Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 26, 2013

P. O'Hanlon
University of Oxford
K. Carlberg
G11

April 24, 2013

Congestion control algorithm for lower latency and lower loss media transport
draft-ohanlon-rmcat-dflow-02

Abstract

This memo provides a design for a congestion control algorithm, for media transport, which aims to provide for lower delay and lower loss communications.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This memo outlines DFlow, a congestion control algorithm that aims to minimise delay and loss by using delay-based techniques. The scheme is based upon TCP Friendly Rate Control (TFRC) [RFC5348], and adds a delay-based congestion detection scheme which feeds into a 'congestion event history' mechanism based upon TFRC's loss history. This then provides for a 'congestion event rate' which drives the TCP equation.

Congestion control that aims to minimise the delay is important for real-time streams as high delay can render the communication unacceptable [ITU.G114.2003]. On today's Internet a number of paths have an excess of buffering which can lead to persistent high latencies, which has become known as the Bufferbloat phenomenon. These problems are particularly apparent with loss-based congestion control schemes such as TCP, as they operate by filling the queues on a path till loss occurs, thus maximising the delay. The unfortunate consequence is that loss-based approaches not only lead to high delay for their own packets but also introduce delays and losses for all other flows that traverse those same filled queues.

Thus when competing with TCP, without the widespread deployment of Active Queue Management that aims to minimise delay, (e.g. Codel [I-D.nichols-tsvwg-codel]), it is not possible to maintain low delay

as TCP will do its best to keep the queues full and maximise the delay.

However there are many paths where the flows are not competing directly with TCP and where delay may be minimised.

The DFlow scheme can transport media with low delay and loss on paths where there is no direct competition with TCP in the same queue. Though we are currently testing some techniques to enable it compete with loss-based schemes (at the expense of delay) but they will be included in a later version of the draft. In simulations it has been seen to be reasonably fair when competing with other DFlow streams.

2. Conventions, Definitions and Acronyms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Background

Whilst the existing standard for media transport, Real-time Transport Protocol (RTP) [RFC3550], suggests that congestion control should be employed, in practice many systems tend to use fixed or variable bit rate UDP and do very little or no adaptation to their network environment. Most of the existing work on real-time congestion control algorithms has been rooted in TCP-friendly approaches but with smoother adaptation cycles. TCP congestion control is unsuitable for interactive media for a number of reasons including the fact that it is loss-based so it maximises the latency on a path, it changes its transmit rate to quickly for multimedia, and favours reliability over timeliness. Various TCP-friendly congestion control algorithms such as TFRC [RFC5348], Sisalem's LDA+ [SisalemLDA.2000], and Choi's TCP Friendly Window Control (TFWC) [ChoiTFWC.2007] have been devised for media transport, that attempt to smooth the short-term variation in sending rate. More recently there have been development of some delay-based schemes which aim to provide for low delay.

3.1. TFRC

TFRC is a rate based receiver driven congestion control algorithm which utilises the Padhye TCP equation to provide a smoothed TCP-friendly rate. The sender explicitly sets the transmission rate, using the TCP equation driven by the loss event rate which is measured and fed back by the receiver, where a loss event consists of one or more packet losses within a single RTT. It utilises a

weighted smoothed loss event rate, and EWMA smoothed RTT, as input to the TCP equation which enables it to achieve a smoother rate adaptation that provides for a more suitable transport for multimedia. TFRC was primarily aimed at streaming media delivery where a smooth rate and TCP-friendliness are more important than low latency operation.

However there are number of issues with TFRC as regards real-time media transport:

Loss-based operation: Firstly since it is a loss-based based scheme the latency is maximised which is a problem for real-time transport over heavily buffered paths. The other problem with loss-based protocols is that they rely on a certain level of packet loss which can be an issue for media traffic since lost media packet cannot usually be retransmitted in time. This problem becomes more of a concern at lower transmission rates since the TCP equation requires a corresponding increase in loss rates.

Bursty media flows: Many media flows exhibit bursty behaviour due to a number of factors. Firstly there may be negative bursts (i.e. gaps) due to silence or low motion which can lead oscillatory behaviours due to the data-limited and/or idle behaviours. Secondly there may be positive bursts (i.e. larger than normal) can also be due to the bursty nature of the media and codec (e.g. I-frames) which can be lead to drops or increased latency. Whilst the current version of TFRC [RFC5348] has attempted to address some of these issues, they are still a concern.

