INTEGRATED END-TO-END BUFFER MANAGEMENT AND CONGESTION CONTROL FOR SCALABLE VIDEO COMMUNICATIONS

Ivan V. Bajić, Omesh Tickoo, Anand Balan, Shivkumar Kalyanaraman, and John W. Woods

ECSE Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

ABSTRACT

In this paper we present a video communication system that integrates end-to-end buffer management and congestion control at the source with the playout adjustment mechanism at the receiver. While each component of the system has been considered independently in the literature, our focus in this work is their integration. The proposed system exploits the fact that when congestion control is implemented at the source, most of the loss occurs at the source and not within the network. Based on this observation, we design the buffer management to trade off random loss for controlled loss of visually less important data. Frame rate is adjusted at the receiver to maximize the visual quality of the displayed video based on the overall loss. We tested our system with both H.26L and a subband/wavelet video coder, and found that it significantly improves the received video quality in both cases.

1. INTRODUCTION

Internet video communications have attracted a lot of research interest in recent years because of the many challenges it poses on the communication system design. Transmission of video typically requires high bandwidth and low delay, while it can tolerate a certain amount of data loss. These requirements are fundamentally mismatched with network protocols such as TCP, that enable lossless data delivery with potentially high delay due to retransmissions. Further, most video coders produce data of varying importance, while networks such as the Internet treat all data equally. To correct this mismatch, several integrated video coding and congestion control approaches have been proposed to simultaneously provide reliable transmission of video and fairness to the competing flows. Some aspects of the interaction between layered video coders and different transport schemes have been studied in [1].

When faced with congestion, transmission rate of the video source needs to be reduced. Many proposed schemes for video transmission rate adaptation implicitly or explicitly make use of SNR scalability (e.g. [2]) which favors the reception of lower quality (SNR) video under unfavorable network conditions. While these approaches may posses certain optimality in a rate-distortion sense, they need not produce the best looking video. For example, a recent study of subjective video quality [3] found that in most cases higher quality low frame-rate video is preferable to the lower quality full frame-rate video. In this work we exploit frame-rate scalability for transmission rate adaptation. In our scheme, as the network conditions deteriorate, receiver is more likely to obtain high quality low frame-rate video.

Contrary to the common belief, we observe that *in a transmission scheme that performs congestion control, most of the packets are dropped at the transmission buffer, while the relative loss inside the network is very low*, as demonstrated in section 2.2. Typically, the packets at the transmission buffer are dropped at random by some congestion avoidance mechanism, which makes the loss at the receiver appear to be random. However, by employing intelligent transmission buffer management, random loss can be traded off for controlled loss which may significantly improve the quality of the received video. We propose a simple buffer management scheme implemented at the transmission source which drops low priority[1] packets in response to congestion. The remaining loss in the network may be handled by other means, such as error concealment (which is the approach we take in this paper) or FEC.

The important contributions of this work are:

- Exploiting frame-rate scalability for adaptation to varying network conditions.
- A simple *generic* end-to-end buffer management scheme that acts as an extension to source coding of video, provides fast adaptation to changing network conditions and converts the random loss a flow suffers to a controlled loss of low priority packets.
- Integration of transmission buffer management and receiver side frame rate adjustment to produce high quality video at the receiver.

This work was supported in part by the ARO grant number DAAD 19-00-1-0559.

[1] In our stystem, priority is related to frame rate.

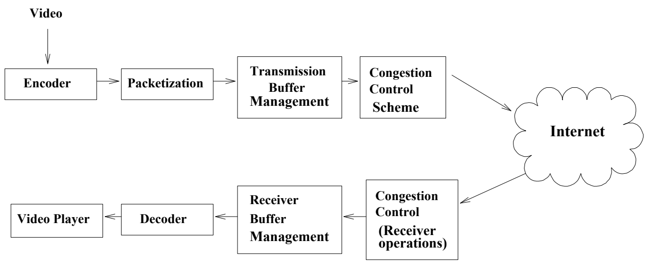


Fig. 1. Video communication system block diagram.

- An integrated video communication system design that produces high quality, low frame-rate video in response to congestion.

The rest of the paper is organized as follows. In Section 2 we describe the components of our video communication system. In Section 3 we report simulation results. Conclusions are given in Section 4.

