MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

An Overview of Scalable Video Streaming

Huifang Sun, Anthony Vetro, Jun Xin

TR2007-007 February 2007

Abstract

During the past two decades, video coding technology has matured and stat-of-the-art coding standards have become very important part of the video industry. Standards such as MPEG-2 [16] and H.264/AVC [20] provide strong support for digital video transmission, storage and streaming applications.

Wireless Communication and Mobile Computing

Published in *Wireless Communications and Mobile Computing, Feb 2007*

An Overview of Scalable Video Streaming

Huifang Sun, Anthony Vetro and Jun Xin

Mitsubishi Electric Research Labs

I. INTRODUCTION

During the past two decades, video coding technology has matured and state-of-the-art coding standards have become very important part of the video industry. Standards such as MPEG-2 [16] and H.264/AVC [20] provide strong support for digital video transmission, storage and streaming applications.

Video streaming addresses the problem of transferring video data as a continuous stream. With streaming, the end-user can start displaying the video data or multimedia data before the entire file has been transmitted. To achieve this, the bandwidth efficiency and flexibility between video servers and equipment of end-users are very important and challenging problems. In response to such challenges, a variety of video coding and streaming techniques have been proposed to provide video streaming services [1]–[10]. In [1]-[3], scalable video streaming over the Internet has been comprehensively investigated. Two streaming approaches were discussed: switching among multiple pre-encoded non-scalable bitstreams and streaming with a single scalable bitstream. In [4], a brief overview of the diverse range of video streaming and communication applications has been introduced. The different classes of video applications provide different sets of constraints and degrees of freedom in system design. The three fundamental challenges in video streaming: unknown and time-varying bandwidth, delay jitter, and loss, must be addressed in video streaming.

The methods of scalable video coding and transcoding have been proposed to provide solutions to these problems. Such techniques aim to adjust the amount of data to be transmitted according to changes in bandwidth. In [6]-[9], the problems of bit allocation and error resilience have been investigated. From the literature, it is evident that the methods of video coding and scalable video distribution are the two key issues for video streaming systems.

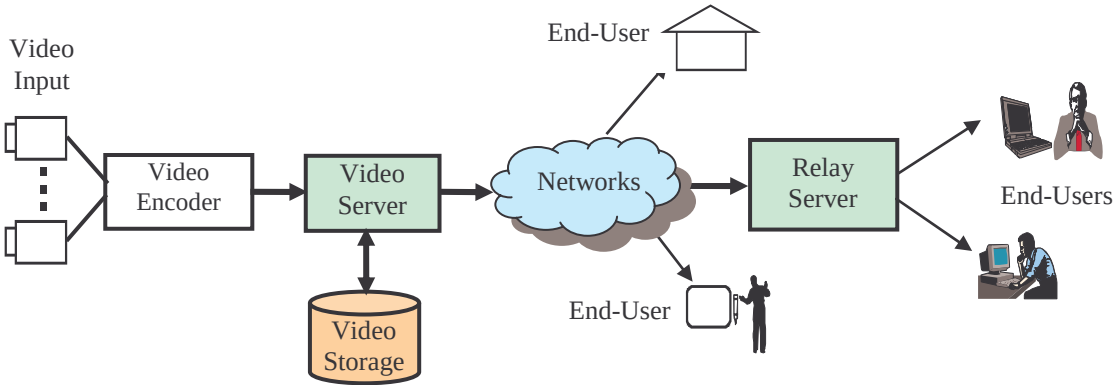

Figure 1. A typical video streaming system

A typical video streaming system is shown in Figure 1, which consists of an encoder, a distribution server with video storage, a relay server and end-users that receive the video data. The distribution server stores the encoded video data and begins to distribute the data at the client's demand. Users can watch the video whenever and wherever by accessing the server over the networks. Encoding and distribution is carried out in real time in the case of live distribution and may not be performed in real time for on-demand type of applications.

For video encoding, there are two ways to compress the video signals: non-scalable video coding and scalable video coding. In non-scalable video coding, the video content is encoded independent of actual channel characteristics. In this method, coding

efficiency is the most important factor and the compression is optimized at a pre-specified rate. The main problem with this method is that it is difficult to adaptively stream non-scalable video contents to heterogeneous client terminals over time-varying communication channels. This is especially true for wireless applications. On the other hand, with scalable video coding, video needs to be encoded only once, then by simply truncating certain layers or bits from the single video stream, lower qualities, spatial resolutions and/or temporal resolutions could be obtained. As an ultimate goal, the scalable representation of video should be achieved without impact on the coding efficiency, i.e., the truncated scalable stream (at lower rate, spatial and/or temporal resolution) should produce the same reconstructed quality as a single-layer bitstream in which the video was coded directly under the same conditions and constraints, notably with the same bit-rate. However, practically all scalable video coders suffer loss in compression efficiency relative to state-of-the-art non-scalable coders.

For the distribution of video bitstreams, the video server and relay server are generally responsible for matching the output data to the available channel resources and ultimately the client's device capabilities. For non-scalable video data, the server may transcode the bitstream to reduce the bit rate, frame rate or spatial resolution [12][13]. Alternatively, it may select the most appropriate bitstream from multiple pre-encoded streams having different quality, spatial resolution, etc. Considering loss characteristics of the networks, the servers may also add error resilience to the output bitstream [14]. Generally speaking, the optimal solution is the one that yields the highest reconstructed video quality at the receiver. For more discussions on error resilience and error concealment, the readers are

referred to [14][44]. Overall complexity in the system, including servers and clients is another important consideration.

Note that this paper discusses streaming techniques mainly from signal processing perspective. Other solutions are not covered in this paper, including content delivery networks such as Akamai's. Readers are referred to [42] for an interesting discussion of both types of solutions.

The rest of this paper is organized as follows. In next section, we review scalable video coding techniques. In Section III, various video streaming methods are presented. Various network related issues for scalable video streaming are covered in Section IV. In Section V, a specific method of scalable video streaming that is based on regions-of-interest is discussed. Finally, the concluding remarks are given in Section VI.

II. Overview of Related Video Coding Technology

A) Video Coding Standards

As mentioned previously, video coding plays an important role in bridging the gap between large amounts of visual data and limited bandwidth networks for video distribution. During the past two decades, several video coding standards have been developed to satisfy industry needs. The video coding standards have been developed by two major groups of standard organizations. One is the Moving Pictures Expert Group (MPEG) of ISO/IEC, and the other is the Video Compression Expert Group (VCEG) of ITU-T. The video coding standards developed by ISO/IEC include MPEG-1 [15], MPEG-2 [16], and MPEG-4 [17]. The standards developed by ITU include H.261 [18], H.262 [16], H.263 [19] and H.264/AVC [20]. It is noted that H.262 is the same as MPEG-2, which is a joint standard of MPEG and ITU. The H.264/AVC video coding

standard is developed by the joint video team (JVT) of MPEG and ITU which is also MPEG-4 Part 10. These standards have found many successful applications such as DTV, DVD, digital telephony and applications on video streaming.