Small RTT environments: When operating in low RTT environments (<5ms), such as a LAN, systems implementing TFRC can have problems with scheduling packet transmissions as inter-packet timings can be lower than application level clock granularity. Whilst the current version of TFRC [RFC5348] has attempted to address these issues, they can still be a concern in some low RTT environments.

Variable packet sizes: As originally designed TFRC will only operate correctly when packet sizes are close to MTU size, and when the packet sizes are much smaller fairness issues arise. Although there have been attempts to address this problem for small packets [RFC4828], it is not clear how to deal with flows that do vary their packet sizes substantially. However this issue is only really a marked problem with lower bit rate video flows or variable packet rate audio.

3.2. Delay-Based schemes

In the last few years there has been a renewed interest in the use of delay based congestion control for media, with a slightly different emphasis to that of the history of TCP based approaches such as Jain's CARD, Wang and Crowcroft's Tri-S, Brakmo's Vegas, Tan et al's Compound TCP, and more recently Budzisz's CxTCP [BudziszCxTCP.2011]. Where the primary goal with media based transports is to actually minimise the latency of the flow, as opposed to just using delay as an early indication of loss. This is of particular relevance on paths with large queues, as is the case with a number of today's Internet paths. In 2007 Ghanbari et al [GhanbariFuzzy.2007] did some pioneering work on delay-based video congestion control using fuzzy logic based systems. Recently there has been on going activity in the IETF as part of the Low Extra Delay Background Transport (LEDBAT) Working Group which aims to provide a less than best effort delay-based transport with lower delay. However [RFC6817] specifies a one-way queuing delay target of 100ms which is quite a high baseline for interactive media, considering the recommended total one-way delay limit for a VoIP call should be less than 150ms [ITU.G114.2003].

4. Objectives

The objectives of DFlow are to provide for low delay and low loss media transport when possible. We also aim to provide (in a future version of the draft) mechanisms to provide for better burst management, and loss-mode operation.

Lower Delay: The one-way delay should be kept well within the acceptable levels of 150ms, and MUST NOT exceed 400ms [ITU.G114.2003].

Lower Loss: For media transport it is important to minimise loss as it is usually not possible to retransmit within the delay budget for many connections. Whilst modern codecs can tolerate some loss it is beneficial to avoid it. The advantage of low delay congestion control is that since it aims to operate within the queuing boundaries it generally avoids loss.

Smoothness: The media rate should aim to be smooth within the constraints of the media, codec, and the network path. A smooth rate generally provides for more palatable media consumption.

Fairness: The system should aim to be reasonably fair with itself and TCP flows. Initially we aim for self fairness, and we will aim to tackle TCP fairness when we have sufficiently robust loss-mode operation.

[Burst Management]: [Due in later rev] We are working on mechanisms to manage the bursty nature of media allowing it maintain a smoother quality.

[Loss-based mode]: [Due in later rev] We are working on mechanisms to allow the system to compete with loss-based congestion control and maintain throughput, though without additional network support it is understood that the delay (and loss) would be largely beyond control.

5. Design Outline

The DFlow scheme aims to primarily utilise delay measurements to drive the congestion control. It currently utilises some of the core aspects of TFRC, such as its rate based operation, utilisation of the TCP equation, and its rate smoothing. It also employs similar signalling mechanisms. However as the design evolves we expect that DFlow may diverge further from TFRC.

5.1. Delay Composition

The total end-to-end one-way delay (OWD) a packet incurs may be considered to consist of four elements; transmission (or serialisation), propagation, processing, and queuing delays. For our purposes the first three elements may be considered together as a largely static component, termed the base delay. The base delay generally does not change significantly unless the node is mobile or the underlying link alters due to something like a route change. The main dynamic element of the delay, which DFlow aims to utilise, is the queuing delay. Taken together with the base delay, the queuing delay provides an indication of the actual path latency and also provides an insight as to the level of congestion on the path.

