An End-to-End Congestion Control Approach Based-on Content-Aware

[1]Hongyun Yang, [2]Xuhui Chen, and [2]Ruiming Hu
[1]*National Engineering Research Center for Multimedia Software*
Wuhan University, Wuhan, Hubei, China
[2]*Computer School, Wuhan University, Wuhan, Hubei, China*
[1]*yhycxh@gmail.com*, [2]*wh.chxh@gmail.com*

Abstract

With the widespread penetration of broadband accesses, many multimedia applications, such as video surveillance, on-demand video streaming, IPTV and news broadcast are emerging. Since video streaming applications on the Internet generally have very high bandwidth requirements and yet are often unresponsive to network congestion. In order to avoid congestion collapse and improve video quality, these applications need to respond to congestion in the network by deploying adaptive control mechanisms. In this paper, we present a receiver-based, bandwidth estimation rate control mechanism with content-aware probability retransmission to limit the burden on the congested network. Like the TCP Friendly Rate Control (TFRC) protocol, we compute the sending rate as a function of the loss event rate and round-trip time. Considering the different importance of three distinct types of frames in Standard MPEG encoders, we divide data packets into three grades coarsely and adopt adaptive probability retransmission strategy to assure video playback quality. It is an extension of TCP friendly congestion control. This paper describes the smooth rate algorithm and probability retransmission mechanism. The result of experiments with competing TFRC specification demonstrates the proposed approach reaches a higher throughput and higher PSNR than TFRC congestion control algorithms especially on the bottleneck links.

Keywords: *Congestion control; Probability retransmission; Content-aware; MPEG*

1. Introduction

With the widespread penetration of broadband accesses, many multimedia applications, such as video surveillance, on-demand video streaming, IPTV and news broadcast are emerging. Since those video streaming applications on the Internet generally have very high bandwidth requirements, constant video quality, low startup delays and small latencies, it is challenging to the best-effort packet- switched network, which is often unresponsive to network congestion. At the same time, the application is loss tolerant to some extent, TCP isn't fit for. UDP protocol is used as transport protocol. UDP does not provide congestion control and QoS guarantee mechanism. To void congestion and get better video playback quality, it is essential to add control mechanism to adaptively adjust each supplying peer's sending rate in the current best-effort unicast datagram service network.

In order to fit for heterogeneity network bandwidth requirements, multirate approaches, such as scalable coding techniques (SVC) and Multiple Descriptive coding (MDC) have been proposed. Those approaches adopt layered adaptation to serve the receiver with different bandwidth capacity and assign packets from the base layer's highest priorities, whereas assign packets from enhancement layers progressively lower priorities. When there isn't enough bandwidth, routers discard the lowest-priority packets, thereby preventing loss of the base layer or high-priority enhancement layers. Nevertheless, it requires routers to implement some priority-based packet scheduling policy, which is considerably more complex and difficult to deploy [1]. At the same time, they are not yet to be deployed in mainstream media players. To date, the single-rate (or single-layer, single-description) video coding remains the most efficient and effective technique [2]. For single-rate video streaming, it is necessary to adopt smaller granularity priority packet scheduling policy to guarantee the quality of service on congestion network.

In the paper under the heuristic approach of Differentiated service, with the motivation to maximize decoded video quality while limiting network congestion, we present a receiver-based, bandwidth estimation rate control mechanism and take single-rate video coding into account. Aiming at four basic types of compressed video frames in standard MPEG (I P, B, S) and according to the dependency of different frames and the effect to the video quality, we classify data packets of frame into three different grades and adopt priority probability retransmission requests method to protect important packets in order to obtain accepted QoS even on network congestion.

The remainder of the paper is organized as follows. In Section 2, we review some related research on end-to-end congestion control methods for multimedia streaming. In Section 3 we describe the receiver-based rate control combining with probability retransmission to limit the incurred network congestion. We also explain the fairness between the method and TCP. Experimental results for MPEG-4 encoded video over NS2 simulator network are analyzed in Section 4. Finally, we draw a conclusion and offer the further research work.

