Towards Improved TCP-Friendly Rate Adaptation for Real-Time Applications using Adaptive Filters

Guy A. Lusilao-Zodi\textsuperscript{1}, Josephine N. Kakande\textsuperscript{1}, Mqhele E. Dlodlo\textsuperscript{1}, Gerhard de Jager\textsuperscript{1} and Keith L. Ferguson\textsuperscript{2}
Department of Electrical Engineering
University of Cape Town\textsuperscript{1}, Private Bag X5, Cape Town, South Africa
Tel: +27 21 6504853
Council for Scientific and Industrial Research\textsuperscript{2}, Pretoria, South Africa
emails: \{gylulls0001, Nakato.Kakande, Mqhele.Dlodlo, Gerhard.DeJager\}@uct.ac.za\textsuperscript{1}; kferguson@csir.co.za\textsuperscript{2}

Abstract- The emergence of multimedia applications has spurred interest in transport protocols with flexible transmission control. In TCP-Friendly Rate Control (TFRC), the multimedia transmission rate is adapted using the TCP throughput equation. This model, defined in terms of the relationship between throughput, round-trip time and loss rate, ensures that the resulting protocol is TCP compatible. However, standard TFRC wastes bandwidth because it is slow to detect changes in the available capacity. In this paper we present a new direction in tackling this problem. The classical TCP filter used in the estimation of the round-trip time is replaced by two filters namely the adaptive TCP filter and adaptive Kalman filter. Based on jitter, delay and throughput results, it is illustrated that the adaptive filters respond better to drastic changes in network round trip time, providing better aggressiveness and responsiveness in rate adaptation at the multimedia server. The Kalman filter provides the best performance, with the adaptive TCP filter also showing significant improvements over the classical TCP filter.

Index Terms— Adaptive filters, Internet, streaming video, QoS control

I. INTRODUCTION

The Internet has since its inception in the early 80’s continued to grow in scope and usage. From initial deployment primarily for data transmission, it has increasingly become a platform for transmission of real-time voice and video traffic. The latter applications are however characterized by stringent delay and bandwidth requirements, which demands are not traditionally catered for by the heterogeneous and best-effort nature of the Internet. Thus schemes such as rate adaptation were devised to allow for fair transmission of bandwidth-hungry multimedia traffic over this ‘no guarantees’ platform. Rate adaptation operates by modifying the sending rate at the sender as a function of packet loss.

In this work modifications are made to the classical estimation of round-trip time (RTT), the accuracy of which is a key factor in achieving timely change of transmission rate during a TCP-friendly real-time streaming session. TCP-Friendly rate control (TFRC) operation wastes bandwidth because it is slow to probe for extra bandwidth by increasing its rate (aggressiveness) and does not react fast enough to increased congestion by reducing its sending rate (responsiveness) [1]. Through simulations it is observed that the slow aggressiveness and responsiveness of the TFRC is a result of the low reaction speed of the classical TCP-Filter.

To address the problem, it is proposed to replace the classical TCP filter with more accurate filters. Two filters were used; an existing adaptive Kalman filter with a change detection scheme [2] and a novel adaptive TCP filter derived from Gustafsson’s work [3] that provides a more accurate estimation during drastic changes of a streaming session than the normal TCP filter.

While some studies have been carried out on using adaptive filters to improve RTT estimation for TCP traffic [2, 4], this study emphasizes the aspect of improving rate adaptation when multimedia is being streamed. Use is made of the Network Simulator version 2 (NS2) implementation enhanced with Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) features plus the added functionality of TCP-friendly video transmission [5]. Through comparison with the jitter, delay and throughput performance when using the traditional TCP estimator, it is demonstrated that adaptive-filters enhance the aggressiveness and responsiveness of TCP-friendly rate control when network conditions change drastically, with the adaptive Kalman filter realizing the best results.

The major contribution of this paper is the introduction of an adaptive TCP filter and illustration of the performance improvement obtained by using the adaptive Kalman and adaptive TCP filters for rate control.

The paper outline is as described below. Section II contains an overview of multimedia streaming and rate adaptation in particular. In section III, the basic concepts of the adaptive filters under investigation are presented. In section IV the simulation setup and results are described, followed by the conclusion in section V.

II. REAL-TIME STREAMING AND RATE ADAPTATION

A. TCP-Friendly behaviour

Multimedia sessions encompass video and audio
transmission. While TCP has attributes that negatively affect the quality of a streaming session such as retransmissions and its congestion control mechanisms which are instigated by packet losses, UDP is also not entirely adequately fashioned for real-time applications either. Given the lack of built-in congestion and flow control mechanisms within UDP and with due consideration that over 90% of Internet traffic is TCP-based [6], it has become of importance to regulate the flow of UDP multimedia traffic so as to achieve fair bandwidth sharing. This necessitates that UDP operation be customized such that if there are k flows being transmitted over a bandwidth R, then each flow is assigned and uses approximately the same long-term average bandwidth as a corresponding TCP flow would use under the same conditions [7]. This is termed as TCP-friendly behavior.

