Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Rate-distortion optimized bitstream switching for peer-to-peer live streaming

Jin-feng ZHANG[†1], Rong-gang WANG[2], Jian-wei NIU[1], Yuan DONG[2], Hai-la WANG[2]

([1]*School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083, China*)

([2]*France Telecom R&D Beijing, Beijing 100080, China*)

[†]E-mail: jinfeng.zhang@buaa.edu.cn

Received June 26, 2007; revision accepted Jan. 4, 2008; published online Mar. 13, 2008

Abstract: Peer-to-peer (P2P) technology provides a cost-effective and scalable way to distribute video data. However, high heterogeneity of the P2P network, which rises not only from heterogeneous link capacity between peers but also from dynamic variation of available bandwidth, brings forward great challenge to video streaming. To attack this problem, an adaptive scheme based on rate-distortion optimization (RDO) is proposed in this paper. While low complexity RDO based frame dropping is exploited to shape bitrate into available bandwidth in peers, the streamed bitstream is dynamically switched among multiple available versions in an RDO way by the streaming server. Simulation results show that the proposed scheme based on RDO achieves great gain in overall perceived quality over simple heuristic schemes.

Key words: Stream switching, Frame dropping, Peer-to-peer (P2P), Rate-distortion optimization (RDO)
doi:10.1631/jzus.A071343 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

With the development of network communication and video compression technologies, real-time video streaming becomes more and more popular in daily life. Generally, for a video streaming system, it is important to support a large scale of concurrent users. This makes all of streaming systems encounter the so-called scalability problem. Unfortunately, the scalability problem cannot be efficiently overcome in the traditional client/server model streaming system. To counter this problem, various media distribution technologies have been intensively studied in the past two decades. IP multicast was firstly proposed and studied in the 1990s (Deering and Cheriton, 1990). However, it has not been deployed extensively because of some serious concerns regarding its scaling, supporting for higher level functionality, and deployment. Then, the application layer multicast technologies were introduced (Banerjee *et al*., 2002; Ganjam and Zhang, 2005), among which, Content Delivery Network (CDN) (Akamai, http://www.akamai.com; Limelight Networks, http://www.limelightnetworks.com) and peer-to-peer (P2P) (Chu *et al*., 2000; Padmanabhan *et al*., 2002; Castro *et al*., 2003; Zhang *et al*., 2005; Huang *et al*., 2006; Guo *et al*., 2007; Liu *et al*., 2008; PPLive, http://www.PPLive.com) are the two most deployed solutions. In CDN, a number of proxy servers are deployed at the edge of the Internet, and clients receive media streaming service from their closest proxy server. The expensive cost of the dedicated proxy servers and high bandwidth required for CDN has limited its broad adoption by the Internet content providers. Differently, in P2P networks, clients are regarded as peers, which not only playback but also relay streaming data to other peers. Since the pre-deployed proxy servers are no longer needed, and much bandwidth is contributed by the peers, the cost of media distribution is greatly reduced. Therefore, today, streaming data distribution through P2P technology attracts more and more attention from academia and industry.

However, P2P video streaming systems pose very different challenges from the traditional client/server streaming systems. On one hand, in P2P streaming system, media data are distributed through one or more multi-hop path(s) from the streaming server to the peers, and high heterogeneity and high dynamics exist in the hop links of the multi-hop paths. The high heterogeneity leads to the existence of some peers whose bandwidth is insufficient to support high quality video, while the high dynamics, especially the unpredictable disconnection of some peers, results in packet loss or too late arrival. What is more, the multi-hop transmission method tends to increase the end-to-end delay, compared with the traditional single hop method. On the other hand, video streaming data have the characteristics of high bitrate, delay sensitivity and loss sensitivity, which make it very challenging to provide high-quality video streaming service in P2P networks.

The overlay of a P2P network can be constructed as a tree, mesh or other hybrid structure. Since "tree" is perhaps the most natural structure and it does not suffer from a latency-overhead tradeoff, it is adopted in the vast majority of the proposals up to date (Chu *et al*., 2000; Liu *et al*., 2008). To exploit the bandwidth diversity in the P2P network, multi-tree overlay has been presented (Padmanabhan *et al*., 2002; Castro *et al*., 2003). In this paper, adaptive video streaming over the single tree structure overlay is studied, and extending it to the multi-tree overlay will be a part of our future work.

To cope with the bandwidth heterogeneity of the clients in the traditional client/server video streaming system, a few of rate adaptation schemes are proposed, including rate control (Li *et al*., 2003), transcoding (Vetro *et al*., 2003), scalable video coding (Schwarz *et al*., 2007), bitstream switching (Krasic *et al*., 2003), and rate shaping (Chakareski *et al*., 2005; Tu *et al*., 2006), etc. Rate control requires so much computation from the streaming server that it is difficult to concurrently support a large scale of clients. What is more important, rate control cannot be applied in the streaming systems in which the bitstreams are compressed off-line. Though transcoding involves lower complexity than rate control, it is still too complex to be applied in a peer with low computation capability. Even though scalable video coding provides inherent priorities among the compressed video data which in turn provides a natural method for selecting which portions of the compressed data to transmit while meeting the transmission rate constraints, it still has high decoding complexity, which makes it not very suitable for the peers with limited computation and memory resources. Moreover, scalable video coding loses 1~2 dB in compression efficiency compared with traditional single layer coding (Schwarz *et al*., 2007). Up to now, scalable video coding has not been extensively adopted in real streaming systems. On the other hand, bitstream switching and rate shaping are broadly adopted in practical streaming systems due to their simplicity and high efficiency.

