Internet Draft
draft-ietf-tcpm-initcwnd-08.txt
Intended status: Experimental

Expiration date: August 2013

J. Chu
N. Dukkipati
Y. Cheng
M. Mathis
Google, Inc.
February 22, 2013

Increasing TCP's Initial Window

Status of this Memo

Copyright Notice

Abstract

This document proposes an experiment to increase the permitted TCP initial window (IW) from between 2 and 4 segments, as specified in RFC 3390, to 10 segments, with a fallback to the existing recommendation when performance issues are detected. It discusses the motivation behind the increase, the advantages and disadvantages of the higher initial window, and presents results from several large scale experiments showing that the higher initial window improves the overall performance of many web services without resulting in a congestion collapse. The document closes with a discussion of usage and deployment for further experimental purpose recommended by the IETF TCP Maintenance and Minor Extensions (TCPM) working group.

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 RFC 2119 [RFC2119].

Table of Contents

1. Introduction

This document proposes to raise the upper bound on TCP's initial window (IW) to 10 segments (maximum 14600B). It is patterned after and borrows heavily from RFC 3390 [RFC3390] and earlier work in this area. Due to lingering concerns about possible side effects to other flows sharing the same network bottleneck, some of the recommendations are conditional on additional monitoring and evaluation.

The primary argument in favor of raising IW follows from the evolving scale of the Internet. Ten segments are likely to fit into queue space available at any broadband access link, even when there are a reasonable number of concurrent connections.

Lower speed links can be treated with environment specific configurations, such that they can be protected from being overwhelmed by large initial window bursts without imposing a suboptimal initial window on the rest of the Internet.

This document reviews the advantages and disadvantages of using a larger initial window, and includes summaries of several large scale experiments showing that an initial window of 10 segments provides benefits across the board for a variety of BW, RTT, and BDP classes. These results show significant benefits for increasing IW for users at much smaller data rates than had been previously anticipated. However, at initial windows larger than 10, the results are mixed. We believe that these mixed results are not intrinsic, but are the consequence of various implementation artifacts, including overly aggressive applications employing many simultaneous connections.

We recommend that all TCP implementations have a settable TCP IW parameter as long as there is a reasonable effort to monitor for possible interactions with other Internet applications and services as described in Section 12. Furthermore, Section 10 details why 10 segments may be an appropriate value, and while that value may continue to rise in the future, this document does not include any supporting evidence for values of IW larger than 10.

In addition, we introduce a minor revision to RFC 3390 and RFC 5681 [RFC5681] to eliminate resetting the initial window when the SYN or SYN/ACK is lost.

The document closes with a discussion of the consensus from the TCPM

working group on the near-term usage and deployment of IW10 in the Internet.

A complementary set of slides for this proposal can be found at [CD10].

2. TCP Modification

This document proposes an increase in the permitted upper bound for TCP's initial window (IW) to 10 segments depending on the MSS. This increase is optional: a TCP MAY start with an initial window that is smaller than 10 segments.

More precisely, the upper bound for the initial window will be

    min (10*MSS, max (2*MSS, 14600))   (1)

This upper bound for the initial window size represents a change from RFC 3390 [RFC3390], which specified that the congestion window be initialized between 2 and 4 segments depending on the MSS.

This change applies to the initial window of the connection in the first round trip time (RTT) of data transmission during or following the TCP three-way handshake. Neither the SYN/ACK nor its acknowledgment (ACK) in the three-way handshake should increase the initial window size.

Note that all the test results described in this document were based on the regular Ethernet MTU of 1500 bytes. Future study of the effect of a different MTU may be needed to fully validate (1) above.

Furthermore, RFC 3390 and RFC 5681 [RFC5681] state that

> "If the SYN or SYN/ACK is lost, the initial window used by a sender after a correctly transmitted SYN MUST be one segment consisting of MSS bytes."

The proposed change to reduce the default RTO to 1 second [RFC6298] increases the chance for spurious SYN or SYN/ACK retransmission, thus unnecessarily penalizing connections with RTT > 1 second if their initial window is reduced to 1 segment. For this reason, it is RECOMMENDED that implementations refrain from resetting the initial window to 1 segment, unless either there have been more than one SYN or SYN/ACK retransmissions, or true loss detection has been made.

TCP implementations use slow start in as many as three different ways: (1) to start a new connection (the initial window); (2) to restart transmission after a long idle period (the restart window);

and (3) to restart transmission after a retransmit timeout (the loss window). The change specified in this document affects the value of the initial window. Optionally, a TCP MAY set the restart window to the minimum of the value used for the initial window and the current value of cwnd (in other words, using a larger value for the restart window should never increase the size of cwnd). These changes do NOT change the loss window, which must remain 1 segment of MSS bytes (to permit the lowest possible window size in the case of severe congestion).

