CS3205: Introduction to Computer Networks

Assignment 1, Submission Deadline: 11/02/24, 23:59PM

Full Marks: 100 **Instructor: Ayon Chakraborty**

Make sure you have the following software installed on your Linux system

- `tcpdump` for capturing packets, `tshark` and `pyshark` for writing packet analysis scripts in Python. Ideally, we want you to use Python so that you can submit an integrated Jupyter notebook containing all the analysis you do.

- A Brave/Chrome/Firefox browser and ability to capture HAR traces. Brave allows custom network throttling profiles. `haralyzer` library for use in your Python scripts. Samples are provided in the class WiKi. Remember to always use a private browsing mode and disable cache in the HAR capturing dashboard.

Activity 1 - Preliminary analysis of packet capture data [40 marks]

*Capture pcap traces and perform analysis as instructed. Submit the notebook ipnyb file (**pcap_analysis.ipynb**). Use pyshark for the analysis.*

A. Play the following 52-second YouTube video (https://www.youtube.com/watch?v=pcSv22DTDUI) in five resolutions (*480p, 720p, 1080p, 2K* and *4K*). Name them youtube_480.pcap, youtube_720.pcap etc. **[5]**

1. Plot a timeline of the network downlink and uplink traffic in terms of packets per second and Kilobytes per second. The timeline can be of 60-seconds plotted in the granularity of 100 milliseconds. For both packets/sec or KBs/sec - plot two graphs, one for uplink the other one for downlink. Each graph should have five subgraphs (line plots) for the five video resolutions. (*Your code will be tested with a pcap on our side*) **[10]**

2. Divide the entire timeline in 100 ms slots. For each such slot, if the total number of downlink packets is greater than a threshold (say, 100), we label that slot as a *burst slot*. For each video resolution, report the fraction of the burst slots (# burst slots / # observed slots) – for all resolutions. **[5]**

B. Load the following webpages in your browser and capture the traffic exactly for 10 seconds (explore -G option in tcpdump for timing). **[5]**

- https://www.deccanherald.com/ deccan.pcap
- https://www.jagran.com/ jagran.pcap
- https://www.mit.edu/ mit.pcap
- https://www.usach.cl/ usach.pcap
- https://www.sinu.edu.sb/ sinu.pcap

1. Report the total time required for the DNS query to complete, for all five websites. Any insights that you can draw from the numbers? **[5]**

2. Report the Time-To-First-Byte (TTFB) for all five websites. This includes the elapsed period from the time when the DNS query is made and the time when first application data (after 'server hello') arrives. **[5]**

3. For a given website, assume a total of N Kilobytes transferred as a part of the downlink traffic in the 10 seconds of your trace. Discretize the timeline into 100 ms slots as before. For each slot, plot the **cumulative** percentage of data that has been transferred till that slot. **[5]**

Activity 2 - Preliminary analysis of HTTP archive data (HAR) [20 Marks]

*Capture HAR traces and perform analysis as instructed. Submit the notebook ipnyb file. (**har_analysis.ipynb**) Use haralyzer for the analysis.*

A. Load the same set of webpages as done in Activity 1 and record the network trace as HAR files. Name them *deccan.har, jagran.har, mit.har, usach.har, simu.har*. **[5]**

For each trace, perform the following analysis.

1. Report the TTFB and page load times. Use `page_load_time` attribute for `HarPage.` **[2]**

2. Report the total number of HTTP GET requests made to the server. **[1]**

3. List the different MIME types observed in the requests. You can read up on common MIME types from here: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types **[2]**

4. What fraction of GET requests correspond to: (a) Images, (b) javascript and (c) HTML and CSS. **[5]**

5. What is the total size of the assets downloaded as a part of the responses? What fraction of the size constitute of the images? What is the mean and median size of the images? **[5]**

Activity 3 – DNS packet parser [40 Marks]

*Write a parser to retrieve the domain name and the corresponding address from a relevant DNS query/answer packet. Submit the notebook ipnyb file. (**dns.ipynb**)*

You have been provided the RFC 1035 that provides documentation on the DNS message structure. In the following, contents of a few DNS packets are shown in the hexadecimal byte stream format.

- If it indicates a query packet, print the domain name that is queried.
- If it is a response packet, print both the domain name and the corresponding IP address.

For this assignment, we only restrict to one answer record per query (although in real life there can be more – e.g., CNAME/alias, multiple IP addresses). Also, we will restrict ourselves to only IPv4 addresses – i.e., 4-byte addresses.

Packet 1

```
4c76010000010000000000010c74696d65736f66696e6469610a696e64696174696
d657303636f6d00000100010000290200000000000000
```

Packet 2

```
629f8180000100010000000103637365046969746d02616302696e0000010001c00
c0001000100001f9600040a0608020000290200000000000000
```

Packet 3

```
00c88b57ec40ec2e98e9046b08004500004b527e40004011802f0a2a52f10a1800c2
e36400350037875137920100000100000000000106636c69656e740764726f70626
f7803636f6d00000100010000290200000000000000
```

Packet 4

```
ec2e98e9046b34e894fa3f5e08004500006b030400003e11f7c4c0a80001c0a80068
003581c300577fa37baf818000010001000000010f7a2d7034322d696e7374616772
616d046331307209696e7374616772616d03636f6d0000010001c00c00010001000
0000c00049df017ae0000290200000000000000
```

Packet 5

```
a0208180000100010000000105666f6e74730a676f6f676c656170697303636f6d00
00010001c00c00010001000000ae00048efab64a0000290200000000000000
```

Note that your code will be tested against any valid or invalid inputs. If the input is invalid, print “Invalid data”. **Marks will be awarded based on passing of other testcases, clarity and clear documentation of the code.**

Submission Instructions

Put all your scripts along with a README file in a directory names <roll_1>_<roll_2>, and compress it in the tar.gz format.

Put the pcap and HAR files in your smail Google drive and include a link of the drive in the README file. Don’t forget to make the link accessible.

Failure in naming the files as instructed will draw additional penalty.

Late Submission Policy

The deadline is at 23:59pm on 11th February 2024.

Late submission by each additional hour will attract a penalty of 1%. For example, if you delay the submission by one complete day, you will face a penalty of 24%.