Project 4 – Advanced TCP/UDP Network Programming

1. Supporting multiple TCP clients (30 Points)

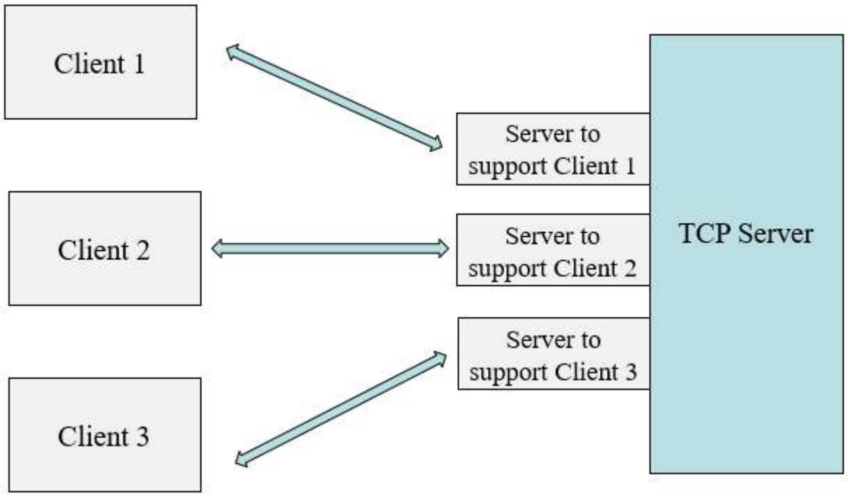
Client 1
Client 2
Client 3
Server to support Client 1
Server to support Client 2
Server to support Client 3
TCP Server

Figure 1. Architecture Question 1.

Design a TCP server capable of handling simultaneous connections from multiple clients, with a maximum limit of five clients. The server's architecture, as depicted in Figure 1, is engineered to enable broadcasting to all connected clients through the TCP server. It is essential that each client can identify the sender of a message. Therefore, assign a unique ID to each client. This design ensures that when a message is broadcasted from one client, all other clients can identify which specific client sent the message.

2. TCP Server and Client on a Multi Hop Network (30 Points)

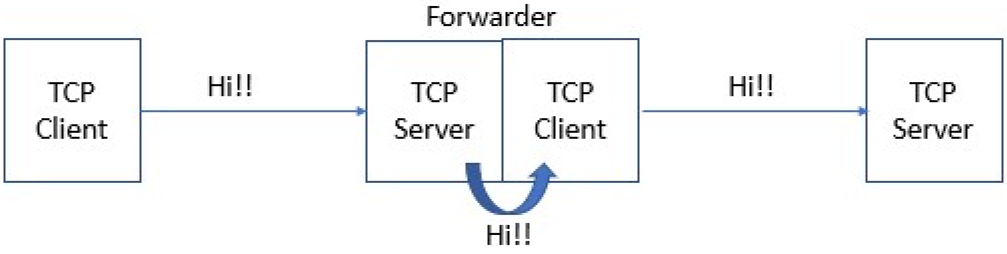

Figure 2. Architecture of Forwarder

You will be working with three Python scripts to set up a network communication model: (1) TCP Server, (2) TCP Client, and (3) Forwarder.py, which acts as a hybrid of both a TCP server and client, as demonstrated in Figure 2. Please refer to the TCP server and client code provided in Project 3 as a starting point. For this task, simultaneous operation of three and four terminals is required for testing.

Design "forwarder.py" as the following:

- A TCP server must start first and wait for a forwarder. The forwarder contains TCP server and clients. A TCP client in the forwarder will establish a connection to the TCP server. For this connection, you must use a TCP client in the forwarder. **(15 Points)**
- Once your forwarder is connected to the TCP server, the TCP server in the forwarder will start and then, a TCP client must start. If you type a word on the TCP client, the TCP client will send this word to the forwarder. The forwarder will forward this word to the TCP server as shown in Figure 1 (1**5 points)**
- Conduct tests for scenarios involving a single forwarder as well as multi-hop networks comprising two and three forwarder stages. These tests are crucial for evaluating the effective handling and forwarding of data across varying network complexities.

3. Combined TCP and UDP (40 Points)

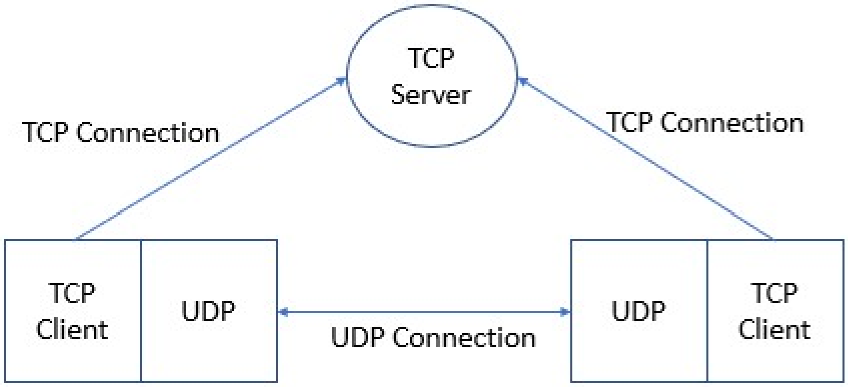

Figure 3. Architecture of Multiple TCP Clients

- This task involves the creation of a network model comprising a TCP server connected to multiple TCP clients. These clients will engage in both TCP and UDP communications as depicted in Figure 3. The core focus is on establishing a versatile communication framework where TCP clients can interact with each other using UDP, in addition to maintaining a TCP connection with the server.
  - Ensure that at least two TCP clients are connected to the TCP server. Subsequently, these clients must establish a UDP communication channel between themselves. This setup is essential for facilitating direct client-to-client interactions. (10 Points)
  - Each TCP client is required to demonstrate the ability to send messages to another client using UDP, while also maintaining the capability to communicate with the TCP server

over TCP. This dual-protocol communication must be seamlessly integrated within your code (20 Points)

- Implement a mechanism whereby if any client sends an "exit" message, it triggers both involved TCP clients to gracefully close their TCP connections and cease all UDP-based communications. This functionality is crucial for ensuring that the network can be orderly disbanded when necessary. (10 Points)

Note:

- Deadline: See the deadline in the submission site on Blackboard.
- Submission: **your python codes and screenshots showing what you have tested for each question**.
- Late Submission: 10% penalty per day