Efficient Congestion Control for Streaming Scalable Video over Unreliable IP Networks

P. B. Ssesanga and M. K. Mandal

Multimedia Computing and Communications Laboratory
Department of Electrical and Computer Engineering
University of Alberta, Edmonton, Alberta, Canada
E-mail: sesbp@ece.ualberta.ca and mandal@ece.ualberta.ca
URL: http://mccl.ece.ualberta.ca/

ABSTRACT

Efficient adaptive Internet video streaming of scalable video requires suitable transport control algorithms to adapt the transmission rate to the fluctuating network conditions. The algorithms implemented in TCP, the most prevalent Internet transport control protocol, are not suitable for video streaming as they create undesirable rate and hence viewing quality fluctuations. In this paper we propose a video streaming congestion control mechanism that uses the receiver buffer to monitor the occurrence of congestion and to determine the degree of its effect on the video connection. This information is used in determining how much the connection should adjust its rate to adapt to the network condition, resulting in a more robust technique of predicting available bandwidth for the video session. The simulation results show that the scheme performs well under congestion and bandwidth probing, and is fair to TCP and other streams and, has a smooth throughput/rate profile.

Keywords: Congestion control, Video Streaming, TCP Friendliness, Scalable Video, Buffer Management.

1. INTRODUCTION

The scalable, low bit rate and error resilient coding capabilities like in FGS MPEG-4 allow for robust video communication over limited and unpredictable channels such as wireless channels and the Internet [1]. This makes scalable MPEG-4 appropriate for efficient video streaming applications in wireless and Internet environments.

Video streaming, which refers to the simultaneous rendering, in real time, of video content as it is being received from the source places quality of service (QoS) constraints on the underlying network. The Internet Protocol has positioned itself as the protocol of choice in the delivery of integrated multimedia services. Being a best effort network with no bandwidth guarantees, however, means that packets are either corrupted in transit or lost due to congestion.

Efficient adaptive video streaming requires suitable transport control algorithms to adapt the transmission rate to the fluctuating network conditions without significantly fluctuating the quality of reception. Although the existing TCP allows for this, it is not suitable as it results in undesirable fluctuations of bit-rate and hence video quality. Modifications of the core TCP congestion control algorithms have been proposed in the literature, but these algorithms still do not provide satisfactory performance.

Most congestion control algorithms are based on two main strategies: model based (also called equation because a single equation is used to model the network) strategy [3], and binomial (window/rate) control strategy [4]. The model based congestion control method provides relative stability and TCP friendliness [5], [6] whereas the binomial congestion control method provides adaptability, good bandwidth probing capabilities, and high sensitivity to the occurrence of congestion.

Of particular interest recently, is the receiver-based congestion control, which takes the congestion detection functions and rate estimation workload from the server to the receiver. There have also been efforts to combine buffer management with congestion control [7] to improve the video quality during congestion.

In this paper we propose a congestion control scheme that is based on the receiver buffer occupancy for the detection of congestion exploiting the fact that congestion leads to a significant reduction in the buffer level as a result of increased queuing delays and/or packet losses at the affected network nodes. The degree of response to congestion is based on the observed throughput as determined from the impact of congestion on the buffer level. The client readily captures the relevant information and periodically passes it on to the server that determines the most appropriate adjustment considering the network conditions in light of the timing and smoothness requirements of the video application.

The rest of the paper is organized as follows. Section 2 presents the proposed receiver buffer based congestion control scheme. We present the performance evaluation in section 3. The conclusion is presented in section 4.

2. RECEIVER BASED CONGESTION CONTROL USING BUFFER OCCUPANCY

The proposed congestion control algorithm is based on the buffer occupancy at the receiver for congestion detection and congestion response. In this congestion control scheme, buffer occupancy is measured both in terms of media temporal length and media size in bits. The use of temporal length is motivated by the fact that video is time rated in frames/second and therefore working at this level ensures that frames are scheduled for transmission and rendered not later than their play out time. At the start of the video session the client determines the temporal length of media to be pre-buffered before play starts. This technique also known as pre-roll is normally used for smoothing out the effects of delay jitter due to the usually variable transmission delay incurred by the video packets as they traverse the packet network. Under perfect network conditions, the buffer should maintain a certain average occupancy. When congestion occurs increased delays and packet losses affect the number of video packets that get through to the client. This results in the buffer occupancy to fall, which in adverse conditions, leads to buffer under-run as the play out rate exceeds the arrival rate. Therefore, monitoring the buffer occupancy together with packet loss enables a video session to detect the incidence of congestion early enough and to take remedial measures before the video play out becomes discontinuous due to missing video packets or their late arrival.

