Internet Engineering Task Force
TCP Implementers Working Group
INTERNET DRAFT
Expires August, 1998

K. Poduri
K. Nichols
Bay Networks
February, 1998

Simulation Studies of Increased Initial TCP Window Size
<draft-tcpimpl-poduri-00.txt>

Status of this Memo

Abstract

An increase in the permissible initial window size of a TCP connection, from one segment to three or four segments, has been under discussion in the tcp-impl working group. This document covers some simulation studies of the effects of increasing the initial window size of TCP. Both long-lived TCP connections (file transfers) and short-lived web-browsing style connections were modeled. The simulations were performed using the publicly available ns-2 simulator and our custom models and files are also available. A pdf version of this document is available and recommended for the figures it contains.

1. Introduction

We present results from a set of simulations with increased TCP initial window (IW). The main objectives were to explore the conditions under which the larger IW was a "win" and to determine the effects, if any, the larger IW might have on other traffic flows using an IW of one segment.

This study was inspired by discussions at the Munich IETF tcp-impl and tcp-sat meetings. A proposal to increase the IW size to about 4K bytes (4380 bytes in the case of 1460 byte segments) was discussed. Concerns about both the utility of the increase and its effect on other traffic were raised. Some studies were presented showing the positive effects of increased IW on individual connections, but no studies were shown with a wide variety of simultaneous traffic flows. It appeared that some of the questions being raised could be addressed in an ns-2 simulation. Early results from our simulations were previously posted to the tcp-impl mailing list and presented at the tcp-impl WG meeting at the December 1997 IETF.

2. Model and Assumptions

We simulated a network topology with a bottleneck link as shown:

```
    10Mb,                                     10Mb,
     (All 4 links)                          (All 4 links)
C   n2__________                              ______ n6     S
l   n3__________\                            /______ n7     e
i               \\              1.5Mb,50ms  //              r
e                n0 ------------------------ n1             v
n   n4___________//                         \ \_____ n8     e
t   n5___________/                           \______ n9     r
s                                                           s
              URLs -->          <--- FTP & Web data
```

File downloading and web-browsing clients are attached to the nodes (n2-n5) on the left-hand side. These clients are served by the FTP and Web servers attached to the nodes (n6-n9) on the right-hand side. The links to and from those nodes are at 10 Mbps. The bottleneck link is between n1 and n0. All links are bi-directional, but only ACKs, SYNs, FINs, and URLs are flowing from left to right. Some simulations were also performed with data traffic flowing from right to left simultaneously, but it had no affect on the results.

In the simulations we assumed that all ftps transferred 1 MB files and that all web pages had exactly three embedded URLs. The web clients are browsing quite aggressively, requesting a new page after a delay uniformly randomly distributed between 1 and 5 seconds. This is not meant to realistically model a single user's web-browsing pattern, but to create a heavy traffic load (many connections opening with the initial window size under test) whose individual TCP connections accurately reflect real web traffic. Some discussion of these models as used in earlier studies is available in references [3] and [4].

The maximum TCP window was set to 11 packets, maximum packet/segment size to 1460 bytes, and buffer sizes were set at 25 packets. (The ns-2 TCPs require setting window sizes and buffer sizes in number of packets. In our tcp-full code most of the internal parameters have been set to be byte-oriented, but external values must still be set in number of packets.) In our simulations, we varied the number of data segments sent into a new TCP connection from one to four, keeping all segments at 1460 bytes. A dropped packet causes a restart window of one segment to be used, just as in current practice.

For ns-2 users: The tcp-full code was modified to use an "application" class and three application client-server pairs were written: a simple file transfer (ftp), a model of http1.0 style web connection and a very rough model of http1.1 style web connection. The required files and scripts for these simulations are available at the site ftp://ftp.ee.lbl.gov/ecn/IW.{tar,tar.Z}, http://www-nrg.ee.lbl.gov/floyd/tcp_init_win_html

Simulations were run with 8, 16, 32 web clients and a number of ftp clients ranging from 0 to 3. The IW was varied from 1 to 4, though the 4-packet case lies beyond what is currently recommended. The figures of merit used were goodput, the median page delay seen by the web clients (time to fetch the primary plus all embedded URLs) and the median file transfer delay seen by the ftp clients. The simulated run time was rather large, 360 seconds, to ensure an adequate sample. (Median values remained stable for twice that time.)

