REAL-TIME NETWORK-AWARE OPTIMAL RATE CONTROL FOR VIDEO COMMUNICATION NETWORKS USING AN AUGMENTED STATE FEEDBACK CONTROLLER

Bulent Cavusoglu, Dan Schonfeld, and Rashid Ansari*

University of Illinois at Chicago
851 S. Morgan St. (M/C 154)
Chicago, IL 60607-7053
*Tel: +1 312 996-5847
{bcavusog, ds, ansari}@ece.uic.edu

ABSTRACT

A real-time adaptive rate control algorithm for video transmission is presented. The proposed algorithm aims to determine the optimal rate at which the video should be sent through the network in order to minimize the expected distortion while meeting the delay requirements. The packet loss rate is assumed to be a function of the transmission rate. The nodes of the network over which video packets are transmitted determine this function, possibly based on the congestion level of the network. Since this function is unknown an augmented state feedback regulator is introduced to control the packet loss rate by adjusting the transmission rate of the video. The expected distortion is calculated by dynamic programming at the source based on the desired packet-loss and transmission rate pair and then the optimal transmission rate is chosen among feasible transmission rates. Simulation results are presented to establish the significant improvement in performance.

1. INTRODUCTION

Reliable transport protocols, such as TCP, may work well for applications that are not constrained by specific delay or jitter requirements. However, TCP is not suitable for multimedia communications, especially video streaming and real-time video, due to its “stop and go” and additive increase multiplicative decrease (AIMD) algorithms. Even though it is unreliable, UDP offers an attractive alternative for delivering multimedia content, as it does not alter the transmission rate due to congestion. However, absence of congestion control mechanisms for multimedia transmission might eventually lead to a communication collapse due to congestion on the Internet [1]. Hence, there has been substantial amount of research to provide congestion control algorithms whose operation is akin to TCP and is also suitable for multimedia communication by using modified TCP or equation-based algorithms. A model-based rate control protocol is given in [2]. A nicely done survey about some of these proposed methods can be found in [3]. Notable among these methods are the Rate Adaptation Protocol (RAP) [4] , Loss-Delay based Adaptation algorithm (LDA) [5], and the TCP-Friendly Rate Control protocol (TFRC) [6]. Another approach to congestion control is network-based. Basically service providers or routers implement algorithms that may isolate unresponsive flows to the network’s congestion and apply penalties to those flows. Random early detection (RED) [7] is a core queue management mechanism in the routers. However, RED does not make any distinction among flows. That is, it drops packets randomly when congestion occurs regardless of the incoming flow to which they belong. This might violate fairness if there are congestion-responsive flows that attempt to reduce their rates. The use of a uniform drop rate for both a responsive and an unresponsive flow in effect penalizes the responsive flow. For this reason other control mechanisms have been developed, which make a distinction among flows and first penalize the unresponsive flows such as CHOKe [8]. It is desirable to effectively allocate available resources such as bandwidth, especially when congestion occurs in the network. The main goal is to minimize the impact of loss of packets due to congestion. For a compressed video, like MPEG-2[9], minimizing packet loss itself does not necessarily mean maximizing video quality. This is due to the fact that the relative importance of content varies significantly among the packets.

2. PROBLEM DEFINITION

We consider the problem of minimizing the expected distortion for video transmission where the packet loss rate (PLR) is a function of video transmission rate. The packet loss rate becomes a function of transmission rate in the presence of a suitable penalty policy enforced by the network for abusive traffics that do not adjust their transmission rate based on the level of congestion in the network, such as UDP. The collection of penalty policies throughout the network determines the behavior of this function. It is not possible to establish an exact closed form relationship between the transmission rate and the PLR due to the possible diversity of network providers and of the congestion control algorithms. It is desirable to be able to control the PLR for the optimization of video quality. This translates to an optimization problem through controlling a dynamic system by using an adaptive controller. The optimization problem is defined as follows,

$$E\left[D_{tot}\right] = \min_{\boldsymbol{\mu}} \sum_{k=1}^{M} E[D_k(\boldsymbol{\mu})] \qquad (1)$$

$$\text{subject to} \quad \sum_{k=1}^{M} T_k(\boldsymbol{\mu}) < T_{\max}$$

where M is the total number of packets that the optimization is performed over, $E\left[D_k(\boldsymbol{\mu})\right]$ is the expected distortion and $T_k(\boldsymbol{\mu})$ is the delay for the k^{th} packet, based on transmission rate $\boldsymbol{\mu}$, and $T_{\max}$ is the maximum allowed delay. For easier representation we will omit $\boldsymbol{\mu}$ from $E[D_k(\boldsymbol{\mu})]$ in our notations and use $E[D_k]$ instead.

