A Novel TOPSIS-based Chunk Scheduling Approach for Layered P2P Streaming

Wei Chen, Sen Su, Fangchun Yang, Kai Shuang and Xinchao Zhao
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
chenwei348@gmail.com, {susen, fcyang, shuangk}@bupt.edu.cn, xcmmrc@gmail.com

***Abstract*—Although layered P2P streaming is perfectly adapted to heterogeneous network environments and heterogeneous user requirements, it suffers from bad delay performance like single-layer P2P streaming. In this paper, we analyze new characteristics of Pull-based P2P chunk scheduling problem caused by layered coding, and propose a Pull-based scheduling problem model aimed at enhancing the delay performance under the guarantee of video quality for layered P2P streaming. And then we put forward a heuristic chunk scheduling algorithm with aperiodic scheduling interval, where technique for order preference by similarity to ideal solution (TOPSIS) is utilized to solve several multiple-attribute decision making problems. Finally, we develop a new metric comprehensively evaluating video playback quality and delay performance, and simulations illustrate that our algorithm can greatly outperform the existing classical related work by a small increase in control overhead.**

Keywords-layered P2P streaming; heterogeneous peers; chunk scheduling; delay performance; MADM-TOPSIS

I. INTRODUCTION

P2P Streaming wins a big success in the large scale streaming systems in recent years due to low deployment cost and high scalability. And the development of Mobile access technologies (like 3G and LTE) and widespread adoption of broadband residential access make it possible that computers, mobile phones and set-top boxes share video streaming in Peer-to-Peer in the emerging scenario, i.e., heterogeneous network resources (especially bandwidth resources) and heterogeneous user requirements (especially subject to user terminals' limit, like display capacities). Layered coding is a promising solution adapted to the emerging scenario [3]. Accordingly, layered P2P streaming has drawn great interest in recent years [6][7][8][9][12].

Mesh-Pull P2P streaming has been proved with higher practicability, scalability, capability of coping with peer churn and bandwidth utilization than other P2P solutions by the success deployments of many commercial softwares, like PPStream [1], PPLive [2] and so on. However, it suffers from bad delay performance [4][5], and this problem would be much worse in layered P2P streaming. The delay origins from two aspects: overlay construction and chunk scheduling. On one hand, there are many related work [6][7] to focus on the overlay construction to meet QoS requirements. On the other hand, much work about chunk scheduling for layered P2P streaming aims to maximize the system's throughput [6][8]. However, to the best of our knowledge, there is no work about Pull-based chunk scheduling to optimize its delay performance under the guarantee of video quality for layered P2P streaming.

In this paper, we focus on the Pull-based chunk scheduling problem of layered P2P streaming to enhance the delay performance under the guarantee of video quality in the emerging scenario of heterogeneous upload and download bandwidth, heterogeneous and dynamic propagation delays, and heterogeneous requirements of video quality, where computers, mobile phones and set-top boxes share video streaming. First, we analyze new challenges brought by layered coding to Mesh-Pull P2P streaming, and then propose a chunk scheduling problem model to minimize the delivery time in a scheduling interval. Secondly, we put forward a TOPSIS-based, variable scheduling interval and heuristic Pull-based scheduling algorithm. Simulations illustrate that our algorithm can outperform the existing classical solutions.

The rest of the paper is organized as follows. Section II presents the related work. Section III describes new challenges and chunk scheduling problem model of layered P2P streaming. Section IV introduces our scheduling algorithm. We show the simulation results in Section V. Finally, Section VI presents our conclusion.

II. RELATED WORK

We briefly review the main chunk scheduling solutions for layered P2P streaming.

PALS [9] focused on the Mesh-Pull chunk scheduling problem, adopted a diagonal chunk priority order method and employed a Round-Robin method to request chunk. [10] aimed at optimizing the streaming transmission performance in mobile ad-hoc network. LION [11] utilized alternative paths and network coding to improve throughput. [12] studied the video-on-demand media distribution problem and put forward a peer-to-peer streaming solution which focused on how to optimally allocate the desired layers among multiple senders. In brief, these works aimed at maximizing the system throughput or delivering the satisfying number of layers according to available bandwidths. However, they ignored an important user experience, i.e., startup delay, and they did not simultaneously optimize delay performance and throughput so as to have longer startup delay. Therefore, we expect to decrease the startup delay caused by chunk scheduling under the guarantee of video play quality.

 ISBN: 978-1-61208-103-8

III. CHUNK SCHEDULING PROBLEM MODEL

In this section, we will analyze new characteristics of layered P2P streaming, and then propose a problem model for layered P2P streaming.