- H.261 was completed in 1990, and it is mainly used for ISDN video conferencing.
- H.263 was completed in 1996 and it is based on the H.261 framework but includes many additional algorithms to increase the coding performance.
- MPEG-1 was completed in 1991. The target application of MPEG-1 is digital storage media, CD-ROM, at bit rates up to 1.5 Mbps.
- MPEG-2, sometimes also referred to as H.262, was completed in 1994. It is an extension of MPEG-1 and allows for greater input format flexibility and higher data rates for both High-definition Television (HDTV) and Standard Definition Television (SDTV). The US ATSC DTV standard and European DTV standard DVB both use MPEG-2 as the source-coding format. The MPEG-2 is also used for Digital Video Disk (DVD).
- MPEG-4 Part 2 was completed in 2000. It is the first object-based video coding standard and is designed to address the highly interactive multimedia applications. The Simple Profile and Advanced Simple Profile of MPEG-4 Part 2 have been used for mobile application and streaming.
- H.264 is also referred to as MPEG-4 Part 10 Advanced Video Coding. It is the latest video coding standard, which has been developed by the joint video team of ISO and ITU. H.264 has greatly improved the coding performance over MPEG-2 and MPEG-4 Part 2. The target applications of H.264 are broadcasting television, high definition DVD, digital storage, and mobile applications.

A summary of these video coding standards is shown in Table 1. Currently, the most popular video coding standards for video streaming include MPEG-2, MPEG-4 Part 2 (Simple Profile and Advanced Simple Profile) and H.264/AVC (Baseline Profile). It should be noted that besides the video coding standards developed by MPEG and VCEG, there are also video coding schemes such as VC-1 (Draft SMPTE Standard) developed by Microsoft, and RealVideo developed by Real Networks. Such media formats are extensively used for video streaming over the Internet.

B) Scalable Video Coding

The efforts on developing scalable video coding (SVC) schemes have been continued for many years in video coding community in response to the emerging applications of video transmission over heterogeneous wired/wireless networks [23]-[25], [40]. The main purpose of scalable video coding is to encode video into a scalable bitstream such that videos of lower qualities, spatial resolutions and/or temporal resolutions can be generated by simply truncating the scalable bitstream. Obviously the scalability makes it easy to meet the bandwidth conditions, terminal capability and quality of service requirement in streaming video applications. In this paper, we focus on standard related activities. We refer readers to [40] for recent advances in scalable video coding research beyond standards.

This effort started from MPEG-2 [16] where the feature of scalable video coding has been developed. In MPEG-2, the video signal is encoded into a base layer and a few enhancement layers, in which the enhancement layers add spatial, temporal, and/or SNR

quality to the reconstructed base layer. The structure of the scalable video coding based on one base layer and several enhancement layers is shown in Figure 2.

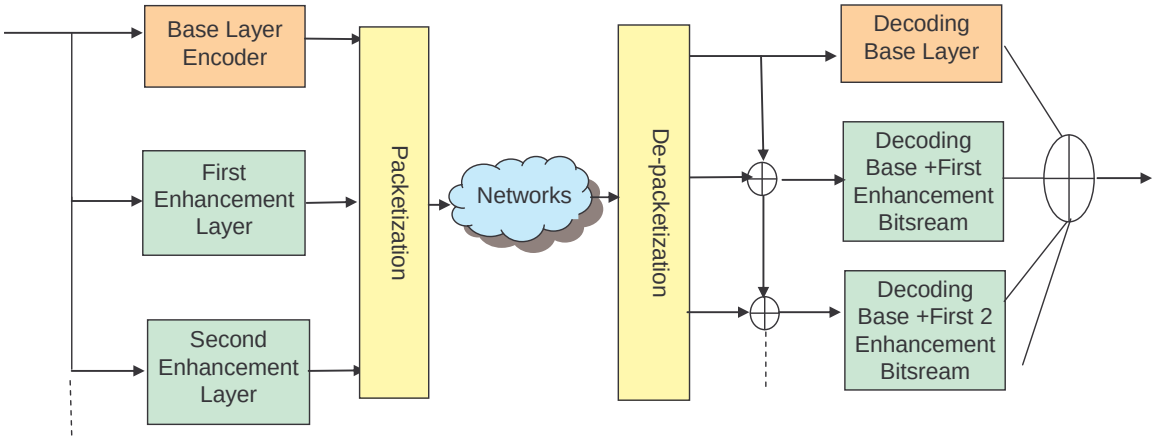

Figure 2. Structure of scalable video format including one base layer and several enhancement layers

Specifically, the enhancement layer in SNR scalability adds refinement data for the DCT coefficients of the base layer. With spatial scalability, the first enhancement layer uses predictions from the base layer without the use of motion vectors. In this case, the layers can have different frame sizes, frame rates, and chrominance formats. In contrast to spatial scalability, the enhancement layer in temporal scalability uses predictions from the base layer using motion vectors, and while the layers must have the same spatial resolution and chrominance formats, they may have different frame rates. The MPEG-2 video standard supports each of these scalable modes, as well as hybrid scalability, which is the combination of two or more types of scalability. It should be noted that the base layer bitstream and enhancement layer bitstreams can be packetized in different packets which can be transmitted with the same channel or different channels depending the network structure.

In MPEG-4 video coding standard, the same concept of scalable video coding has been extended to object-based scalability, which includes spatial, temporal and SNR scalability. Furthermore, a new form of scalability, known as fine granular scalability (FGS), has been developed as part of the MPEG-4 video standard [11][23]. In contrast to conventional scalable coding schemes, FGS allows for a much finer scaling of bits in the enhancement layer. This is accomplished through a bit-plane coding method of DCT coefficients in the enhancement layer, which allows the enhancement layer bit stream to be truncated at any point. In this way, the quality of the reconstructed frames is gradually improved with the number of enhancement bits received. FGS suffers significant compression efficiency loss at higher bitrates since only low quality base layer video frames are used as reference. Enhanced FGS schemes have been proposed to address this problem, including progressive FGS (PFGS) [47] and Motion-compensation FGS (MC-FGS) [48] etc. In PFGS, enhancement layers are allowed to be predicted from either base layer or enhancement layer reference frames. In addition, PFGS also introduces a drifting model to estimate the drifting errors at encoder. As a result, PFGS can improve coding efficiency significantly at higher bitrates. Note that the FGS of new scalable video coding (SVC) standard (see below) contains the above technologies.

Even though the MPEG-4 FGS has certain advantages over the previous scalable video coding schemes, it still has not found practical applications. There may be several reasons. The first is the coding efficiency. The FGS scheme still incurred notable penalties in coding efficiency, which is a sacrifice that content and service providers would not like to make. The second reason is the increase of complexity and therefore cost of decoders.

Despite these issues with prior scalable coding schemes, researchers have continued efforts on developing new scalable video coding techniques since scalability is still a very attractive way to achieve universal multimedia access (UMA) [24]. The MPEG community is now developing new scalable video coding standard [25] and has made significant progresses.

The new scalable video coding (SVC) standard is being designed based on the H.264/AVC coding tools and is still under joint development within MPEG and ITU-T [25]. It is expected that this new standard will overcome much the problem of loss in coding efficiency compared with existing non-scalable coding. An important concept to achieve efficient scalable coding is Motion Compensated Temporal Filtering (MCTF), which is based on the lifting scheme [43]. The lifting scheme insures perfect reconstruction of the input in the absence of quantization of the decomposed signal even if non-linear operations are used during the lifting operation. The benefits of this filtering approach for scalable coding could be found in [40].

