A Network-assisted Scheme for Bandwidth Allocation Among Video Streams

Mythili Suryanarayana Prabhu, Xiaqing Zhu, Rong Pan, Vijay Subramanian and Flavio Bonomi
Advanced Architecture & Research Group, Cisco Systems Inc., San Jose, CA 95134, U.S.A.
{mysuryan, xiaoqzhu, ropan, vijaynsu, flavio}@cisco.com

Abstract

We present a network-assisted scheme for media-aware bandwidth sharing among multiple video streaming sessions. Departing from the conventional paradigm of fair-rate allocation among data traffic flows, our scheme allocates the bottleneck bandwidth among the video streams according to their rate-distortion (R-D) characteristics, with the objective of minimizing the total video distortion of all streams. This demonstration shows that our scheme achieves the optimal rate allocation with fast convergence, efficient bottleneck utilization, and balanced video qualities among the competing streams.

1 Introduction

As video traffic increases in the Internet and competes for limited bandwidth resources, it is important to design bandwidth sharing schemes that account for video characteristics, beyond the traditional paradigm of fair-rate allocation among data flows. Ideally, when multiple video streams compete for a shared network bandwidth, it is desirable that their allocated rates reflect their respective rate-distortion (R-D) characteristics. For instance, a video stream containing scenes from a dynamic action movie needs a higher rate than a stream containing static head-and-shoulder news clips to achieve the same visual quality.

We present a novel scheme for media-aware bandwidth sharing among multiple video streaming sessions. The proposed scheme leverages explicit congestion feedback from the network to aid the sender of each stream in calculating its optimal rate allocation for each video stream. Our demonstration shows that the proposed scheme is simple to implement in a real system, and achieves desirable visual qualities for each video streaming session.

2 Scheme Description

The goal of our media-aware bandwidth sharing algorithm is to minimize the total distortion of all streams, while achieving a target utilization at the bottleneck link. This can be formally expressed as:

\[
\min_{\{r_i\}} \sum_i d_i(r_i) \tag{1}
\]

s.t. \(y_l = \sum_{i: t \in i} \leq \gamma c_l\) \tag{2}

where \(d_i(r_i)\) denotes the R-D tradeoff function of a video stream \(i\). The capacity of the bottleneck is denoted as \(c_l\); the total rate over the link is \(y_l\), comprising of the rates of all video streams traversing that link; the target utilization \(\gamma\) is chosen to be slightly less than unity.

Figure 1 provides an overview of our network-assisted media-aware bandwidth sharing scheme. Each video packet carries a header field into which a network node can insert its congestion information, virtual congestion level. This field is initialized to zero at the sender and can be modified by any network node if its congestion level is greater than the one already in the header. By the time a packet reaches the receiver, its header carries the maximum congestion level along the forward path. The receiver then echoes this information back to the sender in the video acknowledgement (ACK) packet header. The congestion information is then used to calculate the optimal rate based on the video R-D parameters. Note that the intermediate network nodes are oblivious of the video R-D information while each video sender only needs the end-to-end maximum virtual congestion level along the path of its stream.
Details of algorithms steps are described in [1], where we also prove the optimality and stability of the system.

3 Demonstration

We implement the network node as a Click Router module [2] hosted in a Linux environment. It continuously estimates the traversing traffic rate, and periodically updates the virtual congestion level. Upon relaying each video packet, it stamps the updated virtual congestion level into the an RTP extension header field, if its value is greater than the existing value in the header field.

The video sender and receiver are implemented as application agents in Linux. The receiver extracts the value of maximum virtual congestion level along the path, and reports such information in the header of an acknowledgment packet. The sender then adjust the outgoing rate of the video stream accordingly, by dynamically tuning the interval of adjacent packet transmissions. For the purpose of rate adaptation, each video sequence is pre-encoded into multiple quality versions using x264 [3], a fast implementation of the H.264/AVC standard [4].

In our demonstration scenario, two different streams, City and Mother & Daughter, share a bottleneck link with a capacity of 2.5Mbps. The target utilization is chosen at 95%. City starts streaming at time $t = 0$ second and Mother & Daughter starts at time $t = 18$ second. Figure 3 shows the recorded traces of allocated rates at the senders, as well as traces of the virtual congestion level at the network node. Initially when only the City sequence is present in the network, its allocated rate stabilizes at 2.35 Mbps; with the virtual congestion level stabilizing at a corresponding value. Immediately after the Mother & Daughter stream enters the network, the virtual congestion level at the network node quickly increases to a new equilibrium, leading to decreased allocated rate for City and increased allocation for the new stream. Figure 4 shows the visual quality of both sequences, before and after the allocation has converged. It can be noted that while the rate reduction of 0.5 Mbps from City introduced negligible reduction in its visual quality, it greatly improves the quality of the new Mother & Daughter stream.

4 Conclusions

We have demonstrated a network-assisted, media-aware sharing scheme for video streaming. Our results show that the bottleneck bandwidth is shared among competing streams in a media-aware fashion, achieving minimum total video distortion of all streams; and the scheme adapts to transient events with fast convergence, while avoiding steady-state queuing delays or persistent packet losses.

References


