Efficient Scalable Video Streaming over P2P Network

Stefano Asioli, Naeem Ramzan, and Ebroul Izquierdo

School of Electronic Engineering and Computer Science,
Queen Mary University of London,
London, United Kingdom
{Stefano.asioli,naeem.ramzan,ebroul.izquierdo}@elec.qmul.ac.uk

Abstract. In this paper, we exploit the characteristics of scalable video and Peer-to-peer (P2P) network in order to propose an efficient streaming mechanism for scalable video. The scalable video is divided into chunks and prioritized with respect to its significance in the sliding window by an efficient proposed piece picking policy. Furthermore the neighbour selective policy is also proposed to receive the most important chunks from the good peers in the neighbourhood to maintain smooth content delivery of certain Quality of Service for the received video. Experimental evaluation of the proposed system clearly demonstrates the superiority of the proposed approach.

Keywords: Scalable video coding, Peer to peer, Bittorrent.

1 Introduction

Multimedia applications over the Internet are becoming popular due to the widespread deployment of broadband access. However, the conventional client-server architecture [1] severely limits the number of simultaneous users, especially for bandwidth intensive applications such as video streaming. On the other hand, Peer-To-Peer (P2P) networking architectures [2] receive a lot of interest, as they facilitate a range of new applications that can take benefit of the distributed storage and increased computing resources offered by such networks. In addition, P2P systems also represent a scalable and cost effective alternative to classic media delivery services. Their advantage resides in their ability for self organization, bandwidth scalability, and network path redundancy, which are all very attractive features for effective delivery of media streams over networks.

In conventional client server applications, the video server requires video contents of different fidelities, such as high quality material for storage and future editing and lower bit-rate content for distribution. In traditional video communications over heterogeneous channels, the video is usually processed offline. Compression and storage are tailored to the targeted application according to the available bandwidth and potential end-user receiver or display characteristics. However, this process requires either transcoding of compressed content or storage of several different versions of the encoded video.

Scalable Video Coding (SVC) [3] promises to partially solve this problem by "encoding once and decoding many". SVC enables content organization in a hierarchical manner to allow decoding and interactivity at several granularity levels. That is, scalable coded bit-streams can efficiently adapt to the application requirements. The SVC encoded bit-stream can be truncated at different points and decoded. The truncated bit-stream can be further truncated to some lower resolution, frame rate or quality as well. Thus, it is important to tackle the problems inherent to the diversity of bandwidth in heterogeneous networks and in order to provide improved quality of services.

In this paper, we exploit the features of scalable video coding and P2P network to perform an efficient video communication over distributed network. The layered structure of the scalable video bit-stream allowed us to achieve efficient on-the-fly rate adaptation. Additionally, SVC gives us the foundation for efficient use of network bandwidth on a P2P network [4] by enabling intermediate high capacity nodes in the overlay to dynamically extract layers from the scalable bit-stream to serve less capable peers. We also proposed an efficient piece picking policy and neighbourhood selection in P2P network for efficient scalable video streaming.

In this paper, Section 2 explains the proposed framework. The main parts of the proposed frame work and proposed optimization are also explained in Section 2. Section 3 provides the experimental evaluation of the proposed technique. Finally, Section 4 concludes this paper.

2 Proposed Framework for Scalable Video over P2P Network

The proposed system is based on two main modules: scalable video coding and the P2P architecture. In this system, we assume that each peer contains the scalable video coder and the proposed policy of receiving chunk is to make sure that each peer at least receives the base layer of the scalable bit-stream for each group of picture (GOP). Under these circumstances, peers could download different layers from different users, as shown in Figure 1. In this section, first we will explain the used scalable video coding and P2P architecture and then the proposed modification to adapt scalable video in P2P network.

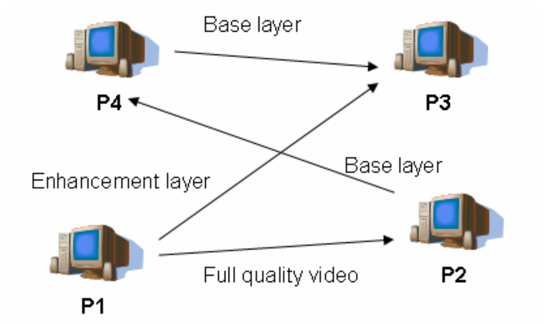

Fig. 1. An example of the proposed system for scalable video coding in P2P network