3. Results

In our simulations, we varied the number of file transfer clients in order to change the congestion of the link. Recall that our ftp clients continuously request 1 Mbyte transfers, so the link utilization is over 90% when even a single ftp client is present. When three file transfer clients are running simultaneously, the resultant congestion is somewhat pathological, with results fairly unstable and having more to do with congestion effects than initial window effects. Though all connections use the same initial window, the effect of a change in the initial window of a 1 Mbyte file transfer is not detectable, thus we focus on the web browsing connections. (In the tables, we use "webs" to indicate number of web clients and "ftps" to indicate the number of file transfer clients attached.) Table 1 shows the median delays experienced by the web transfers with an increase in the TCP IW. There is clearly an improvement in transfer delays for the web connections with increase in the IW, in many cases on the order of 30%. The steepness of the performance improvement going from an IW of 1 to an IW of 2 is mainly due to the distribution of files fetched by each URL (see references [1] and [2]); the median size of both primary and in-line URLs fits completely into two packets. If file distributions change, the shape of this curve may change also.

Table 1: Median web page delay (in ms)

#Webs	#FTPs	IW=1	IW=2	IW=3	IW=4
8	0	0.56	0.48	0.46	0.47
8	1	1.06	0.86	0.79	0.72
8	2	1.18	0.99	0.98	0.84
8	3	1.26	1.11	1.02	0.92
16	0	0.64	0.57	0.54	0.52
16	1	1.04	0.86	0.79	0.67
16	2	1.22	1.01	0.97	0.91
16	3	1.31	1.17	1.03	1.02
32	0	0.92	0.74	0.69	0.68
32	1	1.19	0.98	0.85	0.94
32	2	1.43	1.09	0.93	0.95
32	3	1.56	1.26	1.10	1.04

Table 2 shows the bottleneck link utilization and packet drop percentages of the same experiment. Packet drop rates did increase with IW, but the increase in drop percentage was within one percent of the IW=1 value with the exception of the single most pathological overload.

Table 2: Link utilization and packet drop rates

		Percentage Link Utilization				Packet Drop Rate			
#webs	#FTPs	IW=1	IW=2	IW=3	IW=4	IW=1	IW=2	IW=3	IW=4
8	0	34	37	38	39	0.0	0.0	0.0	0.0
8	1	95	92	93	92	0.6	1.2	1.4	1.3
8	2	98	97	97	96	1.8	2.3	2.3	2.7
8	3	98	98	98	98	2.6	3.0	3.5	3.5
16	0	67	69	69	67	0.1	0.5	0.8	1.0
16	1	96	95	93	92	2.1	2.6	2.9	2.9
16	2	98	98	97	96	3.5	3.6	4.2	4.5
16	3	99	99	98	98	4.5	4.7	5.2	4.9
32	0	92	87	85	84	0.1	0.5	0.8	1.0
32	1	98	97	96	96	2.1	2.6	2.9	2.9
32	2	99	99	98	98	3.5	3.6	4.2	4.5
32	3	100	99	99	98	9.3	8.4	7.7	7.6

To get a more complete picture of performance, we computed the network power, goodput divided by median delay (in Mbytes/ms), and plotted it against IW for all scenarios. (Each scenario is uniquely identified by its number of webs and number of file transfers.) We plot these values in Figure 1, illustrating a general advantage to increasing IW. When a large number of web clients is combined with ftps, particularly multiple ftps, pathological cases result from the extreme congestion. In these cases, there appears to be no particular trend to the results of increasing the IW, in fact simulation results are not particularly stable.