3. APPROACH TO THE PROBLEM

We consider both the expected distortion and the packet loss rate together. The rate-controlled traffic flow is sent to the network and the sender receives information about packet loss rate either through RTCP (Real Time Transport Control Protocol) [10] or through negative acknowledgement (NACK) packets [11]. We use a controller together with a Kalman predictor, depicted in Figure 1, to drive the output to a desired packet loss rate by using observations and a simple loss model. Once the information regarding the previous loss rate is received and the packet loss rate (state of the system) is predicted for the next transmission, we attempt to bring the predicted loss rate to the desired loss rate by increasing or decreasing the video transmission rate. It should be noted that the desired loss rate, and thus desired transmission rate, is determined by solving the optimization problem we defined in the previous section. The distortion per pixel can be given using the mean square error by,

$$E[d_i] = E[\left(q_i - q_i^c\right)^2] \quad (2)$$

where q_i is the actual value of the i^th^ pixel and q_i^c is the pixel value at the receiver after possible concealment. We will define the distortion for intra pixels and for predicted pixels separately. Since the concealment method also plays a role in determination of the reconstructed pixel it should be defined in order to compute the distortion. In the concealment method used here the reference frame's pixel is copied and used at the same spatial position as the lost pixel. It can be shown that the expected distortion per packet for intra and predicted pixels is given by:

$$E_I[d_i] = E\left[d_i^c\right]p_i \qquad \text{, for intra pixels only}$$

$$E_P[d_i] = (1-p_i)E\left[d_i^{ref}\right] + p_i E\left[d_i^c\right] \qquad \text{, for predicted pixels only}$$

$$E[D_k] = p_k E[d^c]_k + \beta(1-p_k)E[d^{ref}]_k \qquad \text{, per packet} \quad (3)$$

where $\beta = N_P/N$ and $E[D_k]$ is the expected distortion in the k^{th} transmitted packet including N_I intra macroblocks, N_P predicted macroblocks, thus total $N = N_I + N_P$ macroblocks. $E[d^c]_k$ *and* $E[d^{ref}]_k$ denote the expected distortion per macroblock after concealment in the k^{th} packet and expected distortion per macroblock after possible concealment in the reference macroblocks of k^{th} packet respectively.

The distortion in (3) needs to be computed recursively. That is, the first frame (I frame) generated and transmitted includes all intra macroblocks and does not depend on any reference frames to be reconstructed. Even though the previous GOP's last predicted frame is used for concealment, it is assumed that previous GOP's last P frame is reconstructed successfully to be able to perform these recursive calculations in real time while streaming. Thus, the starting point of the recursive computations is the transmission of the I frame. Once another I frame is transmitted, in other words a new GOP starts, the recursive calculations reset and resume.

3.1. Control of Packet Loss Rate via Adaptive Augmented State Feedback Controller

It is necessary to observe and predict the relationship between the transmission rate and the loss rate in an adaptive dynamic fashion. Hence, we use a Kalman predictor to estimate p by using observations and a simple loss model. We define the state equations for the Kalman predictor as follows:

$$p_k = p_{k-1} + u_k + w_{k-1}$$
$$z_k = p_k + v_k$$

where w and v are process noise and the measurement noise respectively. Here w is used to model the packet drops due to traffic increase stemming from other sources in the network or due to random packet drops as a result of any other underlying algorithm such as RED, and v is used to model possible measurement errors. Both variables are assumed to be Gaussian random variables. However, we chose σ_v^2, measurement noise variance, much smaller than σ_W^2, process noise variance. Thus, we have greater trust in the measurements. Once the state prediction is available we use the state feedback controller to regulate the packet loss rate. Here u is the control input to the system and is used to adjust the transmission rate until a feasible pair of desired loss rate and transmission rate can be found through this regulation process. We explain the usage of the state feedback controller next.

3.1.1. *Augmented State Feedback Controller*

We will define an augmented state feedback controller, which includes a linear quadratic regulator (LQR) and an adaptive compensator to reach to the desirable state. The state equations without the disturbance can be given in the following form

$$p_k = p_{k-1} + (u_k + r_k)$$
$$z_k = p_k \quad (4)$$

where r_k is our augmented state that is based on tracking of the output. We will explain in the next section how r_k is defined. We would like to first design a LQR to eliminate the additional external disturbances in the system and to enable the system to come to a stable zero equilibrium in the absence of input disturbance (augmented state, r_k).