Furthermore, to limit any negative effect that a larger initial window may have on links with limited bandwidth or buffer space, implementations SHOULD fall back to RFC 3390 for the restart window (RW) if any packet loss is detected during either the initial window, or a restart window, and more than 4KB of data is sent. Implementations must also follow RFC6298 [RFC6298] in order to avoid spurious RTO as described in section 9 later.

3. Implementation Issues

HTTP 1.1 specification allows only two simultaneous connections per domain, while web browsers open more simultaneous TCP connections [Ste08], partly to circumvent the small initial window in order to speed up the loading of web pages as described above.

When web browsers open simultaneous TCP connections to the same destination, they are working against TCP's congestion control mechanisms [FF99]. Combining this behavior with larger initial windows further increases the burstiness and unfairness to other traffic in the network. If a larger initial window causes harm to any other flows then local application tuning will reveal that fewer concurrent connections yields better performance for some users. Any content provider deploying IW10 in conjunction with content distributed across multiple domains is explicitly encouraged to perform measurement experiments to detect such problems, and to consider reducing the number of concurrent connections used to retrieve their content.

Some implementations advertise small initial receive window (Table 2 in [Duk10]), effectively limiting how much window a remote host may use. In order to realize the full benefit of the large initial window, implementations are encouraged to advertise an initial receive window of at least 10 segments, except for the circumstances where a larger initial window is deemed harmful. (See the Mitigation section below.)

TCP SACK option ([RFC2018]) was thought to be required in order for the larger initial window to perform well. But measurements from both

a testbed and live tests showed that IW=10 without the SACK option outperforms IW=3 with the SACK option [CW10].

4. Background

TCP congestion window was introduced as part of the congestion control algorithm by Van Jacobson in 1988 [Jac88]. The initial value of one segment was used as the starting point for newly established connections to probe the available bandwidth on the network.

Today's Internet is dominated by web traffic running on top of short-lived TCP connections [IOR2009]. The relatively small initial window has become a limiting factor for the performance of many web applications.

The global Internet has continued to grow, both in speed and penetration. According to the latest report from Akamai [AKAM10], the global broadband (> 2Mbps) adoption has surpassed 50%, propelling the average connection speed to reach 1.7Mbps, while the narrowband (< 256Kbps) usage has dropped to 5%. In contrast, TCP's initial window has remained 4KB for a decade [RFC2414], corresponding to a bandwidth utilization of less than 200Kbps per connection, assuming an RTT of 200ms.

A large proportion of flows on the Internet are short web transactions over TCP, and complete before exiting TCP slow start. Speeding up the TCP flow startup phase, including circumventing the initial window limit, has been an area of active research [RFC6077, Sch08]. Numerous proposals exist [LAJW07, RFC4782, PRAKS02, PK98]. Some require router support [RFC4782, PK98], hence are not practical for the public Internet. Others suggested bold, but often radical ideas, likely requiring more years of research before standardization and deployment.

In the mean time, applications have responded to TCP's "slow" start. Web sites use multiple sub-domains [Bel10] to circumvent HTTP 1.1 regulation on two connections per physical host [RFC2616]. As of today, major web browsers open multiple connections to the same site (up to six connections per domain [Ste08] and the number is growing). This trend is to remedy HTTP serialized download to achieve parallelism and higher performance. But it also implies today most access links are severely under-utilized, hence having multiple TCP connections improves performance most of the time. While raising the initial congestion window may cause congestion for certain users using these browsers, we argue that the browsers and other application need to respect HTTP 1.1 regulation and stop increasing number of simultaneous TCP connections. We believe a modest increase of the initial window will help to stop this trend, and provide the

best interim solution to improve overall user performance, and reduce the server, client, and network load.

Note that persistent connections and pipelining are designed to address some of the above issues with HTTP [RFC2616]. Their presence does not diminish the need for a larger initial window. E.g., data from the Chrome browser show that 35% of HTTP requests are made on new TCP connections. Our test data also shows significant latency reduction with the large initial window even in conjunction with these two HTTP features ([Duk10]).

Also note that packet pacing has been suggested as a possible mechanism to avoid large bursts and their associated harm [VH97]. Pacing is not required in this proposal due to a strong preference for a simple solution. We suspect for packet bursts of a moderate size, packet pacing will not be necessary. This seems to be confirmed by our test results.

