IEEE Int. Conf. on Multimedia & Expo, Taipei, June 2004

A Two-stream Approach for Adaptive Rate Control in Multimedia Applications

Longin Jan Latecki, Tao Jin and Jaiwant Mulik
Dept. of Computer and Information Sciences, Temple University
Philadelphia, PA 19122, USA
{latecki, jmulik}@temple.edu

ABSTRACT

We propose a two-stream approach for adaptive rate control in multimedia applications. By monitoring a low-rate *monitoring stream*, we keep track of the available bandwidth of the network path and dynamically adjust the sending rate of the *traffic stream* close to the optimal rate. The proposed two-stream approach perfectly meets the requirements of the current best-effort Internet and fits well in multimedia applications. For example, there is no bandwidth overhead for the monitoring stream in peer-to-peer video conferencing, because the monitoring stream is the audio stream. We show in our experiments that both the network and the application can benefit from this approach. The proposed two-stream approach is applicable to monitor the sending rate of the traffic stream over UDP as well as over TCP.

Keywords

Adaptive rate control, TCP friendly congestion control, audio and video streaming, video conferencing, video telephone.

1. INTRODUCTION

We will present a system for continuous adaptive rate control that is useful for real time multimedia streaming, particularly video telephone over the public Internet. While no changes to existing transport protocols are required, it is possible to incorporate our system at the transport layer.

We ensure that audio stream has absolutely higher priority, and transmit video stream using only the remaining bandwidth. [5] has shown that during a multimedia session over a congested network it is, in terms of human perception, more important to maintain a continuous (minimum jitter) audio stream than a video stream. Our system makes real time transmission of video over TCP possible since the one-way delay when video stream is transmitted over TCP with our rate control is on the same level as it would be for video over UDP. We performed a large number of experiments both in the controlled environment as well as on real networks that included a dialup 28.8 kbps and cable modem connections. These experiments verify excellent performance of the proposed system.

Our main contributions are several substantial extensions and modifications of Pathload [2]. They were necessary since Pathload is designed for one-time bandwidth estimation, while we need continuous bandwidth monitoring in the interaction with the sender rate control. Following Pathload, the available bandwidth (AB) is detected at the application level by making use of the *One Way Delay (OWD) property under congestion* (e.g., OWD of stream packets will increase when the network becomes congested). On the level of OWD analysis, we propose a multiscale analysis of OWD properties and a method for detection of decreasing trend (Pathload uses only increasing trend). On the rate control level, we extend the Pathload iterative rate control algorithm to allow for instantaneous and immediate bandwidth adaptation for the traffic stream.

Our modeling framework is different from Pathload. We use the two-stream model introduced in Latecki et al. [2]. All network measurements are performed on so-called *monitoring stream*, called stream A for audio [2], which is assumed to be substantially smaller than the AB, while the main data transport stream is called *traffic stream,* called stream V for video [2]. The main control flow is that the measurements preformed on monitoring stream are used to adjust the sending rate of the traffic stream. We assume that there is at least enough end-to-end bandwidth for the UDP monitoring (audio) stream.

2. RELATED WORK

The area of streaming multimedia has been extensively researched for several years. In this section, we compare and contrast our approach with a representative selection of earlier approaches.

Jain and Dovrolis [1] propose a Self-Loading Periodic Streams (SLoPS) approach and a tool named Pathload to detect the AB of the network path based on the OWD property under congestion. Pathload sends periodic streams into the network path and detect the OWD trends at the receiver's side. The main algorithm in Pathload is the detection of congestion by looking for increasing trend in the OWDs of a stream. Two tests called PCT and PDT are used for OWD increasing trend detection (see Section 3.1).

Bansal and Balakrishnan [3] present a family of innovative TCP congestion control algorithms called binomial algorithms. These algorithms prevent a drastic reduction in transmission rate upon congestion and are designed for streaming audio and video applications. They show that there exist infinitely many deployable TCP-friendly binomial algorithms. Our approach differs from [3] in two important aspects. First, we consider a multimedia application as a whole consisting of simultaneous audio and video streams with an explicit hierarchy between them. In the event of congestion, our goal is to maintain the transmission rate of the audio stream while sacrificing the video stream. Second, we try to prevent the packet loss as an indicator of congestion in TCP, by using delay trends in the audio stream as our congestion signaling mechanism. Third, they require changes to the transport layer, while we do not.

