Delay-based Congestion Control for Low Latency

David A. Hayes* and David Ros†
*University of Oslo, davihay@ifi.uio.no †Institut Mines-Télécom / Télécom Bretagne, david.ros@telecom-bretagne.eu

I. Introduction

For more than 25 years, Delay-based congestion control (DBCC) has promised low latency transmissions with little or no congestion-related losses, but so far it hasn't really delivered. We argue that despite its issues it is still a valuable direction to pursue in the quest for low-latency communication. DBCC may not be *the* solution for achieving low delays, but we do believe it is part of the solution.

A. Why should we use delay-based congestion control?

The position of this paper is that we should use end-to-end delay as a signal for managing congestion and reducing latency. Yes, network delay is a difficult metric to use, but it is a low level measure of the very thing we are trying to reduce: latency. It is also a fundamental property of a packet-switched network, and thus requires no changes to the network to provide the signal.

The delay signal delivers more timely feedback on the state of the network queues than loss or even Explicit Congestion Notification (ECN), though the latter is not widely used at present[1]. This is especially important in high capacity (or high bandwidth-delay product) networks. Jin et al. [3] argue that as capacity increases, and without any sort of explicit signal from the network, delay is the only viable choice for a congestion measure. We argue further that as path delay is a key measure of the quality we are attempting to reduce—i.e., latency—, path delay is a critical metric and input parameter for any congestion controller with that objective.

B. Brief history of delay-based congestion control techniques

DBCC in the Internet has its origins in NETBLT [4]. Jacobson [5] had a footnote concerning rate based AIMD, but the main early work was Jain [6]. Since then there have been a number of innovative approaches, from DUAL [7] and TCP Vegas [8] to more recent proposals such as FAST [9], Compound TCP [10] and LEDBAT [11].

All these methods had much promise, but varying degrees of success. To date, the only widely adopted delay-based mechanism is LEDBAT[2]; however, there is only anecdotal evidence of LEDBAT working as promised, but quite a few research studies pointing to several pitfalls and problems (e.g., [12, 13]).

II. Small queues and ECN

Arguably the best method of reducing latency along a path is to reduce the size of the buffers. With small buffers, early reaction to congestion is even more critical in order to avoid packet loss.

An early ECN-type marking can help with this. This would need to be an ECN-like marking that provides per-packet feedback and marks well before the queue builds up, possibly as soon as a packet needs to be queued. Such an ECN marking does not yet exist, and even if it did, it does not provide a picture of the entire path's latency, but could complement a delay signal well. Packet-based queues have a variable latency depending on the size of packets that they are populated by at any point in time. A delay signal provides thus the best measure of the path latency.

[1]Besides, ECN as currently defined gives a limited one-bit signal. There are promising ways of enhancing / leveraging ECN feedback, but either they are still at the research proposal stage (e.g., [1]) or they can be deployed only in tightly-controlled scenarios (e.g., [2]).

[2]Compound TCP has been shipped in Windows systems since Vista, but it seems to be disabled by default.

III. The delay signal

Two delay signals are commonly used for congestion control, round-trip time T_{RTT} and one-way delay T_{OWD}. The ultimate goal is to obtain a measure of the *queueing* delay T_q between the sender and receiver, but both of these measures include delays within the sender, within the receiver, and the propagation delays along the path.

Most delay-based TCP variants use T_{RTT}, as then only the TCP sender needs to be modified. However, other protocols, such as LEDBAT, use T_{OWD}.

A. Problems with the signal, and possible solutions to some of them

Finding the network queueing delay and using it as a congestion signal is fraught with difficulties, whether T_{RTT} or T_{OWD} is used as a proxy for T_q. However, we argue that ways of tackling some of those issues, and of improving the "quality" of the signal, do exist.

a) Directional congestion: Using T_{RTT} as a measure means that delays in the forward and reverse direction cannot be distinguished from one another. Not only that, but queueing delays in the opposite direction to that of interest are noise on the delay signal in the direction of interest. This can result in the sender reducing its sending rate, even though it is only the returning ACK stream that is experiencing congestion. Loss based congestion control can similarly be affected, though to a lesser extent, when returning ACKs are lost.