More discussion of the increase in initial window, including the choice of 10 segments can be found in [Duk10, CD10].

5. Advantages of Larger Initial Windows

5.1 Reducing Latency

An increase of the initial window from 3 segments to 10 segments reduces the total transfer time for data sets greater than 4KB by up to 4 round trips.

The table below compares the number of round trips between IW=3 and IW=10 for different transfer sizes, assuming infinite bandwidth, no packet loss, and the standard delayed acks with large delayed-ACK timer.

total segments	IW=3	IW=10
3	1	1
6	2	1
10	3	1
12	3	2
21	4	2
25	5	2
33	5	3
46	6	3
51	6	4
78	7	4
79	8	4

120	8	5
127	9	5

For example, with the larger initial window, a transfer of 32 segments of data will require only two rather than five round trips to complete.

5.2 Keeping up with the growth of web object size

RFC 3390 stated that the main motivation for increasing the initial window to 4KB was to speed up connections that only transmit a small amount of data, e.g., email and web. The majority of transfers back then were less than 4KB, and could be completed in a single RTT [All00].

Since RFC 3390 was published, web objects have gotten significantly larger [Chu09, RJ10]. Today only a small percentage of web objects (e.g., 10% of Google's search responses) can fit in the 4KB initial window. The average HTTP response size of gmail.com, a highly scripted web-site, is 8KB (Figure 1. in [Duk10]). The average web page, including all static and dynamic scripted web objects on the page, has seen even greater growth in size [RJ10]. HTTP pipelining [RFC2616] and new web transport protocols such as SPDY [SPDY] allow multiple web objects to be sent in a single transaction, potentially benefiting from an even larger initial window in order to transfer an entire web page in a small number of round trips.

5.3 Recovering faster from loss on under-utilized or wireless links

A greater-than-3-segment initial window increases the chance to recover packet loss through Fast Retransmit rather than the lengthy initial RTO [RFC5681]. This is because the fast retransmit algorithm requires three duplicate ACKs as an indication that a segment has been lost rather than reordered. While newer loss recovery techniques such as Limited Transmit [RFC3042] and Early Retransmit [RFC5827] have been proposed to help speeding up loss recovery from a smaller window, both algorithms can still benefit from the larger initial window because of a better chance to receive more ACKs to react upon.

6. Disadvantages of Larger Initial Windows for the Individual Connection

The larger bursts from an increase in the initial window may cause buffer overrun and packet drop in routers with small buffers, or routers experiencing congestion. This could result in unnecessary retransmit timeouts. For a large-window connection that is able to recover without a retransmit timeout, this could result in an unnecessarily-early transition from the slow-start to the congestion-

avoidance phase of the window increase algorithm.

Premature segment drops are unlikely to occur in uncongested networks with sufficient buffering, or in moderately-congested networks where the congested router uses active queue management (such as Random Early Detection [FJ93, RFC2309, RFC3150]).

Insufficient buffering is more likely to exist in the access routers connecting slower links. A recent study of access router buffer size [DGHS07] reveals the majority of access routers provision enough buffer for 130ms or longer, sufficient to cover a burst of more than 10 packets at 1Mbps speed, but possibly not sufficient for browsers opening simultaneous connections.

A testbed study [CW10] on the effect of the larger initial window with five simultaneously opened connections revealed that, even with limited buffer size on slow links, IW=10 still reduced the total latency of web transactions, although at the cost of higher packet drop rates as compared to IW=3.

Some TCP connections will receive better performance with the larger initial window even if the burstiness of the initial window results in premature segment drops. This will be true if (1) the TCP connection recovers from the segment drop without a retransmit timeout, and (2) the TCP connection is ultimately limited to a small congestion window by either network congestion or by the receiver's advertised window.

7. Disadvantages of Larger Initial Windows for the Network

An increase in the initial window may increase congestion in a network. However, since the increase is one-time only (at the beginning of a connection), and the rest of TCP's congestion backoff mechanism remains in place, it's unlikely the increase by itself will render a network in a persistent state of congestion, or even congestion collapse. This seems to have been confirmed by the large scale web experiments described later.

It should be noted that the above may not hold if applications open a large number of simultaneous connections.

Until this proposal is widely deployed, a fairness issue may exist between flows adopting a larger initial window vs flows that are RFC3390-compliant. Although no severe unfairness has been detected on all the known tests so far, further study on this topic may be warranted.

Some of the discussions from RFC 3390 are still valid for IW=10.

