

Network Working Group  
Request for Comments: 2414  
Category: Experimental

M. Allman  
NASA Lewis/Sterling Software  
S. Floyd  
LBNL  
C. Partridge  
BBN Technologies  
September 1998

## Increasing TCP's Initial Window

### Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

### Abstract

This document specifies an increase in the permitted initial window for TCP from one segment to roughly 4K bytes. This document discusses the advantages and disadvantages of such a change, outlining experimental results that indicate the costs and benefits of such a change to TCP.

### 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](#)].

### 1. TCP Modification

This document specifies an increase in the permitted upper bound for TCP's initial window from one segment to between two and four segments. In most cases, this change results in an upper bound on the initial window of roughly 4K bytes (although given a large segment size, the permitted initial window of two segments could be significantly larger than 4K bytes). The upper bound for the initial window is given more precisely in (1):

$$\min (4 * \text{MSS}, \max (2 * \text{MSS}, 4380 \text{ bytes})) \quad (1)$$

Equivalently, the upper bound for the initial window size is based on the maximum segment size (MSS), as follows:

```
If (MSS <= 1095 bytes)
    then win <= 4 * MSS;
If (1095 bytes < MSS < 2190 bytes)
    then win <= 4380;
If (2190 bytes <= MSS)
    then win <= 2 * MSS;
```

This increased initial window is optional: that a TCP MAY start with a larger initial window, not that it SHOULD.

This upper bound for the initial window size represents a change from [RFC 2001](#) [S97], which specifies that the congestion window be initialized to one segment. If implementation experience proves successful, then the intent is for this change to be incorporated into a revision to [RFC 2001](#).

This change applies to the initial window of the connection in the first round trip time (RTT) of transmission 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 above that outlined in equation (1). If the SYN or SYN/ACK is lost, the initial window used by a sender after a correctly transmitted SYN MUST be one segment.

TCP implementations use slow start in as many as three different ways: (1) to start a new connection (the initial window); (2) to restart a transmission after a long idle period (the restart window); and (3) to restart after a retransmit timeout (the loss window). The change proposed 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 (to permit the lowest possible window size in the case of severe congestion).

## [2.](#) Implementation Issues

When larger initial windows are implemented along with Path MTU Discovery [[MD90](#)], and the MSS being used is found to be too large,

the congestion window `cwnd` SHOULD be reduced to prevent large bursts of smaller segments. Specifically, `cwnd` SHOULD be reduced by the ratio of the old segment size to the new segment size.

When larger initial windows are implemented along with Path MTU Discovery [[MD90](#)], alternatives are to set the "Don't Fragment" (DF) bit in all segments in the initial window, or to set the "Don't Fragment" (DF) bit in one of the segments. It is an open question which of these two alternatives is best; we would hope that implementation experiences will shed light on this. In the first case of setting the DF bit in all segments, if the initial packets are too large, then all of the initial packets will be dropped in the network. In the second case of setting the DF bit in only one segment, if the initial packets are too large, then all but one of the initial packets will be fragmented in the network. When the second case is followed, setting the DF bit in the last segment in the initial window provides the least chance for needless retransmissions when the initial segment size is found to be too large, because it minimizes the chances of duplicate ACKs triggering a Fast Retransmit. However, more attention needs to be paid to the interaction between larger initial windows and Path MTU Discovery.

The larger initial window proposed in this document is not intended as an encouragement for web browsers to open multiple simultaneous TCP connections all with large initial windows. When web browsers open simultaneous TCP connections to the same destination, this works against TCP's congestion control mechanisms [[FF98](#)], regardless of the size of the initial window. Combining this behavior with larger initial windows further increases the unfairness to other traffic in the network.

### 3. Advantages of Larger Initial Windows

1. When the initial window is one segment, a receiver employing delayed ACKs [[Bra89](#)] is forced to wait for a timeout before generating an ACK. With an initial window of at least two segments, the receiver will generate an ACK after the second data segment arrives. This eliminates the wait on the timeout (often up to 200 msec).