Cen et al. [4] describe the Streaming Control Protocol (SCP) which is a TCP-like and TCP-friendly transport protocol designed to prevent the abrupt rate changes of TCP. However, the SCP does not allow inter-stream state sharing that our approach uses in order to provide a priority to the audio stream over the video stream.

3. TWO-STREAM APPROACH

The basic idea is that we use a low rate life-time *monitoring stream* to keep track of the ever-changing network, in order to find at any time the optimal sending rate (close to the end-to-end AB) for the *traffic stream*. The *monitoring stream* must be sent over UDP because the mechanism is based on the OWD properties of the stream packets, and only for UDP packets can we measure the packets OWD within the application layer. The *traffic stream* can be sent over any protocol.

We continuously detect the OWD trends in the *monitoring stream*, use it as an indication of the relationship between the current traffic rate and the AB, and adjust the transmission rate of the *traffic stream* close to the AB. By assigning the low-rate audio stream as the *monitoring stream*, we can make use of the information naturally contained in the in-band traffic without introducing any intrusive traffic. In our approach video can be sent over either UDP or TCP, since in both cases, the monitoring stream keep video below the AB preventing losses (UDP and TCP) and subsequent retransmissions (TCP).

3.1 Detecting OWD Trends in the Continuous Monitoring Stream

The OWD of a packet stream will increase when the traffic rate is above the end-to-end AB [1]. One direct outcome of this property is that there will be a short transition phase in which the OWD shows obvious increasing trend at the beginning when the network becomes congested. If we can detect this transition phase, we can respond to the congestion even before packet loss happens. Our two-stream approach for adaptive rate control is based on this observation.

OWD phases

OWDs of the stream packets show different patterns under different network conditions. We can divide the whole process into the following four types of phases based on the congestion status and OWD trends. R is transmission rate of traffic stream.

- *Increasing phase* (R > AB) The short transition period before entering the congestion. The OWDs show increasing trend.
- *Decreasing phase* (R < AB) The short transition period when recovering from congestion. The OWDs show decreasing trend.
- *Steady phase* (R < AB) The OWDs are stable in this phase.
- *Congested phase* (R > AB) The network is already congested. Even though usually more variant than in the steady phase, the OWDs in this phase are stable.

OWDs are stable in both the *steady phase* and *congested phase*, so it is hard to discriminate these two phases based on the measured OWDs, especially when we only have a small window of measurements in real time environment. Thus, it is crucial that we detect the *increasing phase* before the traffic goes into the *congested phase*.

PCT and PDT

Two complementary statistic metrics, PCT and PDT can be used to detect the OWD trends in the stream. Before calculating the PCT and PDT from K measured OWDs, we pre-process the data because they usually contain a lot of noise and outliers. The K OWDs $\{D_1, D_2, \ldots, D_k\}$ are partitioned into Γ groups, and within each group we use the median value $\hat{D}_i$ as that group's representative value.

The original *Pairwise Comparison Test* (PCT) metric of a stream as defined in [1] is only sensitive to the OWD increasing trend in the stream. Since we need to detect the decreasing phase in the stream we extend PCT as

$$S_{PCT} = \frac{\sum_{k=2}^{\Gamma} I(\hat{D}_k, \hat{D}_{k-1})}{\Gamma - 1}$$

where,

$$I(\hat{D}_k, \hat{D}_{k-1}) = \left\{ \begin{array}{ll} 1, & if\ \hat{D}_k - \hat{D}_{k-1} > \varepsilon \\ 0, & if\ \left|\hat{D}_k - \hat{D}_{k-1}\right| < \varepsilon \\ -1, & if\ \hat{D}_k - \hat{D}_{k-1} < -\varepsilon \end{array} \right\}$$

I(X) is 1 if X holds, and 0 otherwise, ε is a predefined threshold value, which is the granularity of the increasing or decreasing step.

If the OWDs are independent, the expected value of S_{PCT} is 0. If there is a strong increasing trend, S_{PCT} approaches 1. If there is a strong decreasing trend, S_{PCT} approaches -1. Due to our modification, the modified S_{PCT} can detect both increasing and decreasing trends. Moreover, by defining the threshold ε, it is more stable and noise resistant.

The *Pairwise Difference Test* (PDT) metric of a stream is defined in [1] as

$$S_{PDT} = \frac{\hat{D}_\Gamma - \hat{D}_1}{\sum_{k=2}^{\Gamma} |\hat{D}_k - \hat{D}_{k-1}|}$$

If the OWDs are independent, the expected value of S_{PDT} is 0. If there is a strong increasing trend, S_{PDT} approaches 1. If there is a strong decreasing trend, S_{PDT} approaches -1.

