APSIPA ASC 2011 Xi'an

Quality Adaptation of SVC-Based P2P Streaming

Yan Pang, Jiaying Liu, and Zongming Guo
Institute of Computer Science and Technology, Peking University, Beijing
{pangyan, liujiaying, guozongming}@icst.pku.edu.cn

***Abstract*—Peer-to-Peer (P2P) video streaming has become increasingly popular as it is a good enhancement to the traditional Clietn/Server methods in reducing cost and increasing robustness. However, streaming over P2P still suffers from lack of adaptation to the system dynamics. Scalable Video Coding (SVC) shows scalability in partial transmission and decoding. It can support heterogeneous devices in terms of communication bandwidth, display resolution, processing power, and other constraints. In this paper, we propose a P2P streaming architecture leveraging SVC and data scheduling to provide maximum quality adaptation. The proposed quality adaptation mechanisms allow for precise adaptation to device inherent resources limitations and the actual P2P network condition. Moreover, the architecture uses an underlay aware peer selection to improve the P2P overlay topology. The simulation results demonstrate the effectiveness of the quality adaptation of the proposed architecture.**

I. Introduction

Video streaming has become the most traffic intensive application in the Internet and is expected to be the killer application in Next Generation Networks (NGNs). According to [1], the sum of all forms of video (TV, video on demand, Internet, and P2P) will continue to be approximately 90% of global consumer traffic by 2015. In recent years, P2P systems have been successfully used in streaming over the Internet [2], [3]. Compared to the Client/Server architecture, P2P has much better robustness, reconfiguration, and scalability. Although P2P streaming takes advantage of the P2P architecture to alleviate server load, it still faces several challenges. Every P2P client connecting to the Internet has specific resource characteristics, which include the peer inherent resources (screen resolution, processing power, and bandwidth) and the overlay resources (active neighbors, throughput, and network condition). However, video streaming is a demanding application and works only when minimum resource requirements are met. The resources of some weak devices are not sufficient for playing back the video file in maximum quality.

This problem can be overcome by reducing quality such as reducing the delivered frame rate, image quality or resolution. To achieve this, we take advantage of the scalability of SVC [4]. SVC is based on three dimensional scalabilities: spatial, temporal, and SNR scalability. Being encoded into multiple layers each with different quality information, SVC coded streams could be transmitted and decoded by extracting several layers. Each substream extracted from different layers can be rendered as a stream with lower perceived quality than the original complete stream. Compared to the single layer coding, SVC is more convenient for supporting streaming applications with heterogeneous devices in terms of communication bandwidth, display resolution, and CPU processing power.

The challenges of quality adaptation in P2P video streaming are caused by large-scale decentralized, heterogeneous, and dynamic environment. For the decentralized structure, the layer selection is made by each client other than streaming servers. And the peers' available resources are subject to the fluctuation of network, the dynamics of peer churn, and the availability of selected layers in neighbor peers. There is a substantial amount of research on P2P systems with support for adaptation such as [5], [6], [7], [8]. Rejaie *et al.* [5] introduced PALS, a receiver driven P2P video streaming system with quality adaptation playback. However, PALS only considers single dimensional scalability (as the case for many layered streaming systems) and therefore cannot adapt to heterogeneous characteristics of peers. Magharei and Rejaie [6] present PRIME with the goal of minimizing content and bandwidth bottlenecks in mesh-based streaming by deriving proper peer connectivity and an efficient pattern of delivery based on Multiple Description Coding (MDC). By using SPPM (Stanford Peer-to-Peer Multicast), [7] distributed SVC streams over multiple multicast trees with congestion control. Chameleon [8] is a new P2P streaming protocol that combines the advantages of network coding and SVC, and also includes neighbor selection, quality adaptation, receiver-driven peer coordination, and sender selection with different design options.

In this paper we focus on using spatial, temporal, and SNR scalabilities which are inherent in SVC to adapt to different peer resources and network conditions. We propose a P2P streaming architecture to stream a video to peers adaptively by identifying the highest quality level based on their available resources. The quality level selection employs a rate model and a complexity model to calculate the processing power and bandwidth requirements, and the device resources and dynamic network resources are also taken into account. The block selection calculates priority for each block based on playback time and quality level. And the peer selection uses the bandwidth as a metric to improve the performance.

