CS3205 (Semester: Holi 2025) Programming Assignment 1

Submission Deadline: *Friday, 14th February, 2025* Total: *25 Marks*

Exercise 1. [10 marks]. In this assignment, you will analyze network traffic from a recorded packet capture (PCAP) file. The network consists of a server with an IP address of **`192.168.1.109`** and five clients sending uplink traffic to the server. The clients have the following IP addresses:

`192.168.1.57`, `192.168.1.77`, `192.168.1.139`, `192.168.1.141` and **`192.168.1.224`**

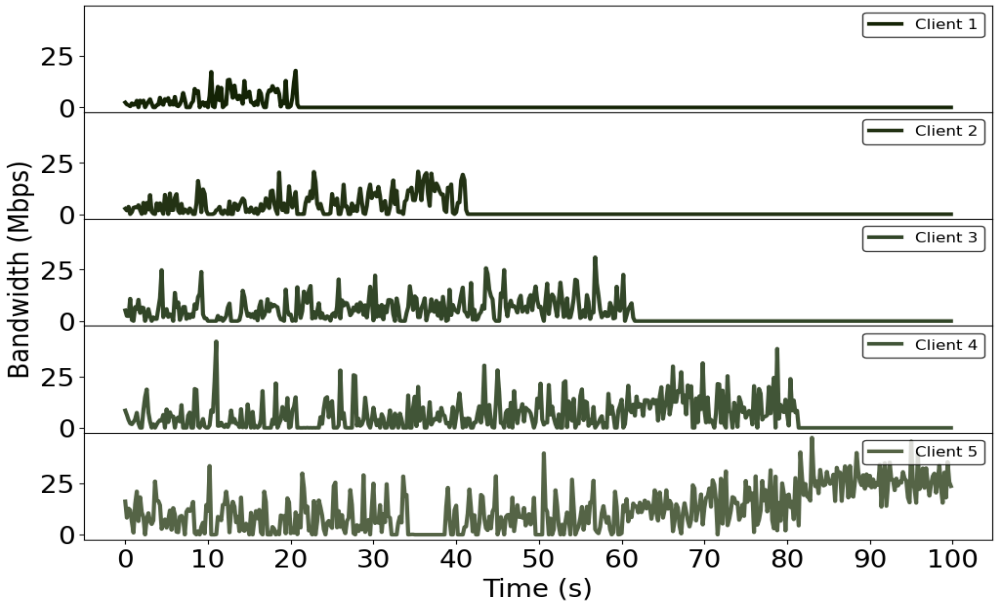
The captured network traffic spans a duration of 100 seconds, and your analysis will focus exclusively on these six IP addresses. All other IPs in the trace should be ignored. Additionally, only packets of size 1514 bytes should be considered for the analysis. If no packets of this size are observed within a given interval, the throughput should be recorded as zero for that interval.

You will write scripts to answer the following questions using the given PCAP file (link). For clarity, create a single Jupyter notebook for the following analysis.

1. ***Per-Client Throughput Timeline:*** Compute and visualize the uplink throughput (bytes per second) for each of the five clients over time. The throughput should be calculated using a time granularity set by the user. Implement a function with the following signature:

   ```
   get_data_transfer_intervals(pcap_file, src_ip, interval_ms=100)
   ```

   The function should return the amount of data transferred by the specified source IP in each interval of the given duration (default: 100ms) as a list. Implement a function, **`plot_timeline`** that takes such lists for the five clients and generates a timeline plot showing the average throughput of each client. A reference figure (computed using a 500ms granularity) is provided for guidance - the client IPs are not mentioned in the plot, of course you need to mention them.



2. ***Total Throughput Plot:*** Calculate and plot the total throughput across all five clients for the entire 100-second period. The x-axis should represent time (in seconds), and the y-axis should represent the aggregated throughput (bytes per second) using the same **`interval_ms`** parameter.

3. ***Throughput Fraction Analysis:*** Compute and state the fraction of total throughput contributed by each client within the following time intervals:

   0 – 20 seconds
   20 – 40 seconds
   40 – 60 seconds
   60 – 80 seconds
   80 – 100 seconds

   Your analysis should focus on how traffic is distributed across clients and whether there are periods of high or low contribution from individual clients.

Suggestion: Use scapy (example work) or pyshark to do the analysis. To improve performance and avoid processing the entire large PCAP file at once, split the file into smaller chunks based on the source IP addresses of the clients. Use the **`tshark`** tool to filter traffic per client and generate five separate PCAP files, each containing packets sent by one of the clients.

```
$ tshark -r input.pcap  -w out_224.pcap -Y "ip.src == 192.168.1.224" -t ad
```

After filtering, use the **`get_data_transfer_intervals`** function on the smaller PCAP files to extract the data transfer timeline for each client and proceed with the rest of the analysis.