To get a clearer picture of what is happening across all the tested scenarios, we normalized the network power values for the non-pathological scenario by the network power for that scenario at IW of one. These results are plotted in Figure 2. As IW is increased from one to four, network power increased by at least 15%, even in a congested scenario dominated by bulk transfer traffic. In simulations where web traffic has a dominant share of the available bandwidth, the increase in network power was up to 60%.

The increase in network power at higher initial window sizes is due to an increase in throughput and a decrease in the delay. Since the (slightly) increased drop rates were accompanied by better performance, drop rate is clearly not an indicator of user level performance

Figure 1. Network Power vs Initial Window Size (W= #webs, F= #ftps)

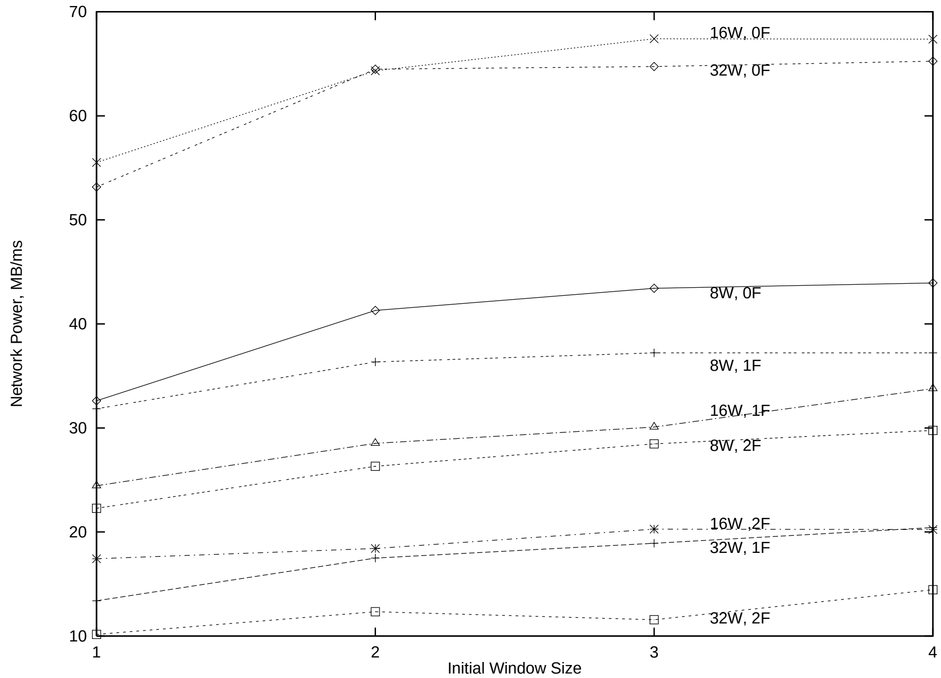
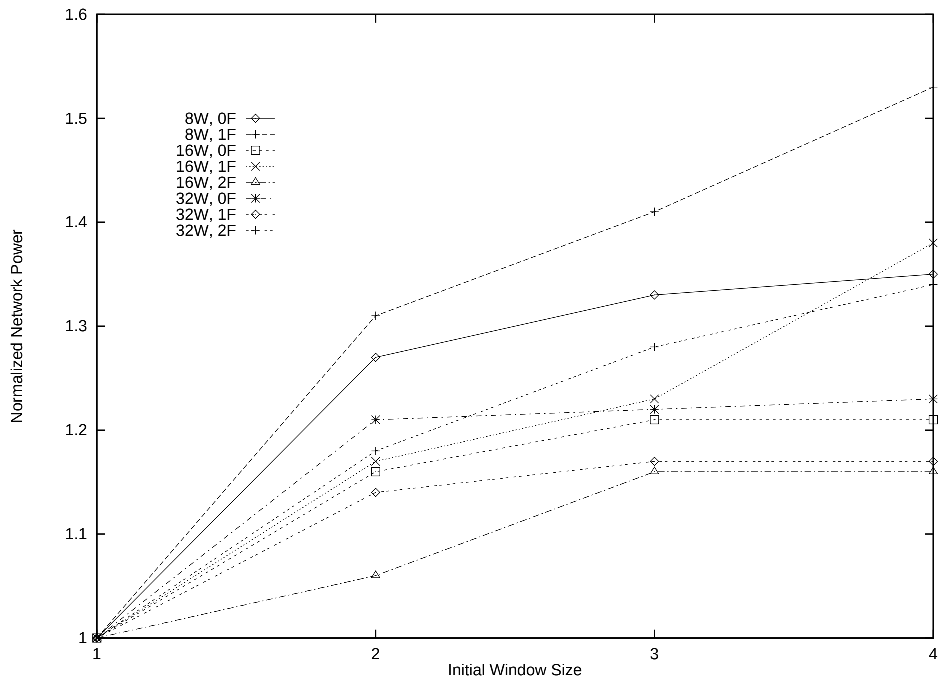


