Proc. IEEE Intern. Conf. Image Processing
ICIP 2002, Rochester, NY, Sept. 2002.

ADVANCES IN CHANNEL-ADAPTIVE VIDEO STREAMING

Bernd Girod, Mark Kalman, Yi J. Liang, Rui Zhang

Information Systems Laboratory
Department of Electrical Engineering, Stanford University
Stanford, CA 94305

Invited Paper

Abstract

We review recent advances in channel-adaptive video streaming. Adaptive media playout at the client can be used to reduce receiver buffering and therefore average latency, and provide a limited rate scalability. Rate-distortion optimized packet scheduling determines the best packet to send given the distortion reduction associated with sending that packet, interpacket dependencies, and the success of past transmissions. Channel-adaptive packet dependency control can greatly improve the error-robustness of streaming video and reduce or eliminate the need for packet retransmissions. Finally we consider three architectures for wireless video streaming and discuss the utility of the discussed techniques for each architecture.

1. INTRODUCTION

Since the introduction of the first commercial products in 1995, Internet video streaming has experienced phenomenal growth. Over a million hours of streaming media contents are being produced every month and served from hundreds of thousands of streaming media servers. Second only to the number-one Web browser, the leading streaming media player has more than 250 million registered users, with more than 200,000 new installations every day. This is happening despite the notorious difficulties of transmitting data packets with a deadline over the Internet, due to variability in throughput, delay and loss. It is not surprising that these challenges, in conjunction with the commercial promise of the technology, has attracted considerable research efforts, particularly directed towards efficient, robust and scalable video coding and transmission [1] [2].

A streaming video systems has four major components: 1. The encoder application (often called the “producer” in commercial systems) that compresses video and audio signals and uploads them to the media server. 2. The media server that stores the compressed media streams and transmits them on demand, often serving hundreds of streams simultaneously. 3. The transport mechanism that delivers media packets from the server to the client for the best possible user experience, while sharing network resources fairly with other users. 4. The client application that decompresses and renders the video and audio packets and implements the interactive user controls. For the best end-to-end performance, these components have to be designed and optimized in concert.

The streaming video client typically employs error detection and concealment techniques to mitigate the effects of lost packets [3]. Unless forced by firewalls, streaming media systems do not rely on TCP for media transport but implement their own application level transport mechanisms to provide the best end-to-end delivery while adapting to the changing network conditions. Common issues include retransmission and buffering of packets [4], generating parity check packets [5], TCP-friendly rate control [6], and receiver-driven adaptation for multicasting [7]. New network architectures, such as DiffServ [8] and the path diversity transmission in packet networks [9], also fall into this category. The media server can help implementing intelligent transport mechanisms, by sending out the right packets at the right time, but the amount of computation that it can perform for each media stream is very limited due to the large number of streams to be served simultaneously. Most of the burden for efficient and robust transmission is therefore on the encoder application that, however, faces the added complication that it cannot adapt to the varying channel conditions but rather has to rely on the media server for this task. Representations that allow easy rate scalability are very important to adapt to varying network throughput without requiring computation at the media server. Multiple redundant representations are an easy way to achieve this task, and they are widely used in commercial systems [4]. To dynamically assemble compressed bit-streams without drift problems, S-frames [10] and, recently, SP-frames [11] have been proposed. Embedded scalable video representations such as FGS [12] would be more elegant for rate adaptation, but they are still considerably less efficient, particularly at low bit-rates. Embedded scalable representations are a special case of multiple description coding of video that can be combined advantageously with packet path diversity [9] [13]. Finally, the source coder can trade-off some compression efficiency for higher error resilience [14]. For live encoding of streaming video, feedback information can be employed to adapt error resiliency, yielding the notion of channel-adaptive source coding. Such schemes have been shown to possess superior performance [15]. For precompressed video stored on a media server, these channel-adaptive source coding techniques can be effected through assembling sequences of appropriately precomputed packets on the fly.

