Internet Engineering Task Force M. Allman
INTERNET-DRAFT ICSI
draft-allman-tcpm-rto-consider-00.txt March 4, 2011

# Retransmission Timeout Considerations

## Status of this Memo



## Copyright Notice



## Abstract

This document provides for high-level guidance for retransmission timeout schemes appropriate for general use in the Internet.

## Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in BCP 14, RFC 2119
[RFC2119].

## 1 Introduction

Despite our best intentions and most robust mechanisms, reliability in networking ultimately requires a timeout and re-try mechanism. Often there are more timely and precise mechanisms (e.g., TCP's selective acknowledgment scheme [RFC2018,RFC3517]), but these require information exchange between components in the system, which cannot be guaranteed. To the contrary, we can always depend on the passage of time and therefore our ultimate backstop to ensuring reliability is a timeout.

Various protocols have defined their own timeout mechanisms (e.g., TCP [RFC2988], SCTP [RFC4960], etc.). Further, while standardized, implementations also add their own subtle tweaks to the process. At this point we recognize that often the specifics are not crucial for network safety. In this document we outline the high-level principles that are crucial for any retransmission timeout scheme to leverage. The intent is to then allow implementations of protocols and applications instantiate mechanisms that best realize their specific goals within this framework. These specific mechanisms could be standardized or ad-hoc, but as long as they adhere to the guidelines given in this document they would be consistent with the standards.

## 2 Guidelines

We now list the four guidelines that apply when utilizing a retransmission timeout (RTO).

(1) In the absence of any knowledge about the round-trip time (RTT) of a path the RTO MUST be conservatively set to no less than 1 second, per TCP's current default RTO [RFC2988bis].

[Note: The above assumes [RFC2988bis] becomes the TCP standard as it seems to the author is likely to happen given that the document is in WGLC and has seen no objections thus far. If it ultimately does not pass the above would be revised to 3 seconds, per RFC 2988.]

(2) In steady state the RTO MUST be set based on recent observations of both the RTT and the variance of the RTT. Also, RTT observations MUST be taken regularly. Finally, RTT samples MUST NOT be ambiguous (i.e., using Karn's algorithm [KP87,RFC2988] retransmitted segments produce ambiguous RTT samples unless they explicitly carry a timestamp).

The exact definition of "regularly" is deliberately left vague.

TCP takes an RTT sample once per RTT, or if using the timestamp option [RFC1323] on each acknowledgment arrival. [AP99] shows that taking an RTT sample from each segment transmitted does not improve the performance of TCP's RTO estimator. However, we are aware of no empirical evidence that explores sampling less frequently than once per RTT.

Therefore, for the purpose of this guideline we state that RTT samples SHOULD be taken at least every RTT or as frequently as data is exchanged and ACKed if that happens less frequently than every RTT. However, we also recognize that it may not always be practical to take an RTT sample this often and so state that RTT samples MUST be taken no more than 1 second apart (assuming the data rate allows).

(3) Each time the RTO fires and causes a retransmission the value of the RTO MUST be exponentially backed off such that the next firing requires a longer interval. The backoff may be removed after a successful transmission.

(4) Retransmission timeouts MUST be taken as indications of congestion in the network and the sending rate adapted using a standard mechanism (e.g., TCP collapses the congestion window to one segment).

## 3 Discussion

We note that research has shown the tension between responsiveness and correctness of TCP's RTO seems to be a fundamental tradeoff [AP99]. That is, making the RTO more aggressive (via the EWMA gains, lowering the minimum RTO, etc.) can reduce the time spent waiting on needed RTOs. However, at the same time such aggressiveness leads to more needless RTOs, as well. Therefore, being as aggressive as the guidelines sketched in the last section allow in any particular situation may not be the best course of action (e.g., because an RTO carries a requirement to slow down).

While the tradeoff between responsiveness and correctness seems fundamental, the tradeoff can be made less relevant if the sender can detect and recover from spurious RTOs. Several mechanisms have been proposed for this purpose, such as Eifel [RFC3522], F-RTO [RFC5682] and DSACK [RFC2883,RFC3708]. Using such mechanisms may allow a data originator to tip towards being more responsive without incurring the attendant costs of needless retransmits.

Also, note, that in addition to the experiments discussed in [AP99], the Linux TCP implementation has been using various non-standard RTO mechanisms for many years seemingly without large scale problems (e.g., using different EWMA gains). Also, a number of implementations use minimum RTOs that are less than the 1 second specified in [RFC2988]. While the precise implications of this may show more spurious retransmits (per [AP99]) we are aware of no large scale problems caused by this change to the minimum RTO.

## 4 Security Considerations

Feh!

Acknowledgments

This document benefits from years of discussions with Sally Floyd, Shawn Ostermann, Vern Paxson and the members of the TCPM and TCP-IMPL working groups.

Normative References

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

Informative References

[AP99] Allman, M., V. Paxson, "On Estimating End-to-End Network Path Properties", Proceedings of the ACM SIGCOMM Technical Symposium, September 1999.

[KP87] Karn, P. and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", SIGCOMM 87.

[RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, October 1996.

[RFC2883] Floyd, S., Mahdavi, J., Mathis, M., and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP", RFC 2883, July 2000.

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

[RFC2988bis] Paxson, V., M. Allman, H.K. Chu, M. Sargent, "Computing TCP's Retransmission Timer", Internet-Draft draft-paxson-tcpm-rfc2988bis-01.txt (work in progress), December 2010.

[RFC3517] Blanton, E., Allman, M., Fall, K., and L. Wang, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", RFC 3517, April 2003.

[RFC3522] Ludwig, R., M. Meyer, "The Eifel Detection Algorithm for TCP", RFC 3522, april 2003.

[RFC3708] Blanton, E., M. Allman, "Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions", RFC 3708, February 2004.

[RFC4960] Stweart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5682] Sarolahti, P., M. Kojo, K. Yamamoto, M. Hata, "Forward

RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious
Retransmission Timeouts with TCP", RFC 5682, September 2009.

Authors' Addresses

Mark Allman
International Computer Science Institute
1947 Center St.  Suite 600
Berkeley, CA  94704

Phone: 440-235-1792
EMail: mallman@icir.org
http://www.icir.org/mallman