A. New Characteristics of Layered P2P Streaming

Layered coding [3] can be adapted to the emerging scenario where mobile terminals, personal computers and set-top boxes share the video streaming in Peer-to-Peer way. And compared with traditional P2P streaming, layered P2P streaming has new characteristics as follows:

- Layer characteristic of a chunk: layer characteristic should be considered because decoding of chunks in upper layer depend on that chunks with the same time identifier have been obtained in all lower layers.
- Heterogeneous video quality requirements: peers have diverse video quality (i.e., number of layers) requirements because of limits of different download bandwidths, different terminals' screen sizes and so on.
- Congestion in download bandwidths of peers: study on traditional P2P streaming in Internet usually assumes that download links of peers are not the bottleneck link [5] so that the number of neighbor peers delivering chunks is generally not limited. However, in the emerging scenario vast diversity among peers' download and upload bandwidths results in that download links of peers may be the bottleneck links. Therefore, a peer can only receive chunks simultaneously from limited number of neighbor peers in a scheduling interval so as not to the congestion in its download link.

These new characteristics must be considered in the model of chunk scheduling problem.

B. Problem Model for Layered P2P Streaming

In Pull-based chunk scheduling approach, each peer generally requests absent chunks in its request window from its neighbor peers autonomously and periodically. We pay close attention to how to get an optimal chunk assignment for a peer in a scheduling interval in which absent chunks as more as possible can be obtained in the shortest time, in order to improve the delay performance under the guarantee of video quality. For the scheduling problem of layered P2P streaming, we will consider the following factors:

- Absent chunks of a peer and their availabilities in its neighbor peers.
- Chunks' time and layer characteristics.
- Heterogeneous video quality requirements of peers.
- Heterogeneous and constant upload and download bandwidth of peers (It can be extended to variable bandwidth scenario by utilizing bandwidth measurement or prediction approaches [19], which is not what we cares about.)
- Congestion avoidance by limiting the number of neighbor peers delivering chunks.
- Heterogeneous and dynamic propagation delays among peers.

Suppose decision variable $\theta_{iljk} \in \{0,1\}$, "$\theta_{iljk}$ =1" means that scheduling peer P_r (any peer in the system) assigns the chunk C_{jk} to the neighbor P_i, and P_i will send C_{jk} as the l th of assigned chunks to P_r; otherwise "θ_{iljk} =0".

According to above consideration, we model Pull-based scheduling problem for layered P2P streaming in merging scenario as an integer programming problem to minimize the delivery time of absent chunks (i.e., the objective function) under the condition of ensuring the number of delivered chunks as more as possible (Constraint k)):

$$\min\{T\} \tag{1}$$

s.t.

$$a) \quad T = \max\{T_1, T_2, ..., T_{N-1}, T_N\}$$

$$b) \quad T_i = d_{ri} + d_{ir} + XNum_i / B_{ir}$$

$$c) \quad Num_i = \sum_{l=1}^{|S_r|} \sum_{j=p_r}^{p_r+W_B+W_C} \sum_{k=1}^{K} \theta_{iljk}$$

$$d) \quad \theta_{iljk} \le H_{ijk}$$

$$e) \quad t_{ijk} = \sum_{l_i=1}^{|S_r|} \sum_{s=1}^{N} (d_{ri} + d_{ir} + Xl_i / B_{ir}) \theta_{sl_i jk}$$

$$f) \quad t_{ijk} < Deadline(C_{jk})$$

$$g) \quad \sum_{i=1}^{N} \sum_{l=1}^{|S_r|} \theta_{iljk} \le 1$$

$$h) \quad \sum_{j=p_r}^{p_r+W_B+W_C} \sum_{k=1}^{K} \theta_{iljk} \le 1$$

$$i) \quad \theta_{iljk} \le \theta_{i(l-1)jk}$$

$$j) \quad \theta_{iljk} \le \theta_{ilj(k-1)}$$

$$k) \quad \sum_{i=1}^{N} \sum_{l=1}^{|S_r|} \sum_{j=p_r}^{p_r+W_B+W_C} \sum_{k=1}^{K} Priority(C_{jk}) \theta_{iljk} \ge \alpha \sum_{j=p_r}^{p_r+W_B+W_C} \sum_{k=1}^{K} Priority(C_{jk}) M_{jk}$$

$$l) \quad \sum_{i=1}^{N} B_{ir} g(Num_i) \le D_r$$

$$m) \quad g(Num_i) = \begin{cases} 1, & \text{if } Num_i \ge 1 \\ 0, & \text{if } Num_i = 0 \end{cases}$$

The symbols are defined in TABLE I. The delivery time in this scheduling interval is the maximum of delivery times of N neighbor peers, as shown in Constraint a). Constraint b) introduces that the propagation delay of request message, transmission delay of chunks and propagation delay of chunks should be considered when evaluating the delivery time of chunks assigned to P_i. Constraint c) shows the number of chunks assigned to neighbor P_i. Constraint d) requires chunk C_{jk} can be assigned to neighbor P_i only if the neighbor P_i has the chunk C_{jk}. Constraint e) introduces how to quantify the delivery time of a chunk C_{jk}, like the quantification of T_i. Constraint f) requires that chunk C_{jk} must be obtained in its deadline. Constraint g) requires a chunk can be assigned to one neighbor peer at most. Constraint h) requires a neighbor P_i delivers one chunk at most to P_r in l th sequence. Constraint i) requires a neighbor

