TCP Congestion Response for Low Latency HTTP Live Streaming

Yousif Humeida, Mike Nilsson and Steve Appleby
BT Technology, Service & Operations
British Telecommunications plc
Adastral Park, Suffolk, IP5 3RE, UK
(yousif.humeida, steve.appleby, mike.nilsson)@bt.com

***Abstract*— Adaptive Bit Rate streaming using HTTP delivery is now widely used to provide live audio-visual content services. However, the quality of experience is often considered inferior to that of conventional broadcast services due to the high end to end latency, which is mostly due to the large amount of buffering required to provide resilience to variable network conditions.**

In this paper we present our initial research to address this issue of high latency with Adaptive Bit Rate streaming. If each segment of content could be delivered in a consistent period of time, the amount of buffering required could be reduced. We present the results of simulations where we have changed the timing of the TCP congestion response to consider the timing requirements of content segments, while trying to retain fairness to competing flows. We show that with this simple modification to TCP the variation in the delivery time of content segments can be reduced and hence much lower end to end latency can be achieved. With no changes to the network or to client devices required, this solution may be straightforward to deploy to make the user experience of services delivered by Adaptive Bit Rate streaming much closer to that of conventional broadcast services.

Keywords—TCP, congestion response, latency, ABR, live streaming

I. Problem Statement

TV content services are now widely deployed using HTTP over TCP delivery. There are many reasons for this including widespread support on client devices and middleware, the ability to operate over domains which extend beyond the network operator, CDN support and efficient delivery of channels with low numbers of viewers.

However the use of TCP for the delivery of content services is complicated by the behaviour of TCP, where the combination of the retransmission of lost packets and the congestion response to ensure fairness to competing flows results in highly variable delivery times for content segments.

Techniques, generally referred to as Adaptive Bit Rate streaming, have been developed to address these issues. Content is divided into temporal segments of duration typically in the range 2s to 10s, and encoded at multiple bit rates. Each segment is delivered using TCP with an encoded bit rate selected to trade-off video quality with timely delivery, to avoid content presentation stalling.

TV services delivered in this way are considered to be of inferior quality to conventional broadcast services, partly due to the significantly higher end to end latency, often more than 30s, that is objectionable for live TV services.

Our research is addressing the question of how to reduce this end to end latency, caused primarily by client buffering to smooth the effect of variable network throughput, without causing presentation to stall and without causing wild fluctuations in video quality. In this paper we present the initial results of this research based on simple ns-3 simulations.

II. Our Proposed Solution to the Problem

A. Overview

Our approach to this problem is to retain the reliability of TCP delivery and to retain the fairness to other TCP flows, but to change the timescale of the congestion response of TCP.

Conventional implementations of TCP respond to congestion quickly, typically within a round trip time of its occurring, and respond without knowledge or consideration of the content service that is being delivered.

With Adaptive Bit Rate streaming, after a content segment has been requested, it must be delivered by a specific time or content presentation will stall.

Our approach is to make the TCP congestion response aware of the content delivery requirements: we delay the response to congestion until delivery of the current content segment has been completed. This potentially allows delivery of the segment to be achieved on time, and prevents content presentation from stalling, while possibly causing some unfairness to competing traffic. The encoded bit rate of the next segment to request is selected by considering the amount of congestion experienced during delivery of preceding content segments.

By doing this we hope to achieve consistent delivery times for content segments, which would allow the amount of client buffering needed to achieve presentation without stalling to be decreased, and hence achieve lower end to end latency, while avoiding sustained unfairness to competing traffic.

978-1-5090-6501-1/17/$31.00

B. TCP Congestion Response for Low Latency services

Rather than change the size of the TCP congestion window, CWND, as soon as packet loss is detected, we keep the value constant until the content segment has been delivered, and then change its value for future content segments based on the recent levels of packet loss observed. We use "Fast Recovery" to enable steady delivery of data after packet loss, but "deflate" CWND back to the initial value chosen for the segment afterwards.