2.1. Congestion Detection

Let T_s be the length of the pre-roll buffer in seconds and $T_c < T_s$ be the selected congestion threshold. When the buffer occupancy falls to T_c, congestion is presumed to have occurred. The session also detects congestion if packet losses are observed that result in a significant reduction in the buffer occupancy. The appropriate buffer occupancy that triggers congestion response is T_{loss}.

$$T_c \leq T_{loss} < T_s \tag{1}$$

2.2. Congestion Response

The efficiency of a congestion control algorithm is determined by how it responds to congestion and of course its bandwidth probing capabilities and; how all these impact the viewer's experience. In addition to setting the temporal length of video available in the buffer the client monitors the size of the video content in bits and calculates the average buffer occupancy in bits. Therefore when congestion is detected the degree of congestion response depends on the buffer emptiness measured in bits. For scalable video such as FGS MPEG-4 the amount of enhancement layer to be transmitted is proportional to the ratio of the buffer occupancy in bits to the average full buffer size. This is because the buffer emptiness represents the penalty that congestion has imposed on the video session. This scheme only adjusts the proportion of the enhancement layer data to allow the video session to remain continuous during congestion. If congestion reaches such a level that the base layer has to be truncated then the stream is simply cut off because the base layer is coarsely encoded with the essential representation of the video pictures and any data dropped from this layer would result in a highly distorted picture.

Let B_{av_size} be the full buffer size in bits and B_{av_obs} the observed average buffer occupancy at the client then

$$\eta = \frac{B_{av\_obs}}{B_{av\_size}} \tag{2}$$

where η is the rate adjustment coefficient to be used as explained above. Given the buffer temporal length the full buffer size in bits (B_{av_size}) can be easily determined at the server by averaging the bit length of data spanning the specified temporal buffer length.

2.3. Server Side Operation

Rate adjustment, in this case, the amount of enhancement data to drop can be done at the temporal level or the SNR level. Subjective quality studies show that viewers prefer slower moving but high quality pictures to fast low quality pictures [8]. Therefore, the SNR FGS layer is usually preferred over the FGS temporal enhancement layer. The server also includes a smoothing buffer that is used to optimally fit the buffered video data in the available bandwidth. Let this smoothing buffer be *1 second* of video data in length. Then the amount of video data scheduled for transmission, after effecting any reductions, is uniformly distributed and transmitted over 1 sec. Let F_k be the size of any frame *k* in a selected 1 sec. window, the amount of video data scheduled for transmission within 1 sec. is given by

$$D_{txn} = \sum_{k=1}^{N} (F_{base_k} + \eta F_{enh_k}) \tag{3}$$

where

$$F_k = F_{base_k} + F_{enh_k} \tag{4}$$

F_{base_k} is the size of the base layer (in bits) of frame *k* and F_{enh_k} is the size (in bits) of the enhancement layer of frame *k*. *N* is the number of frames in the time window.

Matching the transmission rate with the video play out rate and maintaining the transmission schedule at the server ensures that there will always be enough video data to be rendered at the client. Any drift is then easily captured through the receiver buffer emptiness and appropriate measures taken to restore the normal situation. After adjusting the data rate, the server also schedules the data to be transmitted taking into consideration the receiver buffer deficit to be made up. Therefore moments following the initial response, the server may momentarily reduce the amount of data per transmitted frame to achieve this.

Congestion recovery is determined from the fact that in the absence of congestion the receiver temporal buffer length is consistently greater than the congestion threshold. If this is observed over at least 3 feedback updates then the server begins restoring the data rate to the normal rate by gradually increasing the amount of transmitted video frame data. This process also known as bandwidth probing is done according to the following equation.

$$D_{txn} = \sum_{k=1}^{N} (F_{base_k} + [\delta + \eta(1-\delta)] F_{enh_k}) \quad (5)$$

where δ is progressively increased in a sequence similar to 1/8, ¼, ½, ¾, 7/8, 1. This sequence represents the aggressiveness yet smooth bandwidth probing capabilities of the algorithm.