2. For connections transmitting only a small amount of data, a larger initial window reduces the transmission time (assuming at most moderate segment drop rates). For many email (SMTP [[Pos82](#)]) and web page (HTTP [[BLFN96](#), [FJGFBL97](#)]) transfers that are less than 4K bytes, the larger initial window would reduce the data transfer time to a single RTT.
3. For connections that will be able to use large congestion windows, this modification eliminates up to three RTTs and a delayed ACK timeout during the initial slow-start phase. This

would be of particular benefit for high-bandwidth large-propagation-delay TCP connections, such as those over satellite links.

#### 4. Disadvantages of Larger Initial Windows for the Individual Connection

In high-congestion environments, particularly for routers that have a bias against bursty traffic (as in the typical Drop Tail router queues), a TCP connection can sometimes be better off starting with an initial window of one segment. There are scenarios where a TCP connection slow-starting from an initial window of one segment might not have segments dropped, while a TCP connection starting with an initial window of four segments might experience unnecessary retransmits due to the inability of the router to handle small bursts. This could result in an unnecessary retransmit timeout. 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. These 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](#)])).

Some TCP connections will receive better performance with the higher 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.

## 5. Disadvantages of Larger Initial Windows for the Network

In terms of the potential for congestion collapse, we consider two separate potential dangers for the network. The first danger would be a scenario where a large number of segments on congested links were duplicate segments that had already been received at the receiver. The second danger would be a scenario where a large number of segments on congested links were segments that would be dropped later in the network before reaching their final destination.

In terms of the negative effect on other traffic in the network, a potential disadvantage of larger initial windows would be that they increase the general packet drop rate in the network. We discuss these three issues below.

### Duplicate segments:

As described in the previous section, the larger initial window could occasionally result in a segment dropped from the initial window, when that segment might not have been dropped if the sender had slow-started from an initial window of one segment. However, [Appendix A](#) shows that even in this case, the larger initial window would not result in the transmission of a large number of duplicate segments.

### Segments dropped later in the network:

How much would the larger initial window for TCP increase the number of segments on congested links that would be dropped before reaching their final destination? This is a problem that can only occur for connections with multiple congested links, where some segments might use scarce bandwidth on the first congested link along the path, only to be dropped later along the path.

First, many of the TCP connections will have only one congested link along the path. Segments dropped from these connections do

not "waste" scarce bandwidth, and do not contribute to congestion collapse.

However, some network paths will have multiple congested links, and segments dropped from the initial window could use scarce bandwidth along the earlier congested links before ultimately being dropped on subsequent congested links. To the extent that the drop rate is independent of the initial window used by TCP segments, the problem of congested links carrying segments that will be dropped before reaching their destination will be similar for TCP connections that start by sending four segments or one segment.

An increased packet drop rate:

For a network with a high segment drop rate, increasing the TCP initial window could increase the segment drop rate even further. This is in part because routers with Drop Tail queue management have difficulties with bursty traffic in times of congestion. However, given uncorrelated arrivals for TCP connections, the larger TCP initial window should not significantly increase the segment drop rate. Simulation-based explorations of these issues are discussed in [Section 7.2](#).

These potential dangers for the network are explored in simulations and experiments described in the section below. Our judgement would be, while there are dangers of congestion collapse in the current Internet (see [[FF98](#)] for a discussion of the dangers of congestion collapse from an increased deployment of UDP connections without end-to-end congestion control), there is no such danger to the network from increasing the TCP initial window to 4K bytes.

## [6.](#) Typical Levels of Burstiness for TCP Traffic.

Larger TCP initial windows would not dramatically increase the burstiness of TCP traffic in the Internet today, because such traffic is already fairly bursty. Bursts of two and three segments are already typical of TCP [[Flo97](#)]; A delayed ACK (covering two previously unacknowledged segments) received during congestion

avoidance causes the congestion window to slide and two segments to be sent. The same delayed ACK received during slow start causes the window to slide by two segments and then be incremented by one segment, resulting in a three-segment burst. While not necessarily typical, bursts of four and five segments for TCP are not rare. Assuming delayed ACKs, a single dropped ACK causes the subsequent ACK to cover four previously unacknowledged segments. During congestion avoidance this leads to a four-segment burst and during slow start a five-segment burst is generated.