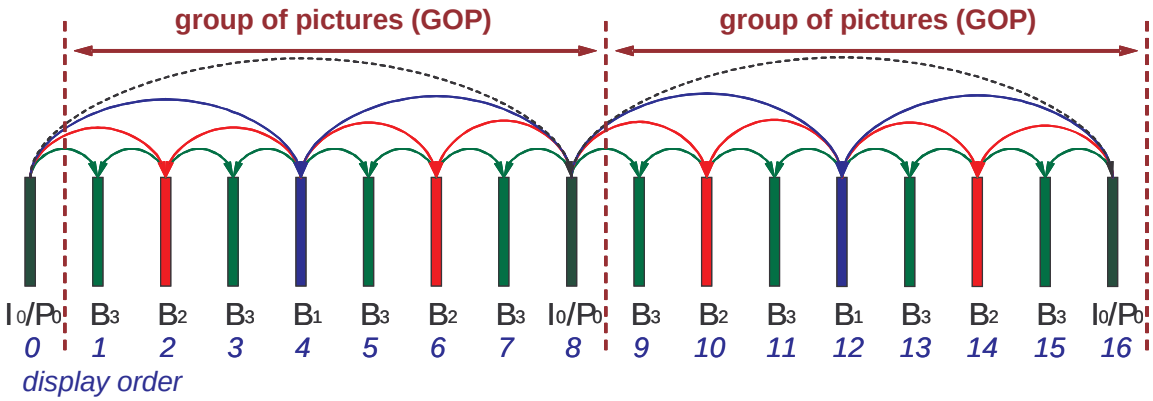

Figure 3. Dyadic hierarchical coding structure with 4 temporal levels and a GOP size of 8. Each B pictures is predicted using 2 reference pictures, which are the nearest pictures of the lower temporal level from the past and the future (from [25])

In the new SVC scheme, the dimensions of scalabilities include spatial, temporal and quality (SNR) scalabilities. Temporal scalability is enabled by hierarchical B pictures, which is illustrated in Figure 3 [25]. In this example there are four temporal scalability levels. The pictures of the coarsest temporal resolution are encoded first, and then B pictures are inserted at the next finer temporal resolution level in a hierarchical manner. Spatial scalability is achieved by using a layered approach, which is the same as in MPEG-2 Video. To achieve SNR scalability, two different approaches are provided: one is the use of embedded quantization for coarse scalability and another is the use of fine grain scalability (FGS), which is based on the principle of sub-bitplane arithmetic coding. When FGS layers are used, two closed motion compensation loops may be used at the encoder side in order to improve the coding efficiency; one loop is used for coding the base layer and the other loop is used for coding the enhancement layers. To achieve better coding efficiency, the reference for coding the enhancement layer corresponds to the highest FGS rate as show in Figure 4 [25].

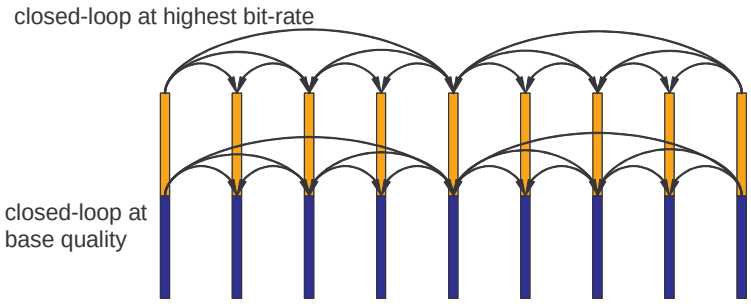

Figure 4. Encoder control with two closed motion compensation loops (from [25])

Finally, we would like to indicate that error resilience technologies are very important for video streams over error-prone wireless or IP-networks. There are several error resilience technologies which have been developed for scalable video coding schemes

[45][46]. These technologies are quite promising. In particular, unequal error protection is a natural fit for protecting FGS video due to the different importance of different layers. Such techniques have been shown to be rather effective [46].

C) Video Transcoding

Theoretically, it is very easy for the server to handle the video streaming process with a scalable compressed video bitstream since this bitstream can be easily truncated to fit the bandwidth requirement. However, due to the reasons mentioned in the previous section, servers will typically store non-scalable bitstreams. In this case, transcoding may be applied to transfer the bitstreams to the proper bandwidth required by the networks or the proper spatial or temporal resolution to match the end-user's device capability. The basic requirements for video transcoding are: 1) the complexity should be as low as possible compared with the cascaded method of full decoding and full re-encoding, and 2) video quality should not be degraded compared to the cascaded full decoding and full re-encoding approach. An example of typical transcoding operations is shown in Figure 5.

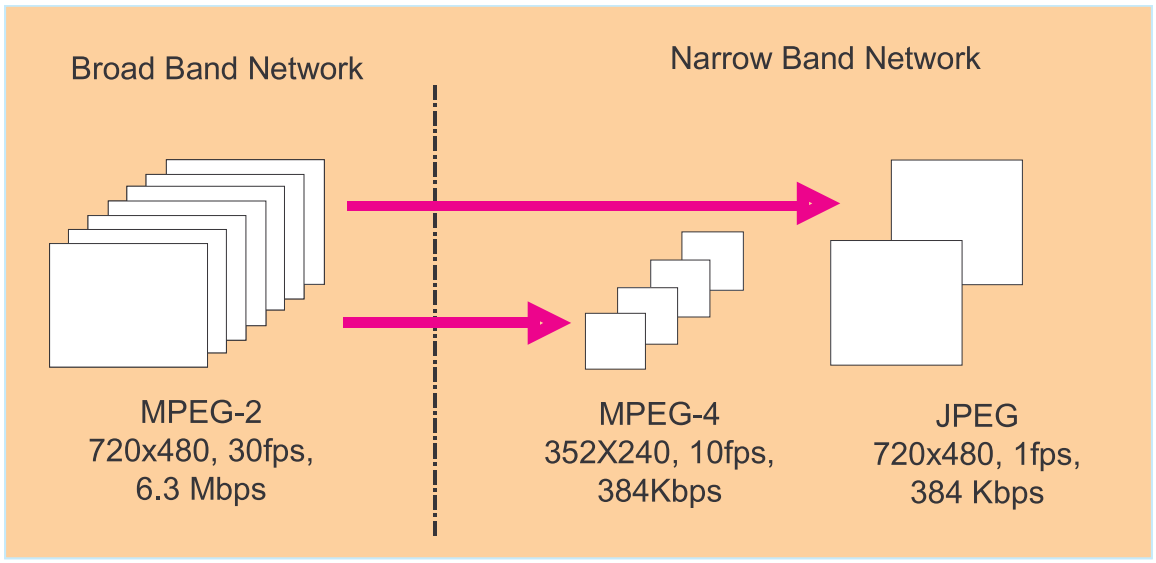

Figure 5. An example of video transcoding: MPEG-2 bitstream is converted to MPEG-4 or JPEG at lower bit rate, lower frame rate and/or lower resolution

The techniques developed for transcoding are aimed at avoiding the full decoding and re-encoding of streams to satisfy network conditions and terminal capabilities. These techniques have greatly reduced the complexity of converting a bitstream, while still maintaining high picture quality. Extensive reviews of transcoding technology exist [12][13], and readers are referred to these articles for further information.

D) Bitstream Switching

Video streaming is an important application over IP networks and 3G wireless networks. However, due to time varying network conditions, the effective bandwidth for a user may vary accordingly. Therefore, the video server should change the bit-rate of the compressed video streams or switch to a more appropriate bitstreams to accommodate the bandwidth variations [28].