2.4. Client Operation

The client measures all the relevant data and is also responsible for determining the occurrence of congestion as explained above. The client is naturally suited for this task as the impact of congestion can be easily determined from the observable effects. The buffer occupancy in bits is a weighted average over the few most recent measurement rounds giving a higher weight to the most recent values while also including the long term behaviour of the video session. The client periodically sends the average buffer occupancy in bits and the proportional temporal occupancy to the server in feedback packets. This information is then used by the server to adjust transmission rates in response to the changing network conditions. The maximum feedback interval is determined from the temporal length of the receiver buffer and is adjusted downwards in proportion to the buffer emptiness to allow for less frequent updates under normal network conditions and more frequent updates when congestion is being experienced. Let T_{fb} be the maximum feedback interval and T_{fb_k} the feedback interval after interval k then,

$$T_{fb_k} = \eta T_{fb} \quad (6)$$

It is reasonable to take T_{fb} as half of the temporal buffer size. It would be desirable to have the update interval longer, but this value ensures that the buffer does not get starved. Frequent updates during congestion enable the session to quickly respond to any changes for the worse or indeed a quick recovery as soon as congestion subsides.

The combination of buffer occupancy and the deviation of the transmission rate from the average value allows the algorithm to appropriately respond to congestion while taking into consideration the long term dynamics of the session, which allows for smoothness and stability. For scalable video, such as FGS coded MPEG-4, a very small reduction in the transmission rate may not translate into a significant quality variation of the rendered video [9].

3. PERFORMANCE EVALUATION

The performance of the proposed congestion control scheme has been evaluated with Network Simulator (NS-2) [10]. As shown in fig. 1, we have used a dumbbell model to realise the bottleneck link in the network simulation with five sessions all going through the link. Two of the five sessions are multimedia UDP streams (using FGS MPEG-4 video traces encoded at CBR 250 kbits/sec) and the other three streams are all TCP streams simulating FTP data transfer. Router 1 uses the tail-drop queue management protocol while Router 2 employs the fairer RED (Random Early Detection) scheme. Of the terminal nodes, the senders on the left hand side and the receivers on the right hand side in a one-to-one mapping. The bottleneck link is 2 Mbps while all the other channels can use up to 5 Mbps each. All data packets flow from left to right and only acknowledgement/feedback packets run in the opposite direction. The channels are full duplex to allow for feedback.

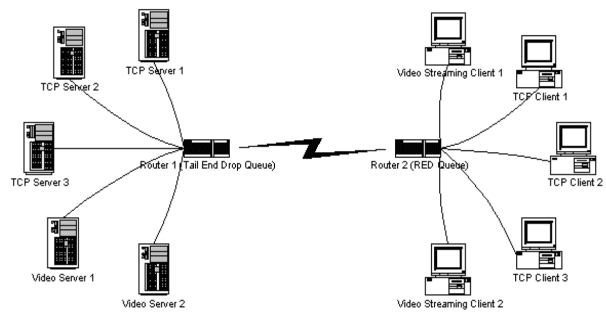

Fig. 1. Simulation Network Layout with 5 parallel sessions.

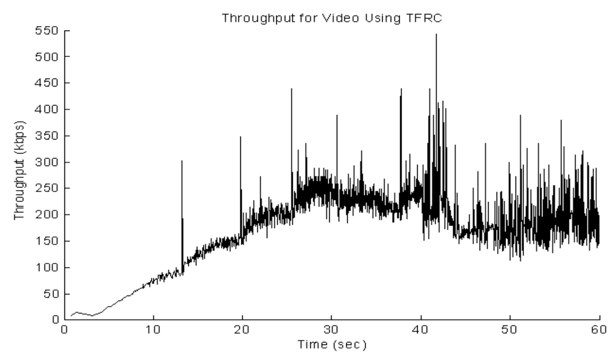

Figure 2: Video Session Using the TFRC congestion control

Table 1: Statistics for the TFRC Session

Mean (kbps)	Standard Deviation (kbps)	Minimum (kbps)	Maximum (kbps)
196.666	58.975	7.492	542.535