2. Related Work

Congestion control is vital in multimedia applications to adapt to dynamic and heterogeneous network environments. Various techniques have been proposed to address network congestion from unresponsive multimedia streams on the Internet[5].Some of the most popular congestion control mechanisms used are window-based, layer-based, rate-based and local recovery based[5,24]. The rate-based regulation is, in principle, a mechanism that keeps the instantaneous rate generated by the sender or received by the receiver below a specific level. It can be partitioned into three classes depending upon the place where the control mechanism is implemented: source-based, receiver-based and hybrid congestion control.

TCP is the dominant and most widely used protocol at the transport layer. It implements flow and congestion control mechanisms in order to avoid receivers' buffers overflowing and network congestion. Because TCP is window-based[6],strict implements congestion control ,which halve the sending rate in response to a single packet drop[7] and does not carry any time related information, it is unfit for transmitting real time streaming.

TCP-Friendly Rate Control (TFRC) [7] is a rate-based mechanism. It directly uses an equation for the allowed sending rate as a function of the loss event rate and round-trip time to match a throughput that is "TCP compatible". Unlike TCP, where the sending rate is controlled by a congestion window that is halved for every lost packet, TFRC refrains from reducing the sending rate in half in response to a single packet-loss.

Instead, the sender explicitly adjusts its sending rate as a function of measured rate of loss events, where a loss event consists of one or more packets lost in a single round trip time. TFRC has a much lower variation of throughput over time than TCP and maintain a relatively steady sending rate while still being responsive to congestion. As a result, it is more suitable for multimedia application. However, the specification doesn't take different importance of different transmission data packets of a streaming into account. It is hard to assure the decoded video quality when data packets of the most important frame were lost.

Standard MPEG encoders generate three distinct types of frames, namely I, P, and B frames. The I frame is encoded in Intra mode, and is essential for the prediction coding of other frames. If part of an I frame is lost, then all frames in the group of pictures (GoP) including this particular frame are impaired. The P frame is encoded in prediction mode, while the B frame is encoded in double prediction mode. As with the I frame, the P frame is also important. If part of a P frame is lost, the impairment propagates the particular P frame, previous B frames, and the following frames in the GoP that includes this P frame [14]. Conversely, if part of a B frame is lost, the impairment propagates solely within that frame.

Considering the different importance of frame to decode, under packetized video transmission, the relative priority assignment for a packet would be best if it can precisely represent its error propagation effect to the video quality at the receiver. For a video stream, a lost packet can lead to the content loss of subsequent packets due to temporal loss propagation as a result of the inter-block and inter-frame correlation. In fact, a raw packet loss rate of 3% in an MPEG encoded bit stream can result in a frame error rate as high as 30% [13].

3. The Adaptive Congestion Control Method

In this section, we describe the receiver-based adaptive rate control algorithm with TCP friendly bandwidth share. Then we explain the probability retransmission method according to the importance of different data packets with the assumption that the sender and receiver have enough buffers and the receiver adopts prefetching way so as to assure the retransmission key parts don't affect decoding and playback of video streaming.

3.1. Receiver-based Adaptive Rate Control

TFRC [7] directly uses a throughput equation for the allowed sending rate as a function of the loss event rate and round-trip time. The throughput equation is:

$$T = \frac{s}{R*\sqrt{2*p/3} + (4*R*(3*\sqrt{3*p/8}*p*(1+32p^2))} \quad (1)$$

where T is the transmit rate in bytes/second, s is the packet size in bytes, R is the round trip time(RTT) in seconds, p is the loss event rate between 0 and 1.0.

3.1.1. RTT Estimations: The sender makes the RTT estimations using the sequence number and timestamps. Sender codes the sending time t_send in packet header and the receiver codes the t_recv in the feedback packet header, where t_recv=t_send and then sends them back to the sender. When a feedback packet is received by the sender at time t_now, it performed the following algorithm to estimate the RTT between it and the receiver:

If no feedback has been received before

RTT =(t_now-t_recv)-t_delay

Else

RTT=q*RTT+ (1-q)*((t_now-t_recv)-t_delay) (2)

where q has a default value of 0.9, t_delay is The amount of time elapsed between the receipt of the last data packet at the receiver, and the generation of this feedback report.

