Network Working Group S. Floyd
Request for Comments: 3742 ICSI
Category: Experimental March 2004

Limited Slow-Start for TCP with Large Congestion Windows

Status of this Memo

Copyright Notice

Abstract

This document describes an optional modification for TCP's slow-start for use with TCP connections with large congestion windows. For TCP connections that are able to use congestion windows of thousands (or tens of thousands) of MSS-sized segments (for MSS the sender's MAXIMUM SEGMENT SIZE), the current slow-start procedure can result in increasing the congestion window by thousands of segments in a single round-trip time. Such an increase can easily result in thousands of packets being dropped in one round-trip time. This is often counter-productive for the TCP flow itself, and is also hard on the rest of the traffic sharing the congested link. This note describes Limited Slow-Start as an optional mechanism for limiting the number of segments by which the congestion window is increased for one window of data during slow-start, in order to improve performance for TCP connections with large congestion windows.

1. Introduction

This note describes an optional modification for TCP's slow-start for use with TCP connections with large congestion windows. For TCP connections that are able to use congestion windows of thousands (or tens of thousands) of MSS-sized segments (for MSS the sender's MAXIMUM SEGMENT SIZE), the current slow-start procedure can result in increasing the congestion window by thousands of segments in a single round-trip time. Such an increase can easily result in thousands of packets being dropped in one round-trip time. This is often counter-productive for the TCP flow itself, and is also hard on the rest of the traffic sharing the congested link. This note describes Limited Slow-Start, limiting the number of segments by which the congestion window is increased for one window of data during slow-start, in order to improve performance for TCP connections with large congestion windows.

When slow-start results in a large increase in the congestion window in one round-trip time, a large number of packets might be dropped in the network (even with carefully-tuned active queue management mechanisms in the routers). This drop of a large number of packets in the network can result in unnecessary retransmit timeouts for the TCP connection. The TCP connection could end up in the congestion avoidance phase with a very small congestion window, and could take a large number of round-trip times to recover its old congestion window. This poor performance is illustrated in [F02].

2. The Proposal for Limited Slow-Start

Limited Slow-Start introduces a parameter, "max_ssthresh", and modifies the slow-start mechanism for values of the congestion window where "cwnd" is greater than "max_ssthresh". That is, during Slow-Start, when

```
cwnd <= max_ssthresh,
```

cwnd is increased by one MSS (MAXIMUM SEGMENT SIZE) for every arriving ACK (acknowledgement) during slow-start, as is always the case. During Limited Slow-Start, when

```
max_ssthresh < cwnd <= ssthresh,
```

the invariant is maintained so that the congestion window is increased during slow-start by at most max_ssthresh/2 MSS per round-trip time. This is done as follows:

```
For each arriving ACK in slow-start:
  If (cwnd <= max_ssthresh)
     cwnd += MSS;
  else
     K = int(cwnd/(0.5 max_ssthresh));
     cwnd += int(MSS/K);
```

Thus, during Limited Slow-Start the window is increased by 1/K MSS for each arriving ACK, for K = int(cwnd/(0.5 max_ssthresh)), instead of by 1 MSS as in standard slow-start [RFC2581].

When

ssthresh < cwnd,

slow-start is exited, and the sender is in the Congestion Avoidance phase.

Our recommendation would be for max_ssthresh to be set to 100 MSS. (This is illustrated in the NS [NS] simulator, for snapshots after May 1, 2002, in the tests "./test-all-tcpHighspeed tcp1A" and "./test-all-tcpHighspeed tcpHighspeed1" in the subdirectory "tcl/lib". Setting max_ssthresh to Infinity causes the TCP connection in NS not to use Limited Slow-Start.)

With Limited Slow-Start, when the congestion window is greater than max_ssthresh, the window is increased by at most 1/2 MSS for each arriving ACK; when the congestion window is greater than 1.5 max_ssthresh, the window is increased by at most 1/3 MSS for each arriving ACK, and so on.

With Limited Slow-Start it takes:

log(max_ssthresh)

round-trip times to reach a congestion window of max_ssthresh, and it takes:

log(max_ssthresh) + (cwnd - max_ssthresh)/(max_ssthresh/2)

round-trip times to reach a congestion window of cwnd, for a congestion window greater than max_ssthresh.

Thus, with Limited Slow-Start with max_ssthresh set to 100 MSS, it would take 836 round-trip times to reach a congestion window of 83,000 packets, compared to 16 round-trip times without Limited Slow-Start (assuming no packet drops). With Limited Slow-Start, the largest transient queue during slow-start would be 100 packets; without Limited Slow-Start, the transient queue during Slow-Start would reach more than 32,000 packets.

By limiting the maximum increase in the congestion window in a round-trip time, Limited Slow-Start can reduce the number of drops during slow-start, and improve the performance of TCP connections with large congestion windows.