Moreover, it is worth noting that although TCP NewReno increases the chance of duplicate segments when trying to recover multiple packet losses from a large window, the wide support of TCP Selective Acknowledgment (SACK) option [RFC2018] in all major OSes today should keep the volume of duplicate segments in check.

Recent measurements [Get11] provide evidence of extremely large queues (in the order of one second or more) at access networks of the Internet. While a significant part of the buffer bloat is contributed by large downloads/uploads such as video files, emails with large attachments, backups and download of movies to disk, some of the problem is also caused by Web browsing of image heavy sites [Get11]. This queuing delay is generally considered harmful for responsiveness of latency sensitive traffic such as DNS queries, ARP, DHCP, VoIP and Gaming. IW=10 can exacerbate this problem when doing short downloads such as Web browsing [Get11-1]. The mitigations proposed for the broader problem of buffer bloating are also applicable in this case, such as the use of ECN, AQM schemes [CoDel] and traffic classification (QoS).

8. Mitigation of Negative Impact

Much of the negative impact from an increase in the initial window is likely to be felt by users behind slow links with limited buffers. The negative impact can be mitigated by hosts directly connected to a low-speed link advertising a smaller initial receive window than 10 segments. This can be achieved either through manual configuration by the users, or through the host stack auto-detecting the low bandwidth links.

Additional suggestions to improve the end-to-end performance of slow links can be found in RFC 3150 [RFC3150].

9. Interactions with the Retransmission Timer

A large initial window increases the chance of spurious RTO on a low-bandwidth path because the packet transmission time will dominate the round-trip time. To minimize spurious retransmissions, implementations MUST follow RFC 6298 [RFC6298] to restart the retransmission timer with the current value of RTO for each ACK received that acknowledges new data.

For a more detailed discussion see RFC3390, section 6.

10. Experimental Results From Large Scale Cluster Tests

In this section we summarize our findings from large scale Internet experiments with an initial window of 10 segments, conducted via

Google's front-end infrastructure serving a diverse set of applications. We present results from two data centers, each chosen because of the specific characteristics of subnets served: AvgDC has connection bandwidths closer to the worldwide average reported in [AKAM10], with a median connection speed of about 1.7Mbps; SlowDC has a larger proportion of traffic from slow bandwidth subnets with nearly 20% of traffic from connections below 100Kbps, and a third below 256Kbps.

Guided by measurements data, we answer two key questions: what is the latency benefit when TCP connections start with a higher initial window, and on the flip side, what is the cost?

10.1 The benefits

The average web search latency improvement over all responses in AvgDC is 11.7% (68 ms) and 8.7% (72 ms) in SlowDC. We further analyzed the data based on traffic characteristics and subnet properties such as bandwidth (BW), round-trip time (RTT), and bandwidth-delay product (BDP). The average response latency improved across the board for a variety of subnets with the largest benefits of over 20% from high RTT and high BDP networks, wherein most responses can fit within the pipe. Correspondingly, responses from low RTT paths experienced the smallest improvements of about 5%.

Contrary to what we expected, responses from low bandwidth subnets experienced the best latency improvements (between 10-20%) in the buckets 0-56Kbps and 56-256Kbps buckets. We speculate low BW networks observe improved latency for two plausible reasons: 1) fewer slow-start rounds: unlike many large BW networks, low BW subnets with dial-up modems have inherently large RTTs; and 2) faster loss recovery: an initial window larger than 3 segments increases the chances of a lost packet to be recovered through Fast Retransmit as opposed to a lengthy RTO.

Responses of different sizes benefited to varying degrees; those larger than 3 segments naturally demonstrated larger improvements, because they finished in fewer rounds in slow start as compared to the baseline. In our experiments, response sizes <= 3 segments also demonstrated small latency benefits.

To find out how individual subnets performed, we analyzed average latency at a /24 subnet level (an approximation to a user base offered similar set of services by a common ISP). We find even at the subnet granularity, latency improved at all quantiles ranging from 5-11%.

10.2 The cost

To quantify the cost of raising the initial window, we analyzed the data specifically for subnets with low bandwidth and BDP, retransmission rates for different kinds of applications, as well as latency for applications operating with multiple concurrent TCP connections. From our measurements we found no evidence of a negative latency impacts that correlate to BW or BDP alone, but in fact both kinds of subnets demonstrated latency improvements across averages and quantiles.