Several rate adaptation schemes combining bitstream switching and rate shaping have been proposed in literature (Schierl and Wiegand, 2004; Stockhammer *et al*., 2006). The existing work mainly focuses on rate adaptation in traditional video streaming scenarios where the streaming server and client communicate directly through one hop connection. For each client, an instance of rate adaptation algorithm is independently performed by the streaming server, and in turn the optimal bitstream to transmit is independently determined. However, the network topology of a P2P system is generally much more complex than that of a traditional system. Most peers cannot directly connect to the streaming server and the multi-hop transmission is involved. In order to maximize the overall video quality of peers in the network, it is not enough to determine the best bitstream by considering only the peers which directly connect to the streaming server by one-hop connection. Fortunately, a peer acts as not only a client but also a server, which makes the stream scheduling algorithms performed in a streaming server can also be applied in peers to further improve the system's overall performance.

In this paper, we firstly propose an adaptive scheme combining bitstream switching and rate shaping for tree-structure P2P overlay. The proposed scheme is to maximize the overall perceived quality in peers, which is achieved by collaboration among the streaming server and peers. The streaming server dynamically selects an optimal bitstream from available bitstreams according to current network condition in the overlay, and disseminates the selected bitstream over the overlay. Before relaying the video data, each peer drops some frames in a rate-distortion optimized way to meet the constraints on the available output link bandwidth and the transmission delay.

The rest of the paper is organized as follows. The problem of optimal bitstream selection for tree-structure overlay and the proposed system model are introduced in Section 2. In Section 3, a single frame dropping module is modeled as a filter, and the rate-distortion optimized frame dropping problem is solved by a low complexity greedy algorithm. In Section 4, the problem of optimal bitstream selection is solved as a series of single frame dropping problems, and complexity issue of the presented solution is also addressed. In Section 5, simulation results are provided and analyzed to verify the performance of the proposed solution. Finally, Section 6 concludes this paper.

PROBLEM AND SYSTEM MODEL

Bitstream selection problem

The bitstream selection problem for tree-structure overlay (BSP-TO) can be described as: among the available multiple bitstreams in the streaming server, which one should be distributed to maximize the overall perceived quality of peers in the overlay?

It is easy to get two simple heuristic solutions for solving the BSP-TO. The first solution, named the minimal rate transmission (MinRate), is that the server sends the bitstream with the rate no more than and as close as possible to the minimal available transmission bandwidth among the hops. Apparently, MinRate scheme tends to waste many resources in the hops with higher bandwidth. The other solution is called the maximal rate transmission (MaxRate). In the MaxRate scheme, the streaming server sends the bitstream with the rate less than and closest to the available bandwidth of the hop from the streaming server to its direct child node, and the intermediate peers perform rate adaptation to shape the bitrate when necessary. Although the MaxRate scheme maximizes the usage efficiency of the network resources, it also cannot guarantee to achieve the best overall perceived quality, which will be shown in Section 5.2.

In this paper, the RDO method is exploited to solve the BSP-TO. The system model is presented in the rest of this section, and the problem is formulated and solved in the subsequent sections.

System model

1. System architecture

The architecture of the tree-structure P2P streaming system is shown in Fig.1. The overlay is a tree rooted at the streaming server, and it is composed of multiple sub-trees rooted at the direct children of the streaming server. For example, there are two sub-trees in the overlay in Fig.1, which root at nodes N1 and N5. We assume that the output bandwidth of the streaming server is much higher than that of the peers, and that the optimal bitstream for each sub-tree is independently determined by the streaming server. According to the variation of network conditions in a sub-tree, the streaming server dynamically switches the streamed bitstream among the multiple available bitstreams to maximize the overall perceived quality of the peers in the sub-tree.

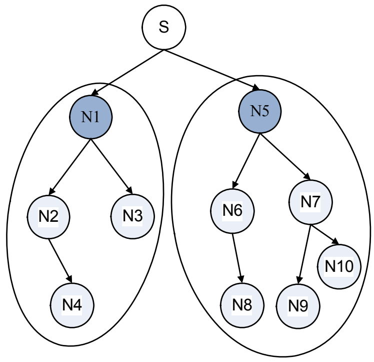

Fig.1 Architecture of the tree-structure P2P streaming system

2. Streaming server

The basic building blocks in the streaming server are illustrated in Fig.2. Before streaming, every video program is encoded into multiple bitstreams with different bitrates, which are represented as S1, S2, …, etc. In addition, the side information, named as rate-distortion hint tracks (RDHTs), is generated for each bitstream (Chakareski *et al*., 2005; Stockhammer *et al*., 2006; Tu *et al*., 2006). The side information mainly includes size and importance of each frame in the bitstream. The Network Prober (NP) is responsible for periodically collecting network parameters of the hops in the overlay. The Stream Switcher (SS) periodically selects the optimal bitstream to transmit among the available bitstreams according to RDHTs and the probed network parameters. The Frame Dropper (FD) selectively drops frames to adapt the output bitrate when necessary. The Streamer is responsible to packetize and send video data.

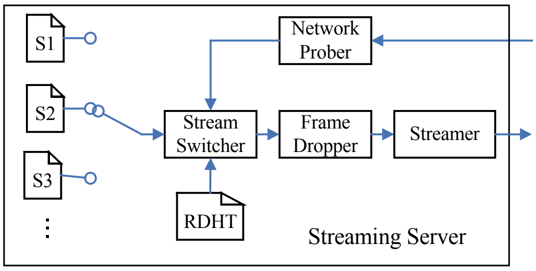

Fig.2 Basic building blocks in the streaming server

3. Peer

The basic building blocks in a peer are shown in Fig.3. A peer assigns a pair of NP and FD to each direct child. Each NP periodically collects the network parameters of the hops in the sub-tree rooted at the direct child to which it is assigned, and then sends the probed results to the peer's parent. Similar to the FD in the streaming server, an FD in a peer shapes the bitrate of bitstream into available transmission bandwidth of the direct child it is assigned to. The FDs are linked into a chain by decreasing order of output bitrates. The output of each FD is not only streamed to the direct child peer the FD is responsible for, but also sent into its next FD in the peer. By this way, each FD's rate shaping is based on that of its preceding FD, which eliminates much redundant computation for intermediate peers with multiple direct children.