In our opinion, the most interesting recent advances in video streaming technology are those that consider several system component jointly and react to the packet loss and delay, thus performing channel-adaptive streaming. In this paper, we review some recent advances in channel-adaptive streaming. As an example of a new receiver-based technique, we discuss adaptive media playout in Section 2, to reduce delay introduced by the client buffer and provide rate scalability in a small range. We then review rate-distortion optimized packet scheduling as the most important recent advance in transport mechanisms in Section 3. An example of a channel-adaptive encoder-server technique we discuss is the new idea of packet dependency control to achieve very low latency in Section 4. All of these techniques are applicable for wireline as well as wireless network. Architectures and the specific challenges arising for wireless video streaming are discussed in the concluding Section 5.

2. ADAPTIVE MEDIA PLAYOUT

Adaptive media playout (AMP) is a new technique that allows a streaming media client, without the involvement of the server, to control the rate at which data is consumed by the playout process. For video, the client simply adjusts the duration that each frame is shown. For audio, the client performs signal processing in conjunction with time scaling to preserve the pitch of the signal. Informal subjective tests have shown that slowing the playout rate of video and audio up to 25% is often un-noticeable, and that time-scale modification is preferable subjectively to halting playout or errors due to missing data [16] [17].

One application of AMP is the reduction of latency for streaming media systems that rely on buffering at the client to protect against the random packet losses and delays. Most noticeable to the user is the pre-roll delay, i.e., the time it takes for the buffer to fill with data and for playout to begin after the user makes a request. However, in streaming of live events or in two-way communication, latency is noticeable throughout the entire session. With AMP, latencies can be reduced for a given level of protection against channel impairments. For instance, pre-roll delays can be reduced by allowing playout to begin with fewer frames of media stored in the buffer. Using slowed playout to reduce the initial consumption rate, the amount of data in the buffer can be grown until sufficient packets are buffered and playout can continue normally. For two-way communication or for live streams, AMP can be used to allow smaller mean buffering delays for a given level of protection against channel impairments. The application was explored for the case of two-way voice communication in [16]. It is easily extended to streaming video. In [18] it is shown that this simple playout control policy can result in latency reductions of 30% for a given level of protection against underflow.

AMP can also be used for outright rate-scalability in a limited range, allowing clients to access streams which are encoded at a higher source rate than their connections would ordinarily allow [18].

3. R-D OPTIMIZED PACKET SCHEDULING

The second advance that we are reviewing in this paper is a transport technique. Because playout buffers are finite, and because there are constraints on allowable instantaneous transmission rates, retransmission attempts for lost packets divert transmission opportunities from subsequent packets and reduce the amount of time that subsequent packets have to successfully cross the channel. A streaming media system must make decisions, therefore, that govern how it will allocate transmission resources among packets.

Recent work of Chou et al. provides a flexible framework to allow the rate-distortion optimized control of packet transmission [19] [20]. The system can allocate time and bandwidth resources among packets in a way that minimizes a Lagrangian cost function of rate and distortion. For example, consider a scenario in which uniformly sized frames of media are placed in individual packets, and one packet is transmitted per discrete transmission interval. A rate-distortion optimized streaming system decides which packet to transmit at each opportunity based on the packets' deadlines, their transmission histories, the channel statistics, feedback information, the packets' interdependencies, and the reduction in distortion yielded by each packet if it is successfully received and decoded.

The framework put forth in [19] is flexible. Using the framework, optimized packet schedules can be computed at the sender or receiver. The authors have also presented simplified methods to compute approximately optimized policies that require low computational complexity. Furthermore, the framework, as shown in [20], appears to be robust against simplifications to the algorithm and approximations of information characterizing the value of individual packets with respect to reconstruction distortion. Low complexity is important for server-based implementation, while robustness is important for receiver-based implementations, where the receiver makes decisions. We have recently extended Chou's framework for adaptive media playout, such that each packet is optimally scheduled, along with a recommended individual playout deadline. For that, the distortion measure is extended by a term that penalizes time-scale modification and delay [18].

4. CHANNEL-ADAPTIVE PACKET DEPENDENCY CONTROL

While for voice transmission over the Internet latencies below 100 ms are achievable, video streaming typically exhibits much higher latencies, even if advanced techniques like adaptive media playout and R-D optimized packet scheduling are used. This is the result of dependency among packets due to interframe prediction. If a packet containing, say, one frame is lost, the decoding of all subsequent frames depending on the lost frame will be affected. Hence, in commercial systems, time for several retransmission attempts is provided to essentially guarantee the error-free reception of each frame, at the cost of higher latency.