 ISBN: 978-1-61208-103-8

TABLE I. NOTATIONS

Symbols	**Description** (all the symbols are confined to a scheduling interval; delivery time is the absolute time; time unit, i.e., K*X/Rate, is the absolute time of K chunks with the same time identifier due to the layered partition of video.)
T	Delivery time of chunks from requested to obtained by scheduling peer P_r
T_i	Delivery time of chunks assigned to neighbor P_i from requested to obtained by scheduling peer P_r
d_{ri}	Propagation delay from scheduling peer P_r to neighbor P_i
d_{ir}	Propagation delay from neighbor P_i to scheduling peer P_r
Num_i	Number of chunks assigned to neighbor P_i
C_{jk}	Identifier of chunk representing jth time unit and kth layer (K is the maximum number of video layers; J is the maximum number of time units of video)
$H_{ijk} \in \{0,1\}$	"H_{ijk} =1" denotes neighbor P_i has the chunk C_{jk}; otherwise, "H_{ijk} =0"
t_{ijk}	Delivery time of chunk C_{jk} by neighbor P_i
X	Size of a chunk
Rate	Bit rate of video playback with K layers
B_{ir}	Available upload bandwidth from neighbor P_i to scheduling peer P_r
D_r	Download bandwidth of scheduling peer P_r
Deadline(C_{jk})	Playback deadline (absolute time) of chunk C_{jk}
Priority(C_{jk})	Priority value of chunk C_{jk}
$p_r \in \{1,2,\ldots,J\}$	Number of time units of chunk being about to be played by scheduling peer P_r
W_A	Number of time units held by A area
W_B	Number of time units held by B (i.e., urgent) area
W_C	Number of time units held by C (i.e., loose) area
S_r	Set of absent chunks in request window of scheduling peer P_r (S_r ={ C_{jk} | H_{ijk} =0, k⩽K, p_r⩽j⩽p_r + W_B + W_C })
$M_{jk} \in \{0,1\}$	"M_{jk} =1" denotes chunk C_{jk} can be provided by some neighbor peer (i.e., $\sum^{N}_{i=1} H_{ijk} \geqslant 1$); otherwise, "$M_{jk}$ =0"

P_i can deliver a chunk in higher sequence to P_r only if there are assigned chunks in all the lower sequences of P_i in order to reduce the delivery time of chunk. Constraint j) requires a chunk can be assigned to some neighbor only if all the chunks in the lower layers have been assigned, in order to decrease the number of chunks which cannot be decoded. Constraint k) requires that sum of all the assigned chunks' priorities should be bigger than α times of sum of all the chunks' priorities, which can be provided by neighbors, which ensures the video play quality; Constraint l) requires sum of upload bandwidths of neighbors delivering chunks to P_r is less than or equal to the download bandwidth of P_r so as not to the congestion in its download link. Constraint m) shows "$g(Num_i)$ =1" represents neighbor P_i will deliver chunks to P_r, otherwise "$g(Num_i)$ =0".

Formula (1) is called distributed and local delay-optimum chunk scheduling problem model under the guarantee of video quality for layered P2P streaming. The integer programming problem is a NP-hard problem with $N^{|Sr|}$ combination solutions [13], where $|S_r|$ is always a large number. Therefore, we propose a distributed and heuristic scheduling algorithm to solve the problem.

IV. TOPSIS-BASED SCHEDULING ALGORITHM

According to the above problem model, we propose a heuristic and TOPSIS-based chunk scheduling algorithm for layered P2P streaming. First, we'd like to emphasize its four important problems:

1) Priority ordering problem: Considering the characteristics of chunks in layer and time (i.e., some chunks must be obtained as soon as possible, and other chunks can be obtained later), scheduling peer P_r should order its absent chunks by their importance to P_r and assign the chunk to some neighbor peer one by one following the chunk priority sequence;

2) Candidate neighbor selection problem: Due to the limit of download bandwidth, P_r should select candidate neighbor peers used to deliver chunks from its neighbor peers in order to avoid the congestion;

3) Chunk assignment problem: P_r should assign a chunk to a neighbor which deliver the chunk and chunks assigned in the shortest time;

4) Aperiodic scheduling interval: too long or too short scheduling interval would result in the longer delay or more repeated chunks, therefore we develop an aperiodic scheduling interval based on delivery time of absent chunks in each scheduling interval to reduce the delay or number of repeated chunks.

TOPSIS [14] is utilized in the following design to solve the MADM problems. Due to the limit of space, we do not introduce the TOPSIS in detail. As described in [14], we can quantify and order the alternatives according to performance data for n alternatives, attributes and their weights.

