

RATE-DISTORTION OPTIMIZED PACKET SCHEDULING OVER BOTTLENECK LINKS

Jacob Chakareski and Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute - LTS4, CH-1015 Lausanne

ABSTRACT

The loss and delay experienced by packets travelling along an Internet network path are mainly governed by the characteristics of a bottleneck link, such as available data rate and queue size. In this work, we propose a framework for rate-distortion optimized packet scheduling with adaptive rate control for media streaming over bandwidth-constrained bottleneck links. The framework computes optimal packet schedules while continuously adapting its instantaneous rate to the following three factors: the available data rate and the current queue size on the bottleneck link, and the congestion that packets transmitted under the schedules will create on the bottleneck link. Experimental results demonstrate that our framework does not lose in rate-distortion performance over rate-distortion optimized packet scheduling without strict rate control, while producing at the same time a much smoother instantaneous rate feeding the bottleneck queue. This in turn contributes to fairness to other flows sharing the bottleneck link and causes less variations in queue size, thereby avoiding queue overflow and unnecessarily long packet delays on the bottleneck link.

1. INTRODUCTION

Communication over a network bottleneck link is quite common in the Internet today. The bottleneck link is often the last hop on the network path between the sender and the receiver. The term bottleneck denotes the fact that the characteristics of this link, such as available transmission rate and queue size, are orders of magnitude smaller than those on the other links along the network path. Therefore, the delay and loss experienced by packets travelling on the network path between the sender and the receiver are mostly governed by the characteristics of the bottleneck link. Moreover, due to the limited data rate and queue size, a packet transmitted on this link will affect the delay and loss experienced by successive packets sent on the link. This is the so called *self-congestion* effect experienced on bottleneck links.

In this paper, we consider the scenario of sender-driven streaming over a network bottleneck link. In this scenario, there is a media stream and a cross-traffic flow, representing the rest of the packets transmitted on the bottleneck link, competing for resources on the bottleneck link. In other words, media packets and cross-traffic packets arrive, possibly simultaneously, at the bottleneck link and enter a buffer queue, in order to be eventually transmitted on the bottleneck link. The queue is emptied at a constant data rate C which represents the available transmission rate offered by the outgoing link. The scenario under consideration is illustrated in Figure 1.

We propose a framework for Rate-Distortion Optimized (RaDiO) streaming over bottleneck links with adaptive rate control.

This work has been partly supported by the Swiss National Science Foundation.

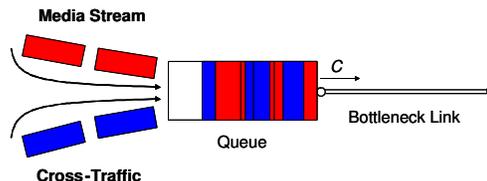


Fig. 1. Streaming over a bottleneck link.

The framework enables computing transmission schedules for media packets, that in addition to being rate-distortion optimal also maintain less variations in instantaneous data rate feeding the bottleneck queue. This is in contrast to earlier works on rate-distortion optimized streaming, such as [1–5], which ignored the effect of self-congestion that in turn arises when streaming is performed over bottleneck links.

In the present paper, we propose to characterize the communication channels in the forward and backward direction on the bottleneck link as a function of the transmission rate of media packets. Furthermore, we propose to incorporate rate-control within the optimization framework for computing packet schedules so that we can take full advantage of these new channel models. Finally, by having the optimization framework and the channel models interact, we ensure that our solutions for the optimal schedules continuously update the effect that transmitted media packets have on the expected congestion (as measured via the packet delay) over the bottleneck link, and on the expected distortion associated with the media stream.