Packet dependency control has been recognized as a powerful tool to increase error-robustness. Earlier work on this topic includes long-term memory prediction for macroblocks for increased error-resilience [21], the reference picture selection (RPS) mode in H.263+ [22] and the emerging H.26L standard [23], and the video redundancy coding (VRC) technique [24]. Those encoding schemes can be applied over multiple transmission channels for path diversity to increase the error-resilience [9] [25], similar to what has been demonstrated for real-time voice communication [26].

In our recent work [27], in order to increase error-resilience and eliminate the need for retransmission, multiple representations of certain frames are pre-stored at the streaming server such that a representation can be chosen that only uses previous frames as reference that may be received with very high probability. We consider the dependency across packets and dynamically control this dependency in adapting to the varying channel conditions. With increased error-resilience, the need for retransmission is eliminated. Buffering is needed only to absorb the packet delay jitter, so that the buffering time can be reduced to a few hundred milliseconds. Due to the trade-off between error-resilience and coding efficiency, we apply *optimal picture type selection (OPTS)* within a rate-distortion (RD) framework, considering video content, channel loss probability and channel feedback (e.g. ACK, NACK, or time-out). This applies to both pre-encoding the video offline and assembling the bitstreams during streaming. In coding each frame, several trials are made, including using the I-frame as well as Inter-coded frames using different reference frames in the long-term memory. The associated rate and expected distortion are obtained to calculate the cost for a particular trial through a Lagrangian formulation. The distortions are obtained through an accurate binary tree modeling considering channel loss rate and error propagation. The optimal picture type is selected such that the minimal RD cost is achieved. Even without retransmission, good quality is still maintained for typical video sequences sent over lossy channels [27]. Thus the excellent robustness achievable through packet-dependency control can be used to reduce or even entirely eliminate retransmission, leading to latencies similar to those for Internet voice transmission.

5. CHALLENGES OF WIRELESS VIDEO STREAMING

In our previous discussion, we have not differentiated between video streaming for the wireline and the wireless Internet. Increasingly, the Internet is accessed from wireless, often mobile terminals, either through wireless LAN, such as IEEE 802.11, or 2.5G or 3G cellular networks. It is expected that in 2004, the number of mobile Internet terminal will exceed the number of fixed terminals for the first time. Wireless video streaming suffers from the same fundamental challenges due to congestion and the resulting best-effort service. Packets still experience variable delay, loss, and throughput, and channel-adaptive techniques as discussed above are important to mitigate these problems.

The mobile radio channel, however, introduces specific additional constraints, and many of the resulting challenges still hold interesting research problems. Fading and shadowing in the mobile radio channel leads to additional packet losses, and hence TCP-style flow control often results in very poor channel utilization. Frame sizes of wireless data services are usually much smaller than the large IP packets preferable for video streaming, hence fragmentation is necessary. Since the loss of any one fragment knocks out an entire IP packet, this effectively amplifies the loss rate of the wireless link. An obvious remedy is to use ARQ for the radio link, trading off throughput and delay for reliability of the wireless link. Most, but not all mobile data services operate in this way.

Other objections against using IP for streaming over mobile radio links is the RTP/UDP/IP encapsulation overhead that can use up a significant portion of the throughput of the expensive wireless link. Moreover, mobility management in IP is lacking, and mobile IP protocols that employ further encapsulation might be even more wasteful. Header compression, however, can very efficiently overcome this problem and will be widely deployed in future radio systems.

Three alternative architectures for wireless video streaming are shown in Fig. 1. The end-to-end architecture in Fig. 1. (a) preserves the Internet paradigm of stateless routing with connection-oriented services implemented in the terminals. Channel-adaptive streaming methods, as discussed above, would be implemented in the client and the server only. We need to distinguish systems with ARQ on the radio link and lossy system. In order to solve the problem of sharing bandwidth fairly both in the wireline and the lossy wireless links, reliable loss differentiation algorithms (LDA) are required that can distinguish loss due to congestion and a deteriorating wireless channel. Some promising research is underway, but the proposed techniques are still limited [28]. ARQ in the radio link can avoid wireless losses altogether, but reduce throughputs and increases delay. For streaming applications where delay is not critical, radio link ARQ is superior.

