Networks Sub-module Assignment

(2023-2024)

1. Assignment introduction

This assignment includes three parts. Part 1 is to code an Instant Messenger in Python 3.11. Part 2 is to analyse a wired and wireless integrated network. For detailed descriptions and tasks, please refer to the individual parts.

You should hand in your completed assignments via Blackboard Ultra. The deadline for your submissions is **2pm Thursday January 11th, 2024**. Each of you should include the following five files in your submission: the client source code (client.py), the server source code (server.py), the log file (server.log), a readme.txt containing any notes about the usage of the program and a PDF file answering the tasks in Part 2.

Please note that all submissions will be subject to plagiarism and collusion checks.

2. Assignment descriptions and tasks

Part 1: Implement an instant messenger (50 marks)

Note that all code should be written in Python 3.11 and must be stored in files named server.py and client.py. All code must run in Windows. You must use the "socket" library directly (i.e., not via any other Python module) for network communications.

You are required to implement a client-server system, which implements an instant messenger **using TCP** allowing users to chat with each other. This instant messenger will consist of a client and a server program. You should be able to invoke server and client as follows:

```
python server.py [port]
python client.py [username] [hostname] [port]
```

For example, `python client.py John 127.0.0.1 12000` would enable a client using the username "`John`" to connect to `127.0.0.1` (i.e., the local system) via port number `12000`.

Please complete the following functions required for the server or clients in your programs.

1) Server and client connection functions (15 marks):

a) When a Client connects, on the Server, print where the connection is coming from (including the IP address and the port number of the Client).
b) When a Client connects, on the Client, display a simple welcome message from the server. Note that this message must be sent over a network socket and should not be hardcoded on the client side.
c) Allow multiple Clients to connect to the same Server.
d) On the Client, provide an input prompt allowing the client to send messages.
e) Allow Client to leave the system by implementing a command.
f) Use the username passed to client.py to print "[username] has joined" or "[username] has left" on all connected Clients whenever a Client connects or disconnects from the server. Note that the leaving messages should be displayed for intentional disconnections (i.e., by issuing the command) as well as unexpected disconnections (say connection interruptions or the client process being terminated by the operating system).
g) One of the connected clients disconnecting (intentionally or unexpectedly) should not cause the server to crash.

2) Messaging functions (20 marks):

a) Client should be able to send multiple messages.
b) Client should be able to broadcast messages to all other clients. Broadcast means that a message is sent to all other clients excluding the one that sends the message.
c) Client should be able to unicast messages to another client individually when there are more than two clients in the system. Unicast means to send messages to a single client.
d) Client should be able to change between the above messaging modes.
e) Create a download folder on Server.
f) Allow Client to request a list of files in the download folder on Sever. This should be displayed on Client prompt. Note that this must be sent over a network socket and should not be hardcoded on the client side.
g) Client will be able to choose a file to download. Downloaded files should be put in a folder named by the username of Client. Note that this must be sent over a network socket and should not be hardcoded on the client side. Also, files with different types of content (texts, images, videos) should be able to be downloaded.

3) Logging functions (8 marks):

a) Produce a log file called server.log when the server is run.
b) This log should contain information about all the above connection and messaging functions and activities. Note that the log file should not include the content of the files downloaded.

4) Documentation (7 marks):

Please submit a README.txt file to include clear and easy-to-follow instructions on how to implement the above functions with your code.

****

Hints for Part 1:

1. Be sure to build up the functionality of your application step by step, for example, do not attempt to write the entire server before starting the client.
2. If you are unsure of how to start, I suggest you to have a look at the lab assignments 2 & 3.
3. I also suggest using a dictionary to store all current connections. Ensure that when a client drops out, that connection is removed from the dictionary.
4. You may find it useful to look at non-blocking sockets and select(), to ensure that all messages can be received. E.g. (note the format of the if statement):

```
r,w,e=select.select([client.connection],[],[client.connection])
   if client.connection in r: [...]
```

5. It is ok to send all messages directly as fixed length byte arrays over the sockets.

****

Part 2: Analyse a wired and wireless integrated network (35 marks)

Refer to the following network scenario.

A wired and wireless integrated network has the following devices (note that the switch and all the devices are cold-started).