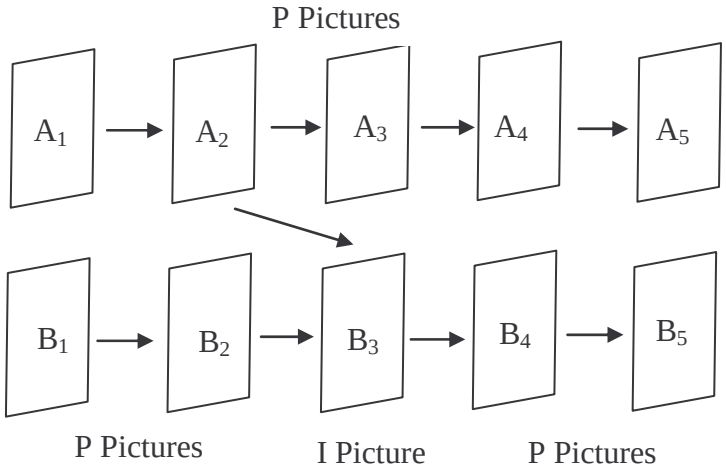

Figure 6. A decoder is decoding Stream A and wants to switch to decoding Stream B

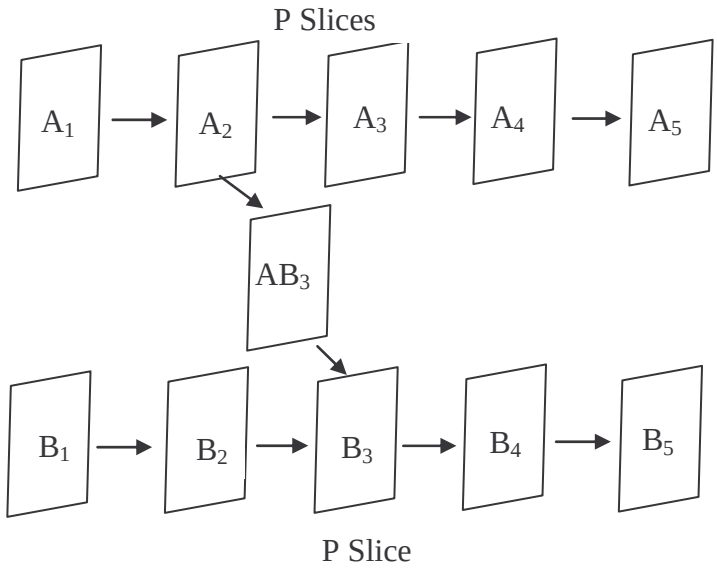

Figure 7. Switching streams using SP-slices

For servers working with non-scalable bitstreams, the switching usually happens at a random access point in a sequence, e.g., an intra-coded frame. This is illustrated in Figure 6. For simplicity, we assume that each frame is predicted from one reference. Also, we

assume that stream A is coded with higher bit rate and stream B is coded with lower bit rate. After decoding P-Pictures A_1 and A_2 in Stream A, the decoder wants to switch to Stream B and decode B_3, B_4 and so on. The main problem in switching bitstreams is avoiding drift. Drift can be explained as a deviation in pixel values from the original video that increases over successively predicted frames. In the context of bitstream switching, attempting to predict a current frame from a different reference frame than originally used for encoding would also cause a mismatch and result in drift. In the context of transcoding, it is usually caused by the loss of high frequency data, which creates a mismatch between the actual reference frame used for prediction in the encoder and the degraded reference frame used for prediction in the transcoder and decoder. In the above example, since B_3 is an intra-coded frame, drift-free switching can be accomplished and loss of frames due to network congestion could be avoided. The server can dynamically switch from the higher rate bitstream to the lower rate bitstream when it detects a drop in the network bandwidth. In this way, the bitstream switching is accomplished by inserting an I-Picture at regular intervals in the coded sequence to create *switching points*.

The problem with this method is that the more random access frames (usually I-frames) that are added to the non-scalable stream, the larger the impact on coding performance since more frequent I-frames will generally cause a decrease in coding efficiency. Also, since the number of bits to code I-frames is generally much larger than the number of bits used to code P-frames or B-frames, the bit-rate tends to spike at each switching point. This variation requires a larger buffer and implies larger delay, which may not be acceptable for certain real-time applications.

In order to address the above problems, Switching P (SP) and Switching I (SI) slices have been proposed in the new video coding standard H.264/AVC [29]. The main purpose of SP and SI slices is to enable efficient switching between video streams and efficient random access for video decoders. With SP slices, it becomes possible to transition from one stream coded at a specific bit-rate to another stream coded at a different bit-rate without causing drift and maintaining a more stable output bit-rate.

For simplicity, assume we have two streams A and B as shown in Figure 7. After decoding P-slices A_1 and A_2 in Stream A, the decoder wants to switch to Stream B and decode B_3, B_4 and so on. The SP-slices are placed at the switching points. As shown in Figure 7, the SP-slice AB_3 is predictively encoded with respect to A_2 to reconstruct B_3. In this way, the SP-slice will not result in a peak in the bitstream since it is coded using motion compensated prediction, which is more efficient than intra coding. Also, this switching between streams will not result in any drift.

For servers working with scalable bitstreams, the use of bitstreams switching for adapting to the changes in network bandwidth is relatively easy. In most scalable video coding schemes, the video sequence is usually encoded into base layer and several enhancement layers. The base layer is encoded to a non-scalable bitstream and the bitstream truncation is performed in the enhancement layers, e.g., a bitstream encoded with MPEG-4 FGS can theoretically be truncated at any point in the enhancement layer bitstream, which is suitable to accommodate network bandwidth variations. However, its coding performance is much lower than the non-scalable video coding because its motion compensation is based on the lowest quality base layer. The new scalable video coding scheme tries to solve this problem by allowing prediction from a high quality reference

picture of the enhancement layer. However, there are also problems with this approach. When truncation happens at enhancement layer, the prediction from a high quality references may be corrupt or invalid. In [28], a scheme for adaptively switching between layers of two scalable bitstreams has been proposed to address this issue. In this scheme, two scalable bitstreams are encoded with different quality base layers and switching is only performed on the base layers of the scalable bitstreams. The advantages of this scheme include the high coding efficiency and drift-free switching.

III. VIDEO STREAMING METHODS

A) Overview

From decoder side, there are two approaches to view video over networks, which have been extensively investigated in recent years. The first is the downloading-based approach, where the complete video file is downloaded to local storage before playback. With this approach, the time to download a video increases with the amount of data, which is proportional to the quality of the video and duration of the video. The network bandwidth also plays a significant role in the downloading time as well. The other way to view video is video streaming, where the video is viewed while it is being transmitted. Video streaming will be the focus of the following discussion.

In video streaming, the end user can start viewing the video almost as soon as it begins downloading with a limited delay. To achieve a seamless playback, the data must be received at a rate that allows the client device to decode and display each frame of the video sequence according to a playback schedule. The video server has two ways to provide the compressed video bitstreams. The first is to select one among multiple non-

scalable bitstreams and the second is the use of a single bitstream, which is encoded with the scalable video encoding or can be transcoded during the streaming [26][27].