2.1 Scalable Video Coding

A wavelet-based scalable video aceSVC [5-6] is employed in this research work. Architecture of aceSVC features spatial, temporal, quality and combined scalability. Temporal scalability is achieved through repeated steps of motion compensated temporal filtering [5]. To achieve spatial scalability, each frame is decomposed using a 2D wavelet transform. Coefficients obtained through spatio-temporal decomposition are coded through the process of bit-plane coding [6] which provides basis for quality scalability. The main features of the used codec are: hierarchical variable size block matching motion estimation; flexible selection of wavelet filters for both spatial and temporal wavelet transforms on each level of decomposition, including the *2D* adaptive wavelet transform in lifting implementation; and efficient bit-plane coder.

aceSVC bit-stream organisation: The input video is initially encoded with the maximum required quality. The compressed bit-stream features a highly scalable yet simple structure. The smallest entity in the compressed bit-stream is called a layer, or Atom, which can be added or removed from the bit-stream. The bit-stream is divided into GOPs. Each GOP is composed of a GOP header, the layers and the allocation table of all layers. Each layer contains the layer header, layer data that can be motion vectors data (some layers do not contain motion vector data) and texture data of a certain sub-band. The bit-stream structure is shown in Figure 2.

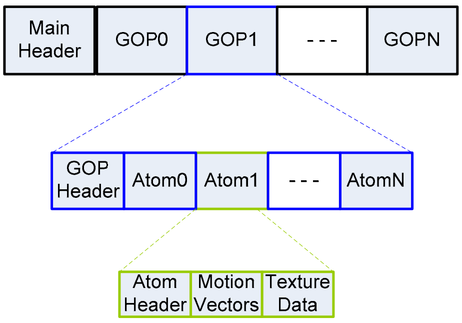

Fig. 2. Detailed description of used scalable bit-stream

In the main header of the aceSVC bit-stream, the organization of the bit-stream is defined so that the truncation is performed at different user points with low complexity.

These atoms are adapted with respect to the user requirements, bandwidth allocation in P2P network for the proposed system.

2.2 P2P Network

BitTorrent [4] is a widely used peer-to-peer protocol developed by Bram Cohen in 2003. The main idea behind it is that users' download rates should be proportional to their upload rates, in order to provide a fair mechanism and motivate users to share more. Free-riding is occasionally accepted, but only if there is enough spare capacity in the system [2]. In the original version of the protocol, this is achieved using a tit-for-tat mechanism, in which peers mainly upload data to peers they are downloading from. Moreover, peers occasionally behave altruistically, in order to discover potentially good neighbours.

In this research work, Tribler [2] is used as a BitTorrent client. It is based on ABC [7] (Yet Another BitTorrent Client), which is itself based on BitTornado [8]. New features include exploitation of social relationships among peers and content discovery through exploration of the network, instead of browsing a torrent repository. Finally, Tribler supports video on demand for non-scalable sequences. In this case a give-to-get algorithm is used, instead of tit-for-tat, which means that peers will upload to peers that have proven to be 'generous' towards third parties, instead of those peers that have a high upload rate.

2.3 Proposed Modification for Efficient Scalable Video Streaming

In this section, we formulate how the scalable layers are prioritized in our proposed system. First we explain how the video segments or chunks are arranged and prioritized in our proposed system and then efficient selection policy of good neighbour for the most important chunks is elucidated.

a. Piece Picking Policy

The proposed solution is a variation of the "Give-To-Get" algorithm [2], already implemented in Tribler. Modifications concern the piece picking and neighbour selection policies.

Scalable video sequences can be split into GOPs and layers as explained in section 2.2, while BitTorrent splits files into pieces [4]. Since there is no correlation between these two divisions, some information is required to map GOPs and layers into pieces and vice versa. This information can be stored inside an index file, which should be transmitted together with the video sequence. Therefore, the first step consists of creating a new torrent that contains both files. It is clear that the index file should have the highest priority and therefore should be downloaded first.

Once the index file is completed, it is opened and information about offsets of different GOPs and layers in the video sequence is extracted. At this point, it is possible to define a sliding window, made of W GOPs and the pre-buffering phase starts. Pieces can only be picked among those inside the window, unless all of them have already been downloaded. In the latter case, the piece picking policy will be the same as the original BitTorrent, which is rarest [piece] first.

Inside the window, pieces have different priorities. First of all, a peer will try to download the base layer, then the first enhancement layer and so on. Pieces from the base layer are downloaded in a sequential order, while all the other pieces are downloaded rarest-first (within the same layer).

The window shifts every $t_{(GOP)}$ seconds, where $t_{(GOP)}$ represents the duration of a GOP. The only exception is given by the first shift, which is performed after the pre-buffering, which lasts W * $t_{(GOP)}$ seconds.