LQR aims to minimize both the input u_k and the state to be controlled p_k. By minimizing the following cost option

$$J = \sum_1^\infty p^T(k)Qp(k) + u^T(k-1)Ru(k-1) \quad , \quad R \geq 0, \ \ Q > 0$$

The solution to this minimization problem is given by [12] $u_k = -K_f p_k$ and $-K_f = R^{-1}P$ where P is the positive definite solution of the Riccati equation. Then we define the desired control rule as

$$u_k^d \triangleq u_k + r_k \triangleq -K_f(\widehat{p}_{k+1} - \widehat{p}_{k+1}^d) \quad (5)$$

where $\widehat{p}_{k+1}$ *and* $\widehat{p}_{k+1}^d$ are the estimated packet loss rate and estimated desired packet loss rate through Kalman predictor for step k+1. Next the desired input u_k^d is defined to compensate the system in order to get the desired output. We choose Q much bigger than R. Thus in our case K_f is very close to 1. We will use K_f as 1 in the rest of the calculations and u_k^d is restricted to ensure stability.

3.1.2. *Determination of input* u_k^d

A loss model to determine u_k^d is defined by assuming that the excess traffic beyond the available bandwidth is dropped. Thus, our proposed loss model is given by,

$$p_k = (\mu_k - C_k)/\mu_k \quad (6)$$

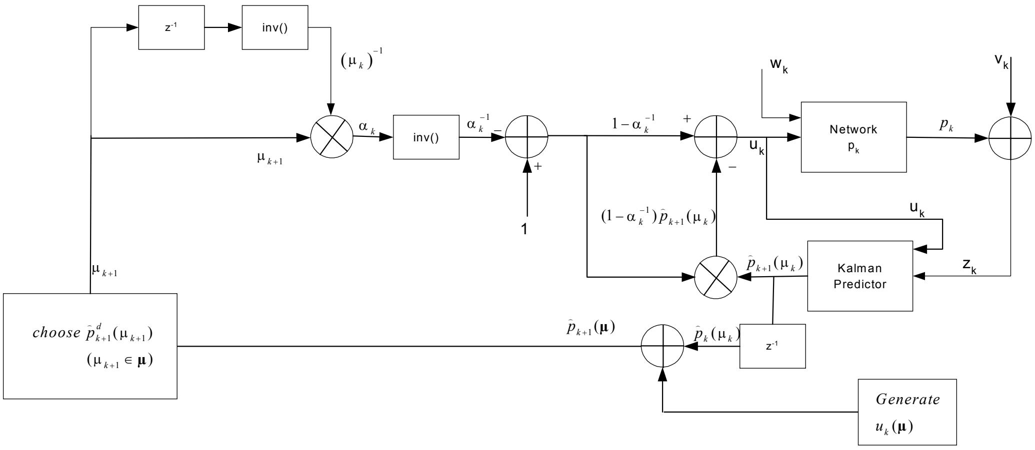

Figure 1. Packet loss control via state feedback controller.

p_k : Probability of loss for the k[th] transmitted packet.

μ_k : Transmission rate for the k[th] packet.

C_k : Available bandwidth for the k[th] packet's transmission.

We will monitor the network and estimate the available bandwidth, thus loss rate by using a Kalman predictor. We assume the relationship between the packet loss rate and the transmission rate to be linear. The linearity assumption of our model seems reasonable in the absence of further knowledge of the network since we are considering diverse network policies.

From (6) we obtain,

$$\hat{p}_{k+1}(\mu_k) = (\mu_k - \hat{C}_{k+1})/\mu_k \quad (7)$$

$$\hat{p}^d_{k+1}(\mu_{k+1}) = (\mu_{k+1} - \hat{C}_{k+1})/\mu_{k+1} \quad (8)$$

From (7) we obtain,

$$\hat{C}_{k+1} = \mu_k(1 - \hat{p}_{k+1}(\mu_k)) \quad (9)$$

If we assume that $\mu_{k+1} = \alpha_k \mu_k$ then

$$u^d_k(\mu_{k+1} \mid \mu_k) = \hat{p}^d_{k+1}(\alpha_k \mu_k) - \hat{p}_{k+1}(\mu_k)$$

