A LIGHT-WEIGHT FEDERATED VIDEO ADAPTATION SYSTEM FOR P2P OVERLAYS

Razib Iqbal, Shervin Shirmohammadi

Distributed and Collaborative Virtual Environments Research Laboratory (DISCOVER Lab)
School of Information Technology and Engineering (SITE), University of Ottawa
800 King Edward Avenue, Ottawa, Ontario, K1N 6N5, Canada
{riqbal | shervin}@site.uottawa.ca

ABSTRACT

This paper presents a lightweight but effective master-sender-driven multiple parent approach for online video adaptation and streaming to heterogeneous clients. The proposed design facilitates the cooperation of participating clients for improving the utilization of the spare resources in an overlay. Our design uses standard H.264/AVC video streams. It enables peers to contribute both CPU and bandwidth in a multi-parent fashion, such that no dedicated adaptation server and streaming server is required, although we need a reliable rendezvous point for the video stream originators and the peers in the overlay. We present the live video processing paradigm using metadata followed by the video distribution overview. A brief performance evaluation supporting the design choices is also presented.

Index Terms— Video Adaptation, Video Streaming.

1. INTRODUCTION

The ubiquitous computing concept has brought a revolution permitting users to think about multimedia contents anytime and in any multimedia capable device. Additionally, 3G and beyond 3G networks make it a reality for small handhelds to receive rich media contents. To lessen the gap between *concept* and *reality*, media adaptation deserves attention to make the presence of the same content in different devices viable. Traditional solutions of dedicated media adaptation and streaming servers suffer from high workload due to frequent requests and available capacity. One solution widely seen today is to store multiple versions of the same content taking into consideration different network types and device capabilities which is not a scalable solution given numerous device types entering the market each day. In this paper, we present a firsthand experience of combining online video adaptation with streaming in Peer-to-Peer (P2P) overlays to serve heterogeneous devices including small handhelds. Participating peers act as the stream source, adaptation engine, and also perform the streaming tasks. Intuitively, there is no need for streaming and/or adaptation servers.

In our previous studies [1][2], we have shown the benefits of using MPEG-21 generic Bitstream Syntax Description (gBSD) [3] for compressed domain authentication and encryption of H.264/AVC video. In this paper, we elaborate our design of *adaptive* video streaming to heterogeneous devices using gBSD. Our design attempts to couple the basics of P2P streaming concept with the mechanism to adapt live non-interactive videos.

In federated video processing for streaming in a P2P overlay, 3 essential components need to be considered: 1) Preparing the video, 2) Data scheduling and Buffering, and 3) Overlay construction and Streaming. Therefore, we have organized the rest of the paper as follows: In Section 2, we highlight our contribution in the context of related work. Section 3 provides brief details of metadata based video adaptation. Section 4 describes buffering and overlay generation to deliver video stream. Section 5 presents some results supporting our design. Finally, we conclude in Section 6.

2. LITERATURE REVIEW

In [4], authors propose adaptive frame scheduling for video streaming. This uses a fixed frame drop set which will hamper the visual quality of the reconstructed video stream, whereas for temporal adaptation, we drop frames based on each frame's importance level. Moreover, instead of delaying sending key frames, we design pictures in groups, called Framesets, in such a way that after transmitting and adaptation, every Frameset is self-contained. In [5], authors propose a video streaming system that uses ordinary computers (peers) as servers rather than using dedicated servers. Authors propose to encode each video into multiple descriptions and to place each description on a different server. However, it is inconvenient for an isolated peer to initiate the streaming session and to place the video descriptions in different participating nodes. In [6], authors propose an unstructured P2P overlay to share the network resources. Their design places dedicated servers and utilizes peers to reduce server load, whereas our goal is to avoid dedicated servers (for adaptation and streaming) and utilize the idle resources of the participating peers instead.

In short, this design utilizes idle computing power of the peers to adapt a video while streaming it. For efficient utilization of idle resources, we further introduce the multiple parent approach. We also present our real-time metadata based compressed domain temporal and spatial adaptation scheme.

3. OVERVIEW OF VIDEO PROCESSING STEPS

3.1. Metadata and H.264 Video

With standard metadata support, complex video processing operations in compressed domain is straightforward. If we know the syntax and semantics of the coded sequence beforehand, then we can avoid the cascaded decoding-adaptation-(re) encoding steps. The MPEG-21 gBSD is essentially a metadata definition of the media, written in XML which we have used extensively for all the video preparation operations in this paper. H.264 is expected to dominate the field of video codecs due to its advanced compression technique and versatility of the codec. Here, we have exploited the multiple reference frame feature of the H.264 video codec. All video streams before and after processing conform to H.264 description and are playable in a standard H.264 video player.