5.2. Delay Measurement

The notional one-way delay is measured for each packet by comparing the sender and receiver timestamps. Whilst the clocks on the sender and receiver are unlikely to be synchronised, it is assumed that their offset is relatively constant as the clock skew is generally quite small. Thus the notional OWD may only be used in a relative context. The notional OWD is measured for each packet over two sampling periods; Firstly over the longer base_period (typically 10*RTT) from which the minima are stored as the base_delay. And secondly it is sampled over a shorter period current_period (typically 50ms), which is also filtered, usually also using a minima filter, and stored as current_delay. The minima of the OWDs are used to reduce noise of the measurements, which can be beneficial in the case of variable link types such as wireless.

5.3. Congestion Detection

The delay-based detection algorithm, outlined in Figure 1, operates by comparing the current_delay to the base_delay, which gives an indication of the queuing delay. If it exceeds a set congestion detection threshold, cd_thresh, then the packet is considered for the next stage of detection. The cd_thresh sets the limit for the queuing delay incurred by the flow, and is typically set at 50ms (we are also investigating automated thresholds). Once a flow has exceeded its cd_thresh then it undergoes a second test which is based upon the gradient of the delay change over two current_period's, indicating that delay is on the increase, if it is positive then a 'congestion event' is flagged.

```
If ((base_delay - current_delay) > cd_thresh AND
          (current_delay - prev_current_delay) > 0)
        DelayCongestionEvent = True
```

Figure 1: Congestion Detection pseudo-code

This algorithm then provides input to the 'congestion interval history' mechanism (based on TFRC's 'loss interval history'), which is combined with normal input from the TFRC packet loss detection mechanisms, from which a 'congestion event rate' is derived which is then fed into the TCP equation to determine the send rate.

Note that we currently disable TFRC's oscillation reduction mechanism from [RFC5348] (Section 4.5) as it adversely affects the delay-based operation.

We have performed a number of simulations of the above mechanism in operation and have found it to be reasonably fair to itself, providing for smooth rates at suitable RTTs.

5.4. Slow Start

The delay based congestion detection is not only used during normal the congestion avoidance phase of the protocol but it also employed during slow start allowing for rapid, lower loss, attainment of the operating rate.

5.5. Loss-mode

We are actively investigating techniques to enable competitive behaviours with loss-based protocol such as TCP. We aim to develop a solution that provides for automatic fallback between loss and delay modes.

6. Further Work

The design is still under active development and there is more work to be done. We are seeking feedback on these ideas and future directions.

7. IANA Considerations

This document makes no requests of IANA.

8. Security Considerations

With a congestion control algorithm an attacker can attempt to interfere with the protocol to cause rate changes. However encryption of the protocol will largely protect it against such threats.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4828] Floyd, S. and E. Kohler, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant", RFC 4828, April 2007.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

9.2. Informative References

[BudziszCxTCP.2011] Budzisz, L., Stanojevic, R., Schlote, A., Shorten, R., and F. Baker, "On the Fair Coexistence of Loss- and Delay-Based TCP", July 2011.

[ChoiTFWC.2007] Choi, S. and M. Handley, "Fairer TCP-friendly congestion control protocol for multimedia streaming applications", Dec 2007.

[GhanbariFuzzy.2007] Jammeh, E., Fleury, M., and M. Ghanbari, "Delay-based congestion avoidance for video communication with fuzzy logic control", Nov 2007.

[I-D.nichols-tsvwg-codel]
Nichols, K. and V. Jacobson, "Controlled Delay Active Queue Management", draft-nichols-tsvwg-codel-01 (work in progress), February 2013.

[ITU.G114.2003]
International Telecommunications Union, "One-way transmission time", ITU-T Recommendation G.707, May 2003.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, December 2012.

[SisalemLDA.2000]
Sisalem, D. and A. Wolisz, "LDA+: A TCP-Friendly Adaptation Scheme for Multimedia Communication", May 2000.

Authors' Addresses

Piers O'Hanlon
University of Oxford
Oxford Internet Institute
1 St Giles
Oxford OX1 3JS
United Kingdom

Email: piers.ohanlon@oii.ox.ac.uk

Ken Carlberg
G11
1600 Clarendon Blvd
Arlington VA
USA

Email: carlberg@g11.org.uk