Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

M. Bagnulo
UC3M
K. De Schepper
Nokia Bell Labs
G. Judd
Morgan Stanley
July 8, 2016

Recommendations for increasing TCP performance in low RTT networks. draft-bagnulo-tcpm-tcp-low-rtt-00

Abstract

This documents compiles a set of issues that negatively affect TCP performance in low RTT networks as well as the recommendations to overcome them.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2017.

Copyright Notice

Table of Contents

1. Introduction

Over the last few years there has been significant operational experience about running TCP in networks with low RTTs. By networks with low RTTs we mean networks with RTTs between a few microsecs and a few hundreds of microsecs. These networks are typically found in datacenters and in addition to a low RTT they usually exhibit a high bandwidth (tens of Gbps). there are a number of reports and papers that show that TCP performance in such environment can be poor and that TCP needs to be tuned and even updated to provide good performance. The goal of this memo is to summarize the set of changes needed to TCP to perform well in these environments.

There are transport protocols, notably DCTCP [I-D.ietf-tcpm-dctcp] that have been specifically designed to perform well in data center environments where low RTT is the norm. However, due to several reasons, many datacenters also need to need to use TCP for their communications (see section 7.1 of [judd-nsdi] for the motivation for using TCP in a production datacenter). This is the reason why the recommendations about how to update TCP to run in these environments are relevant. Some of the recommendations contained in this note may also apply to protocols such as DCTCP, but the main goal of this note is TCP.

We next describe different issues that have been identified and the changes that would be required in the TCP specifications and/or the TCP implementations to address them.

2. Minimum Retransmission timer

Current TCP specification recommend that the minimum retransmission timer (RTOmin) should be at least 1 second. According to [incast], current implementations, RTOmin is set between 200 ms and 400 ms. In a network with RTT in the order of microseconds, this imposes large periods of inactivity when a packet is lost and its loss is detected via the retransmission timeout. This also aggravates the so called TCP incast problem. This issues has been reported in several papers, including [incast-wren], [judd-nsdi], and [incast]. One proposed mitigation to this problem that results in better performance is to reduce RTOmin.

From a specification perspective, RFC 6298 [RFC6298] states that:

> (2.4) Whenever RTO is computed, if it is less than 1 second, then the RTO SHOULD be rounded up to 1 second.

[incast] suggests that using RTOmin equal to 200 microsecs provides significant performance improvement in terms of goodput and that even no RTOmin results in even better performance.

Using a lower RTOmin while it goes against the recommendation included in RFC6298, it is supported as the specification as the RTOmin of 1 ms is not mandatory, just a recommendation. However, it would beneficial to update RFC6298 in this aspect and to provide a recommendation (maybe in the form of BCP) that for low RTT networks, a smaller RTOmin should be used.

This has an implication on the clock granularity when calculating RTO. RFC6298 doe not impose any requirement on the granularity of the clock used to measure the RTT used for the RTO calculation. It does state that finer clock granularities (below 100 ms) perform better. In order to achieve RTOmin of 200 micro secs or less, the granularity must be finer than the the RTOmin allowed. According to [incast] and [judd-nsdi] current linux systems can achieve a RTOmin of 4 ms due to the coarse granularity. so, providing a recommendation in terms of the granularity may also be useful.

3. Delay for Delayed ACKs

[judd-nsdi] reports that the default value for the delay for delayed ACKs ranges between tens and hundreds of ms. For low RTTs, a lower value of delay achieves a higher performance (see [judd-nsdi]) and hence a value of 1 ms or lower should be recommended for low RTT networks.

From a specification perspective, current specifications do not require a minimum waiting time for generating the delayed ACKs. They do impose a maximum waiting time. In particular, RFC 1122 [RFC1122] states that:

> A TCP SHOULD implement a delayed ACK, but an ACK should not be excessively delayed; in particular, the delay MUST be less than 0.5 seconds, and in a stream of full-sized segments there SHOULD be an ACK for at least every second segment.

Also, RFC5681 [RFC5681] states that:

> The delayed ACK algorithm specified in [RFC1122] SHOULD be used by a TCP receiver. When using delayed ACKs, a TCP receiver MUST NOT excessively delay acknowledgments. Specifically, an ACK SHOULD be generated for at least every second full-sized segment, and MUST be generated within 500 ms of the arrival of the first unacknowledged packet.

