y. Guo, Mesh or multiple-tree: A comparative study of P2P live streaming approaches, Proc. IEEE INFOCOM, (2007).

[3] T. Wolf, In-network service for customization in next generation networks, vol. 24, no. 4, 6-12 (IEEE Network, 2010).
[4] M.-T. Lu, J.-C. Wu, K.-J. Peng, P. Huang, J. J. Yao, and H. H. Chen, Design and evaluation of a P2P IPTV system for heterogeneous networks, vol. 9, no. 8, 1568-1579 (Multimedia, IEEE Trans., 2007).
[5] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, A Measurement Study of a Large-Scale P2P IPTV System, vol. 9, no. 8, 1672-1687 (Multimedia, IEEE Trans., 2007).
[6] F. Chen, T. Repantis, and V. Kalogeraki, Coordinated media streaming and transcoding in peer-to-peer systems, Proc. IEEE Int. Parallel Distributed Process. p. 56b (2005).
[7] D. Liu, E. Setton, B. Shen, and S. Chen, PAT: Peer-assisted transcoding for overlay streaming to heterogeneous devices, Proc. Int. Workshop Netw. Oper. Syst. Support Dig. Audio Video (2007).
[8] S. R. Narayanan, D. Braun, J. Buford, R. S. Fish, A. D. Gelman, A. Kaplan, R. Khandelwal, E. Shim, and H. Yu, Peer-to-peer streaming for networked consumer electronics, vol. 45, no. 6, 124-131 (Commun., IEEE Mag., 2007).
[9] J. Noh, M. Makar, and B. Girod. Streaming to mobile users in a peer-to-peer network. Proc. International Mobile Multimedia Communications Conference (2009).
[10] Atenas M, Canovas A, Garcia M, Lloret J. IPTV transcoding to avoid network congestion. Proc. The Sixth International Conference on Networking and Services (2010).
[11] Zixia Huang, Chao Mei, Li Erran Li and Thomas Woo, CloudStream: delivering high-quality streaming video through a cloud-based H.264/SVC proxy, Proc. IEEE INFOCOM, (2011).
[12] J.-C. Wu, P. Huang, J. J. Yao, and H. H. Chen, A collaborative transcoding strategy for live broadcasting over peer-to-peer IPTV networks, vol. 21, no. 2, 220-224 (Circuits Syst. Video Technol, IEEE Trans., 2011).
[13] R. Kumar, Y. Liu, and K. W. Ross, Stochastic Fluid Theory for P2P Streaming Systems, Proc. IEEE INFOCOM, (2007).
[14] S. Liu, R. Z. Shen, W. Jiang, J. Rexford, and M. Chiang, Performance Bounds for Peer-Assisted Live Streaming, Proc. ACM SIGMETRICS (2008).
[15] C. Zhao, X. Lin, and C Wu, The streaming capacity of sparsely-connected P2P systems with distributed control, Proc. IEEE INFOCOM, 1449-1457 (2011).
[16] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P.A. Chou, Peer-to-Peer Streaming Capacity, vol. 57, no. 8, 5072-5087 (Information Theory, IEEE Trans., 2011).
[17] C. Feng and B. Li, On large-scale peer-to-peer streaming systems with network coding, Proc. The 16th ACM international conference on Multimedia, 269-278 (2008).
[18] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, How to model an internetwork, Proc. IEEE INFOCOM, 594-602 (1996).

Shuai Zeng was born in 1982. He received the Ph.D. degree in communication and information engineering from University of Electronic Science and Technology of China in 2013. He is now a lecturer at Chongqing University of Posts and Telecommunications. His research interests include mobile communication, future Internet, content distribution, P2P and network optimization.

Lemin Li was born in 1932. He is a professor and doctoral supervisor with the School of Communication and Information Engineering, UESTC. He is an Academician of Chinese Academy of Engineering. His researches areas covers communication networks and data communications.

Dan Liao received the Ph.D. degree from University of Electronic Science and Technology of China in 2007. He has been an associate professor at UESTC since 2009. His current research interests include P2P networks, wireless networks and cloud computing.