B. Chunks' Priorities Ordering

For layered coding, the video stream is encoded into several layers, and each layer is partitioned into chunks. Thus each chunk (C_{jk}) has a layer ID (i.e., k) and time unit ID (i.e., j). Buffer, shown in Figure.1, is divided into three parts: A area, B area (also called urgent area or playback window) and C area (also called loose area). A area stores video chunks just played. Urgent area is close to playback point, and each new peer cannot start playing the video until obtaining the large part of or all the chunks in this area. Request window is composed of B and C area and a peer

 ISBN: 978-1-61208-103-8

hopes to request absent chunks in request window from its neighbor peers in each scheduling interval.

Quantifying and ordering the absent chunks is a MADM problem and we adopt TOPSIS to solve the problem. According to characteristics of layered coding, four attributes should be considered as follows:

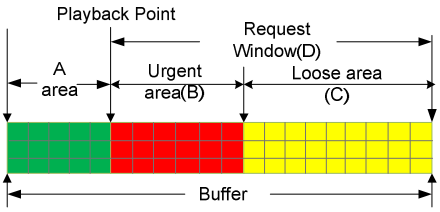

Figure 1. Design of Buffer

- Layer ID of chunk C_{jk} (denoted as $Property_1(C_{jk})$): the priority of C_{jk} increases with the decrease of its layer ID.
  $Property_1(C_{jk})= k$ (2)
- Number of chunks which cannot be decoded due to the absence of chunk C_{jk} (denoted as $Property_2(C_{jk})$): the priority of the chunk increase as the decrease of its $Property_2(C_{jk})$.
  $Property_2(C_{jk})= \sum^{K}_{s=k+1} H_{rjs}$ (3)
- Number of neighbors having chunk C_{jk} (denoted as $Property_3(C_{jk})$): the priority of C_{jk} increases with the decrease of its $Property_3(C_{jk})$, because this strategy can achieve good performance [15][16].
  $Property_3(C_{jk})= \sum^{N}_{s=0} H_{sjk}$ (4)
- Number of time units of the playback deadline of chunk C_{jk} (denoted as $Property_4(C_{jk})$): the priority of C_{jk} increases with the decrease of its $Property_4(C_{jk})$.
  $Property_4(C_{jk})=j-p_r$ (5)

Considering the characteristics of urgent and loose areas, we design the chunk priority algorithm as follows:

1) The priorities of chunks in urgent area are higher than ones of chunks in loose area, because chunks in urgent area are closer to playback point.

2) Scheduling objective of chunks in urgent area is to reduce the number of chunks which cannot be decoded, and to obtain video chunks with more layers (i.e., higher video quality). Therefore, we consider three attributes: $Property_1$、$Property_2$ and $Property_4$. And the corresponding weights are ω_{B1}、ω_{B2} and ω_{B4}, where $\omega_{B2} > \omega_{B4} > \omega_{B1}$ ($\omega_{B2}+\omega_{B1}+\omega_{B4}=1$). ω_{B2} is the largest value in order to decrease the number of chunks which cannot be decoded to improve the video play quality. The set of absent chunks in this area is S_{rB} ($=\{C_{jk} \mid H_{rjk}=0, k\leqslant K, p_r\leqslant j\leqslant p_r + W_B, M_{jk}=1\}$). According to TOPSIS, performance data a_{nm} ($n=|S_{rB}|$, $m=3$) of chunks obtained by Formula (2,3,5), monotonicity and weight of each attribute are put together to quantify the chunks' priorities, denoted as $Priority_{rB}(C_{jk})$, and order the chunks.

3) Scheduling objective of chunks in this area is to obtain more chunks so that these chunks with a high video quality can enter the urgent area. Therefore, we consider three attributes: $Property_1$、$Property_3$ and $Property_4$. And the corresponding weights are ω_{C1}、ω_{C3} and ω_{C4}, where $\omega_{C3} > \omega_{C1}$ and $\omega_{C3} > \omega_{C4}$ ($\omega_{C1}+\omega_{C3}+\omega_{C4}=1$). ω_{C3} is the largest value to increase the number of assigned chunks. The larger ω_{C1} leads to higher video playback quality while the larger ω_{C4} brings higher playback continuity. Therefore, ω_{C1} and ω_{C4} are a tradeoff between playback quality and playback continuity. The set of absent chunks is S_{rC} ($=\{C_{jk} \mid H_{rjk}=0, k \leqslant K, p_r+W_B\leqslant j\leqslant p_r+W_B+W_C, M_{jk}=1\}$). According to TOPSIS, performance data a_{nm} ($n=|S_{rC}|$, $m=3$) of chunks obtained by Formula (2,4,5), monotonicity and weight of each property are put together to quantify the chunks' priorities, denoted as $Priority_{rC}(C_{jk})$, and order the chunks.

Utilizing the above chunk priority algorithm, the priorities of chunks in S_r ($=S_{rB}\cup S_{rC}$) can be quantified, and we can obtain a chunk sequence $(C_1,C_2,\ldots,C_S)$ with $S=|S_r|$.

C. Candidate Neighbor Selection

Scheduling peer should choose as more neighbor peers with higher performances as possible as candidate neighbor peers whose upload bandwidths' sum is less than or equal to its download bandwidth to avoid the congestion.