2. CONGESTION-AWARE PACKET DELAY AND PACKET LOSS PROBABILITIES

Consider first the case when there are only cross-traffic packets entering the queue in the forward direction on the bottleneck link. If the inter arrival time and size of cross-traffic packets are exponentially distributed, then the queue can be modeled using an M/M/1 model [6]. That is, the delay that a packet experiences in the forward direction on the bottleneck link can be modeled using an exponential distribution with a mean $\mu_F = PackSize / (C_F - F_F)$, where $PackSize$ is the packet size in bits, C_F is the capacity of the queue, and F_F is the average cross-traffic rate, both in bits per second (bps). In certain scenarios there is in addition a fixed deterministic delay κ_F that each packet experiences in the forward direction. This delay is typically attributed to processing time of packets in the queue. Then, the forward-trip time (FTT) across the queue is modeled using a shifted exponential distribution with a mean $\mu_F + \kappa_F$ and a right shift κ_F .

Now, consider the case when media packets, in addition to cross-traffic packets, arrive at the queue. The average data rate of the media traffic is R_F (bps). We assume that the forward-

trip time has still a shifted exponential distribution, however with $\mu_F = \text{PacketSize}/(C_F - F_F - R_F)$. That is, the mean of the delay distribution is now a function of the data rate of media packets. This distribution allows us to account for the fact that media traffic arriving at the queue affects the delay that a packet (media or cross-traffic) sent across the link experiences, i.e., it incorporates the effect of congestion. This is illustrated in Figure 2, where a shifted exponential distribution is shown for a few example values of R_F relative to C_F and F_F .

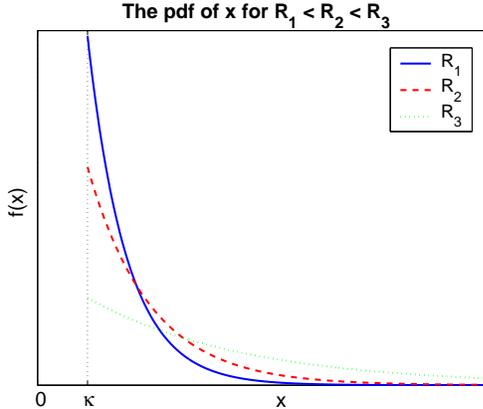


Fig. 2. Shifted exponential distribution $f(x)$ of the transmission delay x across the bottleneck link for different data rates of incoming media traffic.

It can be seen from the figure that as the data rate of media packets increases, the delay distribution becomes increasingly heavy-tailed, i.e., longer delays are more likely to be encountered when sending a packet in the forward direction on the bottleneck link. Finally, random packet loss experienced on the bottleneck link due to transmission errors can also be accounted for in the following manner. Let ϵ_F be the average packet loss rate. That is, a packet sent across the link is either lost with probability ϵ_F , or otherwise (with probability $1 - \epsilon_F$) experiences a random delay distributed according to the shifted exponential distribution described above. Hence, the probability that a packet experiences a delay longer than τ during transmission across the bottleneck link can be computed as $P\{FTT > \tau\} = \epsilon_F + (1 - \epsilon_F) \int_{\tau}^{\infty} f(t) dt$, where $f = \exp(\mu_F, \kappa_F)$ is the shifted exponential distribution.

Analogous analysis applies to the backward direction on the bottleneck link, which due to space considerations is not presented here. Finally, the forward-trip time and the backward-trip time distributions induce a distribution on the round-trip time on the bottleneck link given as $P\{RTT > \tau\} = \epsilon_R + (1 - \epsilon_R) \int_{\tau}^{\infty} f(t) dt$, where $f(t) = \exp(\mu_F, \kappa_F) * \exp(\mu_B, \kappa_B)$ is the convolution of the forward-trip time and the backward-trip time distributions, and $\epsilon_R = 1 - (1 - \epsilon_F)(1 - \epsilon_B)$ is the probability of losing a packet either in the forward or backward direction on the bottleneck link. Note that $P\{RTT > \tau\}$ is the probability that the sender does not receive an acknowledgement packet by time $t + \tau$ for a media packet transmitted at time t .

As discussed in Section 1, none of the prior works on rate-distortion optimized packet scheduling accounts for the effect of congestion through their statistical models of the communication processes employed in the optimization framework, that in turn is used to compute the optimal schedules. In our framework, in or-

der to take advantage of the models described above, we need a mechanism for controlling the transmission rate of packets under the computed schedules. Thus, in the next section, we show how to compute packet schedules that maintain a constant or nearly constant data rate of media packets entering the queue in the forward direction on the bottleneck link.