Fig. 1 (b) shows an architecture with a proxy server separating the wireless and wireline portion of the network. Instead of connecting to the back-end streaming media server directly, the client connects to the proxy server, which in turn connects to the streaming media server. The proxy is responsible for pre-fetching and buffering packets, such that they are available when the mobile client needs them. Channel-adaptive streaming techniques can now be applied to each of the two connections separately. The proxy server might also implement simple transcoding to reduce the bit-rate or increase error resilience for low-delay applications.

Fig. 1 (c) shows an architecture where a gateway between the wireline and wireless part of the network marks the territory of the Internet. For the wireless link, an integrated wireless media protocol, tailored to the needs of wireless audio and video transmission, is used. This integrated wireless media protocol could even be a circuit-switched multimedia protocol stack, such as H.324M [29]. Channel-adaptive streaming techniques would be used between the gateway and the streaming media server, while packet-oriented streaming media techniques, such as dynamic packet scheduling, might not be applicable to the wireless link. With H.324M, error-resilience of the video stream is important, as is rate scalability or rate control to accommodate variable effective throughput even on a nominally fixed-rate link. The 3G-PP consortium has evolved the ITU-T recommendation H.324M into 3G-324M, which also supports MPEG-4 video, in addition to H.263v2, for conversational services.

The streaming architecture in Fig. 1 (c) is actually being implemented by some companies, but it appears to be a short-term solution. The segregation of the world into wireline and wireless terminals is a far too serious drawback. Establishing and tearing down a circuit for each video stream is cumbersome and wasteful, particularly considering that a packet-switched always-on connection will soon be widely available in 2.5G and 3G systems. The open architecture of IP-solutions as shown in Fig. 1 (a) and (b) will undoubtedly prevail.

6. REFERENCES

[1] M. R. Civanlar, A. Luthra, S. Wenger, and W. Zhu (eds.), Special Issue on Streaming Video, IEEE Trans. CSVT, vol. 11, no. 3, Mar. 2001.

[2] C. W. Chen, P. Cosman, N. Kingsbury, J. Liang, and J. W. Modestino (eds.), *Special Issue on Error Resilient Image and Video Transmission, IEEE Journal on Selected Area in Communications*, vol. 18, no. 6, June 2001.

[3] Y. Wang, and Q. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86:5, p. 974-97, May 1998.

[4] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman, and Y. A. Reznik, "Video coding for streaming media delivery on the Internet," *IEEE Trans. CSVT*, vol. 11, no. 3, pp. 269-81, Mar. 2001.

[5] W. Tan, and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. CSVT*, vol. 11, no. 3, pp. 373-87, Mar. 2001.

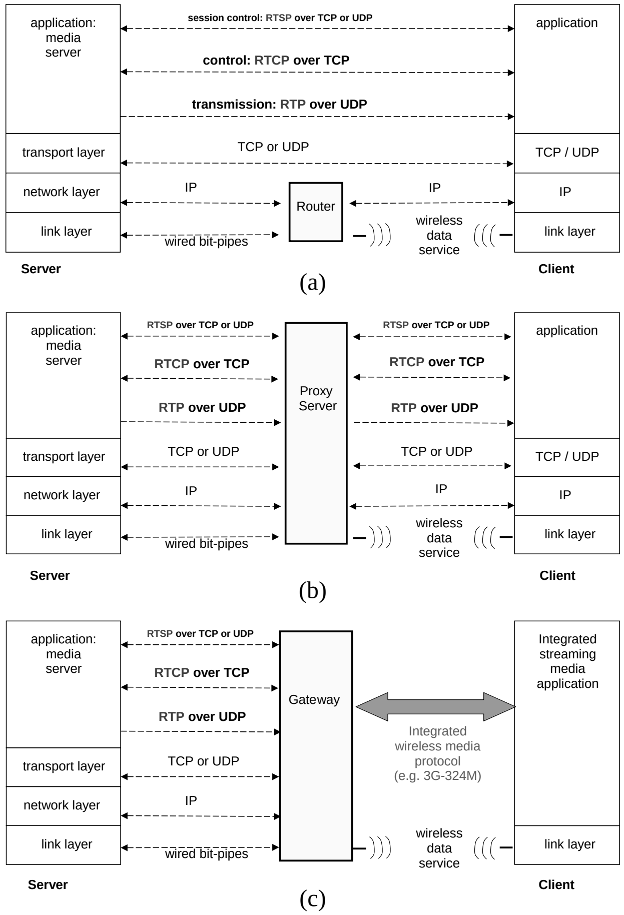

Figure 1: Wireless streaming architectures: (a) end-to-end, (b) proxy server, (c) gateway with integrated media protocol.