We adopt TOPSIS method to quantify neighbors' importance index I_{ri} (i.e., the importance degree of neighbor peer P_i to scheduling peer P_r). Two factors affecting I_{ri} are the chunks owned by neighbor peer and the capability of neighbor peer delivering chunks.

First, we adopt the sum of priorities of chunks needed by P_r and owned by P_i to evaluate the first factor, because different chunks have different priorities or importance for P_r. Considering the design of urgent area and loose area, we take sum of priorities of chunks owned by P_i in urgent area and sum of priorities of chunks owned by P_i in loose area as two attributes (PR_{Bi} and PR_{Ci}):

$$PR_{Bi} = \sum_{C_{jk}\in S_{rB}} H_{ijk}\,Priority_{rB}(C_{jk}) \quad (6)$$

$$PR_{Ci} = \sum_{C_{jk}\in S_{rC}} H_{ijk}\,Priority_{rC}(C_{jk}) \quad (7)$$

The corresponding weights of PR_{Bi} and PR_{Ci} are ω_B and ω_C, where $\omega_B > \omega_C$. That is because chunks in urgent area are closer to playback point and more important to video play quality. With the rise of ω_B or ω_C, I_{ri} increases.

Secondly, the capability of neighbor peer P_i delivering a chunk (denoted as $Performance_i$) is the third attribute, which is evaluated by the propagation delay of request message, the propagation delay of a chunk and transmission delay of P_i:

$Performance_i=X/B_{ir}+d_{ri}+d_{ir}$ (8)

The I_{ri} of neighbor peer decreases with the increase of $Performance_i$. The corresponding weight of this attribute is ω_p. Candidate neighbor peer selection is greatly related to the current scheduling situation and the number of chunks owned by candidate neighbor peers affects the number of chunks which can be delivered, so $\omega_B+\omega_C > \omega_p$ ($\omega_B+\omega_C+\omega_p=1$).

According to TOPSIS, performance data a_{nm} of neighbor peers ($n=N$, $m=3$), monotonicity and weight of each attribute

are put together to quantify the I_{ri}.

By I_{ri}, we can order the neighbor peers and obtain the decreasing sequence ($P_1,P_2,\ldots,P_N$). The candidate neighbor selection algorithm is to choose the first N_r neighbor peers so that sum of these N_r neighbors' upload bandwidths is equal to scheduling peer's download bandwidth. Maybe, the upload bandwidth of P_{Nr} cannot be fully utilized due to the limit of scheduling peer's download bandwidth. Therefore, the set $\{P_1,P_2,\ldots,P_{Nr}\}$ is the candidate neighbor peers set.

D. Delay-Optimum Chunk Assignment

For the chunk sequence ($C_1,C_2,\ldots,C_{|Sr|}$) and candidate neighbor peers set $\{P_1,P_2,\ldots,P_{Nr}\}$, scheduling peer choose a neighbor peer in $\{P_1,P_2,\ldots,P_{Nr}\}$ for each chunk one by one following the chunk sequence. Due to the same chunk size, the delivery time of a chunk only depends on the transmission bandwidth of a neighbor peer and propagation delay between the neighbor and scheduling peer. Therefore, delay-optimum chunk assignment algorithm is that scheduling peer greedily chooses the neighbor peer which meets the formula (9) (i.e., choosing the neighbor peer which delivers the chunk in the shortest time) for each chunk according to the chunk sequence.

$$\min\{t_{ijk}\} = \min\{\sum_{l=1}^{|S_r|}\sum_{s=1}^{N}(d_{ri} + d_{ir} + Xl_i / B_{ir})\theta_{sljk}\} \quad (9)$$

E. Aperiodic Scheduling

Pull-based chunk scheduling algorithms [9][15] usually adopt constant periodical scheduling interval. That's because they cannot quantify the delivery time of each scheduling. If the interval is too long, there exists a span when upload bandwidths of neighbor peers and download bandwidth of the scheduling peer are idle so as to decrease the utilization of bandwidths and increase the video play delay. And if the interval is too short, the scheduling peer may request chunks which have been requested in the last scheduling interval so as to waste the bandwidths of neighbor peers and scheduling peer and also increase the play delay.

In our algorithm, we adopt the aperiodic scheduling interval, and take the delivery time (i.e., $T=\max\{T_1,T_2,\ldots,T_{Nr}\}$) of chunks in a scheduling as the interval between this scheduling and next scheduling. This will reduce the idle time of upload and download bandwidths and the number of repeatedly requested chunks to further improve the delay performance.