$$u^d_k(\mu_{k+1} \mid \mu_k) = [(\alpha_k - 1)/\alpha_k](1 - \hat{p}_{k+1}(\mu_k)) \quad (10)$$

$$\hat{p}^d_{k+1}(\mu_{k+1}) = \hat{p}_k(\mu_k) + [(\alpha_k - 1)/\alpha_k](1 - \hat{p}_{k+1}(\mu_k)) \quad (11)$$

where $-\hat{p}_k(\mu_k)/(\hat{p}_{k+1}(\mu_k) - 1) \le (\alpha_k - 1)/\alpha_k \le 1$

so that $0 < \hat{p}^d_{k+1}(\mu_{k+1}) < 1$

Notice that α_k is adjusted dynamically at each step based on the system model's assumption of linearity.

3.2. Solution of the Optimization Problem

We would like to minimize the expected distortion while meeting the delay requirements. We obtain the packet loss rate and transmission rate pair through the regulator and minimize (1) over M packets. The number M is set to include the number of packets per GOP. Thus, the maximum delay allowed for this particular set-up is the time to encode the whole GOP. However, for stricter delay requirements M can be set to include fewer packets. Thus the maximum allowed delay can be set to a lower value as well. We relax the optimization problem by introducing a Lagrange multiplier λ in order to reduce it to an easier problem to solve. A cost function can then be written in terms of expected distortion and delay as;

$$C = \sum_{k=1}^{M} (E[D_k] + \lambda T_k(\mu_k)) \quad (12)$$

where $T_k = S_k/\mu_k$, S_k : Size of the transmitted packet.

We present the solution in the form of dynamic programming (DP) by recursively calculating the cost given by (12). Figure 2 depicts the possible transmission rates, associated costs with those transmission rates and pruned branches (shown with dashed lines). Once a solution is found it is checked to ensure that the delay criteria is also satisfied. If not, then λ is adjusted until the delay criterion is met. An optimum solution to the original problem can be obtained by appropriately choosing λ .

4. SIMULATIONS

We have performed simulations for scenes from the movie "matrix". The video has 30 frames per second and the streams are generated over different bit rates varying from 1.5Mbps to 3Mbps. The simulations are performed for two different network congestion control schemes, Linear Congestion control (LC) and CHOKe congestion control with Optimal Rate Control Algorithm (ORCA) and without ORCA. The results are also compared to the case where there is no congestion control in the network. When there is no congestion control in the network the loss does not depend on the transmission rate of the video. The first set of simulations assumed that the network operates as per our model. In other words, there is a linear relationship between the excess amount of traffic and the probability of loss.

In our second set of simulations we have used the CHOKe model as our network congestion control algorithm. The explanation of this algorithm and the resulting probability of loss is given in [8]. It should be noted that the packet loss rate is exponentially proportional to the incoming flows arrival rate in CHOKe model. We have assumed that the underlying RED algorithm only drops about 0.5% of the traffic and the traffic we generate makes up for 3% of the total traffic in the node that applies this algorithm. PSNR comparisons are shown in Figure 3. We see that using ORCA in the presence of a congestion control algorithm significantly improves the performance. Moreover, the algorithm does not degrade the quality of the video transmission when there is no congestion control in the network. The algorithm drops some of the packets, thereby reducing its average video transmission rate. Comparisons of transmission rates are shown in Figure 4. The video output rates generated by ORCA are reduced by considerable amounts in the presence of packet losses. However they are still much higher than what the TCP traffic would have allowed at these loss rates. It should be noted that however, if there were an extremely strict control algorithm in the network that drops all of the packets above

TCP rate, this algorithm would reduce its rate towards TCP-friendly rate.

5. DISCUSSIONS AND CONCLUSION

The framework presented in this paper can be extended to include adaptive bit allocation for real-time MPEG encoders, especially for videoconferencing applications. Our study only included the distortion caused during the transmission of the video. By including an adaptive MPEG encoder in the equation a joint distortion measure can be calculated and the rate may be optimized over this measure. This framework can also be extended to a content-based streaming protocol.