[6] W. Tan, and A. Zakhor, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 172-86, June 1999.

[7] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 983-1001, Aug. 1997.

[8] J. Shin, J. Kim, and C.-C. J. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Transactions on Multimedia*, vol.3, no.2, pp.219-31, June 2001.

[9] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," *IEEE Infocom*, July 2002.

[10] N. Färber, and B. Girod, "Robust H.263 Compatible Video Transmission for Mobile Access to Video Servers," *Proceedings ICIP 97*, vol. 2, pp. 73-76, Santa Barbara, Oct. 1997.

[11] M. Karczewicz, and R. Kurceren, "A proposal for SP-frames," Proposal to H.26L, Jan. 01.

[12] M. van der Schaar, and H. Radha, "A hybrid temporal-SNR fine-granular scalability for Internet video," *IEEE Trans. CSVT*, vol. 11, no. 3, pp. 318-31, Mar. 2001.

[13] Y. Wang, M. Orchard, V. Vaishampayan, and A. R. Reibman, "Multiple description coding using pairwise correlating transforms," to appear in *IEEE Trans. Image Processing*.

[14] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966-76, June 2000.

[15] B. Girod, and N. Färber, "Wireless video," in A. Reibman, M.-T. Sun (eds.), Compressed Video over Networks, Marcel Dekker, 2000.

[16] Y. J. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communication," *Proc. ICASSP '01*, Salt Lake City, May 2001.

[17] E. G. Steinbach, N. Färber, and B. Girod, "Adaptive playout for low Latency video streaming," *Proc. International Conference on Image Processing (ICIP-01)*, Thessaloniki, Greece, Oct. 2001.

[18] M. Kalman, E. Steinbach, and B. Girod, "R-D Optimized Media Streaming Enhanced With Adaptive Media Playout," in *Proc. Int'l Conf. Multimedia and Exhibition, Lausanne*, Switzerland, Aug. 2002.

[19] P. A. Chou and Z Miao, "Rate-distortion optimized streaming of packetized media," IEEE Transactions on Multimedia, February 2001. Submitted. http://research.microsoft.com/ pachou

[20] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," Packet Video Workshop, Pittsburg, PA, April 2002.

[21] T. Wiegand, N. Färber, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1050–1062, June 2000.

[22] ITU-T Recommendation H.263 Version 2 (H.263+), *Video coding for low bitrate communication*, Jan. 1998.

[23] ITU-T Video Coding Expert Group, *H.26L Test Model Long Term Number 8*, July 2001, online available at ftp://standard.pictel.com/video-site/h26L/tml8.doc.

[24] S. Wenger, G. D. Knorr, J. Ott, and F. Kossentini, "Error resilience support in h.263+,"*IEEE Journal on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 867–877, Nov. 1998.

[25] S. Lin, S. Mao, Y. Wang, and S. Panwar, "A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths,"*Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, Aug. 2001.

[26] Y. J. Liang, E. G. Steinbach, and B. Girod, "Real-time voice communication over the Internet using packet path diversity," *Proceedings ACM Multimedia 2001*, Oct. 2001, pp. 431–440, Ottawa, Canada.

[27] Y. J. Liang and B. Girod, "Rate-distortion optimized low-latency video streaming using channel-adaptive bitstream assembly," *accepted by IEEE International Conference on Multimedia and Expo*, Aug. 2002.

[28] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *SPIE Multimedia Computing and Networking (MMCN2002)*, San Jose, CA, Jan 18-25, 2002.

[29] N. Färber, B. Girod, and J. Villasenor, "Extensions of the ITU-T Recommendation H.324 for error resilient video transmission," IEEE Communications Magazine, vol. 36, no. 6, pp. 120-128, June 1998.