3.2. Video preparation

We intend to create a video once in its lifetime along with its metadata in the form of MPEG-21 gBSD. gBSD provides codec independence, and as a result, intermediary nodes can perform adaptation operations in a format ignorant way. We enhance the H.264/AVC encoder [7] with gBSD generation mechanism to facilitate temporal and spatial adaptation in the compressed domain. Therefore, gBSD containing the information (e.g. starting byte and length of each frame, slice size, and macroblock information etc.) pertaining to the encoded bitstream are written while encoding a video. The encoded video and the corresponding gBSD are jointly denoted as the Digital Item (DI). For our convenience, we have assumed 'Frameset' to represent the number of frames equal to the frame rate at which the raw video is being encoded (usually 30fps). A Frameset is used to identify a cluster of frames for the encoded bitstream while being represented in the metadata.

For temporal adaptation, the encoder is modified to write the frame information in the gBSD as soon as it performs the compression from the uncompressed video i.e. the YUV input. We exploit the multiple reference frame feature of H.264 video. In gBSD, frame importance level is set based on the following – 1. Reference frame, 2. Motion in the frame, 3. Frame size. The start of each frame in the compressed bitstream and the length of each encoded frame along with the NAL units are written in the gBSD. For spatial adaptation, while encoding, the video frame slices are encoded in fixed size and in a self-contained manner. The offsets of the slices in the encoded frames are also written in the gBSD during encoding. The macroblocks within a self-contained slice will not refer to any macroblock belonging to other slices. This will result in fewer choices for prediction for the macroblocks along the edge of the slice for cropping a frame to achieve a target resolution. This scheme of spatial adaptation is well suited for video-conferencing or news broadcast (i.e. talking head).

3.3. Adapting the Video

To adapt a DI, adaptation is performed in the following two steps: first, transform the metadata characterizing adaptation goal, and second, generate the adapted bitstream using the transformed metadata. End user's preference or device requirement is the input for transformation decision taking mechanism. MPEG-21 Usage Environment Description (UED) tool is used to gather this information.

For temporal adaptation, the encoded bitstream is adapted by discarding the portions to achieve the target resolution or framerate. Frames are discarded based on their importance level. Frames used as a reference frame, or with high motion, have the higher priority than others respectively. Otherwise, frames with low priority or same priority are dropped randomly based on the total size of the discarded frames to achieve the target rate. For low target framerate, to maintain the visual quality, the first frame of a Frameset is always considered as the base frame while decoding on the receiver side. For spatial adaptation, offsets of the slices are parsed from the gBSD. This is used to extract the desired region from the compressed video in the serving nodes. Once the slices have been extracted, the video/frame headers are updated accordingly by setting size of the region. In Figure 1, we summarize the concept of metadata-based video adaptation from video preparation to adapted video generation.

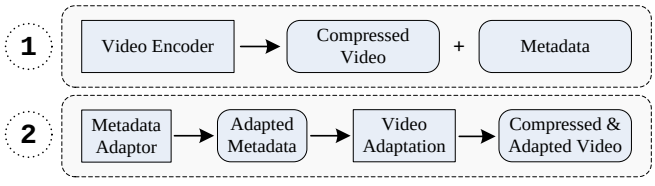

Figure 1. Metadata based compressed-domain video adaptation – Step 1: Video source, Step 2: Intermediary node.

4. DISTRIBUTED ADAPTATION AND STREAMING

Considering the adaptation technique in Section 3 as a utility tool, in this section we briefly explain how, using this tool, peers can adapt, transmit and buffer video streams to match the varying network conditions in a P2P environment.

4.1. Video Buffering

Videos are processed as clips (consisting of Framesets) individually encoded in the H.264 format and with their own gBSD. The adapted clips also conform to the H.264/AVC standard. For continuous playback, an initial buffering is needed on the receiver's side. The buffer is filled up with the received video segments for immediate playback and retransmission to other nodes. Initial buffer size is computed as: *Number_of_Framesets* × *frame_rate* × *average_frame_size*. A sliding window represents the active buffer portion of each peer, and the size varies for different frame consumption rates. While viewing, a sender peer adapts the Framesets in the active buffer for its immediate receiver peer(s), if required, and forwards to them the (adapted) stream.