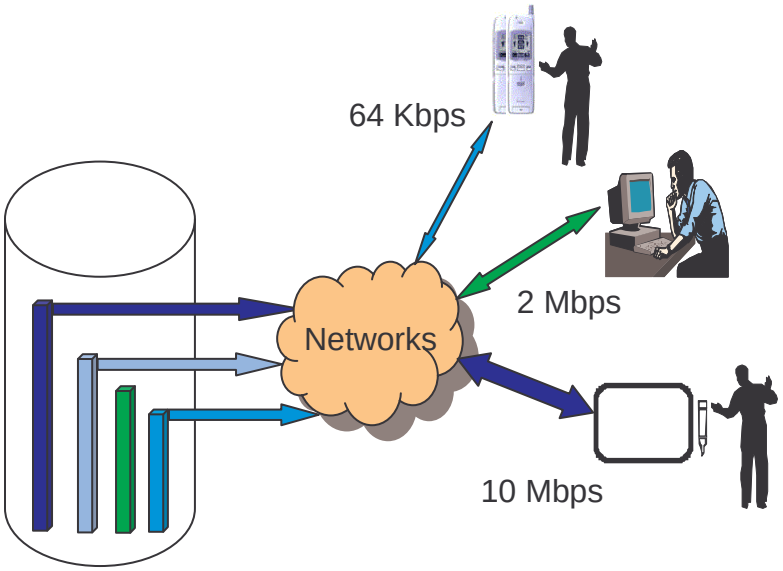

Figure 8. Streaming with multiple pre-encoded bitstreams with different bit rates, frame rates and spatial resolutions

In the first way, several bitstreams for the same video with different bit-rates, which may also have different temporal or spatial resolutions, have been stored in the video server. The end-user can select the bitstream according to its capability and available bandwidth of the network. This method is shown in Figure 8. The advantages of this method are that the compressed bitstream is optimized to the specified user and is the decoder has lower complexity since it only needs to receive and decode a single layer. The main disadvantage is that the video sever must store multiple bitstreams for the same video, which is redundant and could impose significant memory constraints with very large video repositories. Also, the different versions of the video have to be pre-encoded,

which makes real-time applications almost impossible. This approach is also limited in the granularity that it could provide.

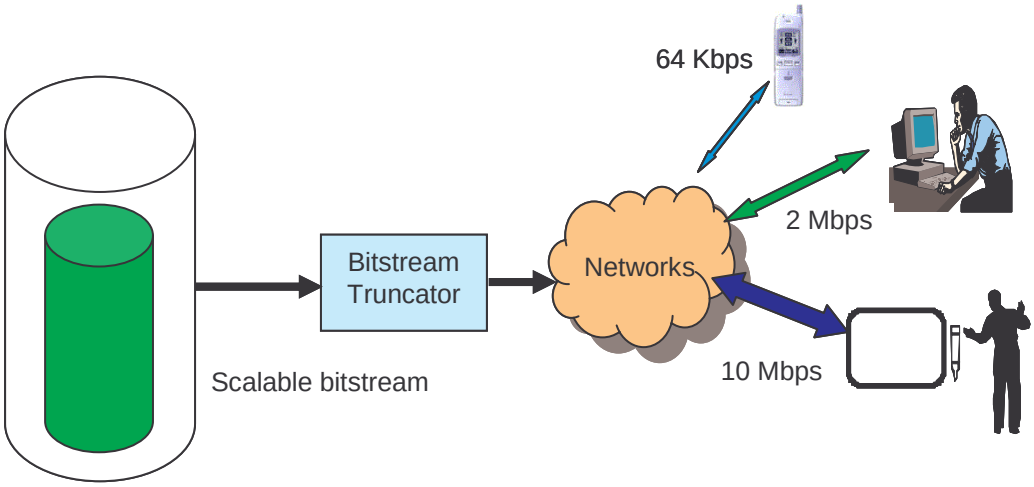

Figure 9. Streaming with bitstream coded with scalable video encoding

The second method of scalable video streaming is implemented with the scalable encoded video bitstreams [23][25]. In this method the video is encoded once and stored in the video sever. The encoded video bitstream can be truncated in ways such as SNR, temporal and spatial scalability based on the requirements of the end user and network conditions as shown in Figure 9. This method is attractive since it provides more flexibility in getting the desired compromise between granular scalability and coding performance. As mentioned previously, this method has to be evaluated by the market. First, the coding technique must not incur significant loss of coding efficiency compared to single layer coding schemes. With significant loss in coding efficiency, it is likely that the content providers would choose not to adopt the coding format. The other issue is decoder complexity. If the scalable decoder is costly to produce, then there may be limited or no deployment of devices capable of receiving a scalable encoded bitstream.

The third method is to use a single encoded bitstreams with higher quality as shown in Figure 10. During the streaming, the bitstreams are converted to match the end user

device and network conditions with a transcoder. The key advantage of this method is that transcoding techniques could be easily installed on servers to satisfy a very diverse set of network and terminal constraints. The transcoding solution offers a layer of flexibility between the content providers who encode the data and consumers that wish to receive the data. The main drawback compared to the scalable coding solution is that transcoding typically requires more computation than simple bitstream truncation. However, advances in the area of transcoding have pushed the complexity much lower than full re-encoding of video without sacrificing quality.

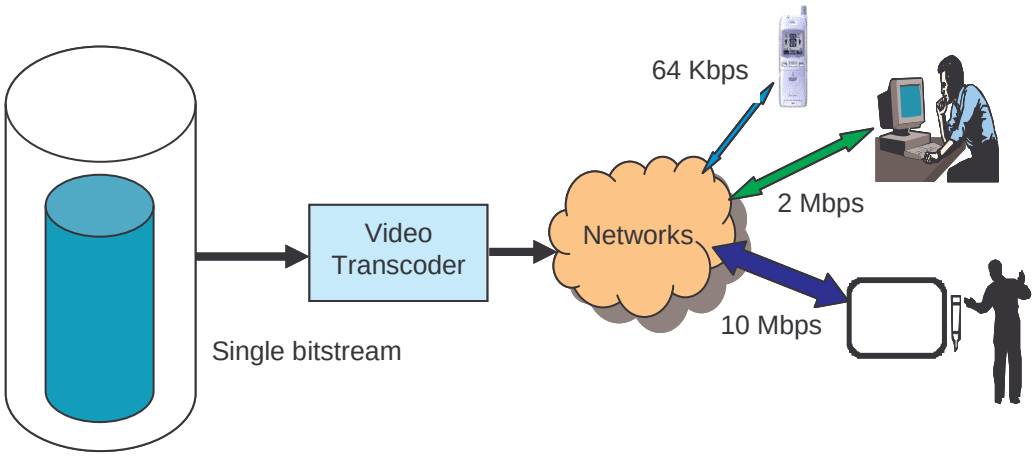

Figure 10. Streaming with transcoding a single bitstream

B) Comparison of Streaming Methods

As described in the previous section, scalable coding specifies the data format at the encoding stage independently of the transmission requirements, while transcoding converts the existing data format to meet the current transmission requirements. With scalable video coding, the video is encoded once, then various qualities, spatial resolutions, and/or temporal resolutions could be extracted. Ideally, this scalable

representation of the video should be achieved without any impact on the coding efficiency.

While coding efficiency is indeed very important, the application space must also be considered. For instance, content providers for high-quality mainstream applications, such as DTV and DVD, have been using single-layer MPEG-2 video coding as the default format, hence a large number of MPEG-2 coded video content already exists, and these industries are now moving towards the H.264/AVC coding format. To access such contents from various devices with varying terminal and network capabilities, transcoding is needed.

A comparison of advantages and disadvantages of the different video streaming methods is given in Table 2. It is indicated that while a single scalable bitstream has small storage needs and facilitates simple bitstream switching, there is a potential loss of coding efficiency and a more complicate decoder would be required. On the other hand, transcoding requires some additional processing; hence there is some additional complexity and potential delay at the transmission side, but the resulting stream could be received by standard single-layer decoders.