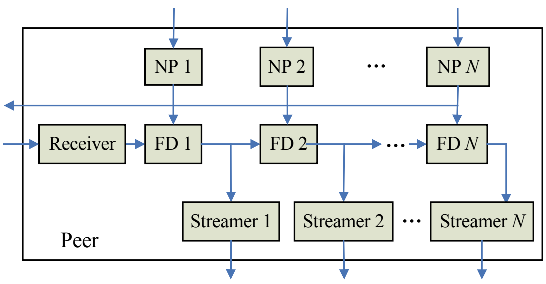

Fig.3 Basic building blocks in the peer

4. Network prober

An NP has two functions. Firstly, it periodically probes the network parameters of the hop from the node to the direct child it is responsible for. Three network parameters, including packet loss rate, round trip time and available bandwidth, are probed as follows. Once a peer successfully receives a data packet from its parent, it will send back an acknowledgment (ACK) of a correctly transmitted packet by an RTCP receiver report to its parent. According to the number of sent packets and that of received ACKs, the packet loss rate can be obtained. Small size packets are periodically sent to probe the round trip time. Based on the propped packet loss rate and round trip time, the available bandwidth is calculated by the TCP-friendly bandwidth formula (Floyd *et al.*, 2000). Secondly, an NP collects the network parameters of all the hops in the sub-tree rooted at the direct child it is responsible for, and then sends the collected network parameters to the NP constructed for it in its parent node.

Each single hop in the overlay is represented by a five-tuple (PID_i, CID_i, B_i, P_i, RTT_i), i=1,2,…,M. PID_i and CID_i are the identifiers of the parent peer and the child peer, respectively; B_i, P_i and RTT_i respectively indicate the available bandwidth, the packet loss rate and the round trip time; M is the count of hops in the overlay. As mentioned previously, the parameters of the hops are periodically updated by NPs to reflect the variation in the overlay.

In this paper, we assume that the overlay network is dynamically optimized by P2P protocols such as PeerCast (Deshpande *et al.*, 2001) and ZigZag (Tran *et al.*, 2003) according to the network condition, so that the available transmission bandwidth of the hop decreases as the distance between the hop and the server increases.

FRAME DROPPER

An FD can be regarded as a filter as in Fig.4. The filter's input is the available or received video data, and its output is a subset of the input video data. The input bandwidth B_{in} and output bandwidth B_{out} are also provided to an FD by its corresponding NP. As a result of overlay optimization, B_{out} is generally no more than B_{in}. So the task of an FD is to extract the output video data to maximize the reconstructed video quality under the rate constraint of B_{out}.

Fig.4 Frame dropper model

To calculate the output video quality of an FD, the term of importance is defined for each frame. For the jth frame in the video stream with version v, the importance for the quality of the reconstructed video stream is denoted by $I(j, v)$. $I(j, v)$ is the total quality increase of the reconstructed video when the jth frame is correctly decoded, and

$$I(j,v)=\frac{1}{L}\left[Q(f_j,f'_{j,v})-Q(f_j,f'_{c(j),v})\right]+\frac{1}{L}\sum_{\substack{k=j+1\\ j\mapsto k}}^{L}\left[Q(f_k,f'_{j,v})-Q(f_k,f'_{c(j),v})\right], \quad (1)$$

where L is the number of frames in the video sequence; f and f' represent the original frame and its reconstruction, respectively; $Q(f, f')$ is the quality measure reflecting the distortion when presenting f by f'; $j\mapsto k$ means f_k is eventually concealed by f_j; $c(j)$ represents f_j's concealment frame, and $c(0)$ represents the default image, e.g., the grey image.

The dropping pattern is defined as a vector $\boldsymbol{\pi}$ with L elements. $\boldsymbol{\pi}(j)$ is the jth element of vector $\boldsymbol{\pi}$, which is 0 when the jth frame is dropped, and 1 when the jth frame is maintained. Then the reconstruction quality of the bitstream S_v is obtained by

$$Q(\boldsymbol{\pi},v)=Q_0+\sum_{j=1}^{L}I(j,v)\boldsymbol{\pi}(j)\prod_{\substack{m=1\\ m\prec j}}^{j-1}\boldsymbol{\pi}(m), \quad (2)$$

where L is also the number of the frames in the video sequence, $Q_0=\sum_{j=1}^{L}Q(f_j,G)\big/L$ denotes the minimum quality when all the frames are presented as the default image, and $m\prec j$ means that the mth frame is required to correctly decode the jth frame.

The problem of frame dropping is formulated into a constrained optimization problem described as

$$\boldsymbol{\pi}^*=\arg\max Q(\boldsymbol{\pi}, v) \quad (3)$$

subject to

$$R(\boldsymbol{\pi},v)=\sum_{j=1}^{n}\boldsymbol{\pi}(j)r(j,v)\le B_{out}, \quad (4)$$

where $r(j, v)$ denotes the size of the jth frame in the video stream.

The constrained optimization problem of Eqs.(3) and (4) can be converted into a non-constrained optimization using a Lagrangian multiplier λ as

$$\boldsymbol{\pi}^*=\arg\max Q(\boldsymbol{\pi}, v)+\lambda R(\boldsymbol{\pi}, v). \quad (5)$$

Eq.(5) is resolved in a low complexity manner as follows. Generally, frames in a compressed video stream are organized in GOP (Group of Pictures) structure. For simplicity, the calculation of a frame's importance is performed independently for each GOP in this paper, and the L of Eq.(1) is set as the length of one GOP. Accordingly, the frame dropping process is also performed in the unit of GOP, and then the computing complexity is greatly reduced.

Applying frame dropping in peers results in that some peers cannot receive the whole bitstream data. When a frame is not available at the decoder in the decoding process, an error concealment method is required. In this paper, "freeze-picture" concealment approach is applied. That is, the dropped frame is represented by the timely nearest available frame. In the case of consecutive frames, concealment is applied recursively. Consequently, the concealment dependencies can be expressed by a directed acyclic graph. An example of frame decoding dependency graph and the corresponding concealment dependency graph is shown in Fig.5. The symbol G represents a standard presentation, e.g., a grey image, which occurs when an I frame is dropped. According to the frame concealment tree, each frame's concealment frame is determined, and then its importance is calculated by Eq.(1).