2. VIDEO COMMUNICATION SYSTEM

Figure 1 shows the block diagram of the video communication system. Video is encoded and packetized into individually decodable packets to prevent the propagation of errors caused by the packet loss. On the *large time scale*, at the Group-Of-Pictures (GOP) level, encoder adapts its encoding rate to the current estimate of the available network rate. On the *smaller time scales* (when the GOP is already encoded, but not yet transmitted), the actual transmission rate is regulated by the transmission buffer. The buffer gets feedback from the congestion control scheme about the current network conditions and sends the most important packets within the available bandwidth. A congestion control scheme serves to minimize burst losses in the network, ensure network stability, and is fair to other flows. At the receiver side, a playout buffer smooths the flow and reduces jitter. Also, frame rate is adjusted appropriately to improve the quality of the displayed video. Individual components of the system are described in the remainder of this section.

2.1. Video coding

The generic video communication system presented in this text can utilize any video coding algorithm which produces data of varying importance i.e. different scalability layers. In our experiments we emphasize frame-rate scalability. Results are reported for the recent H.26L video coder [4] and a robust scalable subband/wavelet video coder from [5]. As the results indicate, in both cases buffer management was found to significantly improve the video quality at the receiver, both visually and in terms of the PSNR.

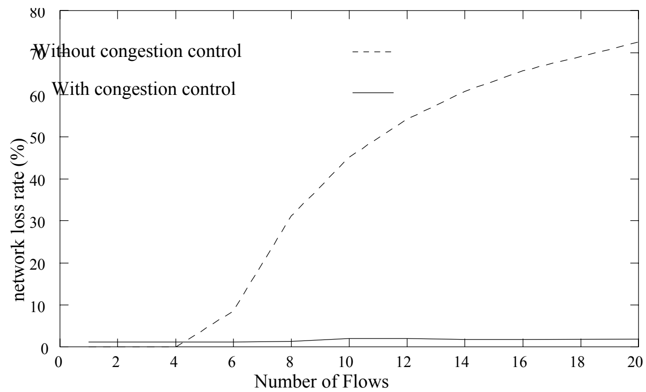


Fig. 2. Comparison of average network loss with and without congestion control.

Common to both coders is that they produce individually decodable packets organized in layers according to their importance (i.e. corresponding frame rate). The importance of the packets in flagged to the transmission buffer management which uses this information to perform congestion control. Our goal here was not to compare the two coders, but to illustrate that the proposed buffer management improves the transmission performance for both of them.

2.2. Source buffer management

Algorithm 1 : source buffer management

for Layer k Packet Arrival **do**
  calculate the queue size $s(t)$
  if $s(t) > q_{max}$ **then**
    Drop the packet
  else
    if $s(t) > T_s$ **then**
      Drop the packet with probability $1 - p_k$
    else
      Enqueue the packet
    end if
  end if
end for

From extensive simulations we infer that for multimedia transmission into a TCP-based network, most loss occurs at the point of transmission i.e. the source, and not at the nodes inside the network. This is *contrary to the belief* that the packet loss in the network due to congestion is the major contributor to the total loss a TCP flow suffers. Our simulations show that in response to congestion the transmission queues at the sources increase which finally leads to packet drops at the source and it is this dropping at the source that is *the major contributor* to the aggregate loss of the flow.

Algorithm 2 : transmission probability assignment

for $(k = K_{max};\ k > 0$ and $\sum_{j=1}^{K_{max}} p_j r_j > R_n(t);\ k - -)$ **do**
  if $\sum_{j=1}^{k-1} p_j r_j > R_n(t)$ **then**
    $p_k = 0$
  else
    $p_k = (R_n(t) - \sum_{j=1}^{k-1} p_j r_j)/r_k$
  end if
end for

As an example, Figure 2 shows the average loss within the network for $N = 1, 2, ..., 20$ flows through a single bottleneck of 5 Mbps bandwidth. With congestion control (TCP) the loss within the network remains fairly low (below 3%) as the number of flows increases, since most of the packets which violate the available rate constraint are dropped at the source. On the other hand, without congestion control (UDP) the loss within the network increases significantly with the number of flows. This provides a *strong incentive to use a suitable end-to-end congestion control for video transmission.*

