Image Transmission over Internet

Mireya García[1] and Alejandro Ramírez[2]

[1] Instituto Politécnico Nacional-CITEDI, Av. Del Parque No.1310, Tijuana BC,
mgarciav@citedi.mx
[2] Dpto R&D, PILIMTEC, Châteaugiron,Francia
alramirez10@yahoo.fr

Abstract. Due to the explosive growth of the internet and increasing demand for multimedia information on the web, streaming video over the Internet has received tremendous attention from academia and industry. Video streaming over Internet represents big challenges due to the fact that this network offers generally the best-effort service. This means that this type of service does not provide a guarantee for the bandwidth, delay (jitter) and losses. These characteristics are uncontrollable and dynamic. The purpose of this article is to give a general overview of video over IP and to examine the challenges that make simultaneous delivery and playback, or streaming, of video difficult over packet networks such as the Internet.

Keywords: Streaming, video, IP, compression.

1 Introduction

Video has been an important media for communications and entertainment for many decades. Initially video was captured and transmitted in analog form, but the advent of digital integrated circuits and computers led to the digitization of video [1]. Digital video enabled a revolution in the compression and communication of video. Video compression became an important area of research in the late 1980's and 1990's and enabled a variety of applications including video storage on DVD's and Video-CD's, video broadcast over digital cable, satellite and terrestrial (over-the-air) digital television (DTV), and video conferencing and videophone over circuit switched networks. The growth and popularity of the Internet in the mid 1990's motivated video communication over best-effort packet networks. Video over best-effort packet networks is complicated by a number of factors including unknown and time-varying bandwidth, delay, and losses, as well as many additional issues such as how to fairly share the network resources amongst many flows and how to efficiently perform one-to-many communication for popular content [2]. The purpose of this article is to give a general overview of video over IP and to examine the challenges that make simultaneous delivery and playback, or streaming, of video difficult over packet networks such as the Internet.

A. Gelbukh, S. Suárez, H. Calvo (Eds.) *Received 07/07/07*
Advances in Computer Science and Engineering *Accepted 19/10/07*
Research in Computing Science 29, 2007, pp. 120-127 *Final version 24/10/07*

Each of the following sections provides a brief description of important factors inside the scheme of video on Internet. The section 2 describes a perspective of the type of applications that they can find inside the scheme of video on IP. The section 3 summarizes the functioning of the compression and its importance in video streaming. Section 4 identifies the three fundamental challenges in video streaming. The section 5 presents the principal current problems of the video on IP. The most important standards used in the video streaming are described in section 6. Standardized media streaming are described in section 7. The article concludes with our vision of the video on IP.

2 Framework Aplications

There exist a very diverse range of different video communications and streaming applications, which have very different operating conditions or properties. Examples of this reality: the video communication application may be for point to point communication or for multicast or broadcast communication, and video may be pre-encoded (stored) or may be encoded in real-time (e.g. interactive videophone or video conferencing). The video channels for communication may also be static or dynamic, packet-switched or circuit-switched, may support a constant or variable bit rate transmission, and may support some form of Quality of Service (QoS) or may only provide best effort support. The specific properties of a video communication application strongly influence the design of the system. Therefore, we continue by briefly discussing some of these properties and their effects on video communication system design.

3 Video Sreaming and Compression

A streaming video system is one in which a source encodes video content and transmits the encoded video stream over a data network (wired or wireless) where one or more receivers can access, decode, and display the video to users in real-time. The presence of the network, which allows the source to be physically distant from the receivers, differentiates streaming video from pre-recorded video used in consumer electronic devices such as DVD players.

Given that uncompressed video has very large bandwidth demands, the need for efficient video compression is paramount in this type of applications.

Video compression is achieved by exploiting the similarities or redundancies that exist in a typical video signal [3]. For example, consecutive frames in a video sequence exhibit temporal redundancy since they typically contain the same objects, perhaps undergoing some movement between frames. Within a single frame there is spatial redundancy as the amplitudes of nearby pixels are often correlated. Similarly, the Red, Green, and Blue color components of a given pixel are often correlated. Another goal of video compression is to reduce the irrelevancy in the video signal that is to only code video features that are perceptually important and not to waste valuable bits of information that is not perceptually important or irrelevant.