V. Simulation and Evaluation

To validate and evaluate our scheduling algorithm, we have conducted extensive simulations based on PeerSim [17]. We only focus on the chunk scheduling algorithm, and so adopt the same overlay construction approach [7]. In the beginning, 100 peers join in the system and make up a mesh with eight neighbor peers. The video with 100s duration is divided into chunks with the same size 1250byte, close to a maximum of MTU, and encoded ten layers with 100kbps of each layer. The buffer has 10s duration with request window of 8s duration. A new peer joins in the system every two seconds and an online peer quits the system every three seconds. Propagation delays among peers are randomly assigned from the delay matrix (2500*2500) in the Internet measurement [18] and reassigned every five seconds. Due to the unpredictability of TCP retransmission delay, UDP is adopted. And we assume packet loss ratio is 2%. The streaming server's upload bandwidth is 2Mbps and the bandwidth distribution and desired video qualities for three kinds of peers are shown in TABLE II.

TABLE II. Bandwidth Distribution and Desired Video Quality

	Mobile terminals	**PCs**	**Set-top boxes**
Upload(kbps)	300	600	1000
Download(kbps)	2000	4000	8000
Video quality	2 layers	6 layers	10 layers
Ratio	30%	40%	30%

A. Metrics

For layered streaming, we adopt the following metrics:

- Layer delivery ratio: Ratio of the number of a peer's video playback layers to one of its desired layers. This metric reflects the peer's video playback quality.
- Useless chunks ratio: Ratio of chunks unable to be decoded for a peer. This metric should be kept low.
- Number of control messages: Number of a peer' control messages, like chunk availability messages, request messages, maintenance messages for peers' departure. This metric reflects the control overhead.
- (λ_1, λ_2)-Startup Delay: A peer usually starts to playback the video after obtaining all or the large part of chunks in playback window. With the rise of playback window, a peer has more time to request the absent chunks so as to have a better video quality, however, have a longer startup delay; otherwise, a peer has a worse video playback quality and a shorter startup delay. This metric represents the minimal startup delay for a peer when it enjoys the video quality with λ_1 layer delivery ratio and λ_2 useless chunk ratio. This metric comprehensively reflects a peer's video quality and delay performance.

B. Effects of algorithm parameters

In our chunk scheduling algorithm, there are several important parameters: the weights [ω_{B2} ω_{B4} ω_{B1}], the weights [ω_{C3} ω_{C4} ω_{C1}] and the weights [ω_B ω_C ω_P].

Figure.2 shows the cumulative distribution function of peers' (0.999,0.001)-Startup Delay with different [ω_{B2} ω_{B4} ω_{B1}], and the performance decreases by the configuration 2, 7, 4, 1, 3, 5, 6. For different configurations, peers have better delay performance with larger ratio of ω_{B2} or ω_{B4}, like configuration 1,2,3,4 and 7; peers have worse delay performance with larger ratio of ω_{B1}. That is because for chunks in urgent area ω_{B2} affects useless chunk ratio, and lower useless chunk ratio may result in higher utilization of download bandwidth, and then higher video playback quality; the larger ω_{B4} makes scheduling peer prioritily request chunks closer to playback point, which improves the video playback quality; the larger ω_{B1} makes scheduling peer prioritily request chunks in lower layers and yet ignore the

 ISBN: 978-1-61208-103-8

absent chunks to be played soon, and then results in a worse video play video, like configuration 5 and 6. Therefore, $\omega_{B2} > \omega_{B1} > \omega_{B4}$ can achieve better delay performance under the guarantee of high video quality and the configuration [0.5 0.3 0.2] is adopted in the following simulation.

Figure. 3 shows the cumulative distribution function of peers' (0.999, 0.001) –Startup delay with different [ω_{C3} ω_{C4} ω_{C1}], and the performance decreases by the configuration 2, 1, 6, 3, 4, 5. When ω_{C3} is bigger, peers have better delay performance, like configuration 1, 2 and 6; otherwise, even if ω_{C3} or ω_{C1} is bigger, peers cannot achieve better delay performance, like configuration 3, 4 and 5. That is because the bigger ω_{C3} is illustrated in [15][16] to achieve better delivery performance. Therefore, $\omega_{C3} > \omega_{C1}$ and $\omega_{C3} > \omega_{C4}$ can achieve better delay performance under the guarantee of high video quality and the configuration [0.8 0.1 0.1] is adopted in the following simulation.

Figure. 4 shows the cumulative distribution function of peers' (0.999, 0.001) –Startup delay with different [ω_B ω_C ω_P], and the performance decreases by the configuration 7, 6, 5, 4, 3, 1, 2. Three weights are important factors of

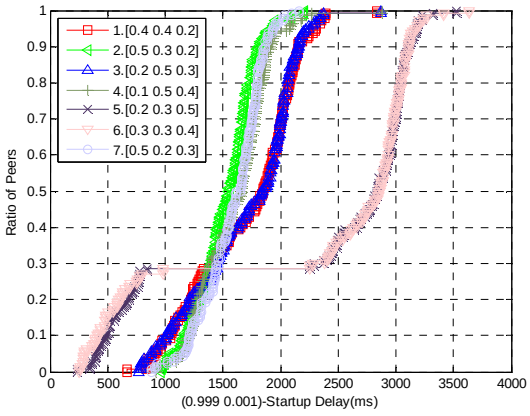

Figure 2. CDF of peers' (0.999,0.001)-Startup Delay with different [ω_{B2} ω_{B4} ω_{B1}]