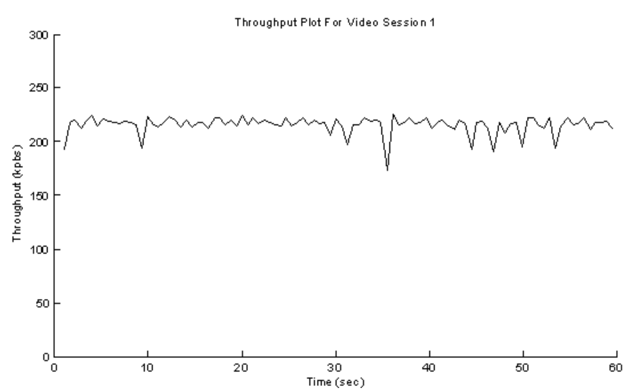

Figure 3: Video Using the Proposed congestion control scheme

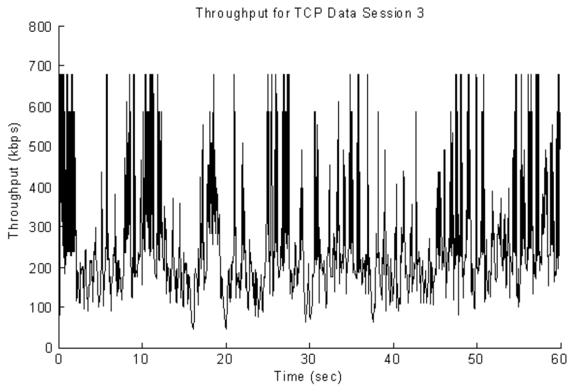

Figure 4: Throughput for a data session using TCP congestion control

Table 2: Statistics of the Experiment with Buffer based Congestion Control

	Mean (kbps)	Standard Deviation (kbps)	Minimum (kbps)	Maximum (kbps)
Video 1	215.720	7.919	173.950	225.979
Video 2	215.354	6.985	192.212	226.286
TCP 1	252.058	133.460	8.672	678.168
TCP 2	263.684	138.649	8.650	678.168
TCP 3	285.130	149.357	44.479	678.168

Fig. 3 shows a smooth low oscillation throughput profile of an FGS MPEG-4 video session. This is obtained with the proposed buffer based congestion control scheme. Fig. 4 is a throughput profile of a concurrent TCP session. Typical of TCP we see that the rate oscillation is high pitched. From Table 2 showing the statistics of the experiment, we observe that despite having a low rate variation the congestion control scheme treats all other streams including TCP with fairness. The two video streams have almost the same throughput while the throughput for the three TCP sessions is of the same order though even slightly higher than for video. The rate deviation and oscillation is very similar, a manifestation of the fairness exhibited by the proposed control strategy. These statistics demonstrate the fairness and smoothness exhibited by the proposed congestion control strategy. Fig. 2 is a comparative experiment using TFRC (TCP Friendly Rate Control) for the video sessions. The TFRC is an equation based congestion control protocol (IETF RFC 3448) that has been proposed for use in video streaming applications. The throughput profile together with the standard deviation show how widely the rate varies during congestion. As we have noted above, this is not great for video streaming applications. The mean throughput obtained is also less than what we obtain with the receiver buffer based congestion control scheme.

4. CONCLUSIONS

In this paper we have proposed a congestion control algorithm based on receiver buffer occupancy to detect the occurrence of congestion and extent of its effect on the video application. The degree of response, which involves dropping some of the video data, is determined by the congestion penalty measured in the effect of congestion on the receiver buffer. The proposed algorithm introduces the long term dynamics of the session which improves smoothness while also allowing for early detection of congestion before it grossly affects the session. It was shown from the simulation results that the proposed technique is fair, reduces overall oscillations and avoids starvation of competing streams. Most of all its implementation takes into consideration the timing requirements of the video application. This scheme makes a very close approximation of the available network bandwidth and because of its integration in the application it can be extended to general congestion management in non-internet scalable video streaming applications.

REFERENCES

[1] R. Koenen, "MPEG-4 Overview - (V.21)," *ISO/IEC JTC1/SC29/ WG11 N4668*, March 2002

[2] C. Lin, J. Zhou, J. Youn and M. Sun, "MPEG Video Streaming with VCR functionality," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 11, No. 3, pp. 414-425, March 2001.

[3] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," *Proc. ACM SIGCOMM*, vol. 30, no. 4, pp. 43-56, Stockholm, Sweden, March 2000.

[4] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms". *Proc. INFOCOM, 20th Annual Joint Conf. of the IEEE Computer and Comm. Societies*, Vol. 2, pp. 631 –640, Anchorage, AK, USA, 2001.

[5] K. Lee, T. Kim and V. Bharghavan, "A Comparison of End-to-End Congestion Control Algorithms: The Case of AIMD and AIPD," *Proc. IEEE GLOBECOM,* Vol. 3, pp. 1580-1584, San Antonio, TX, USA, 2001.

[6] M. Vojnovic and J. Le Boudec, "On the Long-Run Behaviour of Equation-Based Rate Control," *Proc. ACM SIGCOMM,* Pittsburgh, Pennsylivania, USA, Aug. 2002.

[7] A. Balam, O. Tickoo, I. Bajic, S. Kalyanaraman, and J. Woods, "Integrated Buffer Management and Congestion Control for Video Streaming," *Proc. IEEE ICIP, Barcelona, Spain, September 2003*.

[8] M. Masry, and S. S. Hemani, "An Analysis of Subjective Quality in Low Bit Rate Video," *Proc. IEEE ICIP Thesaloniki, Greece, 2001*.

[9] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press 1998.

[10] NS-2, http://www.isi.edu/nsnam/ns.