So, from a specification perspective, current RFCs do not need to be updated, but it may be useful to provide a recommendation in the form of BCP that for low RTT environments, the delay used for delayed ACKs should be tuned accordingly.

4. Minimum Congestion window

Current specifications require that the minimum congestion window is 2MSS. As pointed out in [TCP-sub-mss-w] and [judd-nsdi], in the case of small RTTs, this may result in a considerably large rate, below which TCP becomes unresponsive to congestion. In particular, with a SMSS of 1500 B and a RTT of 50 micro secs, this results in a rate of 240Mbps.

In terms of specifications, according to RFC5681, the CWND in Fast Retransmit and Fast Recovery is calculated as:

> 2. When the third duplicate ACK is received, a TCP MUST set ssthresh to no more than the value given in equation (4).
>
> 6. When the next ACK arrives that acknowledges previously unacknowledged data, a TCP MUST set cwnd to ssthresh (the value set in step 2). This is termed "deflating" the window.
>
> ssthresh = max (FlightSize / 2, 2*SMSS) (4)

In order to address this issue, it is necessary to modify TCP behaviour to function with CWND smaller than 2 MSS. This would require a update to RFC 5681. Several possibilities have been

proposed to accommodate this need. [TCP-sub-mss-w] and [TCP-nice] propose possible solutions.

5. Other issues

[judd-nsdi] identifies that in networks where the propagation delay and the transmission delay are very small, the queuing delay affects the RTT severely resulting in significant changes in the RTT. This has a negative effect in the calculation of the receiver buffer when using autotunning, since the buffer is calculated using the RTT estimation. The result is that it is frequent in these scenarios that the TCP connection is limited by the receiver buffer/RCVWND.

As far i can tell, there is no RFC that defines how to calculate the receive buffer, so no change in any spec would be required to address this, but maybe it is worthwhile to define a mechanism for autotunning for small RTTs and/or to do some recommendation in this regard.

6. Concluding remarks

This document compiles a number of issues that have been previously identified as harming TCP performance in low RTT networks. Some of the issues require updates in the current specifications and probably most of the issues may deserve some form of recommendation in the form of a BCP for using TCP in low RTT networks. It may make sense to work on the changes in the specification and the definition of new specifications (in particular for the case of lower than 1 MSS CWND) and then evolve this document to become the BCP for low RTT environments.

7. Security considerations

TBD, not sure if there is any.

8. IANA Considerations

There are no IANA considerations in this memo.

9. Acknowledgments

TBD

10. Informative References

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <http://www.rfc-editor.org/info/rfc6298>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <http://www.rfc-editor.org/info/rfc5681>.

[I-D.ietf-tcpm-dctcp] Bensley, S., Eggert, L., Thaler, D., Balasubramanian, P., and G. Judd, "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters", draft-ietf-tcpm-dctcp-01 (work in progress), November 2015.

[judd-nsdi] Judd, G., "Attaining the promise and avoiding the pitfalls of TCP in the Datacenter", NSDI 2015, 2015.

[incast] V., V., A., A., H., H., E., E., D., D., G., G., G., G., and B. B., "Safe and Effective Fine-grained TCP Retransmissions for Datacenter Communication", ACM SIGCOMM 2009, 2009.

[incast-wren] Y., Y., R., R., J., J., R., R., and A. A., "Understanding TCP Incast Throughput Collapse in Datacenter Networks", WREN 2009, 2009.

[TCP-nice] A., A., R., R., M., M., R., R., and A. A., "TCPNICE: A mechanism for background transfers", SIGOPS Oper. Syst. Review 2002, 2002.

[TCP-sub-mss-w] Briscoe, B. and K. De Schepper, "Scaling TCP's Congestion Window for Small Round Trip Times", BT Technical Report TR-TUB8-2015-002, May 2015, <http://www.bobbriscoe.net/projects/latency/sub-mss-w.pdf>.

Authors' Addresses

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Koen De Schepper
Nokia Bell Labs
Antwerp
Belgium

Email: koen.de_schepper@nokia.com
URI: https://www.bell-labs.com/usr/koen.de_schepper

Glenn Judd
Morgan Stanley

Email: Glenn.Judd@MorganStanley.com