The rest of this paper is structured as follows. Section II presents the quality adaptive P2P streaming architecture in detail. Simple simulation results are discussed in Section III. We conclude the paper in Section IV.

II. Adaptive P2P Streaming

The main idea of our proposed architecture is to effectively utilize available resources of each peer to maximize delivered stream quality under resources variations. Taking advantage

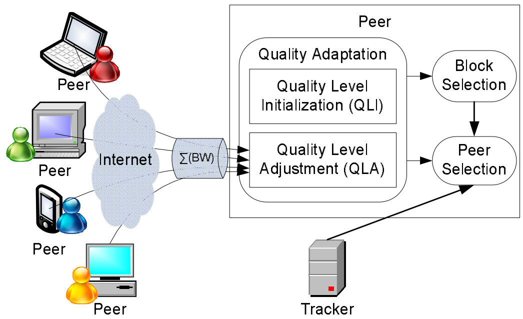


Fig. 1. The quality adaptive P2P streaming architecture

of the scalability of SVC, quality adaptive P2P streaming could make the decision of which layer video is best matching the peers' available resources. Based on the data scheduling, blocks in the selected layers are requested and streamed in the P2P environment.

The proposed architecture for quality adaptive P2P streaming is showed in Fig. 1. Quality adaptation mechanisms are mainly composed by two modules: Quality Level Initialization (QLI) and Quality Level Adjustment (QLA). When a peer wants to view a video, it first invokes the QLI module to choose the quality level best suited to its static resources. Then, the peer joins the swarm and receives a list of potential neighbors provided by peer discovery module. All of these potential neighbors stream same or lower quality video. The underlay information based peer selection module chooses peers to request needed blocks from. After establishing connections with provider peers, video streaming starts to fill the video buffer. During streaming, a feedback is send to QLA module to adjust the parameters with the changing conditions. Therefore, if necessary, block selection is updated in order to support an increased or decreased quality level.

A. *Quality Adaptation*

We now discuss in more detail the structure of the QLI and the QLA modules responsible for quality level selection and their auxiliary components.

1) Rate Model and Complexity Model: Both QLI and QLA have complexity and bitrate adaptation components. The role of the bitrate adaptation component is to take rate requirements for decoding into consideration and to match it with available bandwidth. In this architecture we employ a rate model proposed by [9]. The bitrate of a video R is considered as a function of frame rate T, quantization parameter Q, and spatial resolution D. Here, the T, Q, and D are the functions of the layer index for temporal, quality, and spatial scalability t, q, and d respectively. The bitrate of decoding is given by:

$$\begin{aligned} &R(Q,T,D) = \\ &\quad R_{max} R_q(Q, T_{max}, D_{max}) R_t(Q,T,D_{max}) R_d(Q,T,D), \end{aligned} \tag{1}$$

where $R_{max} = R(Q_{min}, T_{max}, D_{max})$ is the maximum bitrate for a chosen minimum quantization parameter Q_{min}, a chosen highest frame rate T_{max}, and a chosen maximum spatial resolution D_{max}. As shown in [9], the effect of Q, T, and D are independent of each other. Hence the function can be written as $R(Q,T,D) = R_{max}R_q(Q)R_t(T)R_d(D)$. The characteristics of the functions $R_q(Q)$ and $R_t(T)$ were broadly studied in [10]. The function $R_d(D)$ is studied in [9].

$$R_q(Q) = \left(\frac{Q}{Q_{min}}\right)^{\alpha}, \alpha > 1. \tag{2}$$

$$R_t(T) = \left(\frac{T}{T_{max}}\right)^{\beta}, \beta < 1. \tag{3}$$

$$R_d^{slow}(D) = 1 + \eta \ln\left(\frac{D}{D_{max}}\right), \eta < 1. \tag{4}$$

$$R_d^{fast}(D) = \left(\frac{D}{D_{max}}\right)^{\theta}, \theta < 1. \tag{5}$$

Apart from considering the rate requirements for decoding, the processing requirements for decoding is also taken into consideration. The complexity adaptation component uses a complexity model following the approach of [10] that works by mapping every set of quality levels (spatial, temporal, and SNR) into processor cycles required for decoding the SVC coded video stream. Based on definitions in Table I, decoding complexity of an SVC stream can be calculated.