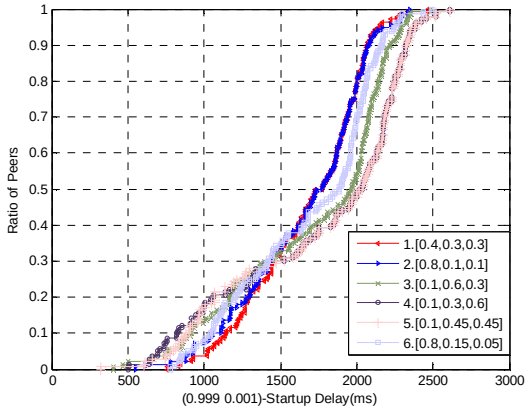

Figure 3. CDF of peers' (0.999,0.001)-Startup Delay with different [ω_{C1} ω_{C3} ω_{C4}]

importance index and when each of them is smaller, peers have the worse delay performances, e.g., configuration 5, 6, 7 are better than configuration 1, 2, 3, 4. That is because ω_B, ω_C and ω_P are respectively responsible for playback quality of video to be played soon, the number of chunks in loose area affecting the video quality of entering urgent area, and the delivery time of chunks. Besides, compared with the other weights, ω_C is a little more important, e.g., configuration 7 is better than 5 and 6. That is because the larger ω_C is to obtain more chunks in this scheduling interval and ensure high video quality entering into urgent area. Therefore, following that ω_C is a little larger than ω_B and ω_P can achieve better delay performance under the guarantee of high video quality. And the configuration [0.3 0.4 0.3] is adopted in the following simulation.

C. *Comparisons*

We compare our algorithm with two classical related work PALS [9] and Random Scheduling [15] to verify our algorithm's performance.

Figure. 5 shows the cumulative distribution function of peers' (0.995, 0.005) –Startup delay with different

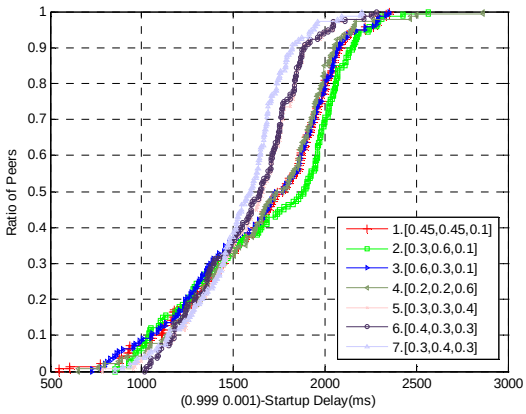

Figure 4. CDF of peers' (0.999,0.001)-Startup Delay with different [ω_B ω_C ω_P]

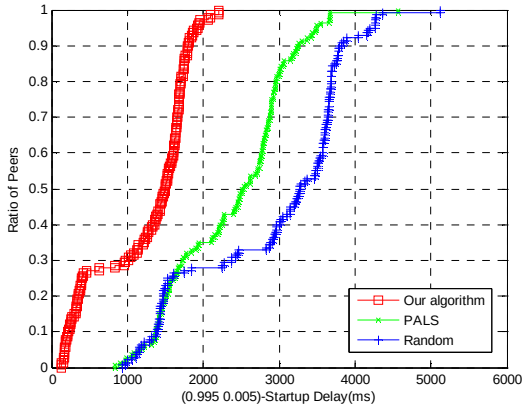

Figure 5. CDF of peers' (0.995,0.005)-Startup Delay with different Scheduling algorithm

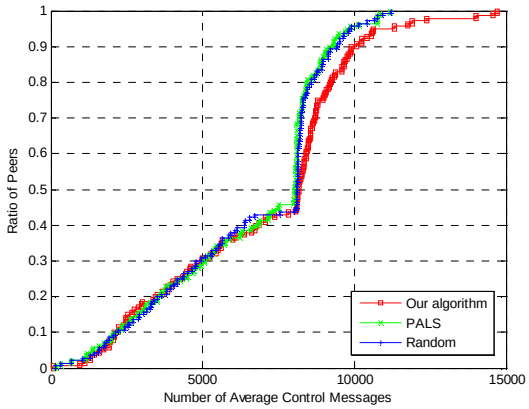

Figure 6. CDF of peers' number of control messages with different scheduling algorithms

scheduling algorithms and our algorithm reduces the startup delay respectively by 48.6% and 57.9% than PALS and random scheduling. PALS adopts Round-Robin scheduling algorithm to ignore the differences among capabilities of neighbor peers delivering chunks; Random scheduling algorithm ignores not only the differences among capabilities of neighbor peers but also the differences among priorities of absent chunks. These result in not achieving higher layer delivery ratio in shorter time for PALS and Random scheduling. Our algorithm proposes a priorities' quantification method based on the characteristics of urgent area and loose area, and prioritily requests chunks with higher priorities from neighbor peers which deliver them in the shortest time. Therefore, our algorithm can achieve higher layer delivery ratio in the shortest time.