As expected, the retransmission rate increased modestly when operating with larger initial congestion window. The overall increase in AvgDC is 0.3% (from 1.98% to 2.29%) and in SlowDC is 0.7% (from 3.54% to 4.21%). In our investigation, with the exception of one application, the larger window resulted in a retransmission increase of < 0.5% for services in the AvgDC. The exception is the Maps application that operates with multiple concurrent TCP connections, which increased its retransmission rate by 0.9% in AvgDC and 1.85% in SlowDC (from 3.94% to 5.79%).

In our experiments, the percentage of traffic experiencing retransmissions did not increase significantly. E.g. 90% of web search and maps experienced zero retransmission in SlowDC (percentages are higher for AvgDC); a break up of retransmissions by percentiles indicate that most increases come from portion of traffic already experiencing retransmissions in the baseline with initial window of 3 segments.

Traffic patterns from applications using multiple concurrent TCP connections all operating with a large initial window represent one of the worst case scenarios where latency can be adversely impacted due to bottleneck buffer overflow. Our investigation shows that such a traffic pattern has not been a problem in AvgDC, where all these applications, specifically maps and image thumbnails, demonstrated improved latencies varying from 2-20%. In the case of SlowDC, while these applications continued showing a latency improvement in the mean, their latencies in higher quantiles (96 and above for maps) indicated instances where latency with larger window is worse than the baseline, e.g. the 99% latency for maps has increased by 2.3% (80ms) when compared to the baseline. There is no evidence from our measurements that such a cost on latency is a result of subnet bandwidth alone. Although we have no way of knowing from our data, we conjecture that the amount of buffering at bottleneck links plays a key role in performance of these applications.

Further details on our experiments and analysis can be found in [Duk10, DCCM10].

11. Other Studies

Besides the large scale Internet experiments described above, a number of other studies have been conducted on the effects of IW10 in various environments. These tests were summarized below, with more discussion in Appendix A.

A complete list of tests conducted, with their results and related studies can be found at the [IW10] link.

1. [Sch08] described an earlier evaluation of various Fast Startup approaches, including the "Initial-Start" of 10 MSS.

2. [DCCM10] presented the result from Google's large scale IW10 experiments, with a focus on areas with highly multiplexed links or limited broadband deployment such as Africa and South America.

3. [CW10] contained a testbed study on IW10 performance over slow links. It also studied how short flows with a larger initial window might affect the throughput performance of other co-existing, long lived, bulk data transfers.

4. [Sch11] compared IW10 against a number of other fast startup schemes, and concluded that IW10 works rather well and is also quite fair.

5. [JNDK10] and later [JNDK10-1] studied the effect of IW10 over cellular networks.

6. [AERG11] studied the effect of larger ICW sizes, among other things, on end users' page load time from Yahoo!'s Content Delivery Network.

12. Usage and Deployment Recommendations

Further experiments are required before a larger initial window shall be enabled by default in the Internet. The existing measurement results indicate that this does not cause significant harm to other traffic. However, widespread use in the Internet could reveal issues not known yet, e.g., regarding fairness or impact on latency-sensitive traffic such as VoIP.

Therefore, special care is needed when using this experimental TCP extension, in particular on large-scale systems originating a significant amount of Internet traffic, or on large numbers of individual consumer-level systems that have similar aggregate impact. Anyone (stack vendors, network administrators, etc.) turning on a larger initial window SHOULD ensure that the performance is monitored before and after that change. A key metric to monitor is the rate of packet losses, ECN marking, or segment retransmissions during the

initial burst. The sender SHOULD cache such information about connection setups using an initial window larger than allowed by RFC 3390, and new connections SHOULD fall back to the initial window allowed by RFC 3390 if there is evidence of performance issues. Further experiments are needed on the design of such a cache and corresponding heuristics.

Other relevant metrics that may indicate a need to reduce the IW include an increased overall percentage of packet loss or segment retransmissions as well as application-level metrics such as reduced data transfer completion times or impaired media quality.

It is important also to take into account hosts that do not implement a larger initial window. Furthermore, any deployment of IW10 should be aware that there are potential side effects to real-time traffic (such as VoIP). If users observe any significant deterioration of performance, they SHOULD fall back to an initial window as allowed by RFC 3390 for safety reasons. An increased initial window MUST NOT be turned on by default on systems without such monitoring capabilities.

The IETF TCPM working group is very much interested in further reports from experiments with this specification and encourages the publication of such measurement data. By now, there are no adequate studies available that either prove or or do not prove impact of IW10 to real-time traffic. Further experimentation in this directions in encouraged.

If no significant harm is reported, a follow-up document may revisit the question on whether a larger initial window can be safely used by default in all Internet hosts. Resolution of these experiments and tighter specifications of the suggestions here might be grounds for a future standards track document on the same topic.