The compression of still images, based on the standard JPEG [4], consists of exploiting the spatial and color redundancy that exists in a single still image. Neighboring pixels in an image are often highly similar, and natural images often have most of their energies concentrated in the low frequencies. JPEG exploits these features by partitioning an image into 8x8 pixel blocks and computing the 2-D Discrete Cosine Transform (DCT) for each block. The motivation for splitting an image into small blocks is that the pixels within a small block are generally more similar to each other than the pixels within a larger block. The DCT compacts most of the signal energy in the block into only a small fraction of the DCT coefficients, where this small fraction of the coefficients are sufficient to reconstruct an accurate version of the image. Each 8x8 block of DCT coefficients is then quantized and processed using a number of techniques known as zigzag scanning, run length coding, and Huffman coding to produce a compressed bitstream.

A video sequence consists of a sequence of video frames or images. Each frame may be coded as a separate image, for example by independently applying JPEG-like coding to each frame. However, since neighboring video frames are typically very similar much higher compression can be achieved by exploiting the similarity between frames.

Currently, the most effective approach to exploit the similarity between frames is by coding a given frame by (1) first predicting it based on a previously coded frame, and then (2) coding the error in this prediction. Consecutive video frames typically contain the same imagery, however possibly at different spatial locations because of motion. Therefore, to improve the predictability it is important to estimate the motion between the frames and then to form an appropriate prediction that compensates for the motion.

The process of estimating the motion between frames is known as *motion estimation (ME)*, and the process of forming a prediction while compensating for the relative motion between two frames is referred to as *motion-compensated prediction (MC-P)*. Block-based ME and MC-prediction is currently the most popular form of ME and MC-prediction: the current frame to be coded is partitioned into 16x16 pixel blocks, and for each block a prediction is formed by finding the best matching block in the previously coded reference frame. The relative motion for the best matching block is referred to as the *motion vector* [5].

There are three basic common types of coded frames: (1) intra-coded frames, or I-frames, where the frames are coded independently of all other frames, (2) predictively coded, or P-frames, where the frame is coded based on a previously coded frame, and (3) bi-directionally predicted frames, or B frames, where the frame is coded using both previous and future coded frames [5]. Figure 1 illustrates the different coded frames and prediction dependencies for an example MPEG Group of Pictures (GOP). It is important to indicate that the selection of prediction dependencies between frames can have a significant effect on video streaming performance, e.g. in terms of compression efficiency and error resilience [6].

It is of relevancy to indicate that the current compression standards achieve compression by applying the same basic principles previously presented.

Summarizing, the temporary redundancy is exploited by applying MC prediction; the spatial redundancy is exploited by applying the DCT. The resulting DCT coefficients are quantized and the nonzero quantized DCT coefficients are run length and Huffman coded to produce the compressed bitstream.

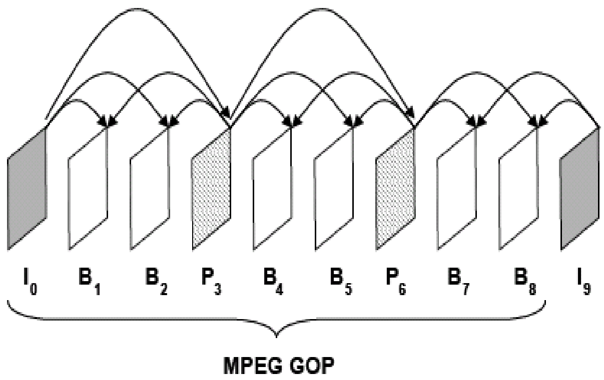

Fig. 1. Example of the prediction dependencies between frames.

At present, the standards of compression that are basically used for communications by video and video streaming are H.263, MPEG-4 and MPEG-4 AVC/H.264 [7].

4 Challenges in Video Streaming

Three fundamental problems exist in video streaming:

4.1 Video Delivery via File Download

Probably the most straightforward approach for video delivery over the Internet is the download, but we refer to it as video download to keep in mind that it is a video and not a generic file. Specifically, video download is similar to a file download, but it is a large file. This approach allows the use of established delivery mechanisms, for example TCP as the transport layer or FTP or HTTP at the higher layers according to OSI model [7].