3. RATE-DISTORTION OPTIMIZATION WITH ADAPTIVE RATE-CONTROL

Suppose there are L data units in the multimedia session. Let π_l be the transmission policy for data unit $l \in \{1, \dots, L\}$ and let $\pi = (\pi_1, \dots, \pi_L)$ be the vector of transmission policies for all L data units. A transmission policy π_l is a schedule according to which the sender (a media server) sends packets over a window of transmission opportunities t_0, t_1, \dots, t_{N-1} to the receiver (a client), for data unit l prior to its delivery deadline $t_N = t_{DTS,l}$.

The transmission policy π_l consists of transmission actions a_i for $i = 0, \dots, N - 1$, where $a_i = 1$ means send a packet with data unit l at transmission opportunity t_i , and $a_i = 0$ signifies the converse. Note that the sender will only send a packet with the data unit for $a_i = 1$ if no acknowledgements arrive at the sender by t_i due to earlier transmissions of the data unit.

Any given policy vector π induces for the multimedia session an expected distortion $D(\pi)$ and a vector of expected transmission rates $R(\pi) = [R_0(\pi) R_1(\pi) \dots R_{N-1}(\pi)]$ over the window of transmission opportunities. We seek the policy vector π that minimizes the expected distortion $D(\pi)$ such that the cumulative transmission rate over the window of transmission opportunities does not create any overflow of the buffer, with available space $\bar{B}(t_0)$, and draining rate R_c (bps), i.e.,

$$\begin{aligned} \pi^* &= \arg \min_{\pi} D(\pi), \\ \text{s.t. } \sum_{j=0}^i R_j(\pi) &\leq R_c(t_{i+1} - t_0) + \bar{B}(t_0), \\ &\text{for } i = 0, \dots, N - 1 \end{aligned} \quad (1)$$

Mathematically, this constrained optimization can be reformulated as unconstrained using the method of Lagrange multipliers. That is, we seek the policy vector π that minimizes the Lagrangian $J(\pi) = D(\pi) + \sum_{i=0}^{N-1} \lambda_i \sum_{j=0}^i R_j(\pi)$ for some vector of positive Lagrange multipliers $\lambda = [\lambda_0 \lambda_1 \dots \lambda_{N-1}]$, and that thus achieves a point on the lower convex hull of the set of all achievable distortion-rate pairs $(D(\pi), R(\pi))$ ¹. By rearranging the Lagrangian multipliers we can also write $J(\pi) = D(\pi) + \sum_{i=0}^{N-1} \lambda'_i R_i(\pi)$, where $\lambda'_i = \sum_{j=i}^{N-1} \lambda_j$, for $i = 0, \dots, N - 1$.

In the following, we explain how $D(\pi)$ and $R(\pi)$ can be computed. The expected transmission rate $R_i(\pi)$ at transmission opportunity t_i , for $i = 0, \dots, N - 1$, is the sum of the expected transmission rates for each data unit $l \in \{1, \dots, L\}$:

$$R_i(\pi) = \sum_l B_l \rho(\pi_l(t_i)), \quad (2)$$

where B_l is the number of bytes in data unit l and $\rho(\pi_l(t_i))$ is the *expected cost* per byte, or the expected number of transmitted bytes per source byte under policy π_l at transmission instance t_i . The expected distortion $D(\pi)$ can be expressed in terms of the *expected error*, or the probability $\epsilon(\pi_l)$ that data unit l does not

¹Equivalently, the set of all achievable distortion-rate $(D, R_0, R_1, \dots, R_{N-1})$ $(N+1)$ -tuples.

arrive at the client on time (under policy π_l). We refer the reader to [5] for an expression for $D(\pi)$.

