TCPM Working Group
Internet Draft

J. Heitz
Cisco
Chuan He
Ericsson

Intended status: Informational
Expires: April 2015

October 19, 2014

TCP Retransmission Timer for Virtual Machines
draft-heitzhe-tcpm-vm-rto-00.txt

Abstract

A Round Trip Time (RTT) estimate that decays performs badly in a bursty environment. A round trip time estimator that does not decay for a period of time is proposed.

It does not require a minimum value to be configured. It works equally well when the typical RTT is 100uS as when it is 10 seconds.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on March 19, 2009.

Table of Contents

1. Introduction

Virtual machines can create bursty environments for TCP, especially if the TCP also runs in a thread within a process of a busy host machine. Round trip time measurements can frequently be hundreds of times the average. A retransmission timer (RTO) calculation that decays even a little bit for each small measured RTT will cause a retransmission for every such outlier RTT.

[RFC6298] requires a minimum RTO of 1 second to avoid spurious retransmissions. RTTs between VMs in a lightly loaded host are regularly less than 1 millisecond. On a heavily loaded host, RTTs do not all get higher. They get a little higher in the median, to a few milliseconds, but the number of spikes of 100s of milliseconds increases. The EWMA algorithm of [RFC6298] is hardly used. It simply defaults to 1 second. When the default is reduced, it gets a spurious retransmission on nearly every spike.

The proposed algorithm is less aggressive than that of [RFC6298] and needs no minimum setting. In fact, it grew out of an attempt to find a better minimum.

The retransmission timer is a compromise. If it is set too low, then spurious retransmissions occur, but if it is set too high, then it takes too long to retransmit when it is really needed.

The right balance is achieved when an acceptably small number of spurious retransmissions occur.

2. The Algorithm

The basis of the algorithm is as follows: Time is divided into intervals. Within each interval, the highest RTT is determined. This RTT forms the RTO to be used for the next interval. The RTO is constant for the duration of an interval.

The end of an interval and the beginning of the next one is determined when any of the following events occur:

1. An RTT is measured that is greater than the maximum RTT from the previous interval. The maximum RTT from the previous interval is the RTT that determines the RTO of the current interval. This measured RTT is the greatest RTT measured for the current interval. It is considered part of the current interval, not of the next one.

2. A large number (suggest 20) of windows of data has been transmitted.

The RTO of one interval is the maximum RTT of the previous interval plus some headroom. The suggested headroom is 1/4, so RTO = 1.25 * (previous max RTT).

A window of data is the largest ever advertised window of a session.

2.1. Initial Interval

The RTO of the first interval should be 1 second. The length of the first interval should be shorter than the others. Suggestion is 3 RTT measurements. The initial RTO may alternatively be determined from a history of previous connections.

An alternative is to run the regular algorithm as in [RFC6298] during the first interval. The RTTs would still be individually measured in preparation for the second interval.

3. Delayed ACK Timer

The delayed ACK timer is not handled differently. If a delayed ACK timer is in effect on the peer, it may cause high RTT measurements. If delayed ACK happens less than every 20 windows, then it will be included as part of the maximum RTT measurement. If it happens after more than 20 windows of data have been transmitted, then a possibly resulting retransmission is not excessive.

4. Backing off the timer

This document is an alternative to section 2 of [RFC6298]. In particular, the backing off mechanism in section 5 remains intact.

5. Security Considerations

No security issues beyond those outlined in [RFC6298] have been identified.

6. IANA Considerations

None

7. References

7.1. Normative References

[RFC6298] V. Paxson, M. Allman, J. Chu and M. Sarent, "Computing TCP's Retransmission Timer", RFC 6298, June 2011.

8. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Jakob Heitz
Cisco
510 McCarthy Blvd,
Milpitas, CA 95035

Email: jheitz@cisco.com

Chuan He
Ericsson
300 Holger Way,
San Jose, CA 95134

Email: chuan.he@ericsson.com