B. Rate Adaptation

Two major categories of rate adaptation schemes exist; congestion window-based and rate-based schemes. Rate-based protocols operate by varying the rate at which the sources transmit based on received notification about congestion levels experienced along the communication path [7, 8]. The implementation of TCP-friendly behavior within video streaming means that the video server will adjust its transmission rate when faced with dynamic conditions along the path to the video receiver. An investigation of the effect of rate smoothing on video traffic found that smoother rate transitions enhanced the user experience [9]. Hence it is important to be able to estimate accurately the path bandwidth, and to effect smooth rate transitions.

TCP-friendly rate control (TFRC) uses packet loss metrics as well as round-trip time (RTT) to compute the acceptable rate at which to transmit video. It is based on an equation that models the amount of traffic consumed by a TCP flow when subjected to varying packet loss conditions. The general equation is such that available bandwidth T in bytes/sec is given by Equation 1,

\[
T = \frac{s}{RTT - \frac{p}{3} + \frac{t_{RTO}}{32} p^2 (1 + 32p^2)} \tag{1}
\]

where \(s\) denotes the packet size, \(RTT\) the round trip time estimate, \(p\) corresponds to the packet loss event rate and \(t_{RTO}\) the TCP retransmit timeout.

A weighted moving average of the packet loss rate is used to minimize the effect of a spurious loss event on the overall calculated value. The RTT is the other key factor in computation of the sending rate. It is generally defined as the time duration between when a packet is sent and when a corresponding acknowledgement for the sent packet arrives at the sender. The most common implementation of RTT estimation is given in Equation 2,

\[
\hat{x}_k = \alpha \hat{x}_{k-1} + (1-\alpha)RTT_{\text{eff},k} \tag{2}
\]

where \(\hat{x}_{k-1}\) is the previous estimate of the RTT, \(\hat{x}_k\) is the new computed value, \(RTT_{\text{eff},k}\) is the \(k^{th}\) sample of the effective RTT and \(\alpha\) is a constant typically chosen between 0.84 and 0.9.

This work utilizes RTT estimation filters as a means of enhancing rate adaptation for improved Quality of service (QoS) of a streaming session.

III. ROUND-TRIP TIME ESTIMATION

To implement rate adaptation, traffic-moderating mechanisms have been designed that operate by predicting the future value of the RTT. This approach reduces the possibility of having the estimated value being unduly skewed by noise in the system such as caused by non-synchronous clocking at the source and receiver [10, 11], or by spikes in the current RTT value. The classical TCP filter, the proposed Kalman filter and the adaptive TCP filter are described briefly in this section.

A. TCP Filter

TCP uses Equation 2 with \(\alpha = 0.845\) to estimate the round trip time. Round trip time estimates produced using the TCP filter present a number of issues, many of which are examined in [4, 11]. In this paper, the focus is on the inadequacy of the filter to quickly adapt to drastic changes in the network round-trip time, leading to bad estimates when such a phenomenon occurs.

B. Adaptive Kalman Filter

Several studies have been undertaken that utilize the Kalman filter to estimate bandwidth availability as well as round-trip time [2, 12]. The Kalman filter operates as follows.

If the RTT is regarded as being composed of a smoothed desired RTT signal together with an additive noise component, then the equations that model the desired RTT as a noise observation of a constant exposed to step change in the mean are summarized in Equation 3 and Equation 4 below [2]:

\[
x_k = x_{k-1} + \delta \nu_k \quad \delta_k \in \{0,1\} \tag{3}
\]

\[
RTT_{\text{eff},k} = \hat{x}_k + e_k \tag{4}
\]

where the noisy characteristics of the RTT are captured by the measurement noise \(e_k\) of variance \(\sigma_e\). The step changes in the desired RTT \(x_k\) are modeled by process noise \(\nu_k\) with variance \(\sigma_\nu\) and the discrete variable \(\delta_k\). If a change occurs at time \(k\), then \(\delta_k = 1\) otherwise \(\delta_k = 0\). To estimate the sequence \(\delta^B\) of instances changes of a cumulative sum (CUSUM) is used.