To solve for the policy vector π that minimizes the expected Lagrangian $J(\pi)$ we employ an iterative descent algorithm, called Iterative Sensitivity Adjustment (ISA), in which we minimize the objective function $J(\pi_1, \dots, \pi_L)$ one variable at a time while keeping the other variables constant, until convergence [1]. It can be shown that the optimal individual policies at iteration n , for $n = 1, 2, \dots$, are given by

$$\pi_l^{(n)} = \arg \min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \sum_{i=0}^{N-1} \lambda'_i B_l \rho(\pi_l(t_i)), \quad (3)$$

where $S_l^{(n)}$ can be regarded as the *sensitivity* to losing data unit l , i.e., the amount by which the expected distortion will increase if data unit l cannot be recovered at the client, given the current transmission policies for the other data units. The optimal transmission policy $\pi_l^* \in \Pi$ for data unit l minimizes the “per data unit” Lagrangian $\epsilon(\pi_l) + \sum_{i=0}^{N-1} \lambda'_{i,l} \rho(\pi_l(t_i))$, where $\lambda'_{i,l} = \lambda'_i B_l / S_l^{(n)}$.

In the following, we provide expressions for the expected error-cost for the family of transmission policies Π corresponding to sender-driven streaming. As described earlier, $\epsilon(\pi_l)$ is the probability that data unit l is not delivered on time given the transmission actions in π_l . Furthermore, the expected cost at transmission opportunity t_i is zero if $\pi_l(t_i) = 0$, and otherwise it is equal to the probability that no acknowledgements arrive at the sender by t_i due to previous transmissions of the data unit. Hence, we can write

$$\begin{aligned} \epsilon(\pi_l) &= \prod_{i:\pi_l(t_i)=1} P\{FTT > t_{DTS} - t_i\}, \quad (4) \\ \rho(\pi_l(t_i)) &= \pi_l(t_i) \prod_{j<i:\pi_l(t_j)=1} P\{RTT > t_i - t_j\}. \end{aligned}$$

Finding the appropriate choice of the Lagrange multipliers λ'_i is performed by iteratively increasing the lower bounds on the multipliers Λ'_i , such that the violation of the transmission rate constraints can be prevented, while adjusting the values of λ'_i until an optimal allocation, where none of the constraints is violated, is found. We initially select $\lambda'_i = \lambda$ for $i = 0, \dots, N-1$, for some $\lambda > 0$. Then, we repeatedly re-run the ISA optimization algorithm till convergence, while adjusting the Lagrange multipliers every time the optimization algorithm converges. The procedure is continued until we adjust accordingly all Lagrange multipliers and therefore achieve a (nearly) constant transmission rate equal to R_c over the whole window of transmission opportunities t_0, t_1, \dots, t_{N-1} . The details of the search procedure for adjusting the Lagrange multipliers can be found in [7–9].

Note that it would be sufficient to determine the optimal transmission policies π_l^* only once at time t_0 , except for the fact that the Lagrange multipliers $\lambda'_{i,l}$ in (3) may be adjusted by the ISA algorithm at each subsequent transmission opportunity t_j , for $j = 1, \dots, N-1$, due to adjusting the sensitivities $S_l^{(n)}$ in (3) in order to take into account feedback from previous transmissions of other data units. Hence, at every following t_j we repeat the optimization procedure described above and recompute the optimal transmission policies. In particular, we only need to update the entries $j, j+1, \dots, N-1$ in π_l^* , i.e., the transmission actions $a_j, a_{j+1}, \dots, a_{N-1}$. Furthermore, when the policy π_l^* is recomputed at t_j the expressions from (4) should be respectively multiplied by appropriate factors ϵ_h and $\rho_h(t_i)$ in order to account

for prospective previous transmissions of data unit l prior to t_j . Specifically, ϵ_h is the probability that none of the previous transmissions arrive at the client on time given that no acknowledgements have arrived at the sender by t_j due to these transmissions, while $\rho_h(t_i)$ is the probability that no acknowledgements due to these transmissions arrive at the sender by t_i (for $t_i > t_j$), given that no acknowledgements have arrived by t_j . Thus, we write

$$\begin{aligned} \epsilon_h &= \prod_{k<j:a_k=1} P\{FTT > t_{DTS} - t_k | RTT > t_j - t_k\} \\ \rho_h(t_i) &= \prod_{k<j:a_k=1} P\{RTT > t_i - t_k | RTT > t_j - t_k\}. \end{aligned}$$