There are also changes in progress that reduce the performance problems posed by moderate traffic bursts. One such change is the deployment of higher-speed links in some parts of the network, where a burst of 4K bytes can represent a small quantity of data. A second change, for routers with sufficient buffering, is the deployment of queue management mechanisms such as RED, which is designed to be tolerant of transient traffic bursts.

## [7.](#) Simulations and Experimental Results

### [7.1](#) Studies of TCP Connections using that Larger Initial Window

This section surveys simulations and experiments that have been used to explore the effect of larger initial windows on the TCP connection using that larger window. The first set of experiments explores performance over satellite links. Larger initial windows have been shown to improve performance of TCP connections over satellite channels [[All97b](#)]. In this study, an initial window of four segments (512 byte MSS) resulted in throughput improvements of up to 30% (depending upon transfer size). [[KAGT98](#)] shows that the use of larger initial windows results in a decrease in transfer time in HTTP tests over the ACTS satellite system. A study involving simulations

of a large number of HTTP transactions over hybrid fiber coax (HFC) indicates that the use of larger initial windows decreases the time required to load WWW pages [[Nic97](#)].

A second set of experiments has explored TCP performance over dialup modem links. In experiments over a 28.8 bps dialup channel [[All97a](#), [AH098](#)], a four-segment initial window decreased the transfer time of a 16KB file by roughly 10%, with no accompanying increase in the drop

rate. A particular area of concern has been TCP performance over low speed tail circuits (e.g., dialup modem links) with routers with small buffers. A simulation study [SP97] investigated the effects of using a larger initial window on a host connected by a slow modem link and a router with a 3 packet buffer. The study concluded that for the scenario investigated, the use of larger initial windows was not harmful to TCP performance. Questions have been raised concerning the effects of larger initial windows on the transfer time for short transfers in this environment, but these effects have not been quantified. A question has also been raised concerning the possible effect on existing TCP connections sharing the link.

## [7.2](#) Studies of Networks using Larger Initial Windows

This section surveys simulations and experiments investigating the impact of the larger window on other TCP connections sharing the path. Experiments in [All97a, AH098] show that for 16 KB transfers to 100 Internet hosts, four-segment initial windows resulted in a small increase in the drop rate of 0.04 segments/transfer. While the drop rate increased slightly, the transfer time was reduced by roughly 25% for transfers using the four-segment (512 byte MSS) initial window when compared to an initial window of one segment.

One scenario of concern is heavily loaded links. For instance, a couple of years ago, one of the trans-Atlantic links was so heavily loaded that the correct congestion window size for a connection was about one segment. In this environment, new connections using larger initial windows would be starting with windows that were four times too big. What would the effects be? Do connections thrash?

A simulation study in [PN98] explores the impact of a larger initial window on competing network traffic. In this investigation, HTTP and FTP flows share a single congested gateway (where the number of HTTP and FTP flows varies from one simulation set to another). For each simulation set, the paper examines aggregate link utilization and packet drop rates, median web page delay, and network power for the FTP transfers. The larger initial window generally resulted in increased throughput, slightly-increased packet drop rates, and an increase in overall network power. With the exception of one scenario, the larger initial window resulted in an increase in the



a one-segment initial window; in this scenario, the drop rate increased from 3.5% with one-segment initial windows, to 4.5% with four-segment initial windows. The overall conclusions were that increasing the TCP initial window to three packets (or 4380 bytes) helps to improve perceived performance.

