COST-DISTORTION OPTIMIZED STREAMING MEDIA OVER DIFFSERV NETWORKS

Anshul Sehgal

University of Illinois
asehgal@uiuc.edu

Philip A. Chou

Microsoft Research
pachou@microsoft.com

ABSTRACT

In this paper, we address the problem of multimedia streaming over DiffServ networks that offer various qualities of service (QoS) at different price points. In particular, we seek cost-distortion optimized transmission policies that minimize the distortion seen by the client under a cost constraint. Our results show that for on-demand streaming of stored media, DiffServ networks provide little or no gain over networks that offers only a single, cost-effective, QoS. However, for real-time conversational communication or multicast streaming, optimized transmission over DiffServ networks can perform better than optimized transmission over any single QoS by over 2 dB at the same cost.

1. INTRODUCTION

The current day Internet does not provide any guarantee on the timely delivery of packets inserted into the network. A packet inserted into the network may be lost, dropped or delayed by a random amount. Nonetheless, the current day Internet works well for "elastic" applications such as file transfer, web browsing and electronic mail. On the other hand, applications such as media streaming (video conferencing, audio / video multicasting, on-demand streaming etc.) have strict delivery deadlines for data units. Data units that arrive at their destination after their deadlines must be discarded. This apparent conflict between the requirements for multimedia transmission and the best-effort service offered by the current Internet has caused many people to recommend augmenting the current Internet with support for additional qualities of service (QoS).

Broadly, two approaches have been suggested for supporting QoS in the Internet: Integrated Services (*IntServ*) and Differentiated Services (*DiffServ*). IntServ supports QoS by reserving resources for individual flows in the network. The main disadvantage of IntServ is that it does not scale well to large networks with thousands of reserved flows, since each router must maintain per-flow state information. DiffServ, on the other hand, supports QoS by the use of multiple service classes. The sender assigns a priority tag to each packet, indicating the QoS class to which the packet belongs. A packet arriving at a router is queued and routed based on the assigned class. Typically, a packet assigned to a high quality-of-service class has a smaller probability of being dropped or delayed at a router than a packet assigned to a low quality-of-service class. These per-hop behaviors induce an end-to-end statistical differentiation between QoS classes.

Maintaining the statistical differentiation between QoS classes in DiffServ depends on limiting the traffic in certain classes or apportioning the traffic between classes, to avoid overloading the higher priority queues in the routers. It is envisioned that such limits or proportions should be specified by service level agreements (SLAs) between the service providers and their customers. An SLA would typically specify for each service level either a hard bit rate constraint (e.g., data transmitted in QoS class X shall not exceed R bits per period T) or a soft, price-based incentive (e.g., data transmitted in QoS class X will cost c cents per megabyte). The latter can be regarded as an unconstrained, Lagrangian version of the former. By adjusting the transmission costs c (the Lagrange multipliers) across QoS classes, users can be induced to transmit in any QoS class at any particular bit rate R. However, this may require averaging over a large period T, over a large number of customers, or over a large amount of data.

In this paper, we examine streaming over a DiffServ network in which the qualities of service are priced differentially, and the sender is charged for each byte that it transmits. No charge is assessed for the bytes that it receives. We assume that the prices are static and are known to the sender a priori. We employ the optimization framework introduced in [1, 2], and seek cost-distortion optimized transmission policies that minimize the distortion seen by the client under a cost constraint. Simulation results demonstrate that if the delay is sufficiently large and retransmissions are allowed, as would be the case for on-demand streaming of stored media, then a cost-distortion (CD) optimized system sends almost all of its traffic over the single most cost-effective QoS available, and hence DiffServ networks provide little or no gain over networks that offer only a single, cost-effective, QoS. However, if the delay is small or retransmissions are not allowed, as would be the case in real-time communication or multicast streaming scenarios, then CD-optimized streaming over DiffServ networks can perform better than optimized streaming over any single QoS by over 2 dB in our experiments.