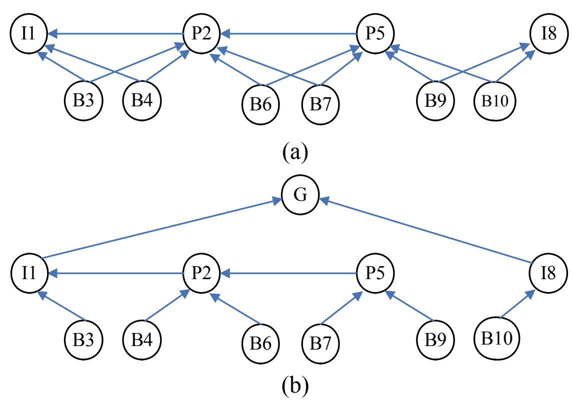

Fig.5 (a) Frame decoding dependency graph; (b) Frame concealment dependency graph

To assure that a remained frame can be correctly decoded after the frame dropping process, all of its ancestors in the frame decoding dependency graph must also be remained. Hence, the solution to the optimization problem Eq.(5) can be regarded as a subset of frames in the bitstream, which maximizes the reconstruction quality of the video sequence under the constraints of meeting the target bitrate and keeping the remained frames decodable.

In the proposed algorithm, two frame sets, such as remained frame set and candidate frame set, are defined. The remained frame set consists of the frames which have been decided so far to be remained, and the candidate frame set consists of the frames under consideration whether they can be put into the remained frame set. The algorithm just greedily updates the two frame sets according to the side information until the available output bandwidth is used up. The frame dropping algorithm is described below:

Step 1: Performing initialization for related variables. For each frame, the utility of importance per bit is calculated as $\lambda(j,v)=I(j,v)/r(j,v)$; the number of bits which can be used to transmit the GOP is calculated as $C=B_{\text{out}}L/f$, where B_{out} is the target bitrate, L is the number of frames in the GOP, and f is the frame rate; the remained frame set S_{r} is set to be empty, and the candidate frame set S_{c} is set to only include I frames.

Step 2: Moving a frame from S_{c} to S_{r}. Search for the frame F with the largest utility $\lambda(F,v)$ among the frames whose sizes are not larger than C in S_{c}. If F is found, move F from S_{c} to S_{r}; otherwise, go to Step 4.

Step 3: Updating related variables. Add frames satisfying the following conditions into S_{c}: the frame is not in S_{r} or S_{c}, and its ancestors in the decoding dependency graph are all in S_{r}; subtract the size of frame F from the bits to transmit the GOP C, $C=C-r(F,v)$; go to Step 2.

Step 4: Performing the frame dropping. Drop the frames which are now not in S_{r}.

The algorithm's complexity is analyzed here. Assume the number of frames considered in the algorithm is L. The algorithm runs in loops. For each loop, a frame is put into the remained frame set. Thus, in the worst condition, the times of the required loops are L. In each loop, Steps 2~4 are performed. The complexities of Steps 2 and 3 are both $O(L)$, and the complexity of Step 4 is $O(1)$. Therefore, the proposed frame dropping algorithm has the complexity of $O(L^2)$.

STREAM SWITCHER

As mentioned previously, before streaming, several compressed versions for a video sequence are prepared, which are denoted by $\{S_v\}$, $v=1,\ldots,V$. For each version of bitstream, side information is generated and stored in an RDHT, which includes frame size, decoding timestamp, frame importance, and dependencies among frames.

Based on the topology of a sub-tree and the bandwidth of its hops, a simulator of FD chain is constructed to simulate the real transmission in the sub-tree. As previously mentioned, each peer creates an FD chain to adapt the video rates to its children, and each child peer is associated with an FD. Through connecting all of the FDs in peers in the sub-tree by decreasing order in the output bandwidth, a big FD chain can be obtained. The big FD chain is called the FD chain of the sub-tree. An example of the construction for a sub-tree's FD chain is shown in Fig.6.

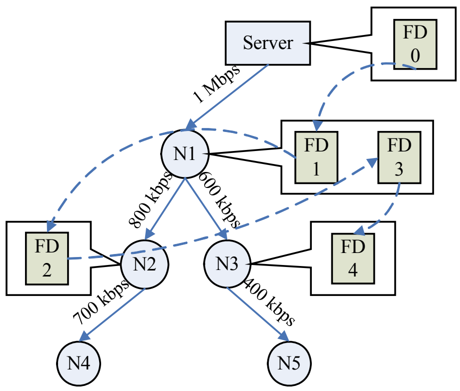

Fig.6 FD chain for a general tree

Each FD in a sub-tree is represented by a two-tuple (PID_i, B_i), $i=1, 2,\ldots ,M$. M is the number of the peers in the sub-tree. PID_i and B_i indicate the ith FD's ID and output bandwidth, respectively. The model of the FD chain is shown in Fig.7. Note that PID_i is the ID of peer i, and B_i is the available receiving bandwidth of peer i.

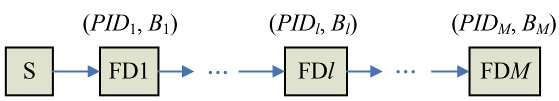

Fig.7 FD chain model

To construct a simulator for an FD chain, the whole real FD chain is divided into several FD segments which contain one or more consecutive FDs, and each segment is implemented as one FD in the simulator. As shown in Fig.8, an FD chain simulator consists of a series of FDs, each of which is represented by a triple (N_l, B_l, W_l), l=1,2,…,N. N is the number of FDs in the simulator; N_l, B_l and W_l indicate the lth FD's ID, output bandwidth, and weight, respectively.