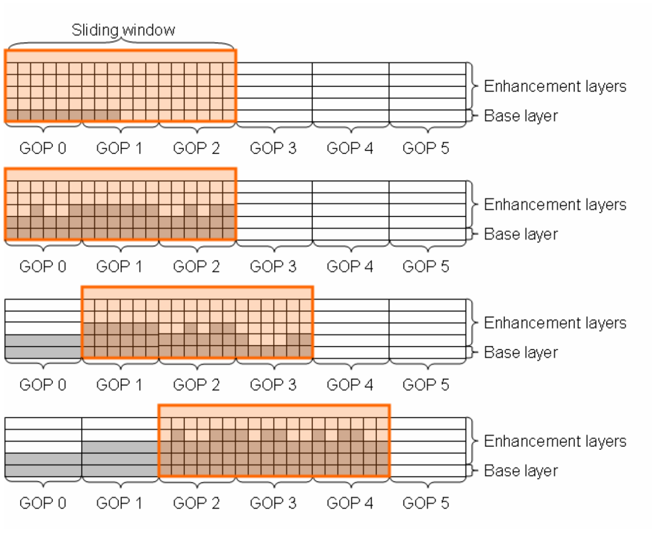

Fig. 3. Sliding window for scalable video bit-stream a) Pre-buffering phase start, b) Pre-buffering phase end, c) The window shifted after GOP0, d) The window shifted after GOP1

Every time the window shifts, two operations are made. First, downloaded pieces are checked, in order to evaluate which layers have been completely downloaded. Second, all pending requests that concern pieces belonging to a GOP that lies before the window are dropped. An important remark is that the window only shifts if at least the base layer has been received, otherwise the system will auto-pause. Figure 3 shows the behaviour of the system with W = 3. An early stage of the pre-buffering phase is showed in Figure 3a. The peer is downloading pieces from the base layer in a sequential way, while in Figure 3b the first two layers have been downloaded and pieces are being picked from the enhancement layer 2 according to a rarest-first policy. These pieces can belong to any of the GOPs in the window. In Figure 3c, the window has shifted as not all the pieces of enhancement layer 2 of GOP 0 have been received; this layer and higher layers are discarded. Inside the window, before downloading any other pieces from GOP 1 or GOP 2, the system will pick pieces from GOP 3 until the quality of the completed layers is the same. In other words, before picking any pieces that belongs to enhancement layer 2, pieces belonging to base layer of GOP 3 and enhancement layer 1 of GOP 3 have to be picked. In Figure 3d all the GOPs have the same number of complete layers and pieces are picked from enhancement layer 3. Another issue is the wise choice of the neighbours.

b. Neighbour Selection Policy

It is extremely important that at least the base layer of each GOP is received before the window shifts. Therefore, pieces belonging to the base layer should be requested from good neighbours. Good neighbours are those peers that own the piece with the highest transfer rates, which alone could provide the current peer with a transfer rate that is above a certain threshold. This threshold is determined by the current number of base layer pieces in the window and it is the minimum rate that allows this layer to be received on time. During the pre-buffering phase, any piece can be requested from any peer. However, every time the window shifts, the current download rates of all the neighbours are evaluated and the peers are ranked. After this operation, the base layer is only requested from the peers that have a download rate above this threshold value. The following algorithm explains the best neighbour policy in detail.

Algorithm: Sort neighbours according to download rate

```
Threshold = (# of pieces in layer 0 inside the window) * (piece length)/ (duration of the window)
for i in sorted list :
    if i owns the piece
        total download rate += download rate [i]
    end if
    if total download rate > Threshold:
        all the pieces with a download rate lower than i are bad peers
    end if
end for
```

3 Experimental Results

The performance of the proposed framework has been extensively evaluated to transmit wavelet-based SVC encoded video over P2P network. The network used for the experiments consists of three peers: two seeders and one leecher. No restriction was applied to download bandwidth of the leecher, while, as far as the two seeders are concerned, upload bandwidth was limited to 50 kbytes/s and 25 kbytes/s respectively. The leecher downloaded three five-minutes video sequences (Crew; City; Soccer) at CIF resolution and 30fps. The results are shown in “Download Rate” of the receiver and the “Received Video Bitrate” against time. Some selected results are shown in Figure 4-6.

All three graphs show similar results. In none of the cases the download rate is high enough to allow the transfer of all the layers and we have quality degradation. Moreover, the quality is not constant and follows the behaviour of the download speed. Finally, in all three cases we have a higher quality immediately after the prebuffering, because these GOPs are in the window for a slightly longer period, and at the end of the sequence, when we cannot shift the window anymore and we need to shrink it.