Intuitively, sending packets over DiffServ is analogous to sending mail using either standard or priority mail delivery services. While the priority mail service is fast and reliable, it comes at a high price. Conversely, the standard mail service is inexpensive, but is slow and mail is liable to be lost. Consequently, when the sender does not have access to feedback from the recipient (analogous to real-time or multicast streaming), she sends the more important documents using the priority service and the less important documents using the standard service. In this way, she is able to better utilize her resources as compared to the case where she has access to either (but only one) of the two services. On the other hand, if the sender does have access to feedback from the recipient, and many round trip delays can be tolerated (analogous to on-demand streaming), then by using appropriate retransmissions and acknowledgements, the sender can be assured of eventually getting a copy of any particular document through to the recipient, even though it may require on average several transmissions per document. In this case, the sender should use only the mail service that is most cost-effective in the sense of offering the lowest average cost per document reliably delivered, or equivalently, the highest throughput per unit cost. Of course, this depends on having an efficient communication protocol between sender and recipient.

Past work on multimedia transmission over DiffServ networks includes [3, 4, 5, 6]. Like our work, these works investigate partitioning the individual units of data in a multimedia stream across different qualities of service, depending on the importance of the data units. However, none of these works are cost-distortion optimized. De Martin et al. [3, 4] show a 2+ dB improvement on speech and video in a low-delay telephony setting, by transmitting only a fraction of the stream across a premium channel. Shin et al. [5, 6] also show a 2+ dB improvement, *in a video on-demand setting*, in conflict with our findings. However, we believe that the systems studied in [5, 6] are suboptimal in that they do not take advantage of the feedback available in the on-demand setting.

2. MODEL DESCRIPTION

In this section, we define our abstraction of the encoding, packetization and communication processes. For further details, the reader is referred to [1, 2], where our framework for the DiffServ problem was introduced.

In a media delivery system, data are encoded and packetized into data units to be transmitted to the client. Depending on the algorithm used for encoding, data units have dependencies between them that can be represented by an acyclic directed graph. Data unit l is said to be dependent on data unit l' if l cannot be decoded without first decoding l'.

Associated with each data unit l is a quantity ΔD_l, which denotes the decrease in distortion if data unit l is decoded on time, and a quantity B_l, which denotes the size of the data unit in bytes. Also associated with each data unit is a decoding time stamp $t_{DTS,l}$, which is the time by which the data unit must be available at the decoder in order to be decoded and played back.

We model the DiffServ scenario as follows: The server can transmit each data unit to the client over one of two channels. If the server transmits a data unit over channel 1, it incurs a cost $c^{(1)}$ per byte transmitted; if it chooses to transmit over channel 2, it incurs a cost $c^{(2)}$ per byte transmitted. The transmitted data unit may either be lost, or reach the client after a random delay, where the loss probability and delay distribution depend on the channel. Every data unit l that arrives at the client on time reduces the distortion of the presentation by ΔD_l, provided that all of the data units l' on which l depends have also arrived at the client. If the client receives the data unit, it acknowledges receipt of the data unit by transmitting an acknowledgment packet at no cost over the same channel on which it received the data unit, and additionally the acknowledgment packet may be lost or delayed. If the server does not receive the acknowledgment in a timely fashion, it may re-transmit the data unit. This process may be repeated until either the server receives an acknowledgment for the data unit, or the delivery deadline of the data unit is reached.

Specifically, the two network paths from the server to the client (the forward channels) and the two network paths from the client to the server (the backward channels) are modeled as independent time-invariant packet erasure channels with random delays. Each packet inserted into either of the channels is independently lost with probability $\epsilon_F^{(k)}$ for forward channel k and $\epsilon_B^{(k)}$ for backward channel k, $k = 1, 2$. If it is not lost, then the packet is delayed by a random forward trip time $FTT^{(k)}$ or backward trip time $BTT^{(k)}$, which are respectively drawn from probability densities $p_F^{(k)}$ and $p_B^{(k)}$, $k = 1, 2$. In our simulations, we use for $p_F^{(k)}$ and $p_B^{(k)}$ shifted Gamma distributions with parameters $(n_F^{(k)}, \alpha^{(k)})$ and $(n_B^{(k)}, \alpha^{(k)})$ and right shifts $\kappa_F^{(k)}$ and $\kappa_B^{(k)}$, respectively. These induce a distribution on the round trip time $RTT^{(k)} = FTT^{(k)} + BTT^{(k)}$, which is a shifted Gamma distribution $p_R^{(k)}$ with parameters $(n_F^{(k)} + n_B^{(k)}, \alpha^{(k)})$ and right shift $\kappa_F^{(k)} + \kappa_B^{(k)}$. Also induced is the round trip loss probability $\epsilon_R^{(k)} = \epsilon_F^{(k)} + (1 - \epsilon_F^{(k)})\epsilon_B^{(k)}$. We define $P\{FTT^{(k)} > \tau)\} = \epsilon_F^{(k)} + (1 - \epsilon_F^{(k)}) \int_\tau^\infty p_F^{(k)}(t)dt$ to be the probability that a packet transmitted by the server to the client at time t on channel k does not reach the client by time $t + \tau$, whether the packet is lost or simply delay by more than τ.

