OMSCS 6250 Project 4: TCP Fast Open

Brian DuVerneay

Introduction

The goal of this experiment is to simulate the effectiveness of the TCP Fast Open (TFO) scheme for transferring data during initial TCP handshaking and compare it standard TCP protocol. The hypothesis is that the handshaking is a significant portion of the page load time. We will compare results of our simulation in this paper to the experiments in the paper "TCP Fast Open", authored by Radhakrishnan, Cheng, Chu, Jain, and Raghavan

Experiment setup

Four URLs were used in this experiment, shown in table below. The "Fetch.py"_script is run to determine the various servers that will need to be accessed to load the entire pages.

URL	Site detail
http://alibi.com	Local news / happenings
http://www.chipublib.org	Public library
http://blog.vandalog.com	Art blog
http://en.wikipedia.org/wiki/Electronic_design_automation	Wikipedia page entry

Simulation Intro

The simulation using Tfo.py attempts to mimic the experiment in the TFO paper, by simulating page download times for standard TCP vs TFO. Across the 4 URLs, 61 unique servers were accessed and Tfo.py runs a mininet simulation that sets up unique servers in a star topology, connected through a single switch. The internet bandwidth of 4Mbps is used in this simulation with RTT of 20, 100, and 200 milliseconds, similar to the TFO paper.

a) Intro to raw experimental data generated from console

The raw output is shown below, formatted into a table. Page Load Times (PLTs) were rounded to the nearest millisecond. For all URLs and RTTs simulated, TFO improved page load time.

Page	RTT (ms)	PLT No TFO (ms)	PLT With TFO (ms)	Improvement
httpwww.chipublib.org	200	22690	3500	84.57%
	100	2821	2363	16.26%
	20	1928	1687	12.52%
httpblog.vandalog.com	200	31834	3129	90.17%
	100	2244	1630	27.35%
	20	1220	922	24.41%
httpen.wikipedia.orgwikiElectronic_design_automation	200	3231	2258	30.12%
	100	1804	1454	19.38%
	20	1165	652	44.01%
httpalibi.com	200	23916	8380	64.96%
	100	7362	5188	29.53%
	20	4962	4152	16.33%

b. Your analysis should answer the following questions for each URL you specified:

i. What effect does TFO have on the timing?

For all URLs and RTTs simulated, TFO improves the page load time.

ii. How does the RTT value affect these results?

Generally, at each increase in RTT, there was a bigger percentage improvement in page load time. As RTT is increased, the handshaking time required to establish a standard TCP connection becomes a more significant length of page load time relative to the amount of data that needs to be transferred from a given server. In one case with the Wikipedia page, there is an outlier from this trend. We can attribute this anomaly to testing environment and variation of internet traffic.

iii. Does the particular content available at this URL lend itself to performance enhancements provided by TFO?

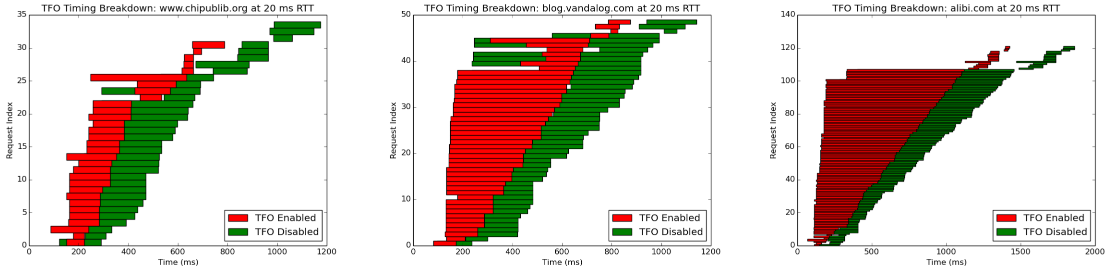
In the TFO paper, it is stated that the “primary goal of TFO is to eliminate one RTT of extra latency, thereby improving the performance of short flows”. So, short flows should be the most improved by TFO. All of these URLs do have a variety of short flows in their makeup, and thus TFO performance enhancements are expected.

iv. Were these results surprising in any way?