In the near-term, scalable coding may satisfy a wide range of video applications such as surveillance and Internet streaming, while transcoding will continue to bridge gaps between legacy content formats and new devices. We believe that the various streaming methods based on scalable coding, video transcoding and bitstream switching should not be viewed as opposing or competing technologies. Instead, they are technologies that meet different needs in a given application space and it is likely that they will coexist.

IV. NETWORK PROTOCOLS FOR SCALABLE VIDEO STREAMING

TCP is the dominant protocol in the Internet for data transfer. In general, TCP could also be used for video streaming over Internet [31]. However, in order to provide reliable and good quality video streaming over TCP, several problems have to be addressed. The first is how to handle the data rate variability. In the Internet, the data rate may have saw-tooth behavior, i.e., additive increase and multiplicative decrease. The second is the end-to-end delay due to retransmission at same time. However, these problems can be addressed with buffering the data. Therefore, the proper buffer size should be decided considering the impact on various performance metrics such as delay, smoothness of playback and data loss. In general, a small buffer size implies smaller delay since the time between the start of transmission and the first picture being displayed is less with a smaller buffer. With regards to smoothness of playback, a larger buffer size will typically ensure smoother playback since larger variations in the bit-rate and transmission time could be tolerated. Larger buffer sizes will also lead to fewer dropped packets in the receiver due to buffer overflow. Given these dependencies, being able to analytically model a video streaming system with TCP is necessary. In [1] and [3], the minimum buffer size requirements for three scenarios have been studied: 1) when TCP throughput matches video encoding rate, 2) when TCP throughput is smaller than the encoding rate, and 3) when TCP throughput is limited by the maximum window size. Another problem with video streaming over TCP is how it handles network layer packet loss. If packets are delayed or damaged, TCP will effectively stop traffic until either the original packets or backup packets arrive. In this sense, TCP is unsuitable for video streaming because TCP

handles the packet loss with the method of retransmission which causes further jitter and skew.

UDP (User Datagram Protocol) is another network protocol for video streaming [32]. UDP handles the packet loss or delay in different way. It allows packets to drop out if these packets are timeout or damaged. This function introduces the packet loss which user can hear or see video damaged, but the stream will continue. With UDP, the error concealment function may be needed in video decoders. Another problem with UDP is that many network firewalls block UDP information. In this case, video streaming over TCP is the only choice since it can get around the firewalls using well-known port numbers (e.g., HTTP or RTSP).

RTP (Real-time Transport Protocol) is alternative choice for video streaming [33]. RTP is an Internet standard protocol for the transport of real-time data, including audio and video. RTP consists of two parts, a data part and a control part which is called RTCP. The data part of RTP supports real-time transmission for continuous media such as video and audio. It provides timing reconstruction, loss detection, security and content identification. The RTCP (RTP control protocol) part provides source identification and support for gateways like audio and video bridges as well as multicast-to-unicast translators. It offers Quality-of-Service (QoS) feedback from receivers to the multicast group as well as support for the synchronization of different media streams. RTP/RTCP is commonly built on the top of UDP and provides some functionality for media transport. But RTP does not guarantee the QoS, address the reservation and negotiate the media format.

RSVP (Resource ReSerVation Protocol) is specially designed for streaming applications [30]. This protocol is suitable for streaming since it permits an application transmitting data over a routed network to request resources at each node and attempt to make a resource reservation for the stream. This feature can be used to ensure the desired quality of service (QoS) with a reliable connection. Another advantage is the scalability. RSVP can scale to very large multicast groups. The disadvantage is for network nodes to support a complicated request mechanism. Also, if routers cannot adequately filter reservations, receivers may experience random packet loss for small reservations. More information about IP networks, including protocols presented above could be found in [41].

V. REGION-OF-INTEREST SCALABLE VIDEO STREAMING

In previous sections we have reviewed the methods of scalable video streaming with different streaming methods and tools. In this section we introduce a related scalable video streaming concept referred to as region-of-interest (ROI) video streaming. Please note that, although earlier ROI techniques such as selective enhancement of MPEG-4 FGS [49] do exist, we use JPEG 2000 [34], which is a scalable image coding format, to illustrate the key points. JPEG 2000 is different from JPEG and other existing scalable video coding schemes that use a non-scalable base layer and which are based on the DCT coding; JPEG2000 is a DWT based scalable coder. The coding scheme employed by JPEG 2000 is often referred to as an embedded coding scheme since the bits that correspond to the various qualities and spatial resolutions can be organized into the bit stream syntax in a manner that allows the progressive reconstruction of images and

arbitrary truncation at any point in the stream. Therefore, JPEG2000 can provide good scalability features including both spatial scalability and SNR scalability.

In order to efficiently access the wavelet coefficients corresponding to a particular spatial region of the image, JPEG2000 introduces the concept of a precinct, which groups code blocks into larger rectangular regions within a resolution level. Each precinct generates one packet. A collection of packets, one from each precinct of each resolution level, comprises the quality layer. The bitstream is organized as a succession of quality layers, which creates a hierarchically structured and embedded bitstream.

Several ROI coding techniques for JPEG 2000 images have been proposed in the past few years. The aim of such methods is to provide a higher quality ROI with lower quality background region. These methods can be classified into two categories: static and dynamic ROI coding. In static ROI coding, the ROI is selected and defined during the encoding procedure. Such methods include the max-shift method [34], a general wavelet coefficient scale up scheme [35], a bitplane-by-bitplane shift method [36], and the partial significant bitplane shift method [37]. The main drawback of these methods is that once the ROI is encoded, it can no longer be changed, which may have limitations for interactive scalable streaming applications that require more flexibility. To overcome these drawbacks, dynamic ROI methods have been developed as described in [38] and [39], which allow for the definition of ROI in an interactive environment by dynamically inserting and rearranging quality layers. With such dynamic ROI methods, we are able to truncate and rearrange the packets of a bitstream to meet rate constraints and variations in the network bandwidth. This is an alternative way to achieve scalable video streaming,

which is quite effective for surveillance applications that require high quality ROI and may operate over bandwidth limited networks.