- A wireless access point (denoted as X) with the MAC address XX-XX-XX-XX-XX-XX
- Two wireless user devices (denoted as A and B respectively) with MAC addresses AA-AA-AA-AA-AA-AA and BB-BB-BB-BB-BB-BB respectively
- A layer-2 8-port switch (denoted as S)
- A wired user device (denoted as C) with the MAC address CC-CC-CC-CC-CC-CC

Assume that, in this network, devices A and B can hear each other's transmissions. They can also hear X and X can hear A and B. The communications between A, B, and X adopt the **CSMA/CA** scheme **without acknowledging** whether a frame is received or not. The wireless access point X connects to the switch S via a cable through port 1. The user device C connects to the switch S via a cable through port 6. Suppose

- At time 0 μs, C is sending a frame to the switch and the destination is some other wireless node covered by X. C completes sending this frame at time 50 *us*.
- At time 20 *us*, a frame becomes available for transmission at A to C. A needs 60 μs to send this frame.
- At time 40 *us*, a frame become available for transmission at B to C. B needs 80 *us* to send its frame.
- Let the value of the backoff timer for A be 10 *us*, the value of the backoff timer for B be 12 *us*, and the value of the backoff timer for X be 8 *us*. (Suppose these backoff timers are fixed.) DIFS and SIFS are 5 *us* for A, B, and X.

Answer the following questions.

1) Sketch a topology to accurately reflect the connections of the network described above. Your topology should include all devices mentioned and their connections. As for wireless devices, you may use coverage to show connections. You may use tools such as Word, Paint, Visio, etc. to complete this topology. (10 marks)

2) Which wireless user devices above can receive the frame sent by C? Why? (4 marks)

3) At what time does A start sending its frame (i.e., putting the frame on the transmission medium) to X? At what time does B start sending its frame to X? Explain. (8 marks)

4) Give the switching table of S at 60 *us*. Explain. (8 marks)

5) If you connect a computer to port 2 of S, which frame(s) can you receive from all the above processes? Explain. (5 marks)

3. How to submit?

You will submit five files:

- server.py
- client.py
- server.log
- readme.txt
- a .pdf file. In this file, you present your answers for Part 2. A template for the .pdf file is in the "Assignment" folder of the Networks sub-module on Blackboard Ultra.

Please compress all files into a single .zip file, and name your .zip file as "YourName_YourID.zip" (make sure to replace "YourName" with your own name and "YourID" with the ID given to you by the University. For example, `JohnDavidson_cgab36.zip`). Please upload your compressed file to Blackboard Ultra for submission.

The submission deadline is **2pm Thursday January 11th 2024**.

4. Collaboration policy

You may discuss your work with anyone, but you must do your work yourself and comply with the University rules regarding plagiarism and collusion (https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/1/). Your submissions will be assessed for collusion and plagiarism.

5. Feedback Sheet

Criterion	Marks	Comments
Part 1 Instant messenger (50)		
Server and client connection functions.	/15	
Messaging functions.	/20	
Log file.	/8	
Documentation.	/7	
Part 2 Analyse a wired and wireless integrated network (35)		
Topology.	/10	
Receiving C's frame.	/4	
Timing of wireless user devices' transmissions.	/8	
Switching table.	/8	
Frame(s) at Port 2.	/5	
Others (15)		
Code quality.	/8	
Presentation and readability (for the .PDF file).	/7	

Code quality (8 marks):

The code should be short, easy to read and understand. In particular, this means that:

- Program code is commented where necessary for understanding. Excessive comments should be avoided.
- Programs are appropriately structured (e.g., correct and consistent indentation style).
- An appropriate naming convention is used for variables, methods/functions and classes throughout the project. See https://www.python.org/dev/peps/pep-0008/.
- Appropriate use is made of external libraries and there is no evidence of redundant code, e.g. importing unused Python modules, functions that are never called, etc.
- Correct/appropriate use of conditional statement and iterative statements.
- The code is not over-engineered, i.e. it does not introduce unnecessary data structures or include overly-generic functions.

Presentation and readability (7 marks):

Content should be concise yet clearly describing the full idea and protocol. In particular, this means that:

- Words should highly focus on answering the questions or describing the protocol steps. Too much background content or other irrelevant content will lose points. Too less words may enable the marker to match the answers to the right questions.

- Avoid very long sentences. Consider using different ways of presenting your ideas (say tables, figures, etc.) as appropriate.
- Font(s) should be legible, particularly for those in a figure or a table if you use figures or tables.
- Format and fonts should be consistent throughout the file.