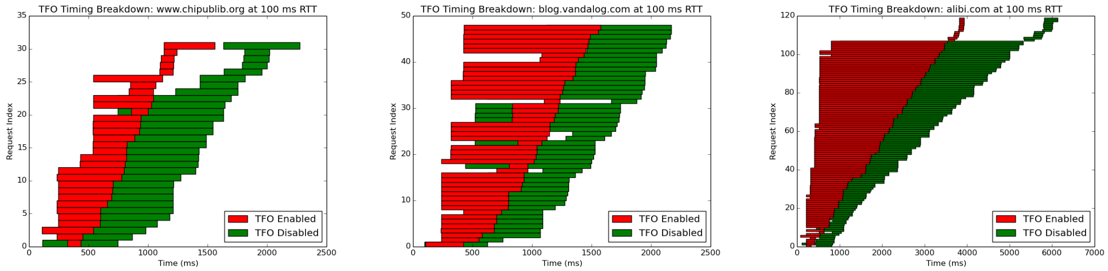
Yes. Re-running the same simulation twice did not produce consistent results from run to run. This can be attributed to variation in testing environment and network traffic. Also, I expected bigger percentage improvements for a simple page like Wikipedia. The paper states that in this case “the browser spends most of its time waiting for network transfers rather than processing the retrieved content, and thus TFO offers significant improvements.” While improvements were seen with the simple Wikipedia page, even greater improvements were observed with more complex sites.

v. Include any relevant graphs from the output-figures folder

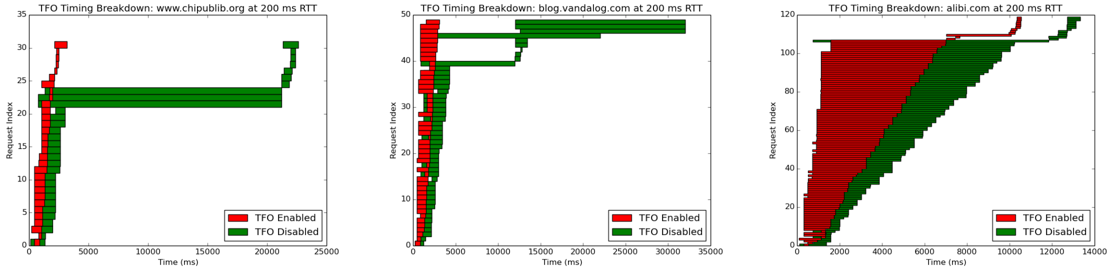
Below are shown graphs for 3 of the URLs: chipublib, blog.vandalog.com, and alibi.com. the 20ms, 100ms, and 200ms for each of the URLs. With TFO enabled, there are improvements across all URLs and RTTs. Note the variety of request indices across URLs, from around 35 for Chicago public library to around 120 for Alibi. For 200ms RTT, in some cases there is a long time to get some specific resources.



20 ms RTT



100 ms RTT



200 ms RTT

c. Include a brief summary of your findings and state what conclusions you can draw based on the results of your experiment.

The result across all URLs is that TFO improves page load time compared to the standard TCP with no TFO. This is consistent with the paper. The RTT for all of these simulations is in the 20ms – 200ms range. Even though TFO is suited better for smaller flows with longer RTTs, according to simulation it is generally better for a range of URLs and flows.

The paper and simulations assigned in this project do not run a simulation with RTT < 20ms. I performed additional simulations and find that the benefit of TFO with RTT in the 10-20ms range become less predictable and less likely to reduce page load times.

d. Based on the reading and your experiment, where do you see TCP Fast Open having the best potential for improvement? What about the worst?

Best potential for improvements with TFO are in high RTT network situations and a large number of small flows. The worst case for TFO is cases where the RTT is only a small fraction of PLT (non-wireless networks, etc), and where the flows are very large.