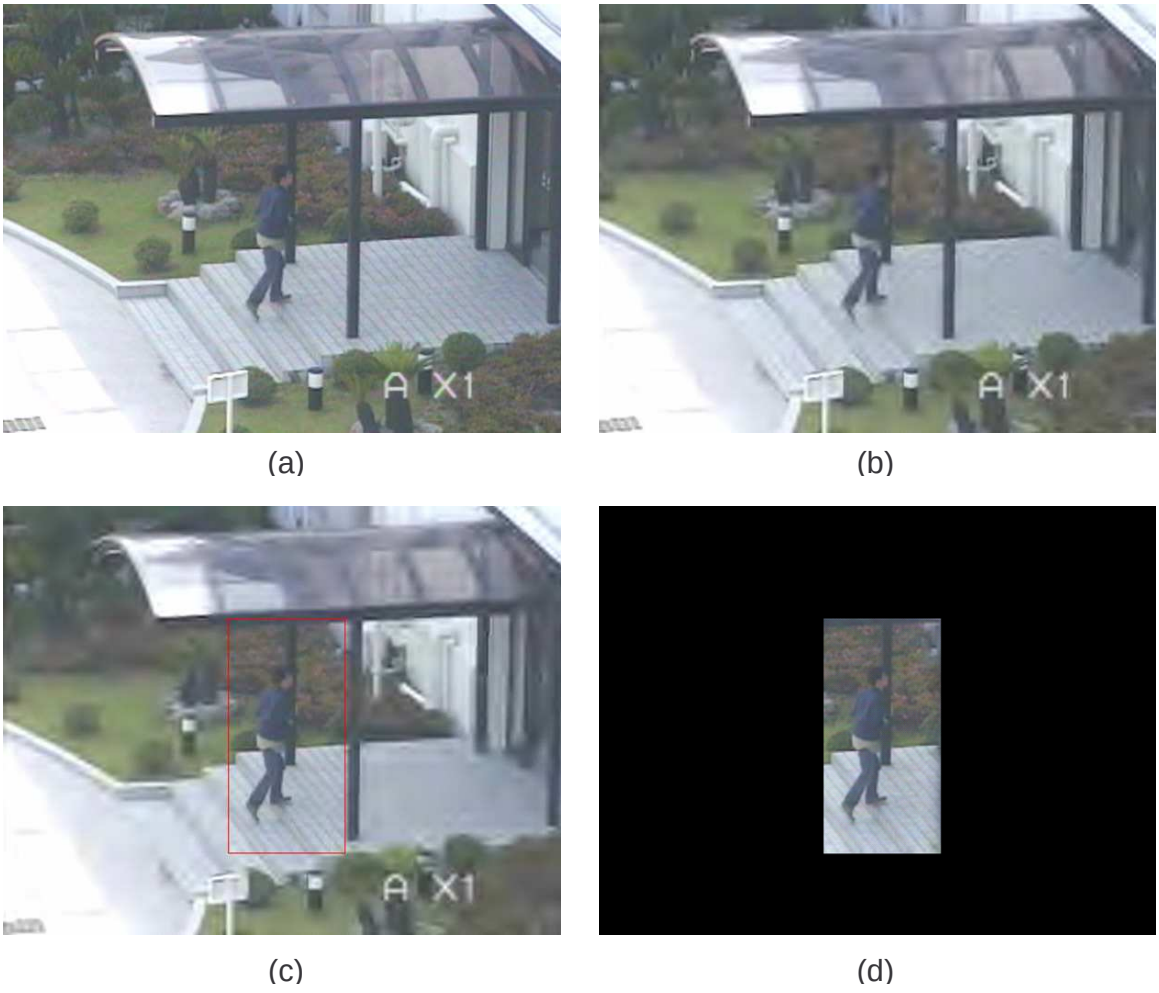

Figure 11. Example JPEG 2000 encodings: (a) original full quality image, (b) all regions coded with same quality, (c) background coded with less quality than ROI, but ROI slightly less than full quality, (c) only ROI at highest quality

Sample JPEG 2000 encodings with ROI examples are shown in Figure 11. In (a), the full image quality is shown, while the images in (b)-(d) have approximately the same reduced rate. The image in (b) allocates equal rate to all regions of the image, while the ROI image in (c) allocates more rate to the ROI and less to the background. Finally, the

image in (d) completely eliminates the background and transmits only the ROI with highest quality. These samples demonstrate that ROI-based streaming is an effective form of scalable streaming compared to uniform scaling of the entire image quality.

VI. CONCLUDING REMARKS

In this article, an overview of scalable video streaming has been presented. The main components of a scalable video streaming system include video server with storage, video encoder, video transcoder or bitstream truncator, and network protocols which enable the transport of video data to end-users. Several technical aspects related to video encoding, streaming methods and network related issues have been discussed. Since video streaming is an extremely broad area, many special topics such as rate control, transmission from multiple servers, peer-to-peer networking, caching strategies, cross-layer design, QoS and DRM issues have not been covered in this article. Nevertheless, we hope this article provides a useful overview for those readers that might not be familiar with this area and gives some useful links to related works.

REFERENCES

[1] T. Kim, “Scalable video streaming over internet”, Ph.D. Thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, Jan. 2005.

[2] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang and J. M. Peha, “Streaming video over the internet: Approaches and directions,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 282–300, Mar. 2001.

[3] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman and Y. Reznik, "Video coding for streaming media delivery on the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 269–281, Mar. 2001.

[4] J.G. Apostolopoulos, W. Tan and S.J. Wee "Video Streaming: Concepts, Algorithms, and Systems" Mobile and Media Systems Laboratory HP Laboratories Palo Alto, HPL-2002-260, Sept. 2002.

[5] X. Sun, F. Wu, S. Li, W. Gao and Y.Q. Zhang, "Seamless Switching of Scalable Video Bitstreams for Efficient Streaming", *IEEE Transaction on Multimedia*, vol. 6, no. 2, pp. 291-303, Apr. 2004.

[6] F. Yang, Q. Zhang, W. Zhu and Y.Q. Zhang, "Bit Allocation for Scalable Video Streaming over Mobile Wireless Internet", *Infocom*, 2004.

[7] F. Ziliani and J-C. Michelou, "Scalable Video Coding in Digital Video Security", White paper, VisioWave, 2005.

[8] Y. Liu, P. Salama, Z. Li and E.J. Delp, "Error Resilient Scalable Video Streaming: Combining Nested Scalability and Parallel Scalability," VIPER Laboratory Report, School of Electrical and Computer Engineering, Purdue University, January 2003. [online at: (accessed on August 25, 2006) http://cobweb.ecn.purdue.edu/~yuxin/resources/Zoe_MDC_Leaky.pdf]

[9] M. Gallant and F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 357–372, Mar. 2001.

[10] W. Tan and A. Zakhor, “Video multicast using layered FEC and scalable compression,” *IEEE Trans Circuits Syst. Video Technol.*, vol. 11, pp. 373–386, Mar. 2001.

[11] W. Li, “Streaming video profile in MPEG-4,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301–317, Mar. 2001

[12] A. Vetro, C. Christopoulos and H. Sun, "An overview of video transcoding architectures and techniques," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18-29, Mar 2003.

[13] J. Xin, C.W. Lin and M.T. Sun, "Digital Video Transcoding," *Proc. IEEE, vol. 93, no. 1, Special Issue on Advances in Video Coding and Delivery*, pp. 84-97, Jan. 2005.

[14] A. Vetro, J. Xin and H. Sun "Error-Resilience Video Transcoding for Wireless Communications", *IEEE Wireless Communications*, vol. 12, no. 4, Aug. 2005.

[15] ISO/IEC JTC1 IS 11172 (MPEG-1), “Coding of moving picture and coding of continuous audio for digital storage media up to 1.5 Mbps,” 1992.

[16] ISO/IEC JTC1 IS 13818 (MPEG-2), “Generic coding of moving pictures and associated audio,” 1994.

[17] ISO/IEC JTC1 IS 14386 (MPEG-4), “Generic Coding of Moving Pictures and Associated Audio, 2000.

[18] ITU-T Recommendation H.261, “Video Codec for Audiovisual Services at px64 Kbit/s,” March 1993.

[19] ITU-T Recommendation H.263, “Video Coding for Low Bit Rate Communication,” Draft H.263, May 2, 1996.

[20] ISO/IEC 14496-10 AVC or ITU-T Rec. H.264, September 2003.

[21] ISO/IEC IS 10918-1:1994 | ITU-T Recommendation T.81 (JPEG), "Digital compression and coding of continuous-tone still images," 1994.

[22] ISO/IEC IS 15444-1:2004 | ITU-T Recommendation T.800, "JPEG 2000 image coding system: Core coding system," 2004.

[23] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301–317, 2001.

[24] A. Vetro and C. Timmerer, "Digital Item Adaptation: Overview of Standardization and Research Activities," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 418-426, June 2005.