Figure 2. Normalized Network Power vs Initial Window Size



The gains in performance seen by the web clients need to be balanced against the performance the file transfers are seeing. We computed ftp network power and show this in Table 3. It appears that the improvements in network power seen by the web connections have negligible effect on the concurrent file transfers.

Table 3: Network power of file transfers with an increase in the TCP IW size

#Webs	#FTPs	IW=1	IW=2	IW=3	IW=4
8	1	4.7	4.2	4.2	4.2
8	2	3.0	2.8	3.0	2.8
8	3	2.2	2.2	2.2	2.2
16	1	2.3	2.4	2.4	2.5
16	2	1.8	2.0	1.8	1.9
16	3	1.4	1.6	1.5	1.7
32	1	0.7	0.9	1.3	0.9
32	2	0.8	1.0	1.3	1.1
32	3	0.7	1.0	1.2	1.0

The above simulations all used http1.0 style web connections, thus, a natural question is to ask how results are affected by migration to http1.1. A rough model of this behavior was simulated by using one connection to send all of the information from both the primary URL and the three embedded, or in-line, URLs. Since the transfer size is now made up of four web files, the steep improvement in performance between IW of 1 and IW of two, noted in the previous results, has been smoothed. Results are shown in Tables 4 & 5 and Figs. 3 & 4. In two scenarios, the increase in IW from 3 to 4 decreases the network power owing to a non-increase or a slight decrease in the throughput. TCP connections opening up with a higher window size into a very congested network might experience some packet drops and consequently a slight decrease in the throughput. This indicates that increase of the initial window sizes to further higher values (>4) may not always result in a favorable network performance. This can be seen clearly in Figure 4 where the network power shows a decrease for the two highly congested cases.

Table 4: Median web page delay (in ms) for http1.1

#Webs	#FTPs	IW=1	IW=2	IW=3	IW=4
8	0	0.47	0.40	0.38	0.37
8	1	0.84	0.69	0.68	0.63
8	2	0.99	0.87	0.77	0.71
8	3	1.04	0.92	0.86	0.80
16	0	0.54	0.50	0.46	0.43
16	1	0.89	0.76	0.70	0.65
16	2	1.02	0.87	0.82	0.76
16	3	1.11	1.01	0.92	0.90
32	0	0.94	0.79	0.66	0.60
32	1	1.23	1.08	0.88	0.97
32	2	139	1.30	1.10	1.22
32	3	1.46	1.40	1.30	1.24

Table 5: Network power of file transfers with an increase in the TCP IW size

#Webs	#FTPs	IW=1	IW=2	IW=3	IW=4
8	1	4.2	4.2	4.2	3.7
8	2	2.7	2.5	2.6	2.3
8	3	2.1	1.9	2.0	2.0
16	1	1.8	1.8	1.5	1.4
16	2	1.5	1.2	1.1	1.5
16	3	1.0	1.0	1.0	1.0
32	1	0.3	0.3	0.5	0.3
32	2	0.4	0.3	0.4	0.4
32	3	0.4	0.3	0.4	0.5