Finally, after the vector of optimal transmission policies π is determined at every t_i , for $i = 0, \dots, N-1$, the average data rate R_F of media packets in the forward direction on the bottleneck link is updated as

$$R_F = \frac{1}{t_{i+1} - t_0} \sum_{j=0}^i R_j(\pi).$$

The corresponding data rate R_B of acknowledgements is updated in a similar fashion. These updated data rates are then used to recompute the late-loss probabilities of media packets described in Section 2, which in turn are used at the next transmission opportunity t_{i+1} by the optimization algorithm described in this section when the optimal transmission policies π_l^* are recomputed. Therefore, the optimization dynamically adjusts the transmission rate of the system based on its previous values so that it maintains an optimal equilibrium point of operation.

4. EXPERIMENTAL RESULTS

Here, we investigate the performance of two systems for streaming packetized video content over a bottleneck link. *ARC RaDiO* is the framework proposed in this paper. *Conv. RaDiO* is a conventional RaDiO streaming system [5]. The video content is a two-layer SNR scalable encoding of the image sequence *Foreman* where each of the layers is encoded at 32 Kbps using an H.263+ [10] codec. The frame rate is 10 fps and the size of the Group of Pictures (GOP) is 10 frames, consisting of an I frame followed by 9 consecutive P frames. The maximum size of the bottleneck queues in the forward and backward directions is set to 30 packets (MTU = 1500 bytes). The queues are drained at constant rates of $C_F = C_B = 96$ Kbps and employ a first-in first-out (FIFO) serving strategy. When queue overflow occurs packets are dropped from the tail of the queue. The mean packet size for the cross traffic in each direction is set to 5 Kbits. The average data rate of the cross-traffic is $F_F = F_B = 32$ Kbps. Finally, the constant processing delay of packets at the queues and the packet loss rates on the bottleneck link are both set to zero, i.e., $\kappa_F = \kappa_B = 0$ and $\epsilon_F = \epsilon_B = 0$. In the experiments we use $T = 100$ ms as the time interval between transmission opportunities and 600 ms for the play-out delay. The quantity $\bar{B}(t_0)$ from (1) is set to 1/4 of the estimate of the available buffer space. We found through our experiments that for the selected simulation parameters setting $\bar{B}(t_0)$ to 25% provides optimal balance between rate-distortion performance and queuing delay experienced on the bottleneck link.

In Figure 3 (left), we show the distortion performances of *Conv. RaDiO* and *ARC RaDiO* as a function of their corresponding data rates in the forward direction on the bottleneck link. It

can be seen from the figure that *ARC RaDiO* exhibits virtually no loss in R-D performance relative to *Conv. RaDiO* over the whole range of rates under consideration. This is very encouraging. Furthermore, we can also see from Figure 3 (left) that as the transmission rate increases beyond 60 Kbps, the distortion performance of *Conv. RaDiO* quickly degrades, while that of *ARC RaDiO* remains steadily at the optimum operating point around 58 Kbps. This is because *Conv. RaDiO* quickly saturates the queue in the forward direction when transmitting at higher rates, thereby causing unacceptably long packet delays and ultimately causing packets loss when packets need to be dropped at the queue due to overflow.

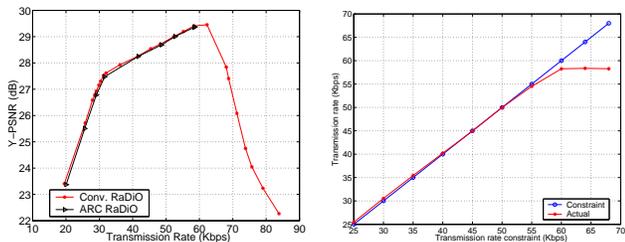


Fig. 3. Rate-distortion performance (left) and Transmission rate of *ARC RaDiO* (right) for streaming *Foreman*.