Exercise 2. [5 marks]. In this exercise, you will capture network traffic using tcpdump, analyze HTTP responses containing **`text/*`** MIME types, extract and reconstruct the payloads, and compare the raw HTTP payload size with the uncompressed (plain text) content size.

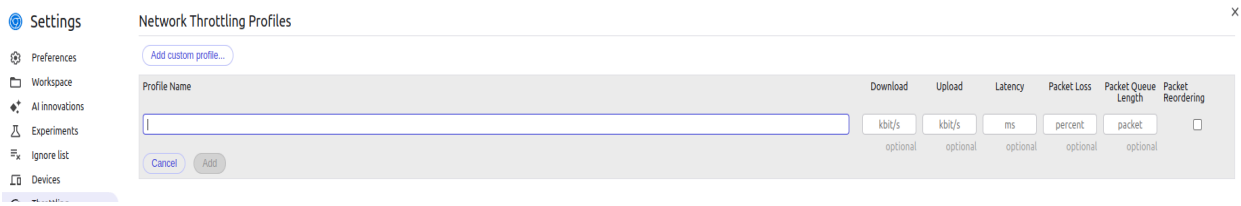
1. ***Data preparation.*** Capture the PCAP using **`tcpdump`**, by visiting this website in incognito mode, http://sneaindia.com/index.php. Use Wireshark to find the PDU corresponding to the homepage HTML content and note the frame numbers of the corresponding reassembled TCP segments. Using **`tshark,`** export each frame as a hex stream and save them in a file - every segment is stored in a new line, sorted according to the frame numbers. Save the file as "**`htmlhexdump.txt`**" (*this entire procedure was explained and demonstrated live inside classroom*)

2. ***PCAP parsing.*** Parse the hexdump file, remove the headers and extract the HTTP response payload from each segment. Do not use any packet parsing

libraries. Join the payloads individually and print the total size of the response data. Next, decompress the data using the python gzip library and print the total size of the extracted text/html response data. What is the compression ratio? (*this entire procedure was explained and demonstrated live inside classroom for both image and text data*)

Exercise 3. [10 marks]. In this exercise, you will analyze HTTP Archive (HAR) files to study web performance metrics. You will record HAR traces for loading a specified website, extract relevant data, and perform an in-depth analysis using Python.

1. ***Collect HAR Traces.***
    a. ***No network throttling.*** Load the following website in a web browser (incognito mode): India Post Website (https://www.indiapost.gov.in/vas/Pages/IndiaPostHome.aspx). Save it as *IndiaPost_nothrottle.har*.
    b. ***With network throttling.*** Next use the network throttling option to configure three custom network profiles, (100Kbps uplink, 100Kbps downlink, 100ms latency)
    → *IndiaPost_Config1.har*
    (1000Kbps uplink, 1000Kbps downlink, 500ms latency)
    → *IndiaPost_Config2.har*
    (1000Kbps uplink, 1000Kbps downlink, 500ms latency, 1% packet loss)
    → *IndiaPost_Config3.har*



2. ***Analysis of requests/responses.*** Write a Jupyter Notebook to analyze the four HAR files. Each of the following questions should be addressed in a separate code cell after you read the files and write utility functions. Use the haralyzer python library as demonstrated in class.

    a. ***Page Load Time.*** Extract and display the total page load time from the HAR files (all four network configurations)
    b. ***Request and Data Transfer Summary.*** Determine the total number of requests made during the page load. Compute the total size of data transferred in response to these requests.
    c. ***Content-Type Analysis.*** Count the number of requests and total data transferred for the following content types: text/html, text/css, application/javascript, image/* (all image formats)

d. ***CDF of Download Times.*** Plot a Cumulative Distribution Function (CDF) for the download time of all response elements. Show all four network configurations in the same plot - use legends.
e. ***CDF of Response Sizes.*** Plot a Cumulative Distribution Function (CDF) for the download size of all response elements.
f. ***Scatter Plot: Size vs. Download Time.*** Generate a scatter plot showing the relationship between resource size and download time for all responses. Analyze whether there is any correlation between these two parameters. Have a 1x4 subplot to show the four scatter plots.

Submission Instructions: Create a folder called CS3205_Assignment1_roll1_roll2. Have three subfolders: ex1, ex2 and ex3. In each folder have the relevant notebook, HAR (only for ex3) and PCAP files (only for ex2). Compress the root folder as a gzipped file and submit CS3205_Assignment1_roll1_roll2.gzip

Late Submission. The deadline for submitting this assignment is 11:00 pm, 14th February. After this there will be a late penalty of 25% for each day for the next four days. You are not required to submit the assignment solutions beyond 18th February.

Viva. We will conduct a viva for every group. The viva will not be limited to the specific questions asked in this assignment, but to the specific topics covered here.