[25] ISO/IEC JTC 1/SC 29/WG 11 N7555, Working Draft 4 of ISO/IEC 14496-10:2005/AMD3 Scalable Video Coding, October 2005, Nice France.

[26] B. Girod, N. Farber and U. Horn, "Scalable codec architecture for internet video on demand," *Proc. Asilomar Conf. Signals and Systems*, vol. 1, pp.357-361, Nov. 1997.

[27] N. Farber and B. Girod, "Robust H.263 Compatible video transmission for mobile access to video servers", *Proc. IEEE Int'l Conf. Image Processing*, vol. 2, pp.73-76, Oct. 1997.

[28] X. Sun, F. Wu, S. Li, W. Gao and Y. Q. Zhang, "Seamless Switching of Scalable Video Bitstreams for Efficient Streaming", *IEEE Multimedia*, vol. 6, no. 2, pp. 291-303, April 2004.

[29] M. Karczewisz and R. Kurceren, “The SP- and SI-Frames Design for H.264/AVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 637-644, July 2003.

[30] Online at: http://www.isi.edu/rsvp/overview.html

[31] J. Widmer, R. Denda, and M. Mauve, “A survey on TCP-Friendly congestion control,” *IEEE Network Magazine*, vol. 15, pp. 28-37, May/June 2001.

[32] J.Postel, "RFC 768 - User Datagram Protocol," August 1980.

[33] H. Schulzrinne et al, “RFC 1889 - RTP: A Transport Protocol for Real-Time Applications,” January 1996.

[34] A. Skodras, C. Christopoulos and T. Ebrabimi, “The JPEG2000 Still Image Compression Stadard,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp.36-58, Sept. 2001.

[35] L. Liu and G. Fan, “A new JPEG 2000 region-of-interest image coding method: partial significant bitplanes shift,” *IEEE Signal Processing Letters*, vol. 10, no. 2, pp. 35-38, Feb. 2003.

[36] Z. Wang, and A.C. Bovik, “Bitplane-by-bitplane shift (BbBShift) – A suggestion for JPEG 2000 region of interest coding,” *IEEE Signal Processing Letters*, vol. 9, pp. 160-162, May 2002.

[37] L. Liu, and G. Fan, “A new JPEG 2000 region-of-interest image coding method: partial significant bitplanes shift,” *IEEE Signal Processing Letters*, vol. 10, no. 2, pp. 35-38, Feb. 2003.

[38] R. Rosenbaum and H. Schumann, “Flexible, dynamic and compliant region of interest coding in JPEG 2000,” *Proc. IEEE Int’l Conference on Image Processing,* pp. 101-104, Rochester, New York, Sept. 2002.

[39] H.S. Kong, A. Vetro, T. Hata and N. Kuwahara, “Fast Region-of-Interest Transcoding for JPEG 2000 Images,” *Proc IEEE Int’l Symp. Circuits and Systems,* Kobe, Japan, May 2005.

[40] J.-R. Ohm, “Advances in Scalable Video Coding”, *Proc. IEEE*, vol. 93, no. 1, Special Issue on Advances in Video Coding and Delivery, pp. 42-56, Jan. 2005.

[41] H. Schulzrinne, "IP Networks," in Compressed Video Over Networks, , Amy Reibman and Ming-Ting Sun (eds.) , Marcel Dekker , 2001.

[42] J. Lu, “Reactive and proactive approaches to media streaming: from scalable coding to content delivery networks,” *Proc. Int’l Conf. Information Technology: Coding and Computing*, April 2-4, 2001, Las Vegas, pp. 5-9.

[43] W. Sweldens, “A custom-design construction of biorthogonal wavelets,” *J. Appl. Comp. Harm. Anal*., vol. 3, no. 2, pp. 186-200, 1996.

[44] Y. Wang and Q.F. Zhu, “Error control and concealment for video communications: A reviews,” *Proc. IEEE*, vol.86, no. 5, pp. 974-997, May 1998.

[45] R. Yan, F. Wu, S. Li and R. Tao,"Error Resilience Methods for FGS Video Enhancement Bitstream", *The First IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM 2000)*, Dec. 13-15, 2000 Sydney, Australia.

[46] H. Cai, B. Zeng, G. Shen, and S. Li, “Error-Resilient Unequal Error Protection of Fine Granularity Scalable Video Bitstreams”, accepted by *EURASIP Journal on*

Applied Signal Processing, special issue on Advanced Video Technologies and Applications for H.264/AVC and Beyond.

[47] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 332-344, Mar 2001.

[48] M. Van der Schaar and H. Radha, "Motion-compensation Fine-Granular-Scalability (MC-FGS) for wireless multimedia", *Proceedings of IEEE Symposium on Multimedia Signal Processing (Special Session on Mobile Multimedia Communications)*, October 2001.

[49] M. Van der Schaar and Y.-T. Lin, "Content-basd selective enhancement for streaming video," *Proc. Int'l Conf. Image Processing*, pp. 977-980, Oct. 2001.

LIST OF TABLES

Table 1. Summary of image and video coding standards

Name	Completion Time	Major Features
JPEG[21]	1992	For still image coding, based on Discrete Cosine Transform (DCT)
JPEG-2000[22]	2000	For still image coding, based on Discrete Wavelet Transform (DWT)
H.261	1990	For videoconferencing, 64Kbps-1.92 Mbps
MPEG-1	1991	For CD-ROM, ≤1.5 Mbps
MPEG-2 (H.262)	1994	For DTV/DVD, 2-15 Mbps; for ATSC HDTV, 19.2 Mbps; most extensively used standard
H.263	1995	For very low bit rate coding, below 64Kbps
MPEG-4 Part 2	1999	For multimedia, content-based coding, its simple profile and advanced simple profile are applied to mobile video and streaming
H.264/AVC (MPEG-4 Part 10)	2005	For many applications with significantly improved coding performance over MPEG-2 and MPEG-4 part 2
VC-1	2005	For many applications, coding performance close to H.264
RealVideo[1]	2000	For many applications, coding performance similar to MPEG-4 part 2

Table 2. Comparison between different streaming methods

Streaming Methods	Advantages	Disadvantages
Multiple bitstreams	• High quality • Simple decoder	• Limited number of streams • Large storage
Single scalable bitstream	• Small storage • Multicast application • Simple bitstream switching	• Complicated decoder • Loss of coding efficiency

[1] Not an official standard, but de facto an industry standard.

Single non-scalable bitstreams with transcoding	• Simple decoder • Small storage • Capable of inserting new information for error resilience	• Drift is possible • Higher complexity • Additional delay

Table 3. Summary of different network protocol for video streaming

Network Protocol	Advantage	Disadvantage
TCP	• Dominate protocol for data transfer of data over the Internet • Streaming through firewall • Reliable	• Typically need large buffer to handle data rate variation • Loss recovery needs retransmission causing further jitter or skew • No support for multicast
UDP	• Suitable for streaming • Allows packet drops; if packets arrive late or damaged, streaming will continue • No retransmission needed	• Many network firewalls block UDP data • Need error concealment for video packet loss • No support for congestion control • Cannot be played using popular stream players such as QuickTime
RTP/RTCP	• Support real-time transmission • Provide timing reconstruction, loss detection, security and content identification • Allows retrieval of very interesting network statistics	• No guarantee for QoS • Header is larger than UDP • More complicated that UDP • No support for congestion control
RSVP	• Reliable connection • Receiver can obtain different levels of service	• Complicated request mechanism • Receivers may experience random packet loss for small reservation

LIST OF FIGURES