However, it has a number of disadvantages. Since videos generally correspond to very large files, the download approach usually requires long download times and large storage spaces. These are important practical constraints. In addition, the entire video must be downloaded before viewing can begin. This requires patience on the viewers part and also reduces flexibility in certain circumstances, e.g. if the viewer is unsure of whether he/she wants to view the video, he must still download the entire video before viewing it and making a decision.

4.2 Video Delivery via Streaming

Video delivery by video streaming attempts to overcome the problems associated with file download, and also provides a significant amount of additional capabilities. The basic idea of video streaming is to split the video into parts, transmit these parts successively, and enable the receiver to decode and playback the video as these parts are received, without having to wait for the entire video to be delivered.

Video streaming can conceptually be thought to consist in the following steps:

1. Partition the compressed video into packets
2. Start delivery of these packets
3. Begin decoding and playback at the receiver while the video is still being delivered

Video streaming enables simultaneous delivery and playback of the video. This is in contrast to file download where the entire video must be delivered before playback can begin. In video streaming there is usually a short delay (usually on the order of 1-15 seconds) between the start of delivery and the beginning of playback at the client. This delay, referred to as the pre-roll delay, provides a number of benefits to smoothly balance any network alteration.

Video streaming provides a number of benefits including low delay before viewing starts and low storage requirements since only a small portion of the video is stored at the client at any point in time. The length of the delay is given by the time duration of the pre-roll buffer, and the required storage is approximately given by the amount of data in the pre-roll buffer.

4.3 Expressing Video Streaming as a Sequence of Constraints

Consider the time interval between displayed frames to be denoted by Δ (Δ is 33ms for 30 frames/s video and 100ms for 10 frames/s video).

Each frame must be delivered and decoded by its playback time; therefore the sequence of frames has an associated sequence of deliver/decode/display deadlines:

Frame N must be delivered and decoded by time T_N

Frame N+1 must be delivered and decoded by time $T_N + \Delta$.

Frame N+2 must be delivered and decoded by time $T_N + 2\Delta$.

Any data that is lost in transmission cannot be used at the receiver. Furthermore, any data that arrives too late is also useless. Specifically, any data that arrives after its decoding and display deadline is too late to be displayed.

Note that data may still be useful even if it arrives after its display time, for example if subsequent data depends on this “late” data. Therefore, an important goal of video streaming is to perform the streaming in a manner so that this sequence of constraints is met.

5 Basic Problems in Video Streaming

There are a number of basic problems that afflict video streaming. For example, video streaming over the Internet is difficult because the Internet only offers best effort service. That is, it provides no guarantees on bandwidth, delay jitter, or loss rate. These characteristics are unknown and dynamic. Therefore, a key goal of video streaming is to design a system to reliably deliver high-quality video over the Internet when dealing with unknown and dynamic.

Since it was mentioned in the previous paragraph, the bandwidth available between two points in the Internet is generally unknown and time-varying. If the sender transmits faster than the available bandwidth then congestion occurs, packets are lost, and there is a severe drop in video quality. If the sender transmits slower than the available bandwidth then the receiver produces suboptimal video quality. The goal to overcome the bandwidth problem is to estimate the available bandwidth and than match the transmitted video bit rate to the available bandwidth.

Additional considerations that make the bandwidth problem very challenging include accurately estimating the available bandwidth, matching the preencoded video to the estimated channel bandwidth, transmitting at a rate that is fair to other concurrent flows in the Internet, and solving this problem in a multicast situation where a single sender streams data to multiple receivers where each may have a different available bandwidth.

The end-to-end delay that a packet experiences may fluctuate from packet to packet. This variation in end-to-end delay is referred to as the delay jitter. Delay jitter is a problem because the receiver must receive/decode/display frames at a constant rate, and any late frames resulting from the delay jitter can produce problems in the reconstructed video, e.g. jerks in the video. This problem is typically addressed by including a playout buffer at the receiver. While the playout buffer can compensate for the delay jitter, it also introduces additional delay.

The third fundamental problem is losses. A number of different types of losses may occur, depending on the particular network under consideration.

For example, wired packet networks such as the Internet are afflicted by packet loss, where an entire packet is erased (lost). On the other hand, the wireless channels are typically afflicted by bit erros or burst errors. Losses can have a very destructive effect on the reconstructed video quality.