S_{PCT} and S_{PDT} are complementary. S_{PCT} detects the ratio of the increasing and decreasing steps, while S_{PDT} quantifies how strong the start-to-end variation is. S_{PCT} and S_{PDT} are combined to determine the current OWD phase. "Increasing phase" and "decreasing phase" are reported when either of the two metrics detect it, and the other one does not disagree. "Steady phase" is reported only when both of the two metrics indicate a "steady phase". All the other situations are reported as "ambiguous phase".

We define two thresholds L_{PCT} and U_{PCT} for S_{PCT}, and L_{PDT} and U_{PDT} for S_{PDT}. Our decision rules are summarized as following:

- If $S_{PCT} > U_{PCT}$ and $S_{PDT} > L_{PDT}$, or $S_{PDT} > U_{PDT}$ and $S_{PCT} > L_{PCT}$, "increasing phase" is reported
- If $S_{PCT} < -U_{PCT}$ and $S_{PDT} < -L_{PDT}$, or $S_{PDT} < -U_{PDT}$ and $S_{PCT} < -L_{PCT}$, "decreasing phase" is reported
- If $-L_{PCT} < S_{PCT} < L_{PCT}$, and $-L_{PDT} < S_{PDT} < L_{PDT}$, "steady phase" is reported
- "Ambiguous phase" is reported in all other situations.

The thresholds can be adjusted depending on how sensitive PCT and PDT should be. In our experiments, L_{PCT} and L_{PDT} were set to 0.25; U_{PCT} and U_{PDT} were set to 0.5.

With these two statistic metrics, we can detect the current OWD phase in the monitoring stream, and use it as an indicator of the relationship between the transmission rate and AB. Figure 1 illustrates detected PCT and PDT when the audio stream was sent at a constant rate of 20.8 kbps, the video stream over UDP at a rate of 120 Kbps from audio packet number 1000 to 2000. AB was 100 kbps, audio packets were sent every 40 ms. PCT and PDT were calculated every 32 measured OWDs in the audio stream.

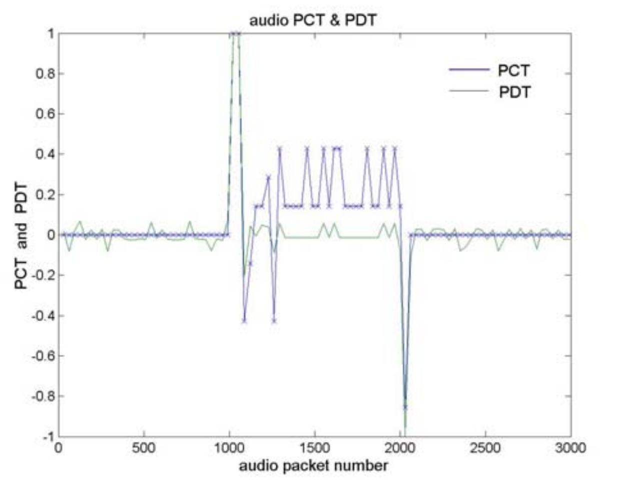

Figure 1: Graph of PCT&PDT

The *detection period*

We continuously compute the PCT and PDT for OWDs of the *monitoring stream.* The *detection period* is the interval during which the PCT and PDT are calculated periodically. It can be expressed either in number of packets or in time. A suitable length of *detection period* is critical in detecting the OWD trends in the continuous monitoring stream. In order for the PCT and PDT metrics be able to detect the OWD transition phases (increasing and decreasing phases), it is important that the detection period is within the OWD transition phase. Experiments show that the length of the OWD transition phase may vary greatly, e.g., when the traffic rate is far above the end-to-end AB, the OWD will quickly increase to the peak; when the traffic rate is slightly above the AB, the increasing phase will be longer. Experiments also show that when the traffic stream is sent over TCP, the OWDs usually only show long-term increasing (or decreasing) trends and display great short-term variance. Our experiments indicate that a short detection period works well under UDP environment, while a longer detection period is more suitable when the traffic is sent over TCP.

