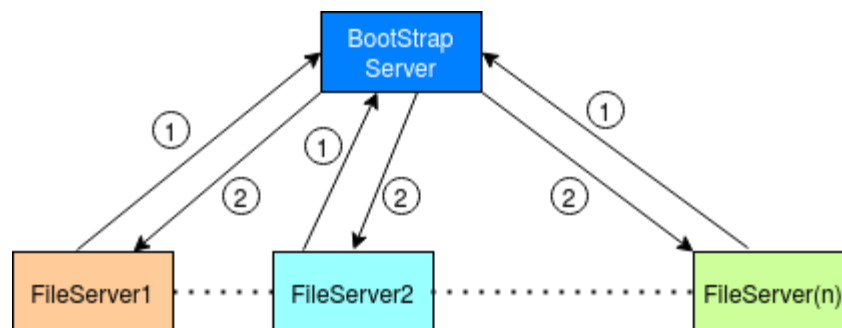


# Socket Programming

## Total Marks: 65 Marks

### Part A

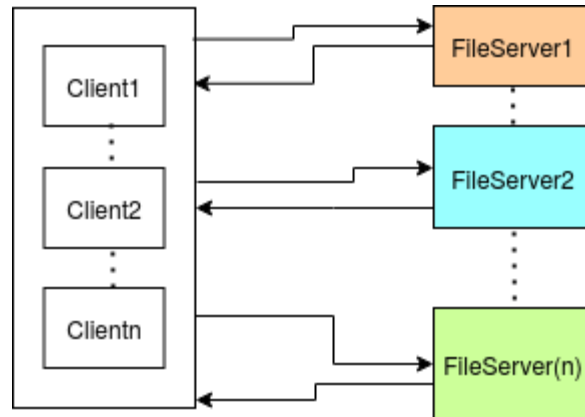
1. Create a Bootstrap server that accepts the serving information from different servers like servicename, IP address, port number, and a service access token on UDP. (e.g: servicename: Videoserver, servicetype: video, IP Address: X.X.X.X, Port number:A, service access token: abcd). Later it should allow a client asking for these servers' information on UDP.



You are free to use either two different sockets (1. Registration of services by the servers and 2. Discovery of the servers by the client) **or** in a single socket define the payload of UDP in a manner to distinguish the incoming message type (MSGTYPE: REGISTRN for registration of services by the servers and MSGTYPE: DISCOVERY for discovering them by the client)

2. Create a file server program.
  - a. Upon starting this file server, it registers to bootstrap server on UDP with the details given in step#1.

## Part B



- a. Extend the file server program in **Part A**. Once it registers to the BootStrap server successfully in Part A, it listens on TCP socket. The server is meant to send a specific file to its clients depending on the service type it is intended for. Like a video server would send a video file type which a client asks for.
- b. Server shall be able to serve multiple clients in parallel. i.e. if two different clients ask for a file, the server shall satisfy both the clients in parallel without making any client wait. Maximum number of clients to honor can be set at the server. You are free to use the file server code you wrote in **Assignment 2**.
- c. When any TCP client arrives asking for file(s), it (server) **first verifies the access token in the data received (sent by the client)** and matches it against its own access token. It proceeds to serve the client only if the access token matches, else it sends a message back to the client saying "Invalid client".
- d. Create a client program
  - i. This client interacts with Bootstrap server on UDP and file servers on TCP.
  - ii. First the client fetches all the servers' information from the BootStrap server using UDP. Client shall get each of the registered server's information with service name, respective IP address, port number, and an accesstoken (e.g: servicename: Videoserver,

servicetype: video, IP Address: X.X.X.X, Port number:A, service access token: abcd) from the BootStrap server.

iii. It then connects to each of the servers returned from the BootStrap server to fetch the list of files , by providing the corresponding servicename/type.

1. Here, the client shall first send the access token. Upon getting the successful answer/confirmation from the server, the client then starts to fetch various files of the corresponding type from the server by sending the file name(s) of the type iteratively. The server in this step shall simply if the respective file type and name matches it/is available with it. If so it sends the file to the client, if not it replies “file doesn’t exist”. You are free to use the client code you wrote in **Assignment 2** where identified.

e. Assume that a valid client knows the ip address, port number and the service access token of each of the file servers.

### **Final Execution**

- a. Create a single BootStrap server, 4 different file servers for video/audio files, .pdf, .text files, image files respectively. Create at least two valid clients mentioned in part B.
- b. Also test the invalid client just by feeding/configuring an invalid access token and sending it any of the file servers. Invalid client shall randomly generate a service access token and send it to the server. However, the server shall be able to reject this client if it is a mismatch of the token received from the client against the required one. The respective file server shall display “Invalid client” on its terminal as well as on the terminal of such an invalid client.

### **Submission**

Add appropriate comments to your code in the programs to make it readable. Prepare a single zip file with name format **<assignment3\_roll no>.zip** including its source files and respective readme of your programs. Submit it to google classroom in the posted assignment section.

### **PLAGIARISM STATEMENT <Include it in your readme>**

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether

they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honor violations by other students if I become aware of it.

**Name of the student**

**Roll No**