3.1.2. Packet Loss Rate Estimation: The receiver measure the packet loss rate based on packet global sequence number. If the expected packet sequence number is equal to packet sequence number received, there is no packet loss. When the expected packet sequence is smaller than the packet sequence received, the packet loss is detected. We recombined received packets at receiver's buffer .According to additional sequence number a packet belonging to a specific frame marked when frames were packed we decided whether lost packets retransmission process was started. In order to prevent a single spurious packet loss having an excessive effect on the packet loss estimation, the receiver smooth the values of packet loss event rate using a filter that weights the n most recent loss event intervals in an average way.

3.2. Content-Aware Probability Retransmission Packet Scheduling

The family of MPEG standards, which includes MPEG-4, encodes video as a stream of I, P, and B frames. Each frame contains varying amount of information regarding the video content and serves a specific purpose in the video representation. The I-frames are still images that contain the maximum amount of information in the video representation. The P-frames contain predictive information that is used to reconstruct the B-frames. The B-frames are the smallest of all the three frame types and contain the least amount of information. Since the MPEG video standards encode some amount of predictive information that can be used to reconstruct lost frames during network transmission, we exploit this feature to selectively retransmission lost I-frames and P-frame without considering retransmission lost B-frames. The encoded sequence contains forward and backward references for the predictive coding of P and B-frames, which can be used to predict these frames when there are P-frames and B-frames loss.

We assume the probability of retransmission I-frame, P-frame, B-frame is PI, PP and PB. In the application layer packet header, we added frame type (f_type), frame sequence number (f_seq), the total number of packets packed a frame (totalPkt), and the sequence number of each packet of a frame (seqPkt). When the receiver received a packet, it estimates the loss event rate and reconstructs frames the packets belonging to. In our algorithm, we assume the sender and receiver have enough buffers and the receiver adopts per-fetch method [22] so as the retransmission doesn't effect video playback. According to the packet sequence number (seqPkt) and the total number (totalPkt), we can judge whether retransmission will start. Every packet has a unique sequence number (uSeqPkt).The probability retransmission algorithm is described in Figure 1.

Here to assure the decode quality, we assume Pro=1.0, 0.5, 0 corresponding to I, P, B frame.

When the expected packet sequence is smaller than the packet sequence received, the packet loss is detected. We recombined received packets at receiver's buffer. According to additional sequence number a packet belonging to a specific frame marked when frames were packed we decided whether lost packets retransmission process was started.

Let Q be a queue of packets for which i has received for Tm time. F donates the type of frame (I, P, B), Pro donates the retransmission probability of packet (P_I, P_P and P_B).

```
while(1)
receive(pkt);
if(pkt.seqPkt==0) enqueue(Q, pkt);
if(pkt.seqPkt>0&&pkt.seqPkt<pkt.totalPkt-1)
{
 H = HeaderOfQueue(Q);
 If(H.f_type==pkt.f_type)
{
 If(H.f_seq==pkt.f_seq&&H.seqPkt==pkt.seqPkt-1) enqueue(Q,pkt);
  If(H.f_seq==pkt.f_seq&&H.seqPkt<pkt.seqPkt-1)
    Retransmit(pkt.uSeqPkt-h.uSeqPkt+1,pkt.uSeqPkt-1,Pro);
 }
 Else  {
  Retransmit(H.uSeqPkt+1,H.uSeqPkt+H.totalPkt-H.seqPkt-1,Pro);
  Retransmit(pkt.uSeqPkt-(pkt.totalPkt-pkt.seqPkt+1),pkt.uSeqPkt-1,Pro);
   }
} else {
 writeToBuffer (Q,pkt.totalPkt);
}
```

Figure 1. Retransmission scheduling algorithm used by receiver i.

4. Experimental Results

Network Simulator 2 (NS-2) [16] is used in the simulations. We compare the proposed scheme with original TFRC proposal [7], since the latter has been integrated into DCCP [17] as one of the possible congestion control method. In order to convenient for presentation, we denoted our scheme as CACC (Content-Aware Congestion Control).

4.1. Simulation model

In our simulations, the simulator network uses a typical "dumb-bell" topology as shown in Figure 2. All links except the bottleneck link are sufficiently provisioned to ensure that any drops/delays that occur are only due to congestion at the bottleneck link.