Figure 3.Network Power vs Initial Window Size (http1.1)

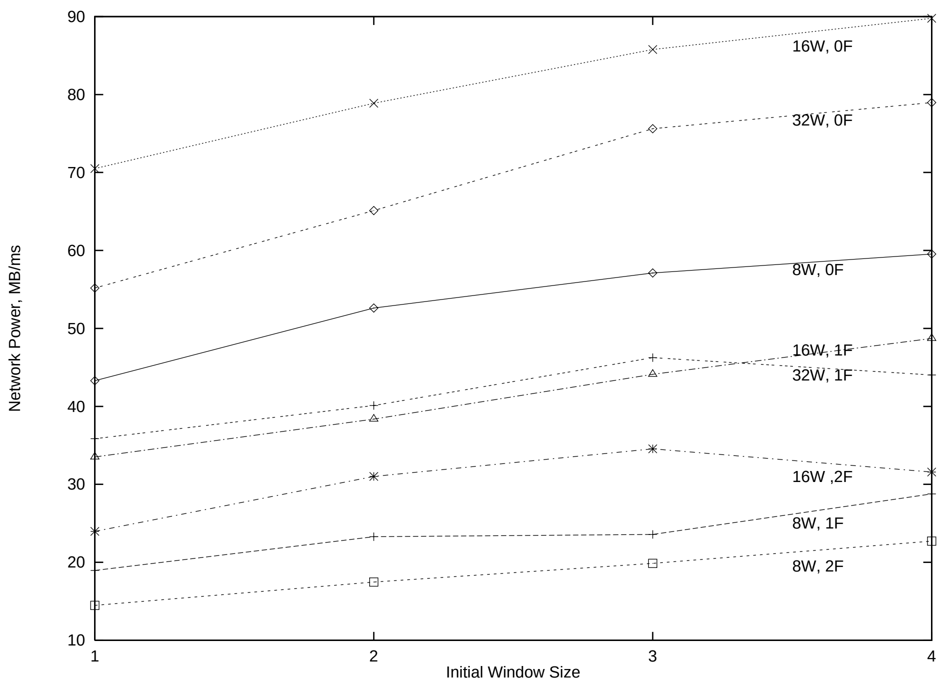
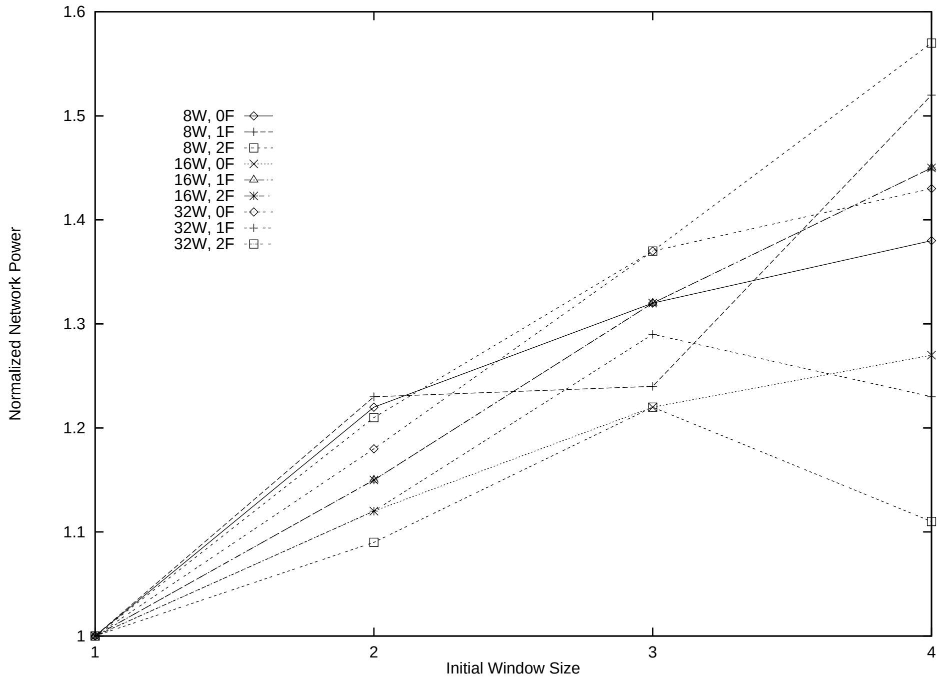


