a. Brief introduction of raw experimental data

Page	RTT(ms)	PLT: no TFO (s)	PLT: TFO (s)	Improv.
httpwww.gatech.edu				
	200	12920.108	2673.726	79.3056993022
	100	2504.771	1564.473	37.5402781332
	20	1169.412	1048.23	10.3626437902
httpwww.google.com				
	200	3577.959	3159.628	11.6918891469
	100	2733.374	2244.174	17.8972946988
	20	1630.531	1716.194	5.25368729573
httpwww.php.net				
	200	2319.281	1828.731	21.1509515233
	100	1326.46	927.054	30.1106705065
	20	610.388	546.636	10.4445041515

We fetched and benchmarked gatech.edu, google.com and php.net, comparing the whole page download performance at local and then ran a HTTP server with given scripts and tested on different RTT for PLT of TFO enabled versus not enabled, the results shows obvious advantage of TFO over TCP.

b i. What effect does TFO have on the timing?

TCP Fast Open decrease HTTP transaction network latency and whole-page load time(TCP) averagely over 11% on google.com, 20% on php.net and on gatech.edu case up to 50%

ii. How does the RTT value affect these results?

The RTT of a web flow largely comprises two components: transmission delay and propagation delay. Propagation delay is largely constrained by the speed of light. Thus reducing RTT is the most effective way to improve the latency of web applications. TFO pages/resource fetches through time and decrease request latency by round trip time (RTT) and improve page load time.

When RTT is high, TFO improves PLT greatly (See below figure When RTT = 200ms on gatech.edu). When RTT is small and network delay is only a small fraction of PLT, the resource processing time would exceed network time, so the gains from TFO are expected to be small(See below figure When RTT = 20ms on google.com).

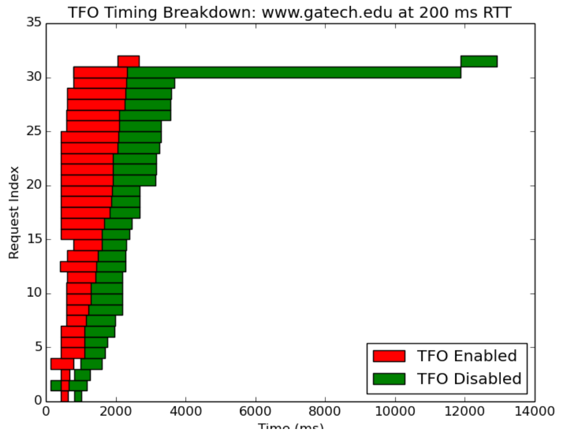
iii. Does the particular content available at this URL lend itself to performance enhancements by TFO?

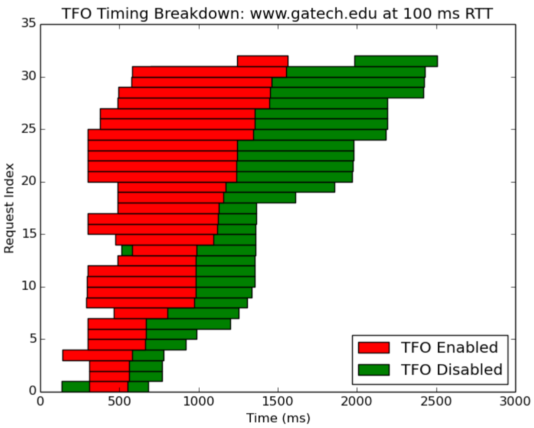
Yes, for example, heavy page with whole-page download or an HTTP GET request / response in SYN packets such as gif and bmp can gain slight improvements with a small RTT, however it improves greatly when higher RTTs(See below figures). TCP Handshake have performance penalty between 10% and 30% of the latency to serve the HTTP request and TFO eliminates this cost and reduce the RTT of extra latency, thus enhance the performance.

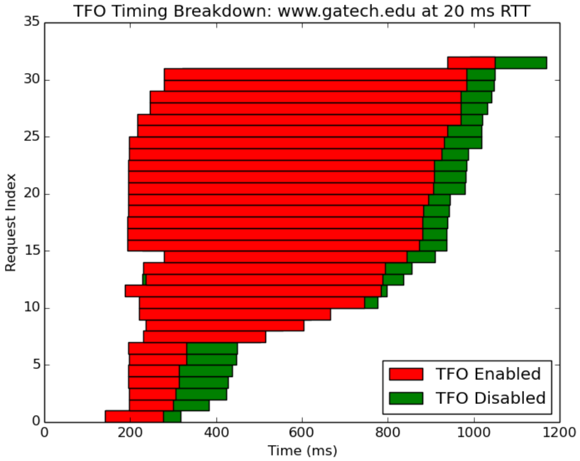
iv. Were these results surprising in any way?

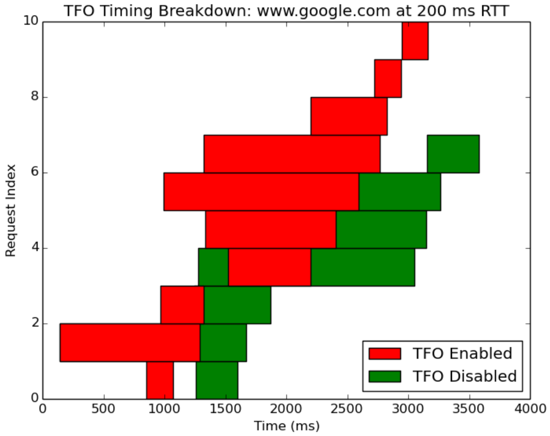
Yes, the presented results support the conclusion from the paper, however it does not appears that the more increase of RTT, the bigger improvements we can get on percentage wise, which is a surprise. For some heavy contented website and big web objects, the increase of the RTT will have a much bigger improvements on percentage, (see figure Gatech.edu vs google.com). If the web content is not heavy enough and request index is low, 20ms RTT would have some non TFO enabled performed the same or even better on PLT from figures of google 20ms RTT and php 20ms RTT.