The bandwidth of the bottleneck link is 1.5Mbps; delay is a random number of 5ms-20ms. Other links' bandwidths are 10Mbps; delays are also a random number of 5-10ms. We adopt Evalvid [18], a simulation and evaluation framework of networked video transmission to estimate the performance of the congestion control method we designed. Reference [19], we modified the original TFRC code in NS2 to allow the reading MPEG-4 video-trace files ,simulation of video transmission, and the generation of appropriate trace file. The simulator last 30s. And the packet size is 1KB.

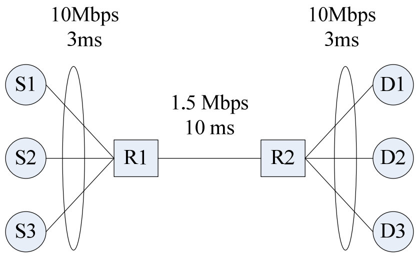


Figure 2. Simulator Network

In the simulator network, node S1 is TCP sender with FTP flow. Node S2 is TFRC sender and node S3 is CACC sender. Node S2 and S3 are both with MPEG-4 video-trace file. The video source is in the YUV QCIF (176 x 144) format. The video streaming codec is NCTU [20].Frame rate is 30fps.Node D1, D2, D3 are the relevant sinks.

4.2. Evaluation Metrics

We provide simulation metrics, including network-level metrics such as delay, jitter, fairness and application-level metrics such as peak-signal noise ratio (PSNR) to estimate the performance of CACC comparing with original TFRC specification.

4.2.1. Inter-arrival Jitter and Delay. Our implementation for delay jitter calculations is fererence on the algorithm defined in RFC 3550. Shortly explaining, let Si is the timestamp of packet i, and Ri is the arriving time in timestamp units of packet i, then for two sequentially packets i and j, delay D may be expressed as:

D= (Rj-Ri)-(Sj-Si) (3)

This delay variation should be calculated for each packet. We use a filter function to avoid temporal fluctuation and the delay jitter is computed with the use of the following equation:

Ji= (15/16) Ji-1+ (1/16) D (4)

4.2.2. Fairness: A flow is TCP-friendliness means that it does not reduce the long-term throughput of any co-existent TCP flow on the same path under the same network conditions. It uses throughput as a scale.

4.2.3. PSNR.PSNR is one of the most widespread objective metrics to assess the application-level QoS of video transmissions. The following equation shows the definition of the PSNR between the luminance component Y of source image S and destination image D:

$$PSNR(n)_{dB} = 20\log_{10}\left[\frac{V_{peak}}{\sqrt{\frac{1}{N_{col}N_{row}}\sum_{i=0}^{N_{col}}\sum_{j=0}^{N_{row}}\left[Y_S(n,i,j)-Y_D(n,i,j)\right]^2}}\right] \qquad (5)$$

where $V_{peak} = 2^k - 1$ and k = (number of bits per pixel (luminance component)). PSNR measures the error between a reconstructed image and the original one. The larger the PSNR value, the better the video quality perceived by the end user.

4.3. Evaluation result

The delay, jitter and average throughput measure is straightforward and is being done NS2 trace file.

Figure 3 shows the comparison of average throughput of TCP, original TFRC and CACC.

The comparison of packets' jitter is shown in Figure 4.Original TFRC and CACC prevent oscillations better than TCP.

Figure 5. shows the delay of the three conditions. Because of selective retransmission, the delay of CACC is a little longer.

Figure 6. shows the comparison of PSNR of original TFRC and CACC.CACC shows a good video quality than original TFRC.