The operation of the source buffer is regulated by Algorithms 1 and 2, where $s(t)$ is the instantaneous source buffer size, $q_{\max}$ is the maximal allowed buffer size, T_s is the source buffer threshold indicating the buffer size at which the drop policy starts being enforced, R_a is the encoding rate, $R_n(t)$ is the current network rate, $K_{\max}$ is the number of layers in the video bitstream, $k = 1, 2, ..., K_{\max}$, is the layer index (with $K_{\max}$ being the least important layer), r_k is the encoding rate for layer k, and p_k is the transmission probability for the packet from layer k. Algorithm 1 describes the selective drop policy enforced when the threshold T_s is exceeded. Packet transmission probabilities are calculated in a greedy manner in Algorithm 2. Details of the algorithms are described in a longer version of this paper [6].

The choice of the source buffer threshold T_s is important for the overall system performance. Having a small threshold will lead to unnecessary packet drops at the source buffer, while having a large threshold will increase the overall delay and eventually cause the receiver buffer underflow. It can be shown [6] that the near-optimal value for the source buffer threshold is $T_s = B - D$, where D is the GOP size in packets and B is the receiver pre-buffer size.

2.3. Congestion control

Our congestion control mechanism is based on binomial algorithms coupled with randomized pacing of packet transmission times [7]. As such, it provides smoothly varying transmission rate suitable for video flows, and helps reduce jitter effects. In [6] we tested video transmission using the randomized versions of IIAD (Inverse Increase and Additive Decrease) and AIMD (Additive Increase and Multiplicative Decrease) schemes, and showed the superiority of IIAD over AIMD schemes in terms of rate variation and the corresponding transmission buffer size. Hence, IIAD is selected for the final system design.

The randomization of packet transmission times was first introduced in [7]. The randomization is shown to reduce bias against flows with higher RTTs, window synchronization, phase effects in flows and correlated losses. The randomization does not send back-to-back packets but spaces successive transmissions with a time interval $\Delta = RTT(1+x)/w_t$, where x is a zero mean random number drawn from a uniform distribution, and w_t is the current window size.

2.4. Video decoding and playout

The overall loss in the video communication system consists of the loss at the source, the loss inside the network, and the loss at the receiver (due to receiver buffer underflow). The source buffer management has been designed to minimize the effects of the loss at the source (by dropping least important data first) and to prevent receiver buffer underflow. The remaining loss, i.e. the loss inside the network, is handled in our case by error concealment, whose task is to improve the reconstructed video quality using the available data. In the case of subband/wavelet video coder, median filtering is employed to recover missing pieces of data. Missing subband samples are estimated as the median of the available neighboring samples, while missing MVs are estimated as a vector median of the available neighboring MVs. Error concealment operations performed by H.26L video decoder are specified in [4].

Due to the source buffer management policy, the loss at the receiver is concentrated in the higher enhancement layers, i.e. those corresponding to higher frame rates. If this loss is high, it may be advantageous to reduce the frame rate of the displayed video, since the lower frame-rate version is received with lower loss and hence higher quality. We propose a simple rule for adjusting the frame rate of the displayed video. In particular, the frame rate of the displayed video is reduced (i.e. less important layers are not decoded/displayed) until the loss in any remaining layer is less than a certain threshold. In the experiments reported in Section 3 this threshold was set to 20%. This value was obtained empirically by visual examination of the two test video sequences (Football and Flower garden) with varying degrees of loss.

3. SIMULATION RESULTS

Our simulations of video transmission were carried out using the *ns*-2 network simulator. We tested the system with

Sequence	Bandwidth	without BMFA	with BMFA
Football	7.5 Mbps	22.1 dB	28.4 dB
Football	6 Mbps	18.4 dB	24.8 dB
F. garden	7.5 Mbps	12.7 dB	29.3 dB
F. garden	6 Mbps	10.2 dB	28.9 dB