To combat the effect of losses, a video streaming system is designed with error control. Approaches for error control can be roughly grouped into four classes: (1) forward error correction (FEC), (2) retransmissions, (3) error concealment, and (4) error-resilient video coding [8].

6 Protocols for Video Streaming

The Internet was developed to connect a heterogeneous mix of networks that employ different packet switching technologies. The Internet Protocol (IP) provides baseline best-effort network delivery for all hosts in the network: providing addressing, best-effort routing, and a global format that can be interpreted by everyone.

On top of IP are the end-to-end transport protocols, where Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the most important [7]. TCP provides reliable byte-stream services. It guarantees delivery via retransmissions and acknowledgements. On the other hand, UDP is simply a user interface to IP, and is therefore unreliable and connectionless. Additional services provided by UDP include checksum and port-numbering for demultiplexing traffic sent to the same destination.

Some of the differences between TCP and UDP that affects streaming applications are:

- TCP operates on a byte stream while UDP is packet oriented.
- TCP guarantees delivery via retransmissions, but because of the retransmissions its delay is unbounded. UDP does not guarantee delivery, but for those packets delivered their delay is more predictable (i.e. one-way delay) and smaller.
- TCP provides flow control and congestion control. UDP provides neither. This provides more flexibility for the application to determine the appropriate flow control and congestion control procedures.
- TCP requires a back channel for the acknowledgements. UDP does not require a back channel.

Web and data traffic are delivered with TCP/IP because guaranteed delivery is far more important than delay or delay jitter. For media streaming the uncontrollable delay of TCP is unacceptable and compressed media data is usually transmitted via UDP/IP despite control information is usually transmitted via TCP/IP.
The entity that specifies the protocols for media delivery, control and description over Internet is IETF (Internet Engineering Task Force) [9].

7 Video Streaming Standards

Standard-based media streaming systems, as specified by 3GPP (3rd Generation Partnership Project) for media over 3G cellular and by ISMA (Internet Streaming Media Alliance) for streaming over the Internet, employ the following protocols [10]:

Media encoding
    MPEG-4 video and audio (AMR for 3GPP), H.263.
Media transport
    RTP for data, usually over UDP/IP
    RTCP for control messages, usually over UDP/IP
Media session control
    RTSP
Media description and announcement
    SDP

The streaming standards do not specify the storage format for the compressed media, but the MP4 file format has been widely used [11].

One advantage of MP4 file format is the ability to include “hint tracks” that simplify various aspects of streaming by providing hints such as packetization boundaries, RTP headers and transmission times.

8 Conclusions

Video communication over packet networks has witnessed much progress in the past few years, from download-and-play to various technologies.

This work presented a general overview of video over IP and examined the challenges that make simultaneous delivery and playback, or streaming, of video difficult over packet networks such as the Internet.

Given this integral vision of the video streaming where big challenges exist to resolving, we believe that video streaming will continue to be a compelling area for exploration, development, and deployment in the future.

Acknowledgments. The authors appreciate the support of SIP-IPN, for the support of this research under grant 20070032.

References

1. B. Girod, J. Chakareski, M. Kalman, Y. Liang, E. Setton and R. Zhang. “Advances in Network-Adaptive Video Streaming” Tyrrhenian Inter. Workshop on Digital Communications, Sept 2002.
2. M.-T. Sun and A. Reibman, eds, Compressed Video over Networks, Marcel Dekker, New York, 2001.
3. Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communications, New Jersey, Prentice-Hall, 2002.
4. G. K. Wallace, “The JPEG Still Picture Compression Standard,” Communications of the ACM, April, 1991.
5. V. Bhaskaran and K. Konstantinides, Image and Video Compression standards: Algorithms and Architectures, Boston, Massachusetts, Kluwer Academic Publishers, 1997.
6. D. Wu, Y. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha. “Streaming Video over the Internet: Approaches and Directions”, IEEE Transactions on Circuits and Systems for Video Technology, March 2001.
7. W. Tan and A. Zakhor, “Real-time Internet Video using Error- Resilient Scalable Compression and TCP-friendly Transport Protocol,” IEEE Trans. on Multimedia, June 1999.
8. Y. Wang and Q. Zhu, “Error control and concealment for video communications: A review”, Proceedings of the IEEE, May 1998.
9. http://www.ietf.org/
10. http://www.isma.tv/.
11. http://www.ww.sinobiometrics.com.