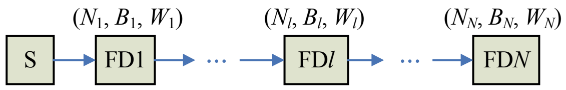

Fig.8 FD chain simulator

The real FD chain is dived into segments by the following steps. Firstly, the number of FDs in the simulator N is determined by the available computational resource in the streaming server. Secondly, the range of the output bandwidth of the real FDs is determined and equally divided into N sections. Generally, if the output bandwidth range is [$B_{\min}$, $B_{\max}$], the lth section is calculated as [$B_{\max}-l(B_{\max}-B_{\min})/N$, $B_{\max}-(l-1)(B_{\max}-B_{\min})/N$], l=1,2,…,N. Finally, the real FDs whose output bandwidth is in the lth section are represented by the lth FD in the simulator.

The output bandwidth of an FD in the simulator is the average output bandwidth of the real FDs in the segment it represents. And the weight of an FD in the simulator is calculated by

$$W_l = M_l / M, \tag{6}$$

where M_l is the number of the real FDs represented by the lth FD in the simulator, and M is the total number of the real FDs in the sub-tree.

When the video stream S_v is distributed over the sub-tree, the average quality of the outputs of the FDs in the simulator, $Q(v)$, is used to predict the real average perceived quality of the peers in the sub-tree. $Q(v)$ is calculated by

$$Q(v) = \sum_{l=1}^{N} Q(N_l, v) W_l, \tag{7}$$

where $Q(N_l, v)$ denotes the quality of the video output from the lth FD in the simulator.

According to the previous definitions and formulations, the problem of BSP-TO can be described as: which v from {S_v} can maximize $Q(v)$? And the problem can be formulated as

$$v^* = \arg\max Q(v). \tag{8}$$

According to Eq.(7), the problem Eq.(8) can be solved as multiple independent frame dropping problems. And the frame dropping problem can be further solved by the proposed algorithm in Section 3.

The complexity of the proposed solution to the problem Eq.(8) is determined by five factors, including the number of considered bitstreams V, the period of bitstream switching T, the number of the frames involved in the simulation L, the number of the FDs in the FD chain simulator N, and the complexity of the frame dropping algorithm. Actually, the trade-off between the complexity and performance can be achieved in the following aspects:

(1) The number of considered bitstreams V. In the practical streaming system, the number of the available bitstream versions for a video sequence is limited, for example, 4~6. When the available computation resource in the streaming server is scarce, the number of bitstream candidates can be reduced to lower the complexity.

(2) The period of bitstream switching T. To respond to the variation in the network condition as soon as possible, the optimal bitstream should be determined at any random access point, for example, once for a GOP. However, frequent running of the bitstream selection algorithm increases the burden of the streaming server. The parameter T can be adjusted to trade-off between the switching complexity and the response time to the network dynamics.

(3) The number of FDs in the FD chain simulator N. The FD chain simulator is an effective tool to scale the required computation in the optimal bitstream selection. An FD in the FD chain simulator can represent one or more real FDs, thus, the time of running frame dropping algorithm can be greatly reduced when necessary. Through adjusting the number of FDs in the simulator, the trade-off between the required computation and performance can be achieved. Besides, the FD chain simulator can be implemented as each FD stores some of intermediate results and feeds them into the FDs following it, and the following FD performs frame dropping based on the

information from the previous FDs. In this way, a following FD's work is based on that of the FDs previous to it, which can remove a great amount of redundant computation.

(4) The number of frames considered in the simulation L. To determine the best bitstream for time interval T, the more the number of frames is considered in the simulation, the more the accuracy of the simulation result is, and the more the computation is required. Therefore, the number of frames involved in the simulation should be skillfully decided to do a good trade-off between the complexity and the accuracy of bitstream selection.

In summary, the computation required by the bitstream switcher in the whole streaming process can be approximated as $O[V\times(1/T)\times N\times L\times L]$. Since V and N both can be defined as constants, the complexity of each run of the bitstream selection algorithm is at the same level as that of frame dropping, which is $O(L^2)$.

SIMULATIONS AND ANALYSIS

Simulation setup

A simulation environment is constructed by C++ for the experiments. Simulations are run over a sub-tree with up to 1000 nodes. The actual number of peers participating in each simulation varies between 800 and 1000. The bandwidth of the hops in the sub-tree distributes in the scope [175, 725] kbps. The nodes are divided into 11 groups according to their receiving bandwidth, and the nodes with receiving bandwidth in the scope $[725-50i, 725-50(i-1)]$, i=1, 2, …, 11, are placed into the ith node group. Accordingly, the FDs of each node group are represented by an FD in the FD chain simulator in the stream switcher.

Three common intermediate format (352×288) test video sequences, News, Foreman and Bus are used. The sequences are encoded at 30 frames/s with JM 11.0 at different rates. The encoding structure is IBBPBBPBB, the length of GOP is 15, the number of B frames between adjacent I/P frames is 2, and every I frame is encoded as an IDR frame. The bitrate of a bitstream is controlled by QP in the encoding process, and each sequence is encoded into 7 bitstreams with bitrates from 180 kbps to 700 kbps. The bitrates of the bitstreams used in the simulations are shown in Table 1.

Table 1 Bitrates of the bitstreams in simulation (kbps)

No.	News	Foreman	Bus
1	672	700	654
2	598	604	569
3	488	520	425
4	387	397	326
5	347	297	252
6	255	262	219
7	184	181	192

In the following figures, symbol "S" represents the streaming server, "Ni" represents the average of the evaluated metric for the nodes in the ith node group, "avg" represents the average of the evaluated metric for all the nodes in the sub-tree, and "Sv" represents the bitstream with No.v.

