UDACITY Nanodegree Catalog Zihan Chen

cn »

View History

Assignment 6 - TCP Fast Open

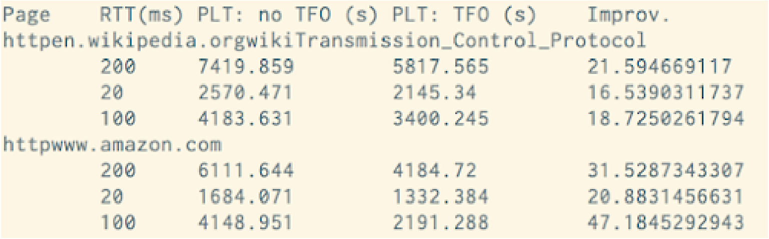
Goal

The goal of this assignment is to learn about a specific change to TCP that reduces transfer latency. In the past lesson you learned about persistent TCP connections which enable a web browser to reuse a connection for multiple data requests. However, many HTTP requests occur over new TCP connections for example because an image for a webpage is located on a separate content server or an ad is being loaded from a third party resource. This led researchers at Google to investigate reducing the setup costs for a TCP connection. They developed TCP Fast Open (TFO) which begins sending data during the TCP handshake thereby reducing the latency of a new connection by one RTT.

You will replicate the experiment from the original paper in Mininet to observe how TCP Fast Open improves round trip times for real websites. The test setup provided uses a modified Chrome binary to download web pages from Mininet hosts. The hosts act as web servers serving mirrored data from real websites. The tests run for different latencies with TCP Fast Open disabled and then enabled. TFO is a now merged in the Linux kernel so it is built into Ubuntu 13.04 which is installed on the virtual machine from assignment 1.

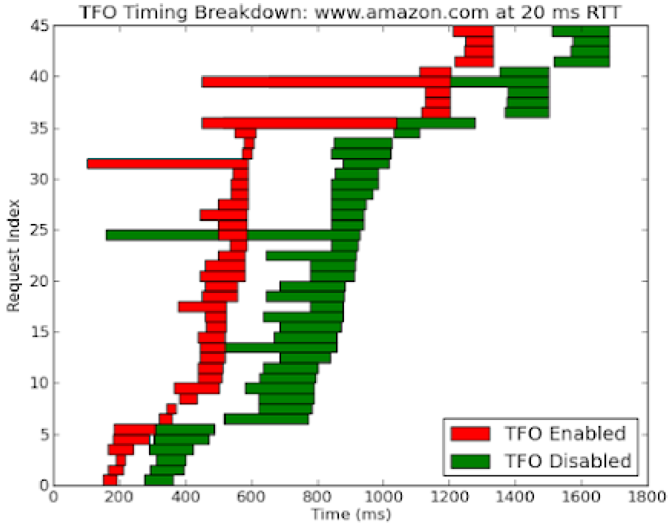
Directions

1. Update to the latest assignment code:
   `git commit -a -m "Saving work"`
   `git pull --rebase`
2. Install a VNC server, termcolor and an additional library:
   `sudo apt-get update`
   `sudo apt-get install vnc4server`
   `sudo apt-get install libnss3-dev`
   `sudo easy_install termcolor`
3. Read the original TCP Fast Open paper. You'll need to read this to answer the quiz questions.
4. Run the experiment to replicate the paper:
   `vnc4server`
   `sudo ./run.sh`
5. The experiment takes some time to complete. When its finished you should see results similar to the image below. Be sure to copy and paste this text into a text file for your final submission.



```
Page    RTT(ms) PLT: no TFO (s) PLT: TFO (s)    Improv.
httpen.wikipedia.orgwikiTransmission_Control_Protocol
        200     7419.859        5817.565        21.594669117
        20      2570.471        2145.34         16.5390311737
        100     4183.631        3400.245        18.7250261794
httpwww.amazon.com
        200     6111.644        4184.72         31.5287343307
        20      1684.071        1332.384        20.8831456631
        100     4148.951        2191.288        47.1845292943
```

   You can also view graphs of the results in the `output-figures` folder.



6. Now, using your own set of websites, run the experiment. To do this, you'll need to create a `.pages` file with a list website URLS each on a single line similar to the `Paper.pages` file. For simplicity, call the file `myURLS.pages`. Then run the `fetch.py` script to download the web pages, modify the `run.sh` script to use your new `.pages` file, and re-run the experiment:

   ```
   ./fetch.py --name myURLS
   ```

   ```
   --- Modify run.sh script on lines 21 and 29 (change --name argument to myURLS) ---
   ```

   ```
   sudo ./run.sh
   ```

   Note that the script has issues with Javascript heavy web pages and avoid using `https` sites as well. The Google and Udacity home pages work well as test sites. Also, if you shut down the virtual machine after step 4, be sure to restart VNC.
7. Submit the log data from step 5, the data replicating the paper's results, and your answers to the quiz questions below to the submission page.

Quiz Questions

1. How does TCP Fast Open improve transfer speed? Explain the specific mechanism that it alters.
2. Why would TCP Fast Open be useful if web browsers reused existing TCP connections or made use of HTTP 1.1 persistent connections? Do web browsers use existing connections efficiently?
3. Why did the authors not use one time cookies in their implementation?
4. How well do your results compare to the paper's results?
5. What improvement if any did you see in your selected sites?

Notes

[1] Based on a reproducing network research post.

This revision was added on 2014/03/12 21:17:24.

POPULAR NANODEGREE PROGRAMS +

STUDENT RESOURCES +

UDACITY +

INQUIRIES +