We consider two scenarios, on-demand streaming of stored media and real-time/multicast streaming. For on-demand streaming, delay requirements are such that the server can use feedback from the client to adaptively retransmit data units. The multicast and real-time streaming scenarios are slightly different; in the multicast scenario, it is infeasible for the server to re-transmit data units to individual clients based on their feedback. In the real-time (conversational) scenario, delay requirements are such that there is no time for retransmissions. Thus, in the real-time/multicast streaming scenario, retransmissions cannot be used to ensure near-reliable delivery.

3. COST-DISTORTION OPTIMIZATION

We assume that communication of each data unit l can be achieved with a policy π_l selected from a family of policies Π. The family Π is determined by the scenario under consideration. In the scenario of real-time/multicast streaming over DiffServ, Π corresponds to the set of service levels. In this case, each data unit l is communicated with QoS π_l. In the scenario of on-demand streaming over DiffServ, Π corresponds to a set of schedules for transmitting a data unit over each QoS, repeatedly if necessary, until an acknowledgment is received. These policies are examined carefully in the next section. In this section, however, it suffices to keep Π and π_l abstract.

Suppose there are L data units in the multimedia session. Let π_l be the transmission policy for data unit $l \in \{1, \ldots, L\}$ and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ be the vector of transmission policies for all L data units. Any given policy vector $\boldsymbol{\pi}$ induces an expected distortion $D(\boldsymbol{\pi})$ and an expected transmission cost $C(\boldsymbol{\pi})$ for the multimedia session. We seek a policy vector $\boldsymbol{\pi}$ that minimizes $D(\boldsymbol{\pi})$ subject to a constraint on $C(\boldsymbol{\pi})$. This can be achieved by minimizing the Lagrangian $D(\boldsymbol{\pi}) + \lambda C(\boldsymbol{\pi})$ for some Lagrange multiplier $\lambda > 0$, thereby finding a point on the lower convex hull of the set of all achievable distortion-cost pairs.

The expected transmission cost $C(\boldsymbol{\pi})$ is the sum of the expected transmission costs for each data unit $l \in \{1, \ldots, L\}$,

$$C(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l), \tag{1}$$

where B_l is the number of bytes in data unit l and $\rho(\pi_l)$ is the *expected cost* per byte under policy π_l. The expected distortion $D(\boldsymbol{\pi})$ is somewhat more complicated to express, but it can be expressed in terms of the *expected error*, or the probability $\epsilon(\pi_l)$ for $l \in \{1, \ldots, L\}$ that data unit l does not arrive at the receiver on time under policy π_l. Specifically, let I_l be the indicator random variable that is 1 if data unit l arrives at the receiver on time, and is 0 otherwise. Then $\prod_{l' \preceq l} I_{l'}$ is 1 if data unit l is decodable by the receiver on time, and is 0 otherwise. Here, $l' \preceq l$ means that l depends directly or indirectly on l'. If data unit l is decodable by the

receiver on time, then the reconstruction error is reduced by the quantity ΔD_l; otherwise the reconstruction error is not reduced. Hence the total reduction in reconstruction error for the presentation is $\sum_l \Delta D_l \prod_{l' \preceq l} I_{l'}$. Subtracting this quantity from the reconstruction error for the presentation if no data units are received, and taking expectations, we have for the expected distortion

$$D(\boldsymbol{\pi}) = d_0 - \sum_l \Delta d_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})), \tag{2}$$

where d_0 is the expected reconstruction error for the presentation if no data units are received and Δd_l is the expected reduction in reconstruction error if data unit l is decoded on time. Here we have used the assumption that the data packet transmission processes are independent, and are independent of the source process, in order to factor the expectation.