4.2. Determining Adaptation/Streaming Capability

We have mapped CPU power to adaptation time. For example, for temporal adaptation, during the client-side application installation, a small video clip will be adapted for different target frame rates (30fps to 1fps) and it will also record the adaptation timings. Based on this timing profile, the controller will decide on the number of adaptations this peer can perform in a certain time frame. Now, if that peer can serve more than one peer at a time, then we lend the CPU to serve adaptation requests in a Time-Division-Multiple-Access fashion. Based on this information, the controller assigns this peer 'adaptation and streaming peer' role or 'streaming peer' role. Devices with limited resources may join as "free riders". For streaming, out-degree is computed from the upload bandwidth offered by each peer and average frame size. A sender keeps track of Framesets since the beginning of transmission with a sequence number. The ACK messages sent over the control channel help the sender peers to determine the transmission rate and adapt the video stream accordingly if not reported by the receiver.

4.3. Overlay generation

A central administering node (Admin node) will have knowledge about the current peers. A separate overlay is formed for each video, in case of multiple video programs. An ordinary peer or a streaming server can be the stream source and registers itself with the Admin node (Figure 2).

Node joining: An incoming peer registers itself with the Admin node and gets a list of available streams. It also passes the preferences in the form of UED descriptions. The Admin node attempts to connect an incoming node requesting certain video quality directly to one of the existing sub-trees (if any). If it fails, then a suitable node is chosen, which can perform the necessary adaptation, to serve the new peer. However, if the incoming node is a 'receiver-only' peer then the selected parent must be able to serve at least one more regular peer (i.e. 'streaming and adaptation' or 'streaming' capable peer). Admin node stores the peer information and sends a copy of the graph to the stream source. Therefore, stream source is the secondary administrative point of contact if the Admin node fails.

Multi-parent approach: Since a single peer may not be able or willing to contribute the required outbound bandwidth or CPU entirely, the Admin node attempts to assign multiple parents (up to a limit). However, the key challenge is coordination among senders. For obvious reasons, first layer peers are served by a single parent (i.e. stream source). However, starting from the second layer, more parents are assigned for each incoming peer. If a peer is served by two parents, then odd and even Framesets are sent by each of the parents, and so on. Now, when a peer is served by more than two peers, each parent is assigned a number. Eventually, a parent forwards those Framesets from a video stream that corresponds to its own number to serve a particular peer. In this regard, every peer synchronizes its virtual clock with that of the stream source. Since each Frameset contains originating timing information, each sender peer identifies which Framesets to forward and when. On the receiver's side, received Framesets are coordinated in the buffer.

Node departure: A peer can depart gracefully or ungracefully by hanging. If the source node has departed, then that overlay is dissolved immediately. If a root node has departed, then the Admin node attempts to assign a parent from the sorted availability list (from the same level or the ancestors of the disconnected parent). Disconnected nodes get higher priority than new node join requests. A peer may report one of its parents' departure or the tracker managed by the Admin node may detect the failure. Now the fact cannot be avoided that there can be few good and available parent choices in the system. Therefore, if the previous parent exists in the system, then the controller will try to reestablish the connection with the previous parent. A receiver peer also reports to the controller if the delay is higher than the program specific threshold.

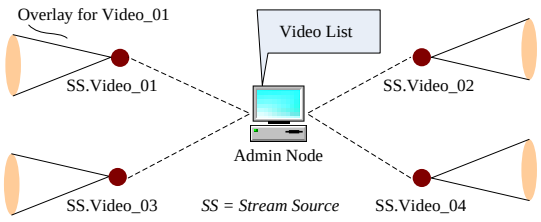

Figure 2. Overlay generation - a media server or an ordinary peer can be the stream source

Mobility of peers: Due to possible mobility, if the upload bandwidth of a parent node decreases, then it will attempt to serve peers according to their peer class (Table 1) and joining order. It may also notify its parent(s) to reduce the bit-rate/frame-rate of the transmitted video. In such a case, the new rate is applied to the next available Frameset.

Tree refinement: Tree refinement places peers with higher aptitude in the upper layer of the tree. Peer switching takes place in such a way that no peer will starve due to this operation. Therefore, the video is prepared ahead of time by the new parent for a particular receiver peer. The refinement procedure is called when end-to-end delay of certain percentage of peers exceeds the predetermined value.