13. Related Proposals

Two other proposals [All10, Tou12] have been published to raise TCP's initial window size over a large timescale. Both aim at reducing the uncertain impact of a larger initial window at an Internet wide scale. Moreover, [Tou12] seeks an algorithm to automate the adjustment of IW safely over long haul period.

Although a modest, static increase of IW to 10 may address the near-term need for better web performance, much work is needed from the TCP research community to find a long term solution to the TCP flow startup problem.

14. Security Considerations

This document discusses the initial congestion window permitted for TCP connections. Although changing this value may cause more packet loss, it is highly unlikely to lead to a persistent state of network congestion or even a congestion collapse. Hence it does not raise any known new security issues with TCP.

15. Conclusion

This document suggests a simple change to TCP that will reduce the application latency over short-lived TCP connections or links with long RTTs (saving several RTTs during the initial slow-start phase) with little or no negative impact over other flows. Extensive tests have been conducted through both testbeds and large data centers with most results showing improved latency with only a small increase in the packet retransmission rate. Based on these results we believe a modest increase of IW to 10 is the best solution for the near-term deployment, while scaling IW over the long run remains a challenge for the TCP research community.

16. IANA Considerations

None

17. Acknowledgments

Many people at Google have helped to make the set of large scale tests possible. We would especially like to acknowledge Amit Agarwal, Tom Herbert, Arvind Jain and Tiziana Refice for their major contributions.

Normative References

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

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3390] Allman, M., Floyd, S. and C. Partridge, "Increasing TCP's Initial Window", RFC 3390, October 2002.

[RFC5681] Allman, M., Paxson, V. and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

[RFC5827] Allman, M., Avrachenkov, K., Ayesta, U., Blanton, J. and P. Hurtig, "Early Retransmit for TCP and SCTP", RFC 5827, May 2010.

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

Informative References

[AKAM10] "The State of the Internet, 3rd Quarter 2009", Akamai Technologies, Inc., January 2010. URL=http://www.akamai.com/html/about/press/releases/2010/press_011310_1.html

[AERG11] Al-Fares, M., Elmeleegy, K., Reed, B. and I. Gashinsky, "Overclocking the Yahoo! CDN for Faster Web Page Loads", Internet Measurement Conference, November 2011.

[All00] Allman, M., "A Web Server's View of the Transport Layer", ACM Computer Communication Review, 30(5), October 2000.

[All10] Allman, M., "Initial Congestion Window Specification", Internet-draft draft-allman-tcpm-bump-initcwnd-00.txt, work in progress, last updated November 2010.

[Bel10] Belshe, M., "A Client-Side Argument For Changing TCP Slow Start", January, 2010. URL http://sites.google.com/a/chromium.org/dev/spdy/An_Argument_For_Changing_TCP_Slow_Start.pdf

[CD10] Chu, J. and N. Dukkipati, "Increasing TCP's Initial Window", Presented to 77th IRTF ICCRG & IETF TCPM working group meetings, March 2010. URL http://www.ietf.org/proceedings/77/slides/tcpm-4.pdf

[Chu09] Chu, J., "Tuning TCP Parameters for the 21st Century", Presented to 75th IETF TCPM working group meeting, July 2009. URL http://www.ietf.org/proceedings/75/slides/tcpm-1.pdf.

[CoDel] Nichols, K. and V. Jacobson, "Controlling Queue Delay", ACM QUEUE, May 6, 2012.

[CW10] Chu, J. and Wang, Y., "A Testbed Study on IW10 vs IW3", Presented to 79th IETF TCPM working group meeting, Nov. 2010. URL http://www.ietf.org/proceedings/79/slides/tcpm-0.pdf.

[DCCM10] Dukkipati, D., Cheng, Y., Chu, J. and M. Mathis, "Increasing TCP initial window", Presented to 78th IRTF ICCRG working group meeting, July 2010. URL http://www.ietf.org/proceedings/78/slides/iccrg-3.pdf

[DGHS07] Dischinger, M., Gummadi, K., Haeberlen, A. and S. Saroiu, "Characterizing Residential Broadband Networks", Internet Measurement Conference, October 24-26, 2007.

[Duk10] Dukkipati, N., Refice, T., Cheng, Y., Chu, J., Sutin, N., Agarwal, A., Herbert, T. and J. Arvind, "An Argument for Increasing TCP's Initial Congestion Window", ACM SIGCOMM Computer Communications Review, vol. 40 (2010), pp. 27-33. July 2010.

[FF99] Floyd, S., and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Networking, August 1999.