Figure. 6 shows the cumulative distribution function of peers' number of control messages when achieving (0.995, 0.005) –Startup delay for different scheduling algorithms. And our algorithm increases respectively by 6% and 5.5% than PALS and Random scheduling. The rise of number of control messages for our algorithm results from the rise of number of request messages which are used to obtain the lost chunk due to link loss, which improves the layer delivery ratio. However, PALS and Random scheduling do not have enough time to request the lost chunks due to the worse performance of their scheduling algorithm.

In summary, our algorithm can greatly (respectively 48.6% and 57.9%) outperform two classical approaches PALS and Random scheduling by control overhead's slight increase (respectively 6% and 5.5%).

I. CONCLUSION

In this paper, we proposed a Pull-based chunk scheduling problem model and TOPSIS-based chunk scheduling algorithm for layered P2P streaming in the emerging scenario to deal with the problem of long startup delay. And simulations illustrated our algorithm could greatly enhance delay performance under the guarantee of high video quality than the existing classical related work by a slight increase of control overhead.

ACKNOWLEDGMENT

This work was supported by National Key Basic Research Program of China (973 Program) (2009CB320504), National High Technology Research and Development Program of China ("863"Program) (2008AA01A317), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 60821001), Research Fund for the Doctoral Program of Higher Education of China (20090005120012), National Key Basic Research Program of China (973 Program) (2009CB320406).

REFERENCES

[1] PPStream, http://www.ppstream.com/ , 07.03.2010

[2] PPLive, http://www.pplive.com/, 07.03.2010

[3] B. Li, J. Liu, "Multirate video multicast over the internet: an overview", IEEE Network, vol. 17, no. 1, pp. 24-29, 2003.

[4] S. Agarwal, J. P. Singh, A. Mavlankar, P. Baccichet, and B. Girod,"Performance and Quality-of-Service Analysis of a Live P2P Video Multicast Session on the Internet," IEEE IwQoS'08, Enschede, NL, pp. 11-19, June 2008.

[5] H. Xiaojun, L. Chao, L. Jian, L. Yong, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," Multimedia, IEEE Transactions on, vol. 9, pp. 1672-1687, 2007.

[6] L. Dai, Y. Cui, and Y. Xue, "Maximizing Throughput in Layered Peer-to-peer Streaming", in Proc. IEEE ICC, Glasgow, Scotland, pp. 1734-1739, 2007.

[7] X. Xiao, Y.C. Shi, B.P. Zhang and Y. Gao, "OCals: A Novel Overlay Construction Approach for Layered Streaming", in Proc. IEEE ICC, Beijing, China, pp. 1807-1812, 2008.

[8] Y. Okada, M. Oguro, et al., "A New Approach for the Construction of ALM Trees using Layered Video Coding", in Proc. P2PMMS, Hilton, Singapore, pp.12-12, 2005.

[9] R. Rejaie, A. Ortega, "PALS: Peer-to-Peer Adaptive Layered Streaming", In Proc. ACM NOSSDAV, Monterey, California, pp. 153-161, 2003.

[10] M. Qin, R. Zimmermann, "Improving Mobile Adhoc Streaming Performance through Adaptive Layer Selection With Scalable Video Coding",in Proc. ACM Multimedia, Augsburg, Germany, pp. 717 - 726, 2007.

[11] J. Zhao, F. Yang, et al, "On Improving the Throughput of Media Delivery Applications in Heterogenous Overlay Network", in Proc. IEEE Globecom, San Francisco, USA, pp. 1-6, 2006.

[12] Y. Cui and K. Nahrstedt, "Layered Peer-to-Peer Streaming", in Proc. ACM NOSSDAV, Monterey, USA, pp.162 -171,2003.

[13] M.R. Garey and D.S. Johnson, Computers and Intractability - A Guide to the Theory of NP-completeness, Freeman, 1979.

[14] 4. K. Yoon and C.L. Hwang, Multiple Attribute Decision Making: An Introduction, Sage, Thousand Oaks, CA,(1995).

[15] V. Pai, K. Kumar, et al., "Chainsaw: Eliminating trees from overlay multicast", in Proc. IEEE INFOCOM 2005, Miami, USA, pp. 127-140, 2005.

[16] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a bittorrent networks performance mechanisms," in IEEE INFOCOM 2006, Barcelona, Spain, pp. 1-12, Apr. 2006.

[17] Jelasity, M., Montresor, A., Jesi, G.P.: Peersim: Peer-to-Peer simulator (2004), http://peersim.sourceforge.net

[18] Meridian node to node latency matrix (2500×2500):http://www.cs.cornell.edu/People/egs/meridian/data.php. Meridian Project, 2005.

[19] S. Floyd et al., "Equation-based Congestion Control for Unicast Applications", In Proc. ACM SIGCOMM 2000, Stockholm, Sweden, pp. 43-56, 2000

 ISBN: 978-1-61208-103-8