The obvious solution to this is to use T_{OWD} as a measure. For TCP, this involves modifications to the receiver to measure and feed the raw or processed signal back to the sender. Although this provides a much cleaner and more accurate measure of the latency, it is harder to deploy.

b) Incorrect estimates of Minimum RTT or Minimum OWD: Most delay based protocols relay on isolating the queueing delay from the path delay measurement by assuming that all other delays are fixed (at least over some time period). The path queueing delay is then:

$$\text{round-trip:} \quad T_{\text{q\_rt}} = T_{\text{RTT}} - T_{\text{RTT\_min}}$$
$$\text{one-way:} \quad T_{\text{q\_ow}} = T_{\text{OWD}} - T_{\text{OWD\_min}}$$

If the estimate of $T_{\text{RTT\_min}}$ or $T_{\text{OWD\_min}}$ is incorrect, then delay-based congestion control schemes tend to leave persistent queues along the network path, or lead to problems such as the so-called latecomer advantage. Work by Leith et al. [14] examines these issues in relation to AIMD control.

As shown by the first author in [15], it is possible to avoid this problem by using the change in delay, or the *delay gradient*, as a measure instead.

It is also worth noting that if the network indicated which packets were *not* queued (e.g., as indicated by the *absence* of an early ECN-type mark that is present only if the packet needs to be queued), $T_{\text{RTT\_min}}$ or $T_{\text{OWD\_min}}$ could be measured with better accuracy.

c) The path delay signal is noisy: TCP delayed ACKs add noise to the delay measurements, but by only measuring immediate responses this noise can be removed. Queues, especially under load, tend to have large variations in their occupancy. This makes the sampled delay signal very noisy, and not as strongly correlated with congestion as packet loss is. McCullagh and Leith [16] examine the issue and point out that it is the aggregate behaviour that is important, and that path delay is an effective measure for congestion control.

As a result of this, DBCC mechanisms need to filter the signal in some manner to provide a stable trigger for congestion detection. Averaging over time is a simple way of doing this, but it reduces the responsiveness of the system. An alternative to this is to use a probabilistic trigger mechanism where a congestion indication is triggered based on a weighted probability proportional to the measured delay [15, 17–19]. Another avenue of research being pursued by the authors is to consider the "noise" of the delay measurements as the signal used to infer congestion, rather than something to be filtered out. We expect that this may be especially useful when buffers are small.

IV. Compatibility with Loss based TCP

TCP sessions using DBCC do not compete well with those using loss-based congestion control (LBCC). Delay-based controls react earlier to congestion, and try to keep delays, and thus queues along the path, small. Loss-based controls try to fill the queues, probing available bandwidth, until there is loss.

A. Fair coexistence

There have been a number of attempts to make delay-based TCPs coexist better loss-based TCPs [15, 18, 19]. They tend to be bimodal, aiming to keep queues small when not coexisting on the path with loss-based TCPs, and then attempting to coexist fairly with loss-based flows (filling queues) when they are detected as sharing the path.

This approach may provide a bridge from LBCC to DBCC. Assume several long-lived flows sharing a bottleneck are using "bimodal" DBCC. If even another single *long-lived*, loss-based flow shares the bottleneck with the latter, then DBCC flows will switch to their compatibility mode, so the bottleneck buffer will remain full and latency will stay high. However, if flows using LBCC are *short-lived*—and so, likely to be *latency-sensitive* ones—, then they will in general not be able to drive buffers full, and they will see low delay *in spite of sharing the bottleneck with long-lived flows*; LBCC will thus be able to attain a high throughput while latency remains low.

B. Less-than-best-effort (LBE) service

LEDBAT attempts to behave in an LBE manner, taking advantage of the early congestion signals provided by the delay signal. Unfortunately, it seem to be prone to some of the issues identified in III-A (see [13]). However, such a scheme could arguably be improved by applying to its design some of the solutions discussed in III-A.