With expressions (1) and (2) for the expected cost and expected distortion for any given policy vector now in hand, we are able to find the policy vector $\boldsymbol{\pi}$ that minimizes the expected Lagrangian $J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda C(\boldsymbol{\pi})$. However, this minimization is complicated by the fact that the terms involving π_l are not independent. We employ an iterative descent algorithm, called the iterative sensitivity adjustment (ISA) algorithm, in which we minimize the objective function $J(\boldsymbol{\pi})$ one component at a time until convergence. Let $\boldsymbol{\pi}^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, \ldots, \pi_L^{(0)})$ be any initial policy vector and let $\boldsymbol{\pi}^{(n-1)}$ the policy vector at iteration $n-1$. At iteration n, select one component $l_n = (n \bmod L)$ to optimize. Then for $l \neq l_n$, let $\pi_l^{(n)} = \pi_l^{(n-1)}$, while for $l = l_n$, let $\pi_l^{(n)} = \arg\min_{\pi_l} J(\pi_1^{(n)}, \ldots, \pi_{l-1}^{(n)}, \pi_l, \pi_{l+1}^{(n)}, \ldots, \pi_L^{(n)})$, or

$$\pi_l^{(n)} = \arg\min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \lambda B_l \rho(\pi_l), \tag{3}$$

where $S_l^{(n)} = \sum_{l' \succeq l} \Delta d_{l'} \prod_{l'' \preceq l' : l'' \neq l} (1 - \epsilon(\pi_{l''}^{(n)}))$ is the *sensitivity* to losing data unit l.

The minimization (3) is now simple, since each data unit l can be considered in isolation. Indeed the problem reduces to minimizing the "per data unit" Lagrangian $\epsilon(\pi_l) + \lambda' \rho(\pi_l)$, where $\lambda' = \lambda B_l / S_l^{(n)}$. Thus, it suffices to know the lower convex hull of the set of points $\{(\epsilon(\pi), \rho(\pi)) : \pi \in \Pi\}$, which we call the *error-cost* function. The next section examines the error-cost function for the scenarios of on-demand and real-time/multicast streaming over DiffServ.

4. ERROR-COST FUNCTION FOR A DATA UNIT

We assume that for each data unit there is a discrete set of N transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to the delivery deadline t_{DTS} at which the data unit may be put into a packet and transmitted using a selected QoS channel. We identify a transmission policy for the data unit with an N-tuple $(a_0, a_1, \ldots, a_{N-1})$, where $a_i = k$ if the data unit is scheduled for transmission at time t_i using QoS channel $k \in \{1, 2\}$, and $a_i = 0$ if the data unit is not scheduled for transmission at time t_i across either channel. The sender transmits the data unit according to the sequence $a_0, a_1, \ldots, a_{N-1}$, until an acknowledgement is received or until the sequence ends. In real-time communication (telephony or conferencing) or multicast streaming, there is only a single transmission opportunity ($N = 1$) at time t_0, because there is no time or opportunity for feedback. In on-demand streaming, however, there

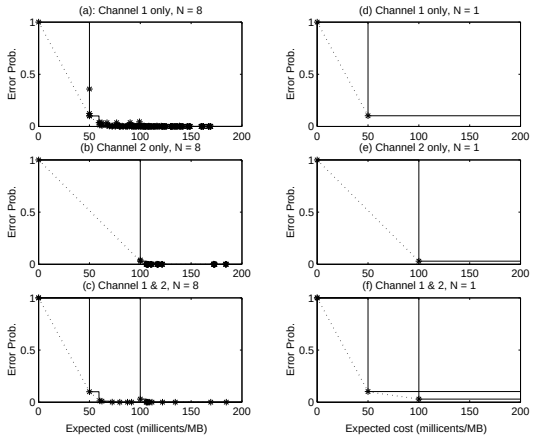

Fig. 1. Error-Cost functions.

may be many opportunities for transmission and feedback before the data unit's delivery deadline t_{DTS}.