We have seen that by implementing an algorithm that evaluates the cost and the benefits of its transmission rate at the sender it can lead to both reducing the traffic on the network when it is necessary and minimizing the expected distortion with the available bandwidth. The loss rate is kept at the desired level by implementing an augmented state feedback controller and the complexity of these calculations is reduced by using a recursive computation to calculate the expected distortion. The source evaluates the best possible rate for the video instead of trying to reduce its transmission rate around TCP-friendly rate. At first it may appear that this algorithm does not help in reducing the network load properly. However, if there is a strict congestion control mechanism in the network to force towards a TCP-friendly rate this algorithm will reduce its rate towards TCP-Friendly rate. On the other hand, when other participants of the networks do not transmit with TCP-friendly rates reducing the transmission rate is not beneficial for the source either. As a result the source moderates its traffic generation based on the control algorithms in place to minimize the expected distortion.

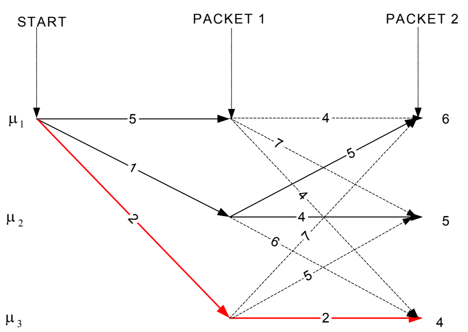

Figure 2. Illustration for DP solution

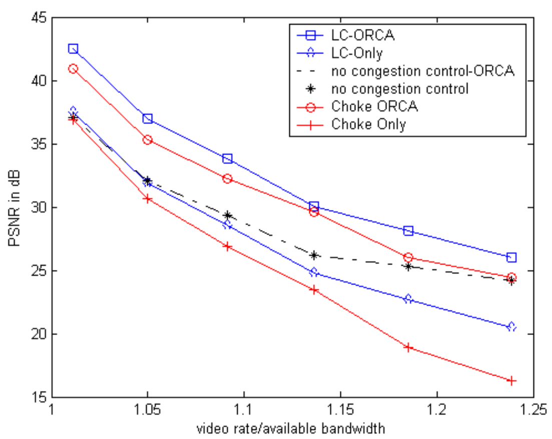

Figure 3. PSNR comparison of the received video.

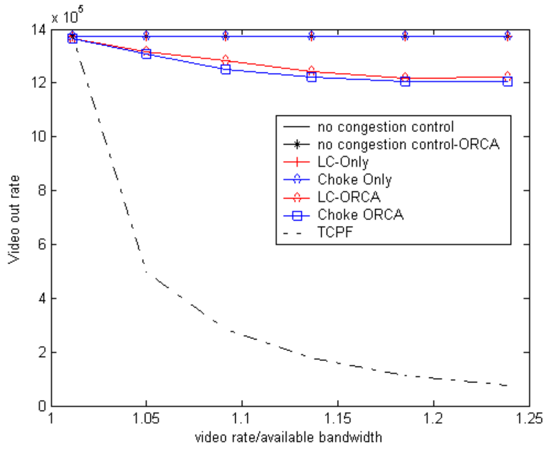

Figure 4. Transmission rates.

REFERENCES

[1] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. on Networking*, vol. 7, pp. 458-472, August 1999.

[2] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A Model Based TCP-Friendly Rate Control Protocol," *in Proceedings of NOSSDAV*, June 1999.

[3] J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-Friendly Congestion Control.," *Technical Report TR-2001-002, Universitaet Mannheim*, February 2001.

[4] R. Rejaie, M. Handley, and D. Estrin, "Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," *Proceedings. IEEE Infocom*, March 1999.

[5] D. Sisalem and A. Wolisz, "LDA+ TCP-friendly adaptation: A measurement and comparison study," *Proc. International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, June 2000.

[6] S. Floyd, M. Handley, J. Padhye, and Widmer, "Equation-based congestion control for unicast applications," *in Proc. ACM SIGCOMM*, pp. 43-56, Aug. 2000.

[7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEE/ACM Transactions on Networking*, vol. 1, August 1993.

[8] A. Tang, J. Wang, and S. H. Low., "Understanding CHOKe," *In Proc. of IEEE Infocom*, April 2003.

[9] "ISO/IEC 13818-2,Generic coding of moving pictures and associated audio information: Video, Recommendation ITU-T H.262," 1995.

[10] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, " RTP: A Transport Protocol for Real-Time Applications," *RFC 1889*, January 1996.

[11] E. Mulabegovic, D. Schonfeld, and R. Ansari, "Lightweight Streaming Protocol (LSP)," *ACM Multimedia Conference, Juan Les Pins, France*, 2002.

[12] H. Schwarz, "Optimale Regelung und Filterung," *Braunschweig, Germany: Vieweg and Sohn*, 1981.