V. A way forward

No one measure provides all the information necessary to effectively control congestion and reduce latency. Small queues, early ECN type marking, other network based controls and signals, and a path delay measurement all provide information that allows transport protocols to best utilise available capacity and coexist in the Internet. Path delay measurements are a critical signal for transport congestion control mechanisms to reduce latency, as they are the only signal that directly measures the quantity we are attempting to reduce. We think not that end-to-end delay should be regarded as the *only* congestion signal to consider, but instead that it *should* be used by congestion controllers in some way.

In homogeneous environments, such as data centers, DBCCs look very promising though work on fine grain measurements is required. Having the right incentives for deployment is of course very important for general adoption. DBCC got some early bad press due to Vegas's coexistence issues with plain loss-based TCP. Bimodal DBCC looks like a good way of ensuring that incentives go in the good direction but, for this to hold, such kind of proposals have to "work right". This issue deserves further study.

Acknowledgment

The authors are funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). The views expressed are solely those of the authors.

References

[1] I. A. Qazi, L. L. H. Andrew, and T. Znati, "Congestion control with multipacket feedback," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1721–1733, 2012.

[2] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data Center TCP (DCTCP)," in *Proceedings of ACM SIGCOMM*, New Delhi, Sep. 2010.

[3] C. Jin, D. Wei, and S. Low, "The case for delay-based congestion control," in *Proceedings of the IEEE 18th Annual Workshop on Computer Communications*, 2003, pp. 99–104.

[4] D. Clark, M. Lambert, and L. Zhang, "NETBLT: A bulk data transfer protocol," RFC 998 (Experimental), Internet Engineering Task Force, Mar. 1987. [Online]. Available: http://www.ietf.org/rfc/rfc998.txt

[5] V. Jacobson, "Congestion avoidance and control," in *Proceedings of ACM SIGCOMM*. New York, NY, USA: ACM, 1988, pp. 314–329.

[6] R. Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," *ACM SIGCOMM Computer Communications Review*, vol. 19, no. 5, pp. 56–71, Oct. 1989.

[7] Z. Wang and J. Crowcroft, "Eliminating periodic packet losses in the 4.3-Tahoe BSD TCP congestion control algorithm," *ACM SIGCOMM Computer Communications Review*, vol. 22, no. 2, pp. 9–16, Apr. 1992.

[8] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995.

[9] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.

[10] K. Tan and J. Song, "Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks," in *in 4th International workshop on Protocols for Fast Long-Distance Networks (PFLDNet), 2006*, May 2006.

[11] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)," RFC 6817 (Experimental), Internet Engineering Task Force, Dec. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6817.txt

[12] J. Schneider, J. Wagner, R. Winter, and H.-J. Kolbe, "Out of my way – evaluating Low Extra Delay Background Transport in an ADSL access network," in *Proceedings of the 22nd International Teletraffic Congress (ITC 22)*, Amsterdam, Sep. 2010.

[13] D. Ros and M. Welzl, "Assessing LEDBAT's delay impact," *IEEE Communications Letters*, vol. 17, no. 5, pp. 1044–1047, May 2013.

[14] D. Leith, R. Shorten, G. McCullagh, L. Dunn, and F. Baker, "Making available base-RTT for use in congestion control applications," *IEEE Communications Letters*, vol. 12, no. 6, pp. 429–431, 2008.

[15] D. Hayes and G. Armitage, "Revisiting TCP congestion control using delay gradients," in *Proceedings of IFIP Networking*, ser. Lecture Notes in Computer Science, vol. 6641. Valencia: Springer, May 2011, pp. 328–341.

[16] G. McCullagh and D. Leith, "Delay-based congestion control: Sampling and correlation issues revisited," Hamilton Institute, National University of Ireland Maynooth, Tech. Rep., 2008.

[17] S. Bhandarkar, A. Narasimha Reddy, Y. Zhang, and D. Loguinov, "Emulating AQM from end hosts," in *Proceedings of ACM SIGCOMM*, Kyoto, Aug. 2007.

[18] L. Budzisz, R. Stanojević, A. Schlote, F. Baker, and R. Shorten, "On the fair coexistence of loss- and delay-based TCP," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1811–1824, Dec. 2011.

[19] D. Hayes and G. Armitage, "Improved coexistence and loss tolerance for delay based TCP congestion control," in *Proceedings of IEEE LCN*, 2010, pp. 24–31.