We now evaluate the expected error and the expected cost. Under policy $\pi = (a_0, a_1, \ldots, a_{N-1})$ for transmitting a data unit, the expected cost is the expected sum of the transmission costs $c^{(a_i)}$ incurred at each time t_i, for $a_i = 1$ or $a_i = 2$, before an acknowledgment is received. Since the probability that an acknowledgement is not received by time t_i is $\prod_{j<i: a_j \neq 0} P\{RTT^{(a_j)} > t_i - t_j\}$, the expected cost is

$$\rho(\pi) = \sum_{i: a_i \neq 0} c^{(a_i)} \prod_{j<i: a_j \neq 0} P\{RTT^{(a_j)} > t_i - t_j\}.$$

Similarly, the expected error is the probability that none of the packets containing the data unit arrive at the client by t_{DTS}, or

$$\epsilon(\pi) = \prod_{i: a_i \neq 0} P\{FTT^{(a_i)} > t_{DTS} - t_i\}.$$

The cost-distortion points $(\rho(\pi), \epsilon(\pi))$ can be enumerated for all 3^N possible policies π, and their lower convex hull can be computed. For comparison, this can also be done for all 2^N possible policies where transmission is restricted to QoS channel 1 only ($a_i \in \{0, 1\}$), or QoS channel 2 only ($a_i \in \{0, 2\}$). This is done in Figure 1. Figures 1a–c respectively show error-cost functions for QoS channel 1 only, channel 2 only, and both channels 1 & 2, in the case of on-demand streaming ($N = 8$). Correspondingly, Figures 1d–f show error-cost functions in the case of real-time communication or multicast streaming ($N = 1$). The channel parameters are chosen as in the next section. Note that in the case of on-demand streaming (Figures 1a–c), the error-cost function for the combination of channels 1 & 2 is essentially the same as the error-cost function for channel 1, which is the more cost-effective of the two channels (over three times the loss rate, but half the price). Thus, if cost-distortion optimized streaming is employed, the server will almost always choose to transmit on channel 1. On the other hand, in the case of real-time/multicast streaming (Figures 1d–f), the error-cost function for the combination of channels 1 & 2 is strictly better than the error-cost functions for either channels 1 or 2 alone. Thus, if cost-distortion optimized streaming is employed, one can anticipate lower distortion delivered to the user under the same cost constraint. In the next section, we show that the improvement can be over 2 dB.

5. EXPERIMENTAL RESULTS

In this section, we report our experimental results for the two communication scenarios, namely, on-demand streaming and real-time or multicast streaming. In each case we use one minute of audio (Sarah McLachlan's *Building a Mystery*) for the evaluations. The audio is compressed with a scalable version of the Windows Media Audio codec, producing a group of twelve 500-byte sequentially-dependent data units every 0.75 seconds for a maximum data rate of 64 Kbps. The twelve data units in the mth group all have $t_{DTS,l} = 0.75m$ and $B_l = 500$, but their ΔD_ls generally decrease according to importance. Two QoS channels are available. For channel 1 (the cheaper channel), $c^{(1)} = 50$ millicents/MB, $\epsilon_F^{(1)} = 0.10$, $n_F^{(1)} = 2$, $1/\alpha_F^{(1)} = 25$ ms, and $\kappa_F^{(1)} = 50$ ms, for a mean RTT of 200 ms. For channel 2 (the more costly channel), $c^{(2)} = 100$ millicents/MB, $\epsilon_F^{(2)} = 0.03$, $n_F^{(2)} = 2$, $1/\alpha_F^{(2)} = 20$ ms, and $\kappa_F^{(2)} = 20$ ms, for a mean RTT of 120 ms. The backward parameters are identical to the forward parameters. A playback delay of $\delta = 750$ ms is used for all simulations. For the on-demand simulations, each data unit has $N = 8$ transmission opportunities spaced by $T = 100$ ms, beginning NT ms before the delivery deadline. For the real-time/multicast simulations, each data unit has only $N = 1$ transmission opportunity, scheduled 1.5 s before the delivery deadline. Transmitted packets are dropped at random and those not dropped receive a random delay according to a shifted Gamma distribution, with the appropriate parameters. Results are averaged over multiple runs to smooth out the effect of any one particular channel realization.

