National University of Sciences & Technology
School of Electrical Engineering and Computer Science
Department of Computing
EE353: Computer Networks, BSCS-6AB Fall 2018

Project Specifications	
CLO 4: Design and implement solutions to contemporary networking issues (through hands on programming)	
Maximum Marks: 20	Instructor: Dr. Arsalan Ahmad
Date: 6th Nov, 2018	Due Date: 23rd Dec, 2018

1 General

This is a group project. Maximum size of the group is restricted to two members.
You are advised to avoid plagiarism; any such case would result in award of zero marks both to the "sharer" and the "acquirer".

2 Learning Objectives

In this project, your task is to build an HTTP file downloader that supports resuming and multiple simultaneous connections. Your HTTP client should be able to download a file present both from a specified location on the internet and the HTTP server that you will set up. Additionally, you will have to report some metrics at specified intervals while the file is being downloaded. On completion of this project, you will gain sufficient expertise in the following skills:

A. Details of HTTP
B. Details of various metrics used to analyze the connection quality
C. Socket programming in python
D. Multi-threading in python

3 Specifications

You are required to use the Python programming language while working on this project. You cannot use any high level HTTP libraries. You can use sockets library for TCP and UDP functionality.

3.1 HTTP Server

You don't have to write code for the HTTP server. You can look up instructions online on how to set one up. Figuring this out is part of the project. The server should be set up in a way that your HTTP client is able to download the file hosted by this HTTP server if its address is given to it. The server should be set up on a different computer than the one on which the client is set up.

3.2 HTTP Client

Your HTTP client should be named client.py. The command line syntax for the client is given below:

```
      client.py -r -n <num_connections> -i <metric_interval> -c <connection_type> -f
      <file_location> -o <output_location>
```

`-n`	(Required)	Total number of simultaneous connections
`-i`	(Required)	Time interval in seconds between metric reporting
`-c`	(Required)	Type of connection: UDP or TCP
`-f`	(Required)	Address pointing to the file location on the web
`-o`	(Required)	Address pointing to the location where the file is downloaded
`-r`	(Optional)	Whether to resume the existing download in progress

As an example, if you want to download a file in current directory (denoted by '.') using 8 simultaneous connections located at http://www.example.com//myfile.png using TCP and report the metrics every 0.5 seconds, you would write:

```
      client.py -n 8 -i 0.5 -c TCP -f http://www.example.com/myfile.png -o .
```

3.3 Download Using Multiple Connections

The client must use the HTTP functionality that allows downloading a file between specific byte ranges to download using multiple connections. You will have to specify it in the HTTP header. A thread should be created for each connection so that they can all download in parallel. Each thread should output a file as a result of its download. After all threads finish downloading, these files should be combined to form a single file.

Note that some HTTP servers will not support byte ranges. You should detect that using HTTP responses and force downloading using a single connection.

3.4 Download Resuming

If the download resuming flag is specified (-r), the client should check the output directory to see the status of current download and attempt to resume it using the HTTP byte ranges. We will test this functionality by disconnecting the server from internet connection, disconnecting the client or simply terminating the client process.

Some HTTP servers do not support download resuming. Your code should detect this, display an output indicating the case and start the download from scratch in such cases.

3.5 Output and metric reporting

The client should print an output that indicates the download status according to the interval specified to -i flag. Suppose the file is being downloaded by using 4 simultaneous connections, the output in this case should be according to the follow format:

```
Connection 1: <downloaded_bytes>/<total_bytes>, download speed: <speed>kb/s
Connection 2: <downloaded_bytes>/<total_bytes>, download speed: <speed>kb/s
Connection 3: <downloaded_bytes>/<total_bytes>, download speed: <speed>kb/s
Connection 4: <downloaded_bytes>/<total_bytes>, download speed: <speed>kb/s
Total: <downloaded_bytes>/<total_bytes>, download speed: <speed>kb/s
```

If the interval specified to -i flag is 2, then this output should be printed every 2 seconds. You should report the download speed of each connection as specified.

4 Grading Criteria

You will be graded based on the following grading criteria:

Criteria	Weightage (%)
Single connection downloading using TCP and UDP	20
Multiple connection downloading using TCP and UDP	35
Download resuming	20
Correct input, output and metric reporting	15
HTTP server setup	5
Code is well documented, clean and readable	5

Late penalty (on your received marks) will be applied as follows:

- 1 day after deadline: 25% reduction
- 2 days after deadline: 50% reduction
- 3 days after deadline: Not accepted

5 Submission and report

You are required to submit your source code and a short report to an upload link that would be made available on the LMS. Zip your source code files and your report document in a file named Project_Surname_Surname (Surname of both Group members). Nominate one of the group members to submit on behalf of the group. Please note that you must upload your submission BEFORE the deadline. The LMS would continue accepting submissions after the due date. Late submissions would carry penalty per day with maximum of 2 days late submission allowed (see section 4). Students who fail to submit would not be allowed to appear in viva and those who miss the viva would not be allowed to retake the viva.

Your submitted code would be checked for similarities and any instances of plagiarism would result in award of ZERO marks for all such students, irrespective of who has shared with whom.

You must write down group members' Registration Nos and Names at the beginning of the report. All reports will be read for marking purposes.

The size of your report MUST be under 2 pages. Your report should briefly document your techniques and methodology used for implementation and how you combat the relevant problems in development. Treat it as a summary document only (point form is acceptable). Your report should contain analysis obtained over a test case where you download a file using TCP and UDP by varying the number of connections. You will be asked to demonstrate your program during your viva.