Optimal bitstream for streaming over static P2P networks

In the experiment, the simulations are run over a static network environment. The bandwidths of the hops in the sub-tree are equally distributed in the scope [175, 725] kbps, and they are kept static in the whole simulation process. Each of the bitstreams of the three sequences is independently streamed over the sub-tree, and the frame dropping algorithm proposed in Section 3 is applied when the rate adaptation is needed. The average perceived quality of each node group and the average quality of all the nodes in the sub-tree are shown in Fig.9. For every sequence, the average perceived quality of all the nodes has a similar apparent trend that firstly increases up to the best quality and then decreases along with the decrease of bitstream bitrate. The bitstream producing the best perceived quality is neither the bitstream with the maximal bitrate, nor the one with the minimal bitrate. For News, Foreman and Bus, the best average quality is achieved at S5, S4 and S4, respectively. And for the three sequences, the best qualities achieved by the optimal bitstreams are higher than the corresponding worst ones by 3.3, 3.7 and 2.5 dB, respectively.

Bitstream switching for streaming over static P2P networks

In this experiment, the static network environment applied in Section 5.2 is also adopted. Three streaming schemes are compared in the throughputs and the perceived qualities in the peers. The first one

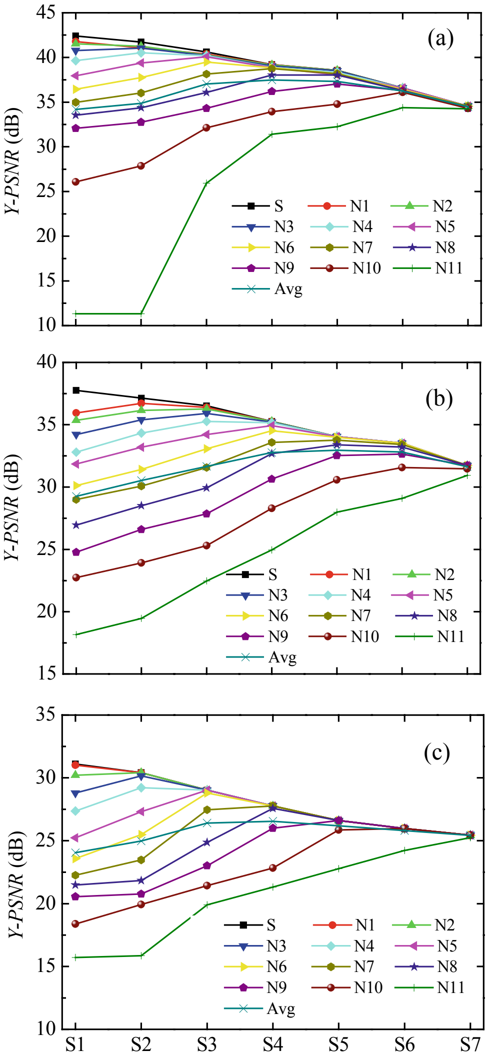

Fig.9 Perceived qualities in the nodes when each bitstream is independently streamed. (a) News; (b) Foreman; (c) Bus

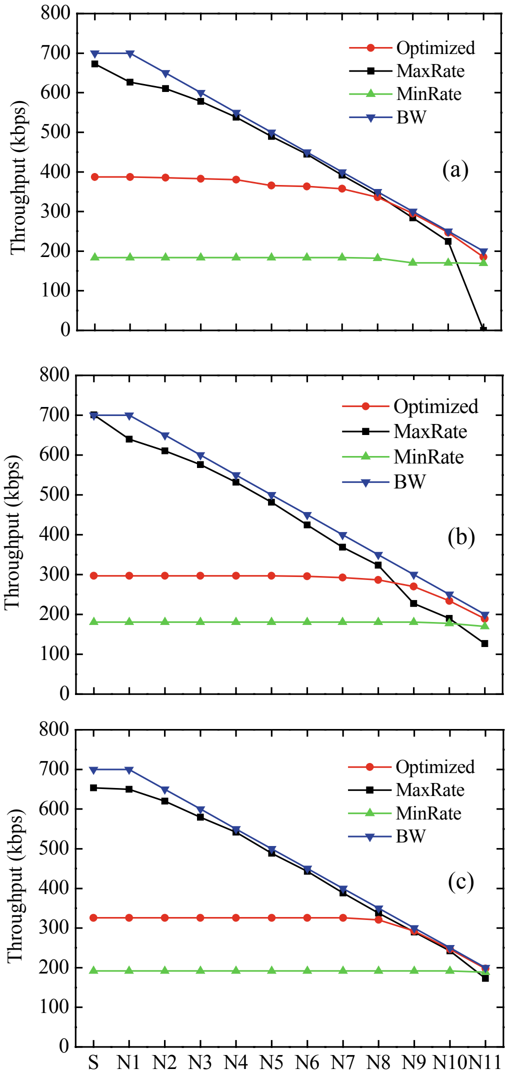

Fig.10 Throughputs in the nodes in static network environment. (a) News; (b) Foreman; (c) Bus

is the proposed Optimized scheme, and the other two are MaxRate and MinRate. In the proposed Optimized scheme, the FDs in each node group are represented by an FD in the FD chain simulator. In the MaxRate scheme, the frame dropping algorithm proposed in Section 3 is also exploited to implement rate adaptation.

The throughputs averaged over node groups are shown in Fig.10. A peer's throughput means the rate in which the peer receives video data. BW represents the available bandwidth for the peer to receive data. It can be seen that the MaxRate scheme maximizes the throughputs, while the MinRate scheme minimizes the throughputs, and the Optimized scheme achieves the throughputs at a medial level between those two schemes.

The perceived qualities of peers averaged over node groups are shown in Fig.11. For the MaxRate scheme, the perceived quality in a peer decreases with the decrease of the receiving bandwidth of the peer. However, the perceived quality in the low bandwidth peer is very poor. Especially, no frame is received by some leaf nodes when News is streamed with the MaxRate scheme. For the MinRate scheme, the perceived quality in a peer keeps at a low and constant level. For the Optimized scheme, the perceived quality in a peer is more stable and better than that of the MaxRate scheme, and is much better than that of

MinRate scheme in most peers. Averagely, the Optimized scheme outperforms about 3 dB over the MaxRate scheme and about 1.5 dB over the MinRate scheme in perceived video quality of peers. It is interesting that the best video quality is obtained by the Optimized scheme, although its bandwidth usage is much lower than that of the MaxRate scheme.

