COL334/672 Computer Networks: 2017-18 Semester 1

Assignment 2

In this assignment, we will look at various HTTP tools to understand page load dynamics.

1. Start with experimenting with the following tools
    a. Chrome developer tools, to log details of all HTTP requests made to download a particular webpage:

       https://developers.google.com/web/tools/chrome-devtools/network-performance/reference

    b. Understand the HAR JSON format, which you will need to parse the HAR files http://www.softwareishard.com/blog/har-12-spec/
2. Webpages to probe
    a. Remember to clear your cache and DNS before starting. In Chrome developer tools, you can also turn off use of the cache
    b. We will study three webpages
        i. http://indianexpress.com which has a lot of small objects and is non-responsive
        ii. http://www.nytimes.com/ which has a lot of small objects and is responsive
        iii. https://www.vox.com/a/internet-maps which too has several objects, but some of them are quite large in size
    c. Turn on the chrome tools and turn on wireshark
    d. Run experiments for each webpage under the standard “fast 3G” profile and “slow 3G” profiles, the unthrottled profile, and for Chrome for Windows and Chrome for Android. Therefore, you will have 6 experiments for each webpage. Please note that for these experiments, you need to install the latest version of Google Chrome on Windows. If your Chrome version is older, you may get profiles with other names like “Regular 3G” and “Good 3G”. It is preferable that you use the latest version to avoid inconsistencies.
    e. For the responsive webpages, observe the difference between when you just access the webpage Vs when you access and scroll over the page. For the analysis below, use the trace for when you quickly scroll down to the bottom of the page, ie. all objects have been requested

f. Save the .har and .pcap dumps for each webpage obtained under different conditions for further analysis.

3. Analysis

   a. Write scripts to parse the HAR JSON, for each webpage, and find out the following:

      i. Total number of objects downloaded, and number of objects downloaded from different domains

      ii. Total size of content downloaded, and size of content downloaded from each domain

      iii. Number of TCP connections opened to each domain, and number and size of objects downloaded on each connection

      iv. Your script should take the HAR file as an argument and produce the above output in a nice readable format

   b. Timing analysis. Extend your scripts to report the following.

      i. Find the page load time from the onLoad entry

      ii. For each domain, find the time spent in the DNS query when opening the first TCP connection. Also confirm that for subsequent TCP connections for each domain, a fresh DNS query was not launched, ie. the IP address was picked up from the DNS cache.

      iii. For each TCP connection in the download tree, use the timing information in the HAR file to find the following:

         1. Connection establishment time to open the TCP connection, reported in the *connect* variable in the HAR

         2. Average time spent in waiting for the server to respond, reported in the *wait* variable in the HAR

         3. Total time spent in receiving data, reported in the *receive* variable in the HAR

         4. Average goodput, calculated as the total size of content downloaded on the TCP connection / total time spent in receiving data

         5. Find the maximum achieved goodput of the connection, by identifying the largest object downloaded on the connection, and then calculating the goodput as the size of the object / receive time for the object

Does this maximum goodput differ from the average goodput you calculated in the previous part?

6. Also find the average achieved goodput of the network, ie. across all domains

7. Do you think your download capacity was utilized well to access the web pages? Hint: Compare the average achieved goodput of the network, with the maximum of the maximum achieved goodput on all connections.

iv. Differences across network conditions

1. Report the differences you observe when emulating a desktop browser Vs a mobile – does the browser open less or more TCP connections to each domain?

2. Similarly, report the difference between slow and fast 3G emulation – does the browser open less or more TCP connections to each domain?

3. Do you see any patterns, such as a cap imposed by the browser on the number of TCP connections opened per domain, or the total number of TCP connections, or the number of objects being downloaded per TCP connection? Do these policies change based on the type of browser and type of connection?

c. We want to see what best page load time could we have obtained

i. Assume that you can collapse on each TCP connection all the GET requests on that connection, so that the objects are received back to back and there are no idle times. Therefore, calculate the total time consumed on the TCP connection as the sum of the *connect* time for the TCP connection, the maximum of the *waiting times* for objects downloaded on that connection, the sum of the *send* times, and the sum of the *receive* times. Now assume that all the connections could have been opened simultaneously (this is a bit unrealistic because it assumes that the edge bandwidth is not a bottleneck), and calculate the maximum of the times for each connection. Add to this the maximum of the *DNS* time.

ii. How does this compare with the page load time?

iii. Run another calculation assuming that all content from a domain can be downloaded on a single TCP connection at the maximum achieved goodput seen for that domain. What is the page load time in this case?

iv. Why is there a difference in the above two estimates, and the page load time actually experienced?

4. Implementation

    a. Build your own web page download program which takes the HAR file as an input, and downloads all the objects. Your program should download and save on disk all the objects downloaded for the web page. You can create folders for each domain name and save the objects in the appropriate folder. Clear the DNS cache before running your program. Arguments your program should accept:

        i. HAR file generated by the Chrome tools

        ii. Maximum number of TCP connections per domain

        iii. Maximum number of objects per TCP connection

    b. Use the Sockets library for this, and not higher layer libraries such as HTTPUrlConnection in Java

    c. Run your program with a range of other parameter values and find the best choice which gives the minimum page load time. How does this compare with the page load times actually observed on the unthrottled network conditions?

    d. What information do you think if you had in advance, you could decide the optimal choice of parameters to use? How could the HTTP protocol, and/or the HTML standard, and/or link/network layer information available, be modified to provide this information to the browser and help it make better scheduling decisions?

5. What to submit

    a. A .tar.gz file with the following folder

        i. /src containing a script (in python or perl or any other language you are comfortable with) and README file, that takes the HAR filename as input and produces as output the various information asked in part 3a, 3b, and 3c. You will also be evaluated on the neatness of the output produced, so spend time thinking this out.

        ii. /doc containing a pdf report on the data, analysis, and insights asked in parts 3a-3c. You should use graphs, timelines, and other tools to present your results – presentation has a high premium to clearly convey your insights

        iii. /src containing a program (in python or perl or any other language) and README file, which takes the parameters listed in

part 4a and produces the page load time output, along with the downloaded objects in a neat directory structure

iv. /doc containing a pdf report on your results as asked in part 4, with neat graphs and/or tables that show which combination of parameters works well. Presentation of research results carries a high premium – do not claim anything that does not follow from the data, and whatever you claim should be obvious from the way you have presented the data.