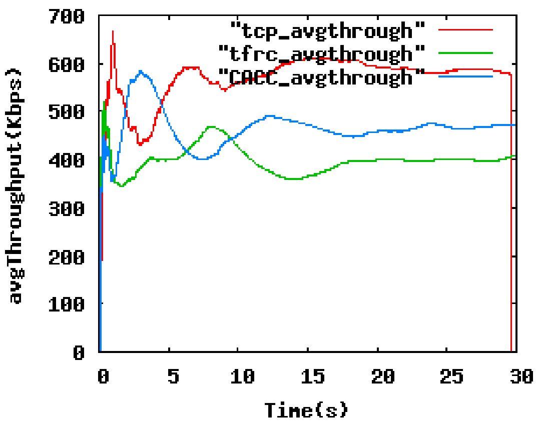


Figure 3. Comparison of average throughput

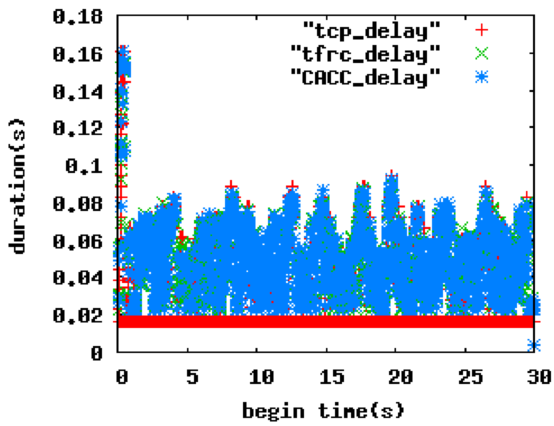


Figure 5. Comparison of delay

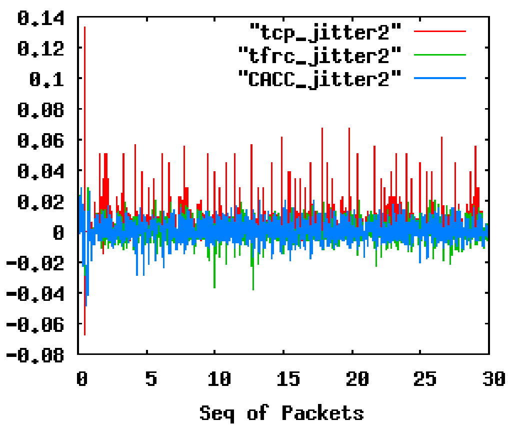


Figure 4. Comparison of jitter

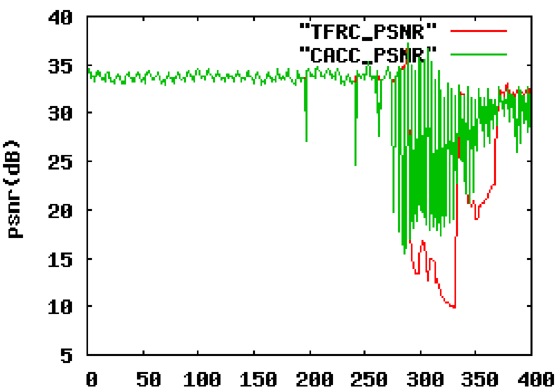


Figure 6. Comparison of PSNR

5. Conclusion and Future Work

In this paper, we extended the original TFRC regulation and proposed a content-aware end-to-end congestion control schemes on “best-effort” packet switched networks for video streaming delivery. According to dependency of different frame when decoding video sequence in standard MPEG family, we adopt probability retransmission packet scheduling mechanism to retransmit lost important packet. The mechanism achieves better PSNR.

In order to reduce the impact of retransmission to video playback quality, appropriate buffer manage and prefetching mechanism will be introduced. At the same time, further research includes using the congestion control method in Peer-to-Peer network environment and evaluating the affect of packets buffer in real time streaming applications.