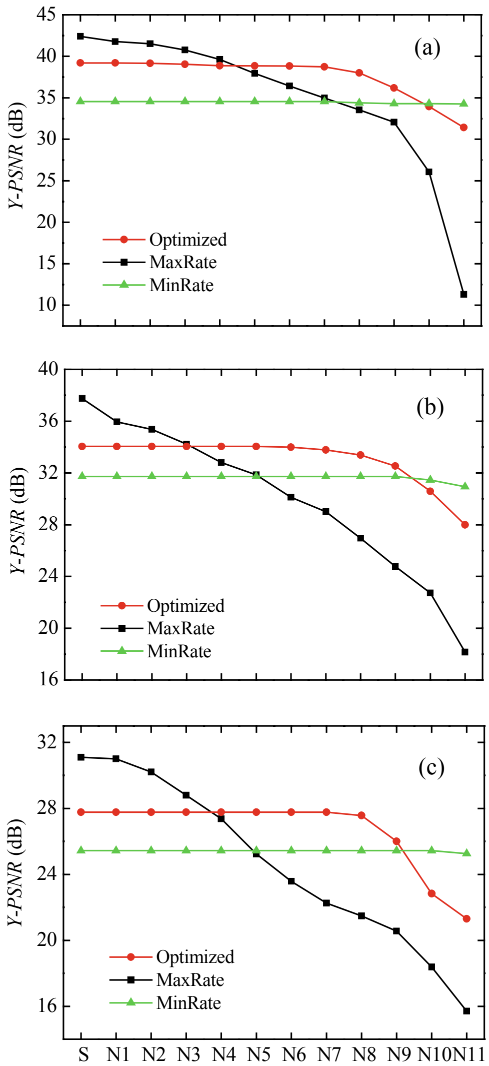

Fig.11 Perceived qualities in the nodes in static network environment. (a) News; (b) Foreman; (c) Bus

Bitstream switching for streaming over dynamic P2P networks

In this experiment, the Optimized scheme, MaxRate scheme and MinRate scheme are compared in a dynamic network environment instead of the static one in Sections 5.2 and 5.3. The three video sequences are streamed for 50 s respectively. The bandwidth of peers changes every 10 s. The ratio of the number of nodes in each group and that of all the nodes in the sub-tree over time is shown in Table 2.

When the video sequences are streamed with each scheme, average perceived quality of peers over time is shown in Fig.12. It is interesting that the MaxRate scheme gets better quality than the MinRate scheme for low motion sequence News, while it gets worse quality than the latter for fast motion sequence Foreman. And for the sequence Bus, the MaxRate scheme is inferior to the MinRate scheme in the early streaming process, and is superior to the MinRate scheme in the later process. This can be explained by that for low motion sequence spatial information contributes more than temporal information to the perceived video quality, and for high motion sequence temporal information contributes more than spatial information. On one hand, the MaxRate scheme tends to drop more frames but the spatial quality of the retained frames is higher, it places more weight on spatial information. On the other hand, the MinRate scheme tends to drop fewer frames but the spatial quality of the retained frames is lower, it places more weight on temporal information. With the help of RDO, the Optimized scheme can optimally allocate available bandwidth between spatial and temporal information, it gains over the MaxRate scheme and the MinRate scheme in perceived quality over all the peers and sequences.

Table 2 Distribution of the nodes in the groups over time (%)

Time (s)	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
0~9	0	0	0	0	0	0	20	20	20	20	20
10~19	0	0	0	20	20	20	20	20	0	0	0
20~29	20	20	20	20	20	0	0	0	0	0	0
30~39	20	0	0	0	20	20	20	20	0	0	0
40~49	20	0	0	0	0	0	0	20	20	20	20

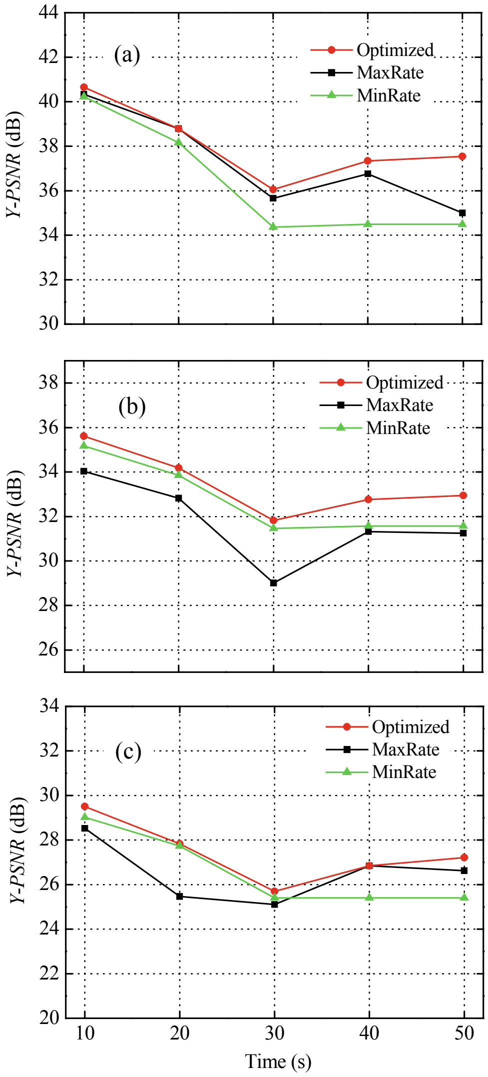

Fig.12 Perceived qualities in the nodes over time in dynamic network environment. (a) News; (b) Foreman; (c) Bus

CONCLUSION

In this paper, a rate-distortion optimized video streaming scheme combining bitstream switching and frame dropping is proposed to cope with the heterogeneity and dynamics of a P2P streaming system. The streamed bitstream is dynamically switched among multiple available versions by the streaming server, and the rate of bitstream is adapted with frame dropping by peers. Both of bitstream switching and frame dropping are performed in a rate-distortion optimized way. Simulation results show that the proposed scheme achieves greater gain in the perceived quality than simple heuristic streaming schemes do.