3. Experimental Results

Tom Dunigan has added Limited Slow-Start to the Linux 2.4.16 Web100 kernel, and performed experiments comparing TCP with and without Limited Slow-Start [D02]. Results so far show improved performance for TCPs using Limited Slow-Start. There are also several experiments comparing different values for max_ssthresh.

4. Related Proposals

There has been considerable research on mechanisms for the TCP sender to learn about the limitations of the available bandwidth, and to exit slow-start before receiving a congestion indication from the network [VEGAS,H96]. Other proposals set TCP's slow-start parameter ssthresh based on information from previous TCP connections to the same destination [WS95,G00]. This document proposes a simple limitation on slow-start that can be effective in some cases even in the absence of such mechanisms. The max_ssthresh parameter does not replace ssthresh, but is an additional parameter. Thus, Limited Slow-Start could be used in addition to mechanisms for setting ssthresh.

Rate-based pacing has also been proposed to improve the performance of TCP during slow-start [VH97,AD98,KCRP99,ASA00]. We believe that rate-based pacing could be of significant benefit, and could be used in addition to the Limited Slow-Start in this proposal.

Appropriate Byte Counting [RFC3465] proposes that TCP increase its congestion window as a function of the number of bytes acknowledged, rather than as a function of the number of ACKs received. Appropriate Byte Counting is largely orthogonal to this proposal for Limited Slow-Start.

Limited Slow-Start is also orthogonal to other proposals to change mechanisms for exiting slow-start. For example, FACK TCP includes an overdamping mechanism to decrease the congestion window somewhat more aggressively when a loss occurs during slow-start [MM96]. It is also true that larger values for the MSS would reduce the size of the congestion window in units of MSS needed to fill a given pipe, and therefore would reduce the size of the transient queue in units of MSS.

5. Acknowledgements

This proposal is part of a larger proposal for HighSpeed TCP for TCP connections with large congestion windows, and resulted from simulations done by Evandro de Souza, in joint work with Deb Agarwal. This proposal for Limited Slow-Start draws in part from discussions with Tom Kelly, who has used a similar modified slow-start in his own research with congestion control for high-bandwidth connections. We also thank Tom Dunigan for his experiments with Limited Slow-Start.

We thank Andrei Gurtov, Reiner Ludwig, members of the End-to-End Research Group, and members of the Transport Area Working Group, for feedback on this document.

6. Security Considerations

This proposal makes no changes to the underlying security of TCP.

7. References

7.1. Normative References

[RFC2581] Allman, M., Paxson, V. and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

[RFC3465] Allman, M., "TCP Congestion Control with Appropriate Byte Counting (ABC)", RFC 3465, February 2003.

7.2. Informative References

[AD98] Mohit Aron and Peter Druschel, "TCP: Improving Start-up Dynamics by Adaptive Timers and Congestion Control"", TR98-318, Rice University, 1998. URL "http://cs-tr.cs.rice.edu/Dienst/UI/2.0/Describe/ncstrl.rice_cs/TR98-318/".

[ASA00] A. Aggarwal, S. Savage, and T. Anderson, "Understanding the Performance of TCP Pacing", Proceedings of the 2000 IEEE Infocom Conference, Tel-Aviv, Israel, March, 2000. URL "http://www.cs.ucsd.edu/~savage/".

[D02] T. Dunigan, "Floyd's TCP slow-start and AIMD mods", 2002. URL "http://www.csm.ornl.gov/~dunigan/net100/floyd.html".

[F02] S. Floyd, "Performance Problems with TCP's Slow-Start", 2002. URL "http://www.icir.org/floyd/hstcp/slowstart/".

[G00] A. Gurtov, "TCP Performance in the Presence of Congestion and Corruption Losses", Master's Thesis, University of Helsinki, Department of Computer Science, Helsinki, December 2000. URL "http://www.cs.helsinki.fi/u/gurtov/papers/ms_thesis.html".

[H96] J. C. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP", SIGCOMM 96, 1996. URL "http://www.acm.org/sigcomm/sigcomm96/program.html".

[KCRP99] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge, "A Simulation Study of Paced TCP", BBN Technical Memorandum No. 1218, 1999. URL "http://www.ir.bbn.com/documents/techmemos/index.html".

[MM96] M. Mathis and J. Mahdavi, "Forward Acknowledgment: Refining TCP Congestion Control", SIGCOMM, August 1996.

[NS] The Network Simulator (NS). URL "http://www.isi.edu/nsnam/ns/".

[VEGAS] Vegas Web Page, University of Arizona. URL "http://www.cs.arizona.edu/protocols/".

[VH97] Vikram Visweswaraiah and John Heidemann, "Rate Based Pacing for TCP", 1997. URL "http://www.isi.edu/lsam/publications/rate_based_pacing/".

[WS95] G. Wright and W. Stevens, "TCP/IP Illustrated", Volume 2, Addison-Wesley Publishing Company, 1995.

Authors' Address

Sally Floyd
ICIR (ICSI Center for Internet Research)

Phone: +1 (510) 666-2989
EMail: floyd@icir.org
URL: http://www.icir.org/floyd/

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.