We compare the performance of cost-distortion optimized streaming over channel 1 only and channel 2 only, referred to as System 1 and System 2, respectively, with cost-distortion optimized streaming over DiffServ, referred to as System 3. Figure 2 compares the performance of the systems in the on-demand scenario, in terms of SNR versus the cost of transmission. The plot shows that System 3 has essentially the same performance as System 1, as expected from the discussion in the previous section.

Figure 3 compares the performance of the systems in the real-time/multicast scenario. For this scenario, we set channel 2's price to $c^{(2)} = 400$ millicents/MB and its forward and backward loss probabilities to $\epsilon_F^{(2)} = \epsilon_B^{(2)} = 0.001$, making Channel 2 very expensive but reliable. The plot shows that cost-distortion optimized streaming over DiffServ outperforms cost-distortion optimized streaming over either of the two channels individually by approximately 2dB, by judiciously choosing the channel for each packet.

6. CONCLUSIONS

We study cost-distortion optimized streaming over networks with multiple QoS classes. We show that if cost-distortion optimization is used for on-demand streaming of stored media, then the ability to use multiple qualities of service offers essentially no advantage over a single QoS having the highest cost-effectiveness (throughput per unit cost). On the other hand, multiple QoS networks can be worthwhile for real-time or multicast communication scenarios in which delay or feedback constraints make it infeasible to communicate reliably.

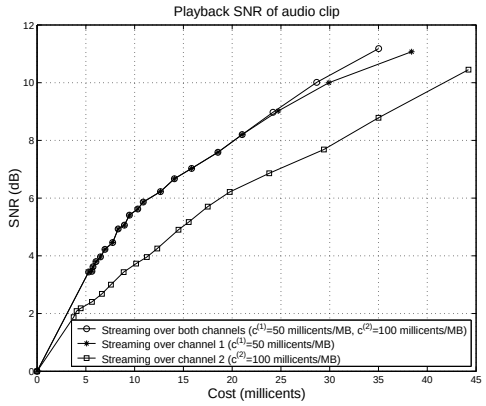

Fig. 2. Performance of CD optimized on-demand streaming.

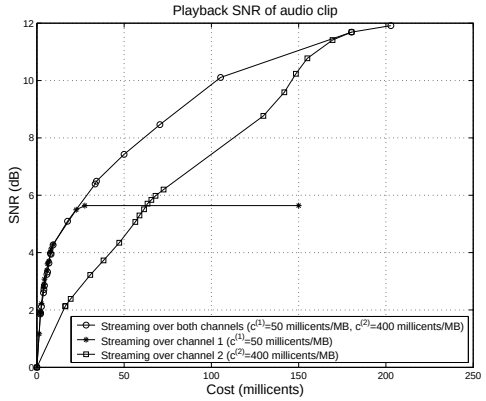

Fig. 3. Performance of CD optimized real-time streaming.

7. REFERENCES

[1] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. Technical Report MSR-TR-2001-35, Microsoft Research, Redmond, WA, February 2001.

[2] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. *IEEE Trans. Multimedia*, 2001. Submitted.

[3] J. C. de Martin. Source-driven packet marking for speech transmission over differentiated-services networks. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001. IEEE.

[4] E. Masala, D. Quaglia, and J. C. de Martin. Adaptive picture slicing for distortion-based classification of video packets. In *Proc. Workshop on Multimedia Signal Processing*, Cannes, France, October 2001. IEEE.

[5] J. Shin, J. Kim, and C.-C. J. Kuo. Relative priority based QoS interaction between video applications and differentiated service networks. In *Proc. Int'l Conf. Image Processing*, Vancouver, Canada, October 2000. IEEE.

[6] J. Shin, J. Kim, and C.-C. J. Kuo. Quality-of-service mapping mechanism for packet video in differentiated services network. *IEEE Trans. Multimedia*, 3(2):219–231, June 2001.