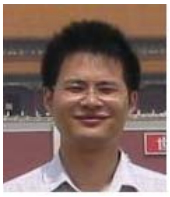
References

[1]L.J.chuan,L.Bo, and Z. Y.Qin, “Adaptive Video Multicast over the Internet” ,IEEE Multimedia,vol.10,no.1,Jan/Mar.2003, pp.22-33.
[2]L.J.chuan, G. R.Sanjay,L. Bo, and Z.Hui, “Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast” ,In Proceeding of the IEEE, invited paper,vol.96,no.1,Jan.2008, pp.11-24.
[3]Y.-H. Chu, S. G. Rao, and H. Zhang. “A case for end system multicast” , In Proceedings of ACM Sigmetrics, June 2000.
[4] L.Qiang, and H. J.Neng, “End-to-End Available Bandwidth Estimation and Time Measurement Adjustment for Multimedia QoS” ,ICME 2003, 2003,pp.373-376.
[5] A. Popescu, D. Constantinescu, D. Erman, and D.Ilie, “ A Survey of Reliable Multicast Communication”, Next Generation Internet Networks, 3rd EuroNGI Conference, May 2007,pp.111-118.
[6] C. Bouras, A. Gkamas, and G.Kioumourtzis, “Extending the Functionality of RTP/RTCP Implementation in Network Simulator (NS-2) to support TCP friendly congestion control”, SIMUTools 2008, Marseille, France,Mar.2008.
[7] M.Handley,S. Floyd, “TCP Friendly Rate Control (TFRC):Protocol Specification”,RFC 3448.
[8] V. Jacobson. “Congestion avoidance and control”,In Proceedings of ACM SIGCOMM’88, Stanford, CA,1988, pp.314-329.
[9] S. R. Subramanya, S. Jagannathan, Mingsheng Peng, “A Scheme for Fair,Rate-based End-to-End Congestion Control of Multimedia Traffic in Packet Switched Network”,ICME 2003,2003, pp.357-360.
[10]The ATM Forum Technical Committee, Traffic Management Specification Version 4.1. http://www.atmforum.com, March 1999, no. AF-TM-0121.000.
[11] D. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” Computer Networks and ISDN Systems, vol. 17, June 1989, pp. 1-14.
[12] J. R. Santos, Y. Turner, G. (John) Janakiraman, “End-to-End Congestion Control for InfiniBand”, Technology report, HPL-2002-359.
[13] J. Boyce and R. Gaglianello. “Packet loss effects on mpeg video sent over the public internet”, In Proceedings of ACM Multimedia ’ 98, Bristol, U.K., Sep. 1998, pp. 181-190.
[14]A. Ziviani, B. E. Wolfinger, J. F. Rezende, O. C. M. B. Duarte, and S. Fdida, “ Joint Adoption of QoS Schemes for MPEG Streams” , Kluwer Academic Publishers, ISSN: 1380-7501, vol. 26, no. 1,May 2005, pp. 59-80.
[15] J. Pandhye, J. Kurose, D. Towsley, and R. Koodli, “ A model based TCP-friendly rate control protocol”, In Proceeding of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, June 1999.
[16] Ns-2 manual, http://www.isi.edu/nsnam/ns/.
[17] E. Kohler, M. Handley, and S. Floyd, “Datagram congestion control protocol (DCCP)” , IETF, Request for Comments 4340, Mar. 2006.
[18] J. Klaue, B.Rathke, and A. Wolisz, “ EvalVid - A Framework for Video Transmission and Quality Evaluation” ,In Proceeding of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, USA, Sep. 2003.
[19] K. Chih-Heng, S.Ce-Kuen, H.Wen-Shyang, and A, Ziviani, “Improving video transmission on the Internet”, IEEE POTENTIALS, 2007.

[20] A. Lo, G. Heijenk, I. Niemegeers, " Performance Evaluation of MPEG-4 Video Streaming over UMTS Networks using an Integrated Tool Environment", In Proceedings SPECTS 2005, 2005 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Philadelphia, PA, USA,July 2005, pp.24-28.
[21] C. Bhadricha, B. Bing, "Scalable MPEG-4 Video Transmission Using Video Quality Metrics", Communication Networks and Services Research Conference (CNSR2008), 2008, pp. 295-299.
[22] A. Sharma, A. Bestavros, and I. Matta. " dPAM: a distributed prefetching protocol for scalable asynchronous multicast in p2p systems", In Proceedings of the IEEE INFOCOM, March 2005.
[23] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.
[24] J.Widmer, R.Denda, and M.Mauve, "A Survey on TCP-Friendly Congestion Control (extended version)", Reihe Informatik, Feb.2001, pp.1-17.

Authors

Hongyun Yang is a Ph.D. candidate at National Engineering Research Center for Multimedia Software in Wuhan University. Her research interests include overlay multicast, peer-to-peer streaming and video surveillance.



Xuhui Chen is a Ph.D.candidate at Computer School in Wuhan University. His research interests include video processing, video surveillance,embedded system.