The algorithm for the computation of the estimates \(\hat{x}\) of desired RTT can be summarized as follows. For the predictor stage, the process current state \(\hat{x}_0\) and covariance \(P_0\) predictions are respectively

\[
\hat{x}_k = \hat{x}_{k-1} + K_k (RTT_{\text{eff},k} - \hat{x}_{k-1}) \tag{5}
\]
The simulation network topology used is shown in Figure 1. The network nodes include one multimedia server feeding multicast receivers over a bottleneck link of 0.5 Mbps that is also shared by TCP flows. It should be noted that the adaptive Kalman filter and adaptive TCP filter in this simulation are applied only to the video flow and not the background TCP flows. From time 0s the video traffic flows via the link Router0-Router1 to the different destinations. At time 250s the link goes down, and automatically the traffic is routed along the path Router0-Router4-Router3-Router1. The link Router0-Router1 is restored at time 350s, at which point the traffic flow reverts to this initial link since NS2 routes traffic through the shortest link. A sudden, significant change in end-to-end delay is thus achieved.

The implementation of NS2 used in the simulation is enhanced with RTP/RTCP as described in [5]. This implementation not only replicates more closely the RTP/RTCP actual behavior and reporting mechanisms, but also facilitates TCP-friendly transmission of multimedia streams. The streaming server used in the simulation is equipped with TCP-friendly rate adaptation capabilities, adjusting its transmission rate based on information gleaned from the RTCP receiver reports. The RTP traffic is set as 50 kbps at the onset, although this later adjusts dynamically, depending on the packet losses and the round-trip time along the path from server to receivers. The RTCP reports are generated every 5 seconds, providing the parameters used to compute the appropriate transmission rate at the source. Values of jitter, throughput and delay for the RTP traffic are obtained and used to assess the performance of the adaptive Kalman and adaptive TCP filters for rate adaptation relative to the TCP filter.

For the Kalman filter, the parameters are set as in [2], with a drift value of 0.005 and a threshold of 0.05. The same values are used for the adaptive TCP filter.

A. RTT results

The simulation is run for 500 seconds and four sets of RTT values are collected. One set is the actual measured RTT for the video traffic, derived from the RTCP reports. The RTT values when the TCP filter is implemented, the
adaptive TCP filter values plus the corresponding values when deploying the Kalman filter are illustrated in Figures 2, 3 and 4. The plots illustrate the rapidly fluctuating values of the measured RTT that are typical of the end-to-end delays encountered on a real network. The estimated RTT values produced when the TCP filter is deployed display smoother value transitions as illustrated in Figures 2 and 3, reflecting the weighting effect of the underlying algorithm that assigns lower significance to more recently measured values. The Kalman filter is found to exhibit the smoothest characteristics as shown in Figure 2, which can be attributed to its error correction capabilities. This however occurs at the expense of precision of RTT estimation, reflecting the trade-off between smoothness and precision. Furthermore, the adaptive Kalman and adaptive TCP filters are found to react more rapidly than the TCP filter to sudden changes of the RTT value, reverting faster to tracking closely the measured RTT value, as occurs at simulation time 250 and 350 seconds in Figures 2 and 3. A comparison of the two adaptive filters, as illustrated in Figure 4, shows that the adaptive TCP filter reacts much faster than the adaptive Kalman to sudden changes in the RTT but also introduces more fluctuations in its RTT estimates.

Increasing the drift for the adaptive Kalman filter produces smoother output and results in higher throughput of the RTP traffic. The higher throughput can be attributed to the associated less conservative RTT estimates, which are interpreted by the rate adapting scheme as being indicative of low congestion and therefore permission for higher bit rates at the server.

Increasing the drift for the adaptive Kalman filter produces smoother output and results in higher throughput of the RTP traffic.

B. Jitter results

The Kalman filter displays a slightly better performance with respect to jitter and delay, than the TCP filter. The average delay experienced by the video flow packets is 1.494s, the corresponding figure in the case of the Kalman filter and the adaptive TCP filter being respectively 1.432s and 1.475s. Of even more importance for real-time applications are the jitter values in the two scenarios. As mentioned in [13], applications being streamed in real-time are more affected by jitter than round trip time values or in other words end-to-end delay. The average jitter for the Kalman prediction filter was 2.678 ms, better than the 2.859 ms and 3.815 ms average experienced by the RTP traffic deploying the adaptive TCP filter and the TCP filter respectively. The jitter is computed as shown in Equation 15.

\[
\text{jitter} = \frac{(R_{j+1} - R_j) - (R_j - R_{j-1})}{16}
\]

where \(R_x\) denotes the arrival time of packet \(x\) at the receiver; \(jitter\) thus being the difference in end-to-end delay for successive packets arriving at the client. Plots of the three sets of instantaneous jitter values are depicted in Figures 5, 6 and 7.