Figure 4.Normalized Network Power vs Initial Window Size (http1.1)



For further insight, we returned to the http1.0 model and mixed some web-browsing connections with IWs of one with those using IWs of three. In this experiment, we first simulated a total of 16 web-browsing connections, all using IW of one. Then the clients were split into two groups of 8 each, one of which uses IW=1 and the other used IW=3. We repeated the simulations for a total of 32 and 64 web-browsing clients, splitting those into groups of 16 and 32 respectively. Table 6 shows these results. We report the goodput (in Mbytes), the web page delays (in milliseconds), the percent utilization of the link and the percent of packets dropped.

Table 6: Results for Half-and-half experiments

	IW=1		IW=3		IW=1		IW=3		Combined	
#webs	Goodpt	Delay	Goodpt	Delay	%util	%drops	%util	% drops	%util	%drops
16	35.5	0.64	36.4	0.54	67	0.1	69	0.7		
8/8	16.9	0.67	18.9	0.52					68	0.5
32	48.9	0.91	44.7	0.68	92	3.5	85	4.3		
16/16	22.8	0.94	22.9	0.71					89	4.6
64	51.9	1.50	47.6	0.86	98	13.0	91	8.6		
32/32	29.0	1.40	22.0	1.20					98	12.0

Unsurprisingly, the non-split experiments are consistent with our earlier results, clients with IW=3 outperform clients with IW=1. The results of running the 8/8 and 16/16 splits show that running a mixture of IW=3 and IW=1 has no negative effect on the IW=1 conversations, while IW=3 conversations maintain their performance. However, the 32/32 split shows that web-browsing connections with IW=3 are adversely affected. We believe this is due to the pathological dynamics of this extremely congested scenario. Since embedded URLs open their connections simultaneously, very large number of TCP connections are arriving at the bottleneck link resulting in multiple packet losses for the IW=3 conversations. The myriad problems of this simultaneous opening strategy is, of course, part of the motivation for the development of http1.1.

4. Discussion

The indications from these results are that increasing the initial window size to 3 packets (or 4380 bytes) helps to improve perceived performance. Many further variations on these simulation scenarios are possible so we’ve made our simulation models and scripts available in order to facilitate others' experiments.

We also used the RED queue management included with ns-2 to perform some other simulation studies. We have not reported on those results here since we don't consider the studies complete. We found that by adding RED to the bottleneck link, we achieved similar performance gains (with an IW of 1) to those we found with increased IWs without RED. Others may wish to investigate this further.

5. References

[1] B. Mah, "An Empirical Model of HTTP Network Traffic", Proceedings of INFOCOM '97, Kobe, Japan, April 7-11, 1997.
[2] C.R. Cunha, A. Bestavros, M.E. Crovella, "Characteristics of WWW Client-based Traces", Boston University Computer Science Technical Report BU-CS-95-010, July 18, 1995.
[3] K.M. Nichols and M. Laubach, "Tiers of Service for Data Access in a HFC Architecture", Proceedings of SCTE Convergence Conference, January, 1997.
[4] K.M. Nichols, "Improving Network Simulation with Feedback", available from knichols@baynetworks.com

6. Acknowledgements

This work benefited from discussions with and comments from Van Jacobson.

7. Authors' addresses

Kedarnath Poduri
Bay Networks
4401 Great America Parkway, SC01-04
Santa Clara, CA 95052-8185
Voice: +1-408-495-2463
Fax: +1-408-495-1299
kpoduri@Baynetworks.com

Kathleen Nichols
Bay Networks
4401 Great America Parkway, SC01-04
Santa Clara, CA 95052-8185
knichols@baynetworks.com