[FJ93] Floyd, S. and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.

[Get11] Gettys, J., "Bufferbloat: Dark buffers in the Internet", Presented to 80th IETF TSV Area meeting, March 2011. URL http://www.ietf.org/proceedings/80/slides/tsvarea-1.pdf

[Get11-1] Gettys, J., "IW10 Considered Harmful", Internet-draft draft-gettys-iw10-considered-harmful-00, work in progress, August 2011.

[IOR2009] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J. Jahanian, F. and M. Karir, "Atlas Internet Observatory 2009 Annual Report", 47th NANOG Conference, October 2009.

[IW10] "TCP IW10 links", URL http://code.google.com/speed/protocols/tcpm-IW10.html

[Jac88] Jacobson, V., "Congestion Avoidance and Control", Computer Communication Review, vol. 18, no. 4, pp. 314-329, Aug. 1988.

[JNDK10] Jarvinen, I., Nyrhinen. A., Ding, A. and M. Kojo, "A Simulation Study on Increasing TCP's IW", Presented to 78th IRTF ICCRG working group meeting, July 2010. URL http://www.ietf.org/proceedings/78/slides/iccrg-7.pdf

[JNDK10-1] Jarvinen, I., Nyrhinen. A., Ding, A. and M. Kojo, "Effect of IW and Initial RTO changes", Presented to 79th IETF TCPM working group meeting, Nov. 2010. URL http://www.ietf.org/proceedings/79/slides/tcpm-1.pdf

[LAJW07] Liu, D., Allman, M., Jin, S. and L. Wang, "Congestion Control Without a Startup Phase", Protocols for Fast, Long Distance Networks (PFLDnet) Workshop, February 2007. URL http://www.icir.org/mallman/papers/jumpstart-pfldnet07.pdf

[PK98] Padmanabhan V.N. and R. Katz, "TCP Fast Start: A technique for speeding up web transfers", in Proceedings of IEEE Globecom '98 Internet Mini-Conference, 1998.

[PRAKS02] Partridge, C., Rockwell, D., Allman, M., Krishnan, R. and J. Sterbenz, "A Swifter Start for TCP", Technical Report No. 8339, BBN Technologies, March 2002.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J. and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[RFC2414] Allman, M., Floyd, S. and C. Partridge, "Increasing TCP's Initial Window", RFC 2414, September 1998.

[RFC3042] Allman, M., Balakrishnan, H. and S. Floyd, "Enhancing TCP's Loss Recovery Using Limited Transmit", RFC 3042, January 2001.

[RFC3150] Dawkins, S., Montenegro, G., Kojo, M. and V. Magret, "End-to-end Performance Implications of Slow Links", BCP 0048, July 2001.

[RFC4782] Floyd, S., Allman, M., Jain, A. and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, January 2007.

[RFC6077] Papadimitriou, D., Welzl, M., Scharf, M. and B. Briscoe, "Open Research Issues in Internet Congestion Control", section 3.4, RFC 6077, February 2011.

[RJ10] Ramachandran, S. and A. Jain, "Aggregate Statistics of Size Related Metrics of Web Pages metrics", May 2010. URL http://code.google.com/speed/articles/web-metrics.html

[Sch08] Scharf, M., "Quick-Start, Jump-Start, and Other Fast Startup Approaches", Internet Research Task Force ICCRG, November 17, 2008. URL http://www.ietf.org/proceedings/73/slides/iccrg-2.pdf

[Sch11] Scharf, M., "Performance and Fairness Evaluation of IW10 and Other Fast Startup Schemes", Internet Research Task Force ICCRG, March 2011. URL http://www.ietf.org/proceedings/80/slides/iccrg-1.pdf

[Sch11-1] Scharf, M., "Comparison of end-to-end and network-supported fast startup congestion control schemes", Computer Networks, Feb. 2011. URL http://dx.doi.org/10.1016/j.comnet.2011.02.002

[SPDY] "SPDY: An experimental protocol for a faster web", URL http://dev.chromium.org/spdy

[Ste08] Sounders S., "Roundup on Parallel Connections", High Performance Web Sites blog. March 2008. URL http://www.stevesouders.com/blog/2008/03/20/roundup-on-parallel-connections

[Tou12] Touch, J., "Automating the Initial Window in TCP", Internet-draft draft-touch-tcpm-automatic-iw-03.txt, work in progress, July 16, 2012.

[VH97] Visweswaraiah, V. and J. Heidemann, "Improving Restart of Idle TCP Connections", Technical Report 97-661, University of Southern California, November 1997.

