Reproducing: An Argument For Increasing TCP’s Initial Congestion Window

Introduction

This is a presentation paper of the reproducing results of the original paper $An\ Argument\ For\ Increasing\ TCP's\ Initial\ Congestion\ Window$ by $Google$ in 2010. The original paper is proposed to increase the TCP's initial congestion window to at least ten segments in order to complete web requests quicker. We have basically reproduced the original paper, and the result is quite close to the result of the original paper.

About the original paper

The original paper $An\ Argument\ For\ Increasing\ TCP's\ Initial\ Congestion\ Window$ was released by $Google$ in 2010. It said that TCP flows start with an initial congestion window of at most four segments or approximately 4KB of data. Because most Web transactions are short-lived, the initial congestion window is a critical TCP parameter in determining how quickly flows can finish. While the global network access speeds increased dramatically on average in the past decade, the standard value of TCP's initial congestion window has remained unchanged.

The origin paper proposed to increase TCP's initial congestion window to at least ten segments or approximately 15KB of data. Through large-scale Internet experiments, the paper quantified the latency benefits and costs of using a larger window, as functions of network bandwidth, round-trip time (RTT), bandwidth-delay product (BDP), and nature of applications. It shows that the average latency of $HTTP$ responses improved by approximately 10% with the largest benefits being demonstrated in high RTT and BDP networks. The latency of low bandwidth networks also improved by a significant amount in our experiments. The average retransmission rate increased by a modest 0.5%, with most of the increase coming from applications that effectively circumvent TCP's slow start algorithm by using multiple concurrent connections.

Reason to choose the paper

Before the paper was released, the amount of the initial congestion window has long remained four, while the paper had a penetrating look on the problem and find that the amount of the initial congestion windows do contribute to the efficiency of computer network. The paper attracts more attention on the slow start of the computer network, and promoted the series of subsequent developments.

The paper analyzed the different affects on the efficiency of the network between different amounts of the congestion windows in detail, which makes the work of reproduction much easier and more clear. Among most of the classical papers in the area of computer network, the original paper is quite basic and easy to understand, which is suitable for the beginners, while the results of the work has already been reproduced, the successfully reproducing can be promising.

Goal and Motivation

The main results we want to reproduce is the first two graphs of paper’s $Figure\ 5$, which are the average response latency for Web search bucketed by RTT, and BW at $AvgDC$.

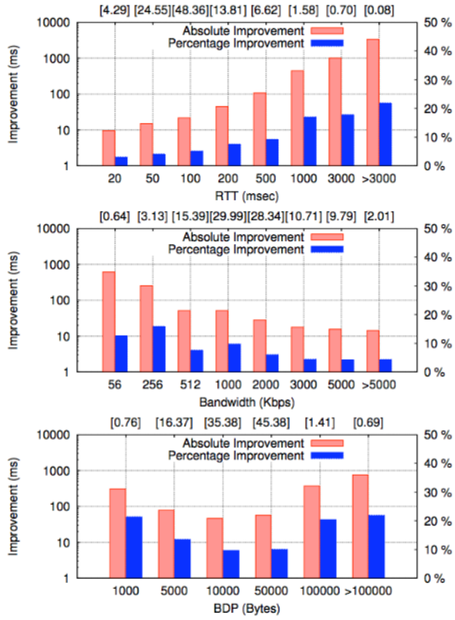


Figure 5: Average response latency for Web search bucketed by RTT, BW, and BDP at AvgDC. The left y-axis shows the absolute latency improvement (in ms), and the right y-axis shows the percentage improvement. The buckets are represented in the lower x-axis scale, while the numbers at the top of the plot represent the percentage responses in each bucket. The buckets are named by the upper end of RTT, BW, BDP.

We did not reproduce the third graph which shows the improvement bucketed by BDP mainly because we don't know the bandwidth and delay they used for each BDP. In fact, since Google monitored real traffic in data center, each BDP result they got used a lot of different bandwidth and delay combination, and we do not have the joint probability distribution of it.

Experiment setup

We used $Mininet$ to set up the experiment on a Google Cloud virtual instance. The $Mininet$ topology consists of a server, a client, and a switch connecting them. The initial congestion window (3 or 10) is configured using the $initcwnd$ option in the $ip\ route$ command, and the default congestion control algorithm we used is $TCP\ CUBIC$. We also set the initial receive window to 27 segments (40.5KB), which is roughly equal to the Windows 7's average $rwnd$. We did this because Linux has an average $rwnd$ of 10KB which is not large enough.

If another reproduction would like to be done, you can import our $Github$: https://github.com/935963004/Reproducing-Results-from-A-Networking-Paper. You can cd into the repo and run $chmod\ +x *.sh$ to make shell scripts executable. Run $sudo\ ./install.sh$ to install $Mininet$, $Python\ 2.7$, $numpy$, $matplotlib$, $argparse$. Run $sudo\ ./run.sh$ to run the simulation. It should take about 15 - 20 minutes. Once done, the two figures for RTT and BW will be in the figures folder.