Morris [Mor97] investigated larger initial windows in a very congested network with transfers of size 20K. The loss rate in networks where all TCP connections use an initial window of four segments is shown to be 1-2% greater than in a network where all connections use an initial window of one segment. This relationship held in scenarios where the loss rates with one-segment initial windows ranged from 1% to 11%. In addition, in networks where connections used an initial window of four segments, TCP connections spent more time waiting for the retransmit timer (RTO) to expire to resend a segment than was spent when using an initial window of one segment. The time spent waiting for the RTO timer to expire represents idle time when no useful work was being accomplished for that connection. These results show that in a very congested environment, where each connection's share of the bottleneck bandwidth is close to one segment, using a larger initial window can cause a perceptible increase in both loss rates and retransmit timeouts.

## 8. Security Considerations

This document discusses the initial congestion window permitted for TCP connections. Changing this value does not raise any known new security issues with TCP.

## 9. Conclusion

This document proposes a small change to TCP that may be beneficial to short-lived TCP connections and those over links with long RTTs (saving several RTTs during the initial slow-start phase).

## 10. Acknowledgments

We would like to acknowledge Vern Paxson, Tim Shepard, members of the End-to-End-Interest Mailing List, and members of the IETF TCP Implementation Working Group for continuing discussions of these issues for discussions and feedback on this document.

11. References

- [All97a] Mark Allman. An Evaluation of TCP with Larger Initial Windows. 40th IETF Meeting -- TCP Implementations WG. December, 1997. Washington, DC.
- [AH098] Mark Allman, Chris Hayes, and Shawn Ostermann, An Evaluation of TCP with Larger Initial Windows, March 1998. Submitted to ACM Computer Communication Review. URL: "<http://gigahertz.lerc.nasa.gov/~mallman/papers/initwin.ps>".
- [All97b] Mark Allman. Improving TCP Performance Over Satellite Channels. Master's thesis, Ohio University, June 1997.
- [BLFN96] Berners-Lee, T., Fielding, R., and H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0", [RFC 1945](#), May 1996.
- [Bra89] Braden, R., "Requirements for Internet Hosts -- Communication Layers", STD 3, [RFC 1122](#), October 1989.
- [FF96] Fall, K., and Floyd, S., Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. Computer Communication Review, 26(3), July 1996.
- [FF98] Sally Floyd, Kevin Fall. Promoting the Use of End-to-End Congestion Control in the Internet. Submitted to IEEE Transactions on Networking. URL "<http://www-nrg.ee.lbl.gov/floyd/end2end-paper.html>".
- [FJGFBL97] Fielding, R., Mogul, J., Gettys, J., Frystyk, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2068](#), January 1997.
- [FJ93] Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.
- [Flo94] Floyd, S., TCP and Explicit Congestion Notification. Computer Communication Review, 24(5):10-23, October 1994.
- [Flo96] Floyd, S., Issues of TCP with SACK. Technical report, January 1996. Available from <http://www-nrg.ee.lbl.gov/floyd/>.
- [Flo97] Floyd, S., Increasing TCP's Initial Window. Viewgraphs,

[RFC 2414](#)                      Increasing TCP's Initial Window                      September 1998

- [KAGT98]      Hans Kruse, Mark Allman, Jim Griner, Diepchi Tran. HTTP Page Transfer Rates Over Geo-Stationary Satellite Links. March 1998. Proceedings of the Sixth International Conference on Telecommunication Systems. URL "<http://gigahertz.lerc.nasa.gov/~mallman/papers/nash98.ps>".
- [MD90]        Mogul, J., and S. Deering, "Path MTU Discovery", [RFC 1191](#), November 1990.
- [MMFR96]     Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", [RFC 2018](#), October 1996.
- [Mor97]       Robert Morris. Private communication, 1997. Cited for acknowledgement purposes only.
- [Nic97]       Kathleen Nichols. Improving Network Simulation with Feedback. Com21, Inc. Technical Report. Available from <http://www.com21.com/pages/papers/068.pdf>.
- [PN98]        Poduri, K., and K. Nichols, "Simulation Studies of Increased Initial TCP Window Size", [RFC 2415](#), September 1998.
- [Pos82]       Postel, J., "Simple Mail Transfer Protocol", STD 10, [RFC 821](#), August 1982.
- [RF97]        Ramakrishnan, K., and S. Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IPv6 and to TCP", Work in Progress.
- [RFC2119]     Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [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.