Appendix A - List of Concerns and Corresponding Test Results

Concerns have been raised since this proposal was first published based on a set of large scale experiments. To better understand the impact of a larger initial window in order to confirm or dismiss these concerns, additional tests have been conducted using either large scale clusters, simulations, or real testbeds. The following attempts to compile the list of concerns and summarize findings from relevant tests.

o How complete are various tests in covering many different traffic patterns?

The large scale Internet experiments conducted at Google front-end infrastructure covered a large portfolio of services beyond web search. It includes Gmail, Google Maps, Photos, News, Sites, Images,..., etc, covering a wide variety of traffic sizes and patterns. One notable exception is YouTube because we don't think the large initial window will have much material impact, either positive or negative, on bulk data services.

[CW10] contains some result from a testbed study on how short flows with a larger initial window might affect the throughput performance of other co-existing, long lived, bulk data transfers.

o Larger bursts from the increase in the initial window cause significantly more packet drops

All the tests conducted on this subject [Duk10, Sch11, Sch11-1, CW10] so far have shown only modest increase on packet drops. The only exception is from the testbed study [CW10] when under extremely high load and/or simultaneous opens. But under those conditions both IW=3 and IW=10 suffered very high packet loss rates though.

o A large initial window may severely impact TCP performance over highly multiplexed links still common in developing regions

Our large scale experiments described in section 10 above also covered Africa and South America. Measurement data from those regions [DCCM10] revealed improved latency even for those services that employ multiple simultaneous connections, at the cost of small increase in the retransmission rate. It seems that the round trip savings from a larger initial window more than make up the time spent on recovering more lost packets.

Similar phenomenon have also been observed from testbed study [CW10].

o Why 10 segments?

Questions have been raised on how the number 10 was picked. We have tried different sizes in our large scale experiments, and found that 10 segments seem to give most of the benefits for the services we tested while not causing significant increase in the retransmission rates. Going forward 10 segments may turn out to be too small when the average of web object sizes continue to grow. But a scheme to right size the initial window automatically over long timescales has yet to be developed.

o Need more thorough analysis of the impact on slow links

Although [Duk10] showed the large initial window reduced the average latency even for the dialup link class of only 56Kbps in bandwidth, more studied were needed in order to understand the effect of IW10 on slow links at the microscopic level. [CW10] was conducted for this purpose.

Testbeds in [CW10] emulated a 300ms RTT, bottleneck link bandwidth as low as 64Kbps, and route queue size as low as 40 packets. A large combination of test parameters were used. Almost all tests showed varying degree of latency improvement from IW=10, with only a modest increase in the packet drop rate until a very high load was injected. The testbed result was consistent with both the large scale data center experiments [CD10, DCCM10] and a separate study using NSC simulations [Sch11, Sch11-1].

o How will the larger initial window affect flows with initial windows 4KB or less?

Flows with the larger initial window will likely grab more bandwidth from a bottleneck link when competing against flows with smaller initial window, at least initially. How long will this "unfairness" last? Will there be any "capture effect" where flows with larger initial window possess a disproportional share of bandwidth beyond just a few round trips?

If there is any "unfairness" issue from flows with different initial windows, it did not show up in the large scale experiments, as the average latency for the bucket of all responses < 4KB did not seem to be affected by the presence of many other larger responses employing large initial window. As a matter of fact they seemed to benefit from the large initial window too, as shown in Figure 7 of [Duk10].

The same phenomenon seems to exist in the testbed experiments [CW10]. Flows with IW=3 only suffered slightly when competing

against flows with IW=10 in light to median loads. Under high load both flows' latency improved when mixed together. Also long-lived, background bulk-data flows seemed to enjoy higher throughput when running against many foreground short flows of IW=10 than against short flows of IW=3. One plausible explanation was IW=10 enabled short flows to complete sooner, leaving more room for the long-lived, background flows.

A study using NSC simulator has also concluded that IW=10 works rather well and is quite fair against IW=3 [Sch11, Sch11-1].

o How will a larger initial window perform over cellular networks?

Some simulation studies [JNDK10, JNDK10-1] have been conducted to study the effect of a larger initial window on wireless links from 2G to 4G networks (EGDE/HSPA/LTE). The overall result seems mixed in both raw performance and the fairness index.

Author's Addresses

Jerry Chu
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA
EMail: hkchu@google.com

Nandita Dukkipati
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA
EMail: nanditad@google.com

Yuchung Cheng
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA
EMail: ycheng@google.com

Matt Mathis
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA
EMail: mattmathis@google.com

Acknowledgment

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