Multiscale and Sliding-window Measurement

Multiscale means that we calculate the PCT and PDT over multiple scales of detection periods simultaneously. We used 3-scale measurements over 32, 64 and 128 packets. The decision rules are summarized as follows

- If any of the 3-scale measurements reports "increasing phase", and the other two do not disagree (i.e., report "decreasing phase"), then "increasing phase" is reported
- If any of the 3-scale measurements reports "decreasing phase", and the other two do not disagree (i.e., report "increasing phase"), then "decreasing phase" is reported

- If at least 2 of the 3-scale measurements reports "steady phase", then "steady phase" is reported
- Otherwise "ambiguous phase" is reported

We also use sliding and overlapping windows to make the measurements more frequent without making the detection period too short.

3.2 An Interactive Adaptive Rate Control Algorithm in the Two-stream Environment

The advantage of having a lifetime *monitoring stream* is that we can detect the AB at any moment in the traffic.

The adaptive rate control mechanism is designed to be able to detect congestion and adjust the sending rate close to AB at any moment of the traffic's lifetime. The whole process of the traffic consists of two alternative stages: *rate-adaptation stage* followed by *steady stage.* The algorithm searches for the AB during the *rate-adaptation stage*. Once the sending rate converges to the optimal rate, it enters the *steady stage*. During the *steady stage*, we still continuously keep track of the network status through the *monitoring stream*. If there is any congestion detected during the *steady stage*, or we decide to try a higher rate, the traffic changes into the *rate-adaptation stage* and search for the AB again.

The algorithm used for searching the AB in the *rate-adaptation stage* is very similar to the iterative algorithm used in the Pathload [1]. We denote the sending rate at time *n* as *R(n)*, the lower and upper bounds for the AB as R_{min} and R_{max}. At time *n,* we calculate R(n+1) as:

If $R(n) > AB$, $R_{max} = R(n)$;

If $R(n) \leq AB$, $R_{min} = R(n)$;

$R(n+1) = (R_{max}+R_{min})/2$;

But we need to do the following adjustments, because our traffic is continuous while Pathload only sends short streams.

- When an "increasing phase" is reported ($R(n) > AB$), we add a *break* in traffic stream before trying the next sending rate, so that the OWD can drop to the normal level (the queues in the routers can be cleaned) before we send more traffic
- The *break* last as long as the "decreasing phase" is detected. We resume sending of the traffic stream when the decreasing phase is finished
- When "ambiguous phase" is reported for sufficiently long time, we regard it as a symptom of slight congestion, so slightly decrease the sending rate

4. CONCLUSIONS

The proposed adaptive rate control algorithm is suitable for real time multimedia streaming over UDP as well as over TCP. When video was send over TCP without our rate control the audio OWD increased by 1221 ms, while with our rate control it increased by 2 ms. Hence our approach significantly improves real time performance of TCP congestion control, so that real time video transmission over TCP is possible. The reason is that our rate control (that does not introduce any packet loss) suppresses TCP rate control, and consequently the packet loss due to the TCP increasing trend is kept very close to zero. Our experiments demonstrate that not only the performance of the video stream but also the audio stream is drastically improved. Our real network experiments indicate that a small amount of retransmissions due to sporadic packet loss (it was below 0.01%) has only minor influence on OWD. Similarly when video was sent using UDP without rate control audio loss was 4.77% but with our rate control it was 0%.

5. ACKNOWLEDGMENTS

We would like to thank Phillip Conrad for his helpful comments. We are grateful to Jay Lepreau and the support staff of Netbed (formerly known as Emulab), the Utah Network Emulation Test bed (which is primarily supported by NSF grant ANI-00-82493 and Cisco Systems) for making their facilities available for our experiments.

6. REFERENCES

[1] M. Jain and C. Dovrolis. End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. *Proc. ACM SIGCOMM*, August 2002.

[2] L. Latecki, K. Kulkarni, and J. Mulik. Better Audio Performance When Video Stream Is Monitored By TCP Congestion Control. *Proc. IEEE Multimedia and Expo*, July 2003.

[3] Deepak Bansal and Hari Balakrishnan, Binomial Congestion Control Algorithms. *Proc. IEEE INFOCOM 2001,* pp. 631-640, Volume 2, Anchorage, Alaska, April 22-24, 2001.

[4] S. Cen, C. Pu, and J. Walpole. Flow and congestion control for Internet streaming applications. *Proc. Multimedia Computing and Networking (MMCN)*, 1998.

[5] Ralf Steinmetz, Human Perception of Jitter and Media Synchronization, *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 1, January 1996, pp. 61 – 72