[S97] Stevens, W., "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", [RFC 2001](#), January 1997.

[SP97] Shepard, T., and C. Partridge, "When TCP Starts Up With Four Packets Into Only Three Buffers", [RFC 2416](#), September 1998.

Allman, et. al.

Experimental

[Page 10]

---

[RFC 2414](#)

Increasing TCP's Initial Window

September 1998

## 12. Author's Addresses

Mark Allman  
NASA Lewis Research Center/Sterling Software  
21000 Brookpark Road  
MS 54-2  
Cleveland, OH 44135

EMail: [mallman@lerc.nasa.gov](mailto:mallman@lerc.nasa.gov)  
<http://gigahertz.lerc.nasa.gov/~mallman/>

Sally Floyd  
Lawrence Berkeley National Laboratory  
One Cyclotron Road  
Berkeley, CA 94720

EMail: [floyd@ee.lbl.gov](mailto:floyd@ee.lbl.gov)

Craig Partridge  
BBN Technologies  
10 Moulton Street  
Cambridge, MA 02138

EMail: [craig@bbn.com](mailto:craig@bbn.com)

### 13. Appendix - Duplicate Segments

In the current environment (without Explicit Congestion Notification [[Flo94](#)] [[RF97](#)]), all TCPs use segment drops as indications from the network about the limits of available bandwidth. We argue here that the change to a larger initial window should not result in the sender retransmitting a large number of duplicate segments that have already been received at the receiver.

If one segment is dropped from the initial window, there are three different ways for TCP to recover: (1) Slow-starting from a window of one segment, as is done after a retransmit timeout, or after Fast Retransmit in Tahoe TCP; (2) Fast Recovery without selective acknowledgments (SACK), as is done after three duplicate ACKs in Reno TCP; and (3) Fast Recovery with SACK, for TCP where both the sender and the receiver support the SACK option [[MMFR96](#)]. In all three cases, if a single segment is dropped from the initial window, no duplicate segments (i.e., segments that have already been received at the receiver) are transmitted. Note that for a TCP sending four 512-byte segments in the initial window, a single segment drop will not require a retransmit timeout, but can be recovered from using the Fast Retransmit algorithm (unless the retransmit timer expires prematurely). In addition, a single segment dropped from an initial window of three segments might be repaired using the fast retransmit algorithm, depending on which segment is dropped and whether or not

delayed ACKs are used. For example, dropping the first segment of a three segment initial window will always require waiting for a timeout. However, dropping the third segment will always allow recovery via the fast retransmit algorithm, as long as no ACKs are lost.

Next we consider scenarios where the initial window contains two to four segments, and at least two of those segments are dropped. If all segments in the initial window are dropped, then clearly no duplicate segments are retransmitted, as the receiver has not yet received any segments. (It is still a possibility that these dropped segments used scarce bandwidth on the way to their drop point; this issue was discussed in [Section 5](#).)

When two segments are dropped from an initial window of three segments, the sender will only send a duplicate segment if the first two of the three segments were dropped, and the sender does not receive a packet with the SACK option acknowledging the third segment.

When two segments are dropped from an initial window of four segments, an examination of the six possible scenarios (which we don't go through here) shows that, depending on the position of the

dropped packets, in the absence of SACK the sender might send one duplicate segment. There are no scenarios in which the sender sends two duplicate segments.

When three segments are dropped from an initial window of four segments, then, in the absence of SACK, it is possible that one duplicate segment will be sent, depending on the position of the dropped segments.

The summary is that in the absence of SACK, there are some scenarios with multiple segment drops from the initial window where one duplicate segment will be transmitted. There are no scenarios where more than one duplicate segment will be transmitted. Our conclusion is that the number of duplicate segments transmitted as a result of a larger initial window should be small.

#### 14. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing

the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.