C. Selection of the Encoded Bit Rate of Content Segments

Round trip time and packet loss rate are measured as content segments are delivered. A fair share bit rate, R, is calculated as a function of the segment size, s, the average round trip time, RTT, and the average packet loss rate, p, using the model developed by Matthew Mathis et al. [1]:

$$R = \frac{\sqrt{3}s}{RTT\sqrt{2p}} \quad (1)$$

This fair share bit rate is used to determine the encoded bit rate of the next content segment: the maximum encoded size of the next content segment, D_{MAX}, is determined from R, the deadline for delivery of the segment T_d, and the current time t:

$$D_{MAX} = R \times (T_d - t) \quad (2)$$

The actual media segment to be requested is determined subject to its encoded size, D, being no greater than D_{MAX}:

$$D \leq D_{MAX} \quad (3)$$

The value of CWND for delivery of the next content segment is calculated to try to ensure delivery by the time T_d:

$$CWND \geq (RTT \times D) / (T_d - t) \quad (4)$$

III. Adaptive Bit Rate Streaming Simulation in ns-3

A. Simulation Configuration

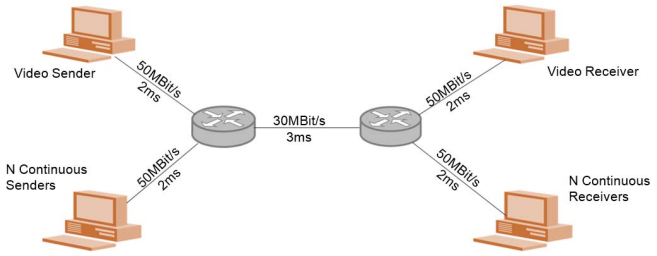

Fig. 1. The network configuration implemented in ns-3.

To assess the performance of our modified TCP congestion response for live content delivery, we implemented an Adaptive Bit Rate streaming system in the ns-3 network simulator [2] and simulated the delivery of 30 minutes of content through the network shown in Fig. 1. The content segments, each of presentation duration 10s, were transmitted every 10s using either NewReno TCP [3] or TCP with the congestion response described above. They competed against 4, 9, or 14 continuous data transfers each of which started 30s before the first content segment. The size of each content segment was selected from representations encoded at bit rates of 750KBit/s, 1.0MBit/s, 2.5MBit/s, 3.8MBit/s and 4.5MBit/s. The receivers were all standard TCP receivers.

B. Results

The aim of our Adaptive Bit Rate streaming system is to achieve consistent content segment delivery times. Fig. 2 shows the cumulative distribution of content segment delivery times. When our TCP congestion response, labelled TCP_LLLS for Low Latency Live Streaming, is used, regardless of the level of competing traffic, all content segments are delivered on time within 10s, and hence have no adverse effect on the user experience. But when NewReno TCP is used, 31 of the segments, out of a total of 540, are late, taking longer than 10s to be delivered, and hence would affect the user experience with playback stalling unless sufficient additional latency had been introduced at the start of playback.

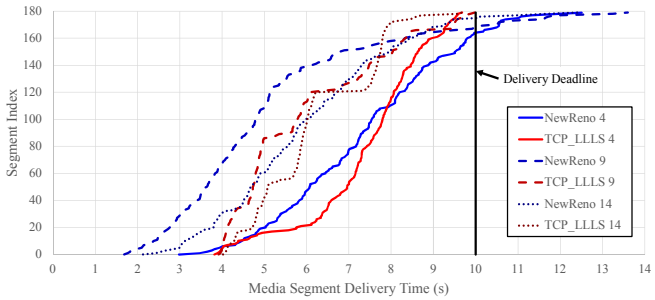

Fig. 2. The cumulative distribution of media segment delivery times.

IV. Conclusion

These results show that by changing the timing of the TCP congestion response to consider the timing requirements of content segments, consistent segment delivery times are achieved, enabling buffering to be reduced, which in turn reduces end to end latency, which in turn improves the quality of experience for live content services. This solution may be straightforward to deploy as no changes are required to the network or to client devices.

Our research is on-going to study the impact of more complex network configurations and a wider variety of competing traffic, and to include comparisons against using the widely deployed TCP CUBIC [4] for content segment delivery.

References

[1] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm", ACM SIGCOMM Computer Communication Review, 27(3), 1997, pp.67-82.

[2] "ns-3 : A Discrete-Event Network Simulator", manual available at https://www.nsnam.org/docs/manual/html/index.html.

[3] "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 6582, 2012, available at https://tools.ietf.org/html/rfc6582.

[4] B. Levasseur, M. Claypool, and R. Kinicki, "A TCP CUBIC implementation in ns-3", in Proceedings of the 2014 Workshop on ns-3, New York, NY, USA, 2014, p. 3:1–3:8.