TABLE I
SYMBOLS FOR ANALYTICAL COMPLEXITY MODEL

Notation	Description
$C_I/C_P/C_B$	Average macroblock decoding complexity of I-/P-/B-picture
C_S/C_Q	Average macroblock decoding complexity at spatial/quality enhancement layers
$T/D/Q$	Total layer number for temporal-/spatial-/quality-scalability
$t/d/q$	Layer index for temporal-/spatial-/quality-scalability
M	Number of macroblocks per picture

The complexity for decoding scalable streams is given by:

$$\begin{aligned} C_{GOP\_Dec} =& M_0(\alpha C_I + (1-\alpha)C_P + (2^{T(0)} - 1)C_B) \\ &+ \frac{8^{D+1} - 1}{7} 2^{T(0)} M_0 Q C_Q \\ &+ 4\frac{8^D - 1}{7} 2^{T(0)} M_0 (C_S + C_B). \end{aligned} \tag{6}$$

In order to reduce the computation complexity of each client, the decoding complexity and the total rate at any combination of spatial, temporal, and SNR scalability are precalculated and saved in the metadata file when the initial peer (typically streaming server) shares the stream. The client only needs to compares its resource parameters with complexity and bitrate in the metadata file and gets the proper quality level parameters simply.

2) Quality Level Initialization (QLI): As shown in Fig. 2, the QLI module evaluates the current resources and requirements in order to match them with achievable quality. This module mainly handles static parameters, such as screen resolution, bandwidth, currently available CPU power, and user preference.

An initial quality set with base layer quality level parameters d_0, t_0, and q_0 is populated at first. And then, the spatial, bitrate, and complexity adaptation modules select out all compatible quality level based on screen resolution, bandwidth, and

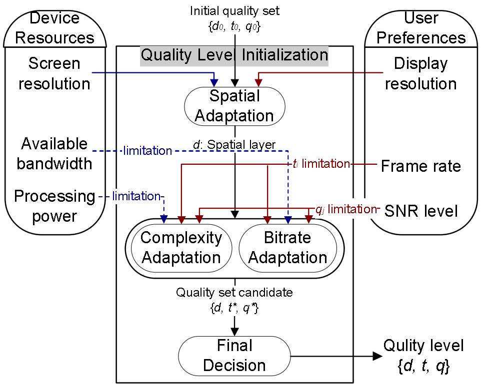


Fig. 2. Quality Level Initialization (QLI) module

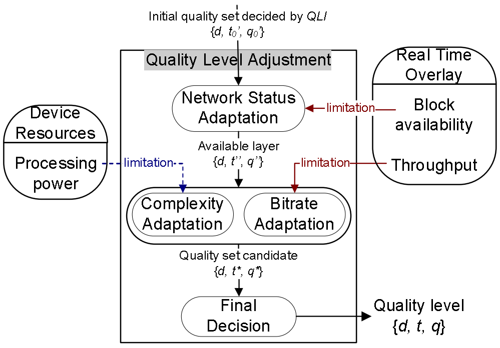


Fig. 3. Quality Level Adjustment (QLA) module

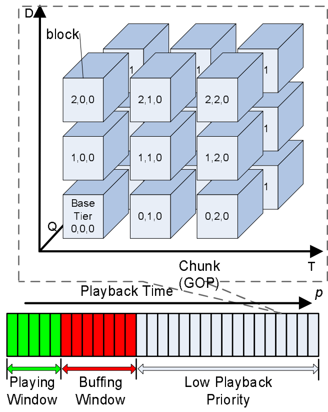


Fig. 4. Chunk and block structure

processing power respectively considering the user preference limitations. All the compatible combinations are appended as candidates. The final decision is made by selecting the item $\{d, t, q\}$ which values of all three dimensions are at their maximum or doing some complex tradeoff between temporal and SNR dimensions. Because recently user surveys show that, for a given resolution, users prefer a video higher image quality and low frame rate instead of a video with medium picture quality and high frame rate [11], the QLI final decision is prefer items with higher SNR value.

The proposed quality level initialization algorithm is shown in Algorithm 1.