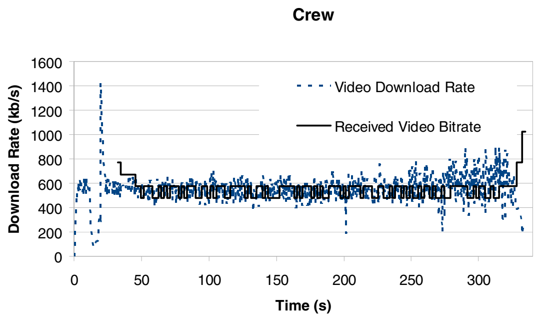

Fig. 4. Received download rate and received video bitrate for Crew CIF sequence

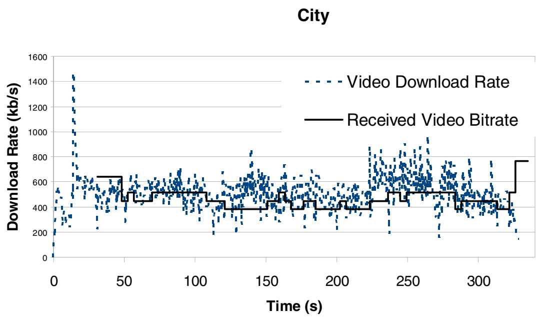

Fig. 5. Received download rate and video bitrate for City CIF sequence

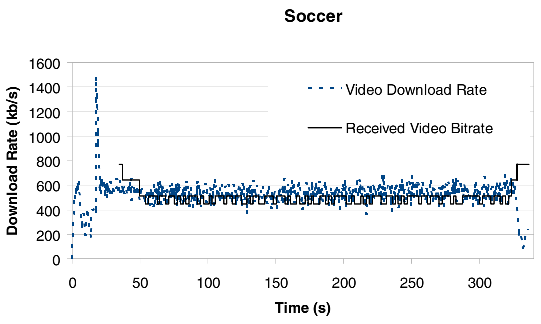

Fig. 6. Received download rate and video bitrate for Soccer CIF sequence

These are the preliminary results of the proposed technique when we have enough download rates to download the video. However we are working on scenarios when we don't have enough download speed to always download the base layer of scalable video. One solution will be to autopause the video until we get the full quality or we can downgrade the video to its lower resolution or frame rate. As these are the open questions in the community, so we are also working in this direction.

4 Conclusion

In this paper, we have presented a novel technique for scalable video streaming over P2P network. We proposed the new piece picking policy and neighbour selection policy in P2P network for efficient streaming of P2P network. Some selected results were presented, which show the efficiency of the proposed system. The future work will concentrate on how to efficiently receive layers of scalable video when the network doesn't allow us enough download speed to download the base layer of the scalable video.

References

1. Ramzan, N., Zgaljic, T., Izquierdo, E.: An Efficient Optimisation Scheme for Scalable Surveillance Centric Video Communications. Signal Processing: Image Communication 24, 510–523 (2009)
2. Pouwelse, J.A., Garbacki, P., Wang J., Yang, J., Iosup, A., Epema, D., Reinders, M., van Steen, M.R., Sips, H.J.: Tribler: A social-based based peer to peer system. In: 5th Int'l Workshop on Peer-to-Peer Systems (IPTPS) (February 2006), `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.8696`
3. Mrak, M., Sprljan, N., Zgaljic, T., Ramzan, N., Wan, S., Izquierdo, E.: Performance Evidence of Software Proposal for Wavelet Video Coding Exploration Group. Technical Report, ISO/IEC JTC1/SC29/WG11/MPEG2006/M13146 (2006)
4. Cohen, B.: Incentives build robustness in BitTorrent. In: Proc. of First Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA (June 2003)
5. Mrak, M., Izquierdo, E.: Spatially Adaptive Wavelet Transform for Video Coding with Multi-Scale Motion Compensation. In: IEEE International Conference on Image Processing, September 2007, vol. 2, pp. 317–320 (2007)
6. Zgaljic, T., Sprljan, N., Izquierdo, E.: Bit-Stream Allocation Methods for Scalable Video Coding Supporting Wireless Communications. Signal Processing: Image Communications 22, 298–316 (2007)
7. Tucker, T., Pate, D., Rattanapoka, C.: Yet Another Bittorrent Client, `http://pingpongabc.sourceforge.net/`
8. `http://www.bittornado.com`