Figure 2: Measured, TCP-filter and Kalman-filter RTT values

Figure 3: Measured, TCP-filter and Adaptive TCP-filter RTT values

Figure 4: Measured, Adaptive TCP-filter and Kalman-filter RTT values

Figure 5: Jitter for the TCP-filter

B. Jitter results

The Kalman filter displays a slightly better performance with respect to jitter and delay, than the TCP filter. The average delay experienced by the video flow packets is 1.494s, the corresponding figure in the case of the Kalman filter and the adaptive TCP filter being respectively 1.432s and 1.475s. Of even more importance for real-time applications are the jitter values in the two scenarios. As mentioned in [13], applications being streamed in real-time are more affected by jitter than round trip time values or in other words end-to-end delay. The average jitter for the Kalman prediction filter was 2.678 ms, better than the 2.859 ms and 3.815 ms average experienced by the RTP traffic deploying the adaptive TCP filter and the TCP filter respectively. The jitter is computed as shown in Equation 15.

\[
\text{jitter} = \frac{(R_{j+1} - R_j) - (R_j - R_{j-1})}{16}
\]

where \(R_x\) denotes the arrival time of packet \(x\) at the receiver; \(jitter\) thus being the difference in end-to-end delay for successive packets arriving at the client. Plots of the three sets of instantaneous jitter values are depicted in Figures 5, 6 and 7.
C. Throughput results

Figures 8 and 9 show the throughput for the RTP stream and one of the background TCP flows for the adaptive TCP filter and Kalman filter respectively. The average throughput for the Kalman case was 730.297 kbps, higher than the 708.108 kbps and 698.613 kbps value obtained when using the adaptive TCP filter and the classical TCP filter respectively. It can be seen from the throughput flows that the video flow using adaptive filters behaves in a bandwidth-friendly manner, varying in much the same fashion as the TCP flow, but fluctuating more smoothly than the TCP flow.

In Figures 10, 11 and 12 the video traffic instantaneous throughput for the Kalman, adaptive TCP and TCP filters are compared against each other.

The results in Figures 10 and 11 show that the TFRC scheme with the adaptive filters is faster in adapting its bit rate than TFRC with the classical TCP filter. This is a direct result of the change detection scheme that allows the adaptive filters to detect any changes in the network conditions much faster than the TCP filter. At time 250s for instance, as the bottleneck link goes down the TFRC algorithms that use adaptive filters substantially reduce their bit rate which results in lower throughput values observed almost immediately after simulation time 250s.

As soon as the network capacity improves, as is the case at time 350s, the TFRC flows enhanced with adaptive filters once again react much faster than the TCP-filter in increasing their bit rate. As a result a much higher throughput is observed almost immediately after simulation time 350s. The classical TFRC, i.e. with TCP-filter, is slow in detecting this change and therefore adapts its rate slowly, hampered by the lower reaction speed of the TCP-filter.
Figure 12 shows that the TFRC with the adaptive TCP filter is the fastest in detecting the improvement in the network capacity, but thereafter the Kalman filter performs better, giving higher throughput values. The implication is that this is due to the Kalman filter having the lowest average RTT values of all the three filters, hence higher instantaneous throughput values are achieved.

![Figure 12: Video throughput for ATCP and Kalman filters](image)

V. CONCLUSION

In this work an investigation of the adaptive TCP and adaptive Kalman filters’ capability to provide more responsive and aggressive rate adaptation for real-time flows as network condition suddenly change has been conducted. The video traffic, despite flowing over UDP, is adapted to display TCP-friendly bandwidth utilization characteristics. Thereafter the Kalman filter and an adaptive TCP filter are incorporated into the rate adaptation model for the video flow, with the goal of achieving smoother and more precise estimation of RTT and hence variation of the sending rate at the server.

It is shown that the adaptive TCP and Kalman filters provide smoother operation than the TCP filter, allowing for a better viewing experience and lower likelihood of artefacts. It is illustrated that when there is a sudden change in network traffic levels, such as due to the collapse of a section of the streaming path and therefore traffic diversion to a new route, the adaptive filters respond more quickly and adeptly to the change, hence creating better responses by the source. The adaptive Kalman filter realizing generally much better results than the adaptive TCP filter.

This work reduces wastage of bandwidth by the TFRC model by providing the classical TCP filter with a change detection scheme, and initial results show improved performance was successfully achieved.

Future work will involve investigation of the adaptive filters under varying network conditions, to determine any instances under which the proposed change detection schemes fail. Performance evaluation with objective video quality measures will also be undertaken.

ACKNOWLEDGEMENT

This work is funded by the DST-Innovation Fund under project no. T70026: Bandwidth Adaptive Real-Time Video Broadcasting over Internet.

REFERENCES


Guy-Alain Lusilao-Zodi earned his Master degree in computer science in 2008 from the University of Stellenbosch, South Africa and is presently studying towards his PhD at the University of Cape Town. His research interests include multimedia streaming, rate adaptation, network analysis and traffic engineering.

Nakato Josephine Kakande is studying towards her Master of Science in Electrical Engineering at the University of Cape Town. Her research interests include multimedia streaming, and communication protocols.