Algorithm 1: Quality level initialization algorithm

Input: Initial quality set candidate $CandidateQS$ with base layer quality level $\{d_0, t_0, q_0\}$
Output: Quality level d, t, q suited to the resources limitation

```
d=min(Peer.Resolution, UserPereference.Resolution);
CandidateQS.append({d, t_0, q_0});
if t_i ≤ UserPereference.FrameRate then
    if q_j ≤ UserPereference.QualityLevel then
        if (Complexity(d, t_i, q_j) ≤ Peer.CPUPower) ∧
        (BitRate(d, t_i, q_j) ≤ Peer.Bandwidth) then
            CandidateQS.append({d, t_i, q_j});
        end
        q_j ← q_j + 1;
    end
    t_i ← t_i + 1;
end
return maximize item in CandidateQS {d, t, q};
```

3) Quality Level Adjustment (QLA): The structure of the QLA is shown in Fig. 3. This module adapts to changes in network conditions in order to maximize available quality at the receiver. It is executed periodically. Other than using static resources information as discussed for the QLI, the QLA relies on real-time overlay status reflected from current throughput and block availability. This allows the peer to quickly react to changes in the P2P network, such as peer churn or a sudden change in throughput. Only temporal and SNR adaptation are handled by the QLA because typically the peer display resolution is unchangeable.

The QLA starts from the QLI output quality level parameters $\{d, t_0', q_0'\}$. The network status, bitrate, and complexity adaptation components adjust all compatible quality level based on block availability, throughput, and processing power respectively. In order to obtain the block availability, we extend the common buffer map in BitTorrent to the block availability indicator which provides availability information of SVC blocks instead of whole chunks.

B. Block Selection

To be transmitted across P2P network, an SVC stream needs to be segmented. In our proposed P2P streaming architecture, a video stream is first divided into chunks. Each video chunk consists of one GOP (Group of Pictures). A chunk is further divided into blocks as depicted in Fig. 3. Each chunk contains layers in the three dimensional quality spaces, and each block is the smallest quality unit with one of the three dimensional quality spaces various. Video block is the basic unit for exchanging data across the network.

Considering the layers selected by the QLI and QLA, block selection module makes a decision on which blocks to request. It calculates priority for each block based on its urgency of playback time and quality level in SVC. The base layer blocks have the highest priority. The priority decreases with playback time p, and decreases for increasing enhancement layers in any dimension.

$$P_{block}(p, d, t, q) = -Ap - B(ad + bt + cq). \tag{7}$$

If a peer is more interested in smoother playback, the parameter A which is the weight of urgency of playback should be increased. If higher quality is preferred, B, the weight of quality, is increased. The parameters a, b, c define the weights for the different quality dimensions.

C. Peer Selection

We utilize a tracker-based approach for peer discovery. In order to support the proposed layer adaptation mechanisms, the tracker manages the list of active peers together with the layers they are currently streaming. The peers send a request to the tracker when they join the system or get the quality level decision after they perform the quality level initialization and adjustment. The request contains the list of quality level initially need (only base layer supported by default) or currently not properly supported by the neighbors. The tracker responds with a list of peers supporting these quality levels. Then the peers set up connections with these neighbors and exchange currently availability information of blocks to each other.

Based on the priorities of the different blocks, peer selection module will select peers to request needed blocks. In this paper, we prefer underlay bandwidth as the peer selection metric. Whenever a peer has a choice between more than one provider peer, the one that can offer more bandwidth is selected. This is achieved by requiring all peers to provide bandwidth information to other peers they contact with. Therefore, fast peers will tend to get their blocks from similarly fast peers. The peers with similar capability clustered. It will enhance the performance of the streaming system.

III. Experimental Result

In this section, we present a preliminary evaluation of our proposed quality adaptation architecture. We simulate the vibration of throughput, block availability, and processing power and see how the QLA reacts to them. There are 3 layers (0, 1, and 2) for both spatial and temporal scalability and 2 layers (0 and 1) for SNR scalability. This leads to the total of 18 possible layer combinations. We also suppose that the QLI has already decided on a basic spatial level, i.e. $d = 0$.