Results

We used a fixed response size of 9KB to reproduce these two graphs. We can see in the figure that about 60% of Google web search response size is around 8KB to 10KB, so 9KB is pretty representative.

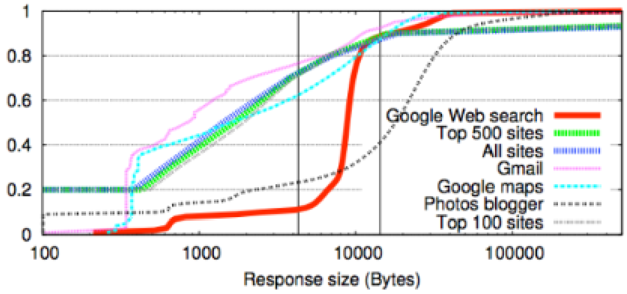
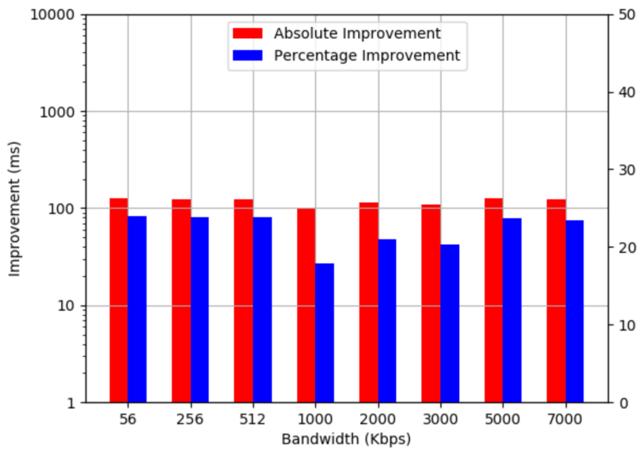


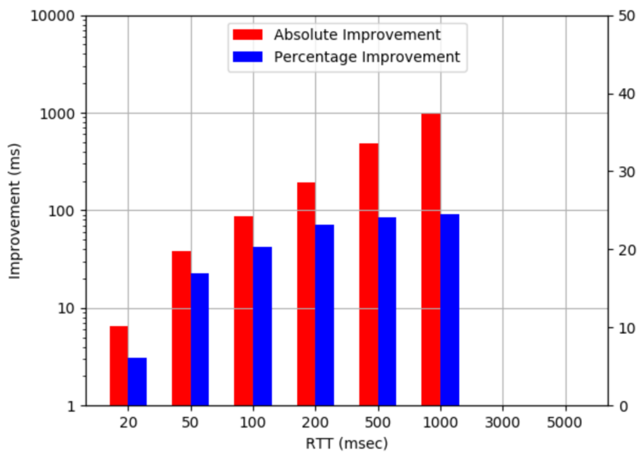
Figure 1: CDF of HTTP response sizes for top 100 sites, top 500 sites, all the Web, and for a few popular Google services. Vertical lines highlight response sizes of 3 and 10 segments.

When reproducing results for a fixed RTT, we actually varied the bandwidth, and different bandwidth are given different weights to calculate the final result.

BW (kbps)	56	256	512	1000	2000	3000	5000	>5000
percentage	0.64	3.13	15.39	29.99	28.34	10.71	9.79	2.01
RTT (ms)	20	50	100	200	500	1000	3000	>3000
percentage	4.29	24.55	48.36	13.81	6.62	1.58	0.70	0.08

And the results we get has shown as below.





The RTT graph is very similar to the one in the paper. However, for RTT = 1000 and 3000ms, there is no improvement. This is because the initial retransmission timeout (RTO) is set to 1000ms, and when retransmission happens $cwnd$ falls and basically cancel out the effect of having higher initial $cwnd$.

For the BW graph we've reproduced, it doesn't match with the original graph. The reason may be the difference of real world network between ours and Google's. But we can still see the higher the bandwidth is, the more improvement exists.

Teamwork

The reproduction of the paper $An\ Argument\ For\ Increasing\ TCP's\ Initial\ Congestion\ Window$ is done in teamwork by Zihua Zhao and Weibang Jiang, in which Weibang Jiang did the reproduction of the paper results and Zihua Zhao completed this paper. The overall work is quite well-off since there's already previous reproduction of the paper. We have basically reproduced the results of the original paper, which is also satisfying.

References

Dukkipati, Nandita, Tiziana Refice, Yuchung Cheng, Jerry Chu, Tom Herbert, Amit Agarwal, Arvind Jain, and Natalia Sutin. "An argument for increasing TCP's initial congestion window." *Computer Communication Review* 40, no. 3 (2010): 26-33.