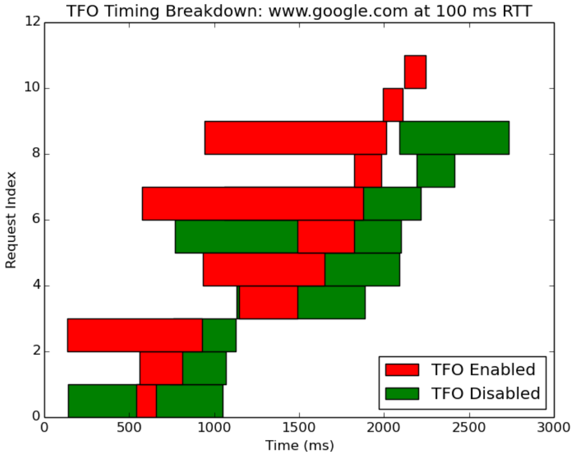
v. Include relevant graphs from the output-figures folder (images do not count toward your two page limit). There should be at least one graph per URL analyzed.

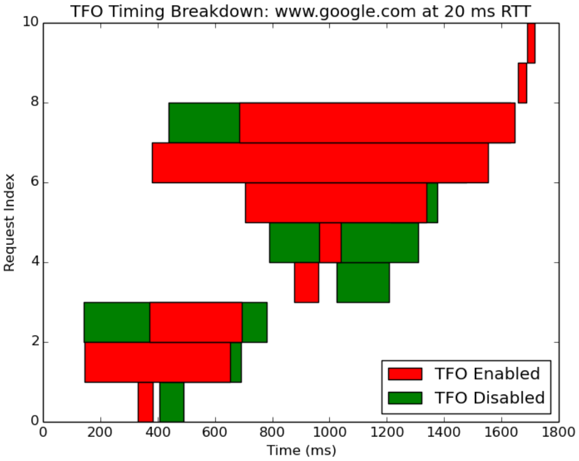


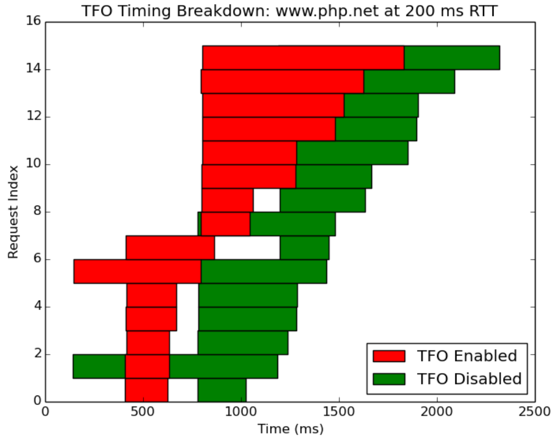


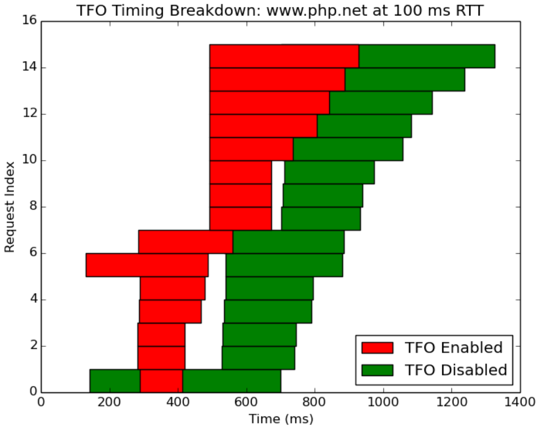


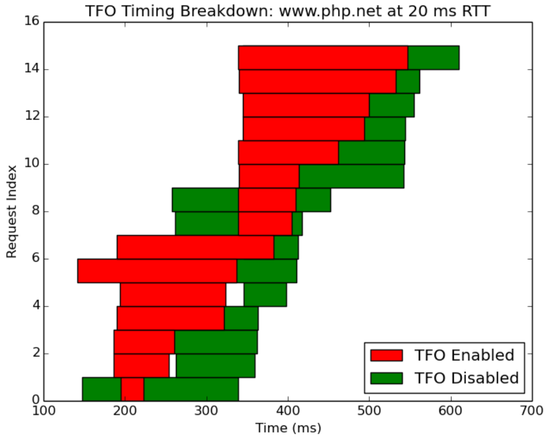












c. Include a brief summary of your findings and state what conclusions you can draw based on the results of your experiment.

Conclusion can be drawn that content type and effectiveness has huge improvement to TCP protocol. However cookie handle has little effect over TCP. Also my findings are the similar from the paper that TFO improves PLT when RTT is high for all the sites tested, when the RTT is small, the gain from TFO are expected to be small. Even for pages heavy on content and with short emulated RTT (See below figure When RTT = 20ms Gatech.edu), TFO accelerates PLT by 10 to 11%. Simpler pages such as php.com, if the browser spends most of its time waiting for network transfers rather than processing the retrieved content, and TFO offers also significant improvements of 21% and 30% with 100ms and 200ms RTTS(See above figure When RTT = 100ms/200ms on php.net)

d. Based on the reading and your experiment, in which of your website experiment scenarios do you see TCP Fast Open having the most performance gain over TCP? What about the worst?

Gatech.edu has the most performance gain while Google.com seems to have the worst performance gain because the content such as page pictures, flash/html content used on the website would largely determine how much performance gain over TCP, the simpler website where spending little time of waiting for network transfers is the worst performance gain in the TFO.