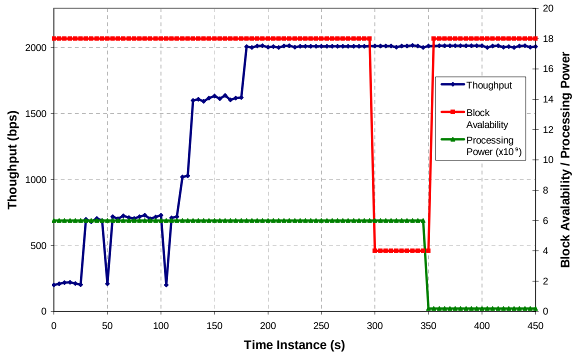
We simulated a changing throughput varied slightly (as shown in Fig. 5(a)) for the first 250 time instances (seconds), then the throughput is fixed at 2Mbps. Blocks of layer 5 to 18 are no longer available from time instant 300 to 350. A sudden drop in processing power occurs at time instant 350.

These test scenarios and the results represented by the instantaneous decision on d, t, and q are shown in Fig. 5(b). The results show that our mechanisms are able to quickly react to different changes in network and in peer resources.

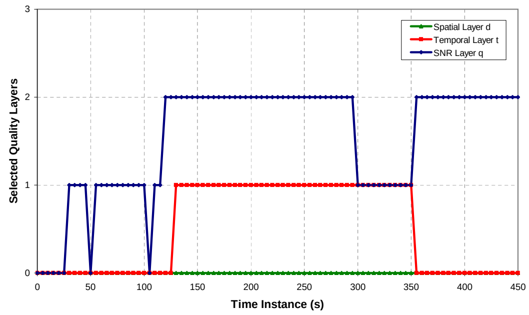
IV. Conclusion

In this paper, we considered quality adaptation in P2P streaming by leveraging SVC. In the proposed architecture, quality adaptation, block selection, and peer selection modules collaborate to providing best quality in the dynamic and heterogeneous P2P environment. Quality adaptation module uses a rate model and a complexity model to select proper layers to achieve the highest supported quality considering both static device resources and dynamic network conditions. Block selection module assigns priority to each block in the selected layers. Moreover, the bandwidth-based underlay aware peer selection improves the streaming performance. The preliminary simulation results show that our architecture reacts quickly to various system changes while providing best quality.

As future work, we will take incentive factors into consideration. Moreover, we will implement a quality adaptation P2P streaming system in practice to validate the effectiveness of our proposed architecture.



(a) Simulation scenarios



(b) Selected quality layers

Fig. 5. Evaluation of the QLA module

Acknowledgment

This work was supported by National Natural Science Foundation of China under contract No. 60833013, National Basic Research Program (973 Program) of China under contract No. 2009CB320907, and National Development and Reform Commission High-tech Program of China under Grant No. [2010]3044.

References

[1] (2011) Cisco Visual Networking Index: Forecast and Methodology, 2010-2015. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[2] X. Zhang, J. Liu, and T.-S. Yum, "CoolStreaming/DONet: a Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming," in *Proc. IEEE INFOCOM*, 2005.

[3] (2010) PPLive. [Online]. Available: http://www.pplive.com

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1103–1120, Sep. 2007.

[5] R. Rejaie and A. Ortega, "PALS: Peer-to-Peer Adaptive Layered Streaming," in *Proc. ACM NOSSDAV*, 2003.

[6] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIver MEsh-Based Streaming," in *Proc. IEEE INFOCOM*, 2007.

[7] P. Baccichet, T. Schierl, T. Wiegand, and B. Girod, "Low-delay Peer-to-Peer Streaming Using Scalable Video Coding," in *Proc. Packet Video*, 2007.

[8] A. T. Nguyen, B. Li, and F. Eliassen, "Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding," in *Proc. IEEE INFOCOM*, 2010.

[9] S. A. Azad, W. Song, and D. Tjondronegoro, "Bitrate Modeling of Scalable Videos Using Quantization Parameter, Frame Rate and Spatial Resolution," in *Proc. IEEE ICASSP*, 2010.

[10] Z. Ma and Y. Wang, "Complexity Modeling of Scalable Video Decoding," in *Proc. IEEE ICASSP*, 2008.

[11] S. Buchinger and H. Hlavacs, "Subjective Quality of Mobile MPEG-4 Videos with Different Frame Rates," *Journal of Mobile Multimedia*, vol. 1, pp. 327–341, 2006.