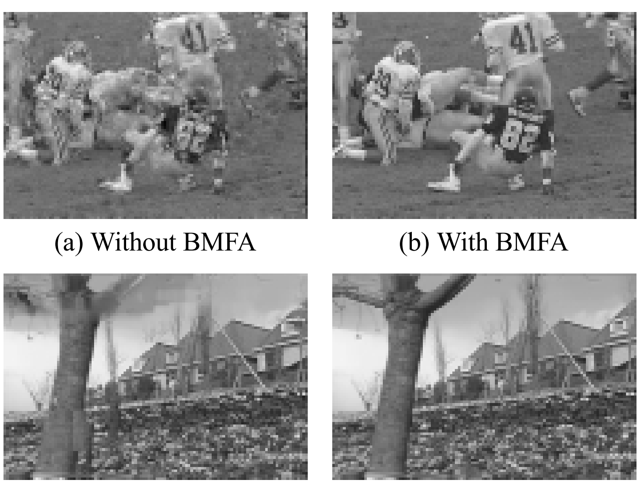
Table 1. Average PSNR comparison for single bottleneck simulations.

Sequence	without BMFA	with BMFA
Football	18.5 dB	27.9 dB
F. garden	17.1 dB	32.0 dB

Table 2. Average PSNR comparison for multiple bottleneck simulations.

two network topologies - single bottleneck and multiple bottleneck topology from [2]. Two video sequences were used in the simulations: Football sequence (encoded with the subband/wavelet coder) and Flower garden (encoded with H.26L). Both were grayscale, SIF resolution, at 30 fps.

In the single bottleneck case, 5 senders simultaneously transmit the video sequence encoded at 1.7 Mbps through the bottleneck whose bandwidth is set at a value less than the total required bandwidth. Average PSNR results for two values of bottleneck bandwidth are reported in Table 1 for the case without BMFA (Buffer Management and Frame rate Adjustment) and with BMFA. Significant PSNR gains of over 6 dB are obtained by BMFA. Visual improvement is illustrated in Figure 3. Results for the multiple bottleneck case are qualitatively the same and are reported in Table 2. Sample video clips may be found at [8].



(a) Without BMFA (b) With BMFA

(c) Without BMFA (d) With BMFA

Fig. 3. Illustration of visual quality improvement brought by buffer management and frame-rate adaptation.

4. CONCLUSIONS

An integrated video communication system that controls the packet drops at the source, along with an intelligent choice of congestion control was proposed to solve the problems faced by a video flow in a congested network. It was shown that most of the packet drops occur at the source buffer when congestion control is employed. This allows us to design a suitable end-to-end buffer management scheme for the video flow. The transmission buffer management works together with receiver side frame rate adjustment mechanism to provide high quality low frame-rate video in response to congestion. The proposed buffer management and frame rate adjustment scheme was tested with two different video coders and different network topologies, and has shown significant improvements in objective and subjective video quality in all cases.

5. REFERENCES

[1] N. Feamster, D. Bansal, and H. Balakrishnan, ”On the interactions between layered quality adaptation and congestion control for streaming video,” *Proc. 11th Packet Video Workshop (PV2001)*, April 2001.

[2] R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, ”An integrated source transcoding and congestion control paradigm for video streaming in the Internet,” *IEEE. Trans. Multimedia*, vol. 3, no. 1, pp. 18-32, March 2001.

[3] M. Masry and S. S. Hemami, ”An analysis of subjective quality in low bit rate video,” *Proc. IEEE ICIP'01*, October 2001.

[4] Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, ”Joint model number 1, revision 1 (JM-1R1),” JVT-A003r1, January 2002. available at ftp://standard.pictel.com/video-site/h26L/

[5] I. V. Bajić and J. W. Woods, ”Domain-based multiple description coding of images and video,” *Proc. SPIE 4671 (VCIP'02)*, pp. 124-135, January 2002.

[6] I. V. Bajić, O. Tickoo, A. Balan, S. Kalyanaraman, and J. W. Woods, ”Integrated end-to-end buffer management and congestion control for scalable video communications,” submitted for publication, December 2002.

[7] K. Chandrayana, S. Ramakrishanan, B. Sikdar, S. Kalyanaraman, A. Balan, and O. Tickoo, ”On randomizing the sending times in TCP and other window based algorithms,” *RPI ECSE Networks Lab Technical Report*, ECSE-NET-2001-1, July 2001.

[8] http://networks.ecse.rpi.edu/~balana/sequences