More importantly, a framework to cope with the heterogeneity in P2P live streaming network is proposed. With the help of FD module, the peer is endowed with the ability to adapt the video rate to the heterogeneous and dynamic network conditions. Other advanced rate adaptation techniques, such as data partition based packet dropping or scalable video coding (SVC), can be easily integrated into the proposed framework by just replacing the FD module with a new rate adaptation module.

In the future, we will investigate the scheme of multi-bitstream simulcasting (Liu *et al*., 2006) in the overlay to improve the bandwidth utilization efficiency in high bandwidth hops. Furthermore, single-tree overlay will be extended to multi-tree overlay to improve both error resilience to network dynamics and bandwidth utilization efficiency.

References

Banerjee, S., Bhattacharjee, B., Kommareddy, C., 2002. Scalable application layer multicast. *ACM SIGCOMM Computer Commun. Rev.*, **32**(4):205-217. [doi:10.1145/633025.633045]

Castro, M., Druschel, P., Kermarrec, A.M., Nandi, A., Rowstron, A., Singh, A., 2003. SplitStream: High-Bandwidth Multicast in Cooperative Environments. Proc. 9th ACM Symp. on Operating Systems Principles, p.298-313. [doi:10.1145/945445.945474]

Chakareski, J., Apostolopoulos, J.G., Wee, S., Tan, W., Girod, B., 2005. Rate-distortion hint tracks for adaptive video streaming. *IEEE Trans. on Circuits Syst. Video Technol.*, **15**(10):1257-1269. [doi:10.1109/TCSVT.2005.854227]

Chu, Y.H., Rao, S.G., Zhang, H., 2000. A case for end system multicast. *ACM SIGMETRICS Performance Evaluation Review*, **28**(1):1-12. [doi:10.1145/345063.339337]

Deering, S., Cheriton, D., 1990. Multicast routing in datagram Internet works and extended LANs. *ACM Tran. on Computer Syst.*, **8**(2):85-110. [doi:10.1145/78952.78953]

Deshpande, H., Bawa, M., Hector, G.M., 2001. Streaming Live Media Over a Peer-to-Peer Network. Stanford Database Group Technical Report (2001-20).

Floyd, S., Handley, M., Padhye, J., Widmer, J., 2000. Equation-based Congestion Control for Unicast Applications. ICSI Technical Report (TR-00-03).

Ganjam, A., Zhang, H., 2005. Internet multicast video delivery. *Proc. IEEE*, **93**(1):159-170. [doi:10.1109/JPROC.2004.839602]

Guo, P.H., Yang, Y., Li, X.Y., 2007. A P2P streaming service architecture with distributed caching. *J. Zhejiang Univ. Sci. A*, **8**(4):605-614. [doi:10.1631/jzus.2007.A0605]

Huang, C., Chou, P.A., Li, J., Zhang, C., 2006. Adaptive peer-to-peer streaming with MutualCast. *J. Zhejiang Univ. Sci. A*, **7**(5):737-748. [doi:10.1631/jzus.2006.A0737]

Krasic, C., Walpole, J., Feng, W., 2003. Quality-Adaptive Media Streaming by Priority Drop. Proc. 13th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, p.112-121. [doi:10.1145/776322.776341]

Li, Z.G., Pan, F., Lim, K.P., Feng, G.N., Lin, X., Rahardja, S., Wu, D.J., 2003. Adaptive Frame Layer Rate Control for H.264. Proc. Int. Conf. on Multimedia and Expo, **1**:581-584. [doi:10.1109/ICME.2003.1220984]

Liu, J.C., Li, B., Zhang, Y.Q., 2006. Optimal stream replication for video simulcasting. *IEEE Trans. on Circuits Syst. Video Technol.*, **8**(1):162-169. [doi:10.1109/TMM.2005.861279]

Liu, J.C., Rao, S.G., Li, B., Zhang, H., 2008. Opportunities and challenges of peer-to-peer Internet video broadcast. *Proc. IEEE*, **96**(1):11-24. [doi:10.1109/JPROC.2007.909 921]

Padmanabhan, V.N., Wang, H.J., Chou, P.A., Sripanidkulchai, K., 2002. Distributing Streaming Media Content Using Cooperative Networking. Proc. 12th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, p.177-186. [doi:10.1145/507670.507695]

Schierl, T., Wiegand, T., 2004. H.264/AVC Rate Adaptation for Internet Streaming. Proc. 14th Int. Packet Video Workshop. Irvine, CA, USA.

Schwarz, H., Marpe, D., Wiegand, T., 2007. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. on Circuits Syst. Video Technol.*, **17**(9):1103-1120. [doi:10.1109/TCSVT.2007.905532]

Stockhammer, T., Liebl, G., Walter, M., 2006. Optimized H.264/AVC-based bit stream switching for mobile video streaming. *EURASIP J. Appl. Signal Processing*, p.1-19. [doi:10.1155/ASP/2006/91797]

Tran, D.A., Hua, K.A., Do, T.T., 2003. ZigZag: An Efficient Peer-to-Peer Scheme for Media Streaming. Proc. INFOCOM, p.1283-1292.

Tu, W., Chakareski, J., Steinbach, E., 2006. Rate-distortion optimized frame dropping and scheduling for multi-user conversational and streaming video. *J. Zhejiang Univ. Sci. A*, **7**(5):864-872. [doi:10.1631/jzus.2006.A0864]

Vetro, A., Christopoulos, C., Sun, H., 2003. Video transcoding architectures and techniques: an overview. *IEEE Signal Processing Magazine*, **20**(2):18-29. [doi:10.1109/MSP.2003.1184336]

Zhang, X., Liu, J., Li, B., Yum, T.P., 2005. DONET: A Data-Driven Overlay Network for Efficient Live Media Streaming. Proc. INFOCOM, p.2102-2111.