Packet loss considerations: In case of high adaptation ratio, e.g. 3fps, if a key frame cannot be reconstructed due to packet loss during transmission then instead of retransmitting that frame we have designed to skip that Frameset and go to the next Frameset on the receiver side. If the sliding window detects too many missing Framesets then either the administrative tasks (e.g. checking existence of the parent) starts or the video is paused and buffering continues until the number of missing Framesets drop to the acceptable limit. From quality of experience point of view, viewer will experience freezing of video for those Framesets which were suppose to be forwarded by the departed parent. When the number of parents is higher then the missing Frameset problem will cause comparatively less annoyance to the viewer since the Frameset drop will occur after couple of seconds and until a new parent has been assigned.

5. IMPLEMENTATION AND EVALUATION

Core components of this system are mainly divided into receiver side functions, sender side functions, and adaptation engine. However, for small handhelds, the application can have only the receiver side functions. A peer failure is detected by monitoring the TCP control channel established between the receiver and each of the sending peers. Evaluation results presented here include both aspects of our scheme covering metadata-based adaptation and building the overlay. For overlay streaming, our results are collected over a simulated network for 150 peers where maximum five simultaneous parents are allowed to serve a peer.

Table 1. Adaptation performance (1frameset: 30 frames; Size: CIF)

Peer Class	Adaptation Time
Tier-1 (Pentium IV, 3.0 Ghz, 1GB RAM or Higher)	52 - 135 Milliseconds
Tier-2 (AMD Athlon XP, 1.4 Ghz, 512MB RAM or equivalent)	128 - 300 Milliseconds
Tier-3 (Pentium III, 700 Mhz, 256 MB RAM or equivalent)	210 - 415 Milliseconds

Table 1 presents the average adaptation time. We have classified peers into 3-tiers which can do adaptation and streaming, and rest is considered as Tier-4. From the table, we can claim that the adaptation time is quite acceptable considering the fact that processing is being done in the compressed domain and in a live fashion. Figure 3 illustrates both new node joining and internal node joining (i.e. reconnection) success rates. We can see that both the success rates are consistent during different runs considering peer heterogeneity and random arrival/departure of peers. Please recall that disconnected nodes get higher priority than new node join requests. However, we cannot avoid the fact that density of Tier-1 and Tier-2 parents is an important factor which drives the success of a particular overlay.

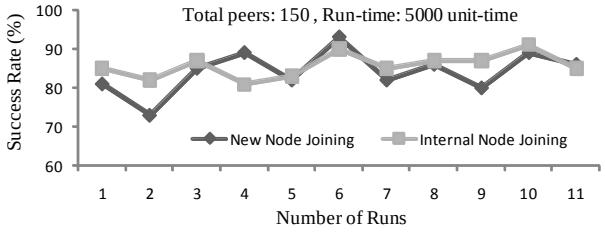

Figure 3. Node joining and Reconnection success rate

The tree refinement operation also enhances the system performance by shifting resourceful peers to the top of the tree. In this experiment, before refinement, the highest tree depth is 7, and after refinement, it is 5.

6. CONCLUSION AND FUTURE WORK

In this paper, we described metadata-based adaptation, and then proposed a system for real time distributed adaptation with multiple parents. Benefit of multiple parent approach and splitting video streams into Framesets is that the overall resiliency is improved since a node will not completely starve by the failure of a parent. As losses are often temporarily correlated along each path, splitting the video Framesets between different independent routes can be seen as a way to protect bitstream from consecutive losses. Our future research is focused on region-of-interest based selection and cropping of frames in compressed-domain.

7. REFERENCES

[1] R. Iqbal, S. Shirmohammadi, J. Zhao, "Hard Authentication of H.264 Video Applying MPEG-21 Generic Bitstream Syntax Description (GBSD)," in Proc. of ICME, 2007.

[2] R. Iqbal, S. Shirmohammadi, A. El Saddik, "Compressed-domain Encryption of Adapted H.264 Video," in Proc. of IEEE MultiSec, 2006.

[3] ISO/IEC 21000-7:2004, Information Technology – Multimedia Framework – Part 7: Digital Item Adaptation.

[4] S. G. Deshpande, "High quality video streaming using content-aware adaptive frame scheduling with explicit deadline adjustment," in Proc. of ACM Multimedia, 2008.

[5] X. Xiaofeng et al. "A peer-to-peer video-on-demand system using multiple description coding and server diversity," in Proc. of ICIP, 2004.

[6] G. Exarchakos, N. Antonopoulos, "Resource Sharing Architecture for Cooperative Heterogeneous P2P Overlays," in Jnl. of Networks & Systems Management, vol. 15, pp. 311-334, 2007.

[7] http://ftp3.itu.ch/av-arch/jvt-site/reference_software/