This behaviour of *ARC RaDiO* is apparent from Figure 3 (right) where the actual average transmission rate of *ARC RaDiO* is shown as a function of the rate constraint R_c . It can be seen from Figure 3 (right) that once *ARC RaDiO* achieves an average transmission rate close to 60 Kbps, it maintains it at that level, even though the rate constraint increases beyond 60 Kbps. This is achieved with the closed-loop design of the optimization algorithm for computing the optimal transmission policies which continuously updates the packet delay/loss probabilities according to its most recent instantaneous transmission rate. These in turn are then used to update the transmission policies at the next transmission instance, as explained in Section 3, thereby causing *ARC RaDiO* to be constantly aware regarding how its current transmission rate affects congestion on the bottleneck link, and ultimately its distortion performance as a function of that rate.

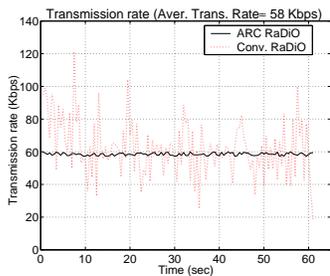


Fig. 4. Instantaneous transmission rate (Kbps).

Finally, we briefly examine the variations in transmission rate for the two systems shown in Figure 4 for the case when the average data rate is 58 Kbps. It can be seen from Figure 4 that *ARC RaDiO* performs rate control quite successfully as its transmission rate exhibits very little variations, in particular relative to the transmission rate of *Conv. RaDiO* which is quite bursty. Such a behaviour of *ARC RaDiO* can be quite beneficial as it contributes to fairness to other flows sharing the bottleneck link, and it prevents

queue overflow and unnecessarily long packet delays on the bottleneck link.

5. CONCLUSIONS

Perhaps the most important lesson of this work is that the effect of (self) congestion needs to be taken into consideration when streaming over a network bottleneck link. Specifically, when packets sent across a network link affect its own transmission delay as well as the delay of packets associated with other network flows existing on the link, it is very important to account properly for this phenomenon in order to improve streaming performance of media packets, while providing at the same time fairness to the other existing flows. The three key ingredients of the work presented in this paper are the congestion-aware packet delay/loss probabilities, the optimization algorithm for computing transmission policies with instantaneous rate control and the dynamic update of the delay/loss probabilities based on the current transmission rate of the optimization algorithm. These three allow us to virtually maintain the same rate-distortion performance when streaming over a bottleneck link relative to a *RaDiO* system that is oblivious to the effect of (self) congestion. At the same time they also provide us with the benefits of a smooth instantaneous rate over the bottleneck link, which is not present in conventional *RaDiO* streaming.

6. REFERENCES

- [1] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," Tech. Rep. MSR-TR-2001-35, Microsoft Research, Redmond, WA, Feb. 2001.
- [2] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *Proc. Int'l Packet Video Workshop*, Pittsburg, PA, Apr. 2002.
- [3] J. Chakareski and P.A. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002, IEEE, vol. 3, pp. 2513–2516.
- [4] 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, IEEE, vol. 1, pp. 869–872.
- [5] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conference*, Snowbird, UT, Mar. 2003, IEEE Computer Society, pp. 203–212.
- [6] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, Englewood Cliffs, N.J., 2nd edition, 1992.
- [7] A. Ortega, "Optimal bit allocation under multiple rate constraints," in *Proc. Data Compression Conference*, Snowbird, UT, Apr. 1996, IEEE Computer Society, pp. 349–358.
- [8] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer Academic Publishers, Mar. 1997.
- [9] C.-Y. Hsu, A. Ortega, and M. Khansar, "Rate control for robust video transmission over burst-error wireless channels," *IEEE Journal on Selec. Areas in Comm.*, vol. 17, no. 5, pp. 756–773, May 1999.
- [10] Standardization Sector of ITU, "Video coding for low bitrate communication," *ITU-T Recommendation H.263 Ver. 2*, Feb. 1998.