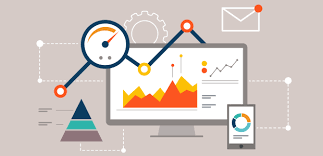
**CS 4713**

**INTRODUCTION TO THE INTERNET: ARCHITECTURE AND PROTOCOLS**

**ASSIGNMENT 2:**

**ANALYSING WEB PAGES OVER DIFFERENT NETWORKS**

**Deadline: Saturday, 3rd November 2018 at 11:50pm**

**Course Policy about Plagiarism**

* This assignment is suppose to be done individually.
* Students must not share any scripts or their analysis report with other students.
* Students must be prepared to explain any program code or report they submit.
* Students must indicate with their submission any assistance received.
* All submissions are subject to plagiarism detection.
* Students are strongly advised that any act of plagiarism will be reported to the Disciplinary Committee

**INTRODUCTION**

Over the years, web pages have evolved from simple text-based pages coming from a single server to diverse content types served by multiple servers. In this assignment, you will analyze the impact of network conditions and web page complexity on the time it takes a page to load completely.

**MAHIMAHI**

Mahimahi is a framework to record and replay traffic from HTTP-based applications over emulated network conditions. It is structured as a set of four UNIX shells, namely, the record, replay, delay and link shells. The framework isolates its traffic from the rest of the host system, allowing multiple instances of its shells to run in parallel without interference.

The four shells are as follows:

1. Record Shell: Records HTTP traffic and stores it for subsequent replays
2. Replay Shell: Replays previously recorded HTTP content (possibly over different emulated network conditions)
3. Delay Shell: Imposes a user specified delay on all packets that originate from it.
4. Link Shell: Emulates a network link by delivering packets according to a particular trace.

You will use mahimahi to emulate different network conditions and observe changes in page load metrics. For the purpose of this assignment, you will require only the last two shells but feel free to play around with them all. For more information, you can take a look at [this](http://mahimahi.mit.edu/mahimahi_atc.pdf) paper.

**SETUP**

**Installing Mahimahi:**

* You will require Ubuntu version 14.04 or later.
* Run the following commands

sudo add-apt-repository ppa:keithw/mahimahi

sudo apt-get update

sudo apt-get install mahimahi

**Setting up Mahimahi for runs:**

* Initial setup

sudo sysctl -w net.ipv4.ip\_forward=1 (enable port forwarding)

* You are provided with network traces over which you will conduct your runs. Change the directory to the one containing the traces
* For the link shell:

mm-link [uplink trace file name] [downlink trace file name] (use the flag --meter-all flag to show graphs of throughput)

* For the delay shell:

mm-delay [delay value]

* You will now be inside a link/delay shell. To launch the browser from within it, do

google-chrome

* For examples of shell setups, visit [this](http://mahimahi.mit.edu/) link.

**Gathering Data**

You will be parsing HAR[[1]](#footnote-1) files generated when a web page is loaded to analyze web page metrics.

To retrieve and save HAR files:

* Open Chrome Dev Tools (Right Click + Inspect OR Ctrl + Shift + I), network tab.
* Ensure cache is disabled.
* Ensure that page is fully loaded before you proceed to the next step.
* Right Click any entry and select “Save HAR with content”

Figure 1 shows a snippet of a HAR file. If your HAR file does not have a “pages” entry, you will need to reload and resave the HAR file.



**Parsing the HAR file**

* You will use the Haralyzer python module to parse the HAR files. To install Haralyzer:

Sudo pip install haralyzer

* You are also provided starter code to help you with the parsing.

Python starter\_code.py sitename

* HAR file must be stored as sitename.har e.g. www.xyz.com.har
* For help purposes, you can install the performance-analyzer extension to visualize the results.

**Web Pages for Testing**

* www.washingtonpost.com
* www.cnbc.com
* www.reddit.com
* www.espncricinfo.com
* www.nfl.com
* www.skysports.com
* www.alibaba.com
* www.businessinsider.com
* www.spscommerce.com
* www.imdb.com
* www.cnn.com
* www.metacritic.com

**Network Traces**

* You are given 3 wifi and 2 cellular traces to perform runs.
* You only need the link shell to run these traces but feel free to play around with delay values.
* In case of an error in launching the link shell, remove the “.txt” extension from the trace files.
* The [--meter-all] flag will give you 4 real time graphs showing the uplink/downlink delay and throughput values. They might be useful for the analysis.

**ANALYSIS TASKS**

Design and implement experimental runs that will allow you to comment on the following aspects of page load process. Please note that grading in this assignment is dependent on how thoroughly you conduct runs and, in light of their results, how confidently you can draw conclusions.

1. **Web Page Complexity**

How does the number of objects, the type of objects and size of objects vary across webpages?

1. **Network Throughput and Page Load Time**

How does the page load time of each page vary across throughput traces?

1. **Web Page Components and Page Load Time**

Across the same throughput trace, how does the number of objects, page size and number of servers’ impact page load time?

1. **Role of External Servers**

How much of the content, in general, is requested from external servers? Is there any type of content (img, css etc.) that is specifically requested from external servers? Comment on your findings.

1. **Impact of Web Page Components on Rendering**

How is rendering time impacted by the number of objects on a web page?

1. **More Significant Impact**

Which of throughput and object types has a greater impact on page load and rendering times? Does this depend on the web page?

1. **DNS Requests**

Analyze the number of DNS requests made by a web page. What aspect of a web page primarily determines the number of DNS requests made?

1. **Object with the Greatest Impact**

For each web page, which object, if removed, would most significantly decrease the load time?

1. **Possible Improvements**

What would be considered a lighter version of a web page? With reference to the traces and pages provided, suggest a control law to decide where efforts to serve lighter web pages should be made.

The assignment is pretty open-ended and grading will depend upon the thoroughness of design and analysis of runs.

**SUBMISSION**

You are required to submit

1. All the scripts that you used to analyze the HAR files.
2. A report of your analysis and all relevant graph plots.

1. The HTTP Archive (HAR) format, is a JSON-formatted archive file format for logging of a web browser's interaction with a site.  [↑](#footnote-ref-1)