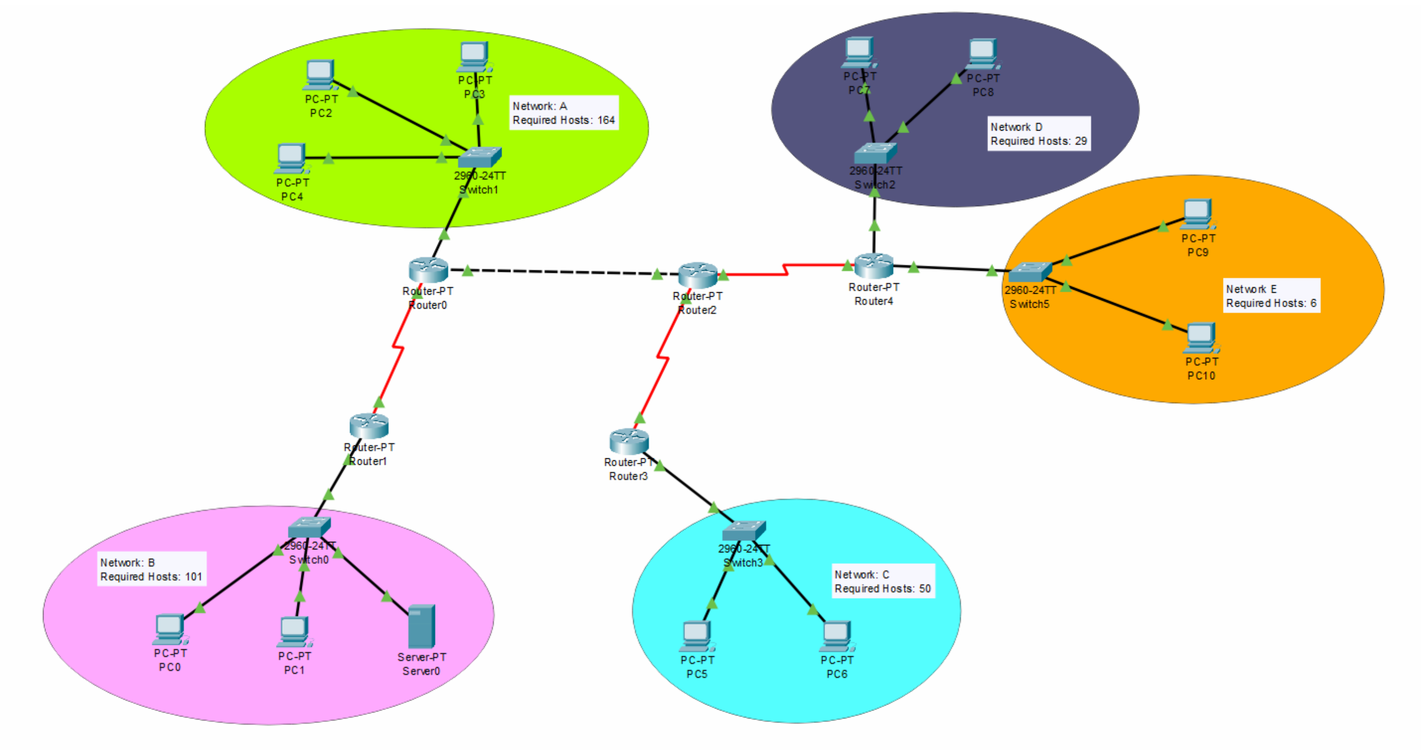
COMSATS UNIVERSITY ISLAMABAD
Lahore Campus
Department of Computer Science

☐ **Mid-Term Examination** ■ **Terminal Examination Fall 2024**

Course Title:	Computer Networks				Course Code:	CSC340	Credit Hours:	4(3,1)
Course Instructor/s:	Inzmam ul haq				Programme	BS Software Engineering		
Semester:	4th	Batch:	SP23	Section:	Name:	Date:	26-12-2024	
Time Allowed:	**150 Minutes**				**Maximum Marks:**		**50**	
Student's Name:					Reg. No.			

Q1. Consider the scenario given below. : [CLO: 6; Bloom Taxonomy Level <Applying>]

(20 Marks)



Use the given class C IP address for this network scenario:

192.168.0.0

192.168.1.0

A. Configure and assign IP addresses for each network. For this fill out the subnetting table required fields given below. **(10 marks)**

Network Name	Network IP Address	First Usable IP Address	Last Usable IP Address	Broadcast IP Address	Prefix	Subnet Mask

B. Apply static routing. **(7 marks)**

C. Apply ACL on R1 by restricting traffic from R0 (Switch 1). **(3 marks)**

Q2. Consider the Trace file given and answer the following questions:
[CLO: 6; Bloom Taxonomy Level <Applying>] (18 Marks)

A: Complete the TCP three way handshake on the port that completes the data sending from client to server: **(4 Marks)**

Three way Handshake	Packet Number	Source Port	Destination Port	Flags On	Sequence number (relative)	Acknowledgment number (relative)
First Handshake						
Second Handshake						
Third Handshake						

B: Consider the Acknowledgment number **41029** in packet **157** as the first segment. Now look for the next 5 segments that receive the acknowledgment upto **63337** ack and calculate the difference between them also calculate the estimated RTT for each of the **6 ack packets**? **(6 Marks)**

C: Are there any TCP Duplicate Ack segments in the entire trace file? If yes, determine the very first TCP Duplicate Ack packet in the provided data. **(1 Marks)**

D: What is the **throughput** (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value? **(3 Marks)**

E: What's the amount of available buffer space advertised at the Sender end for the entire trace??

(1 Marks)

F: What is the length of each of the six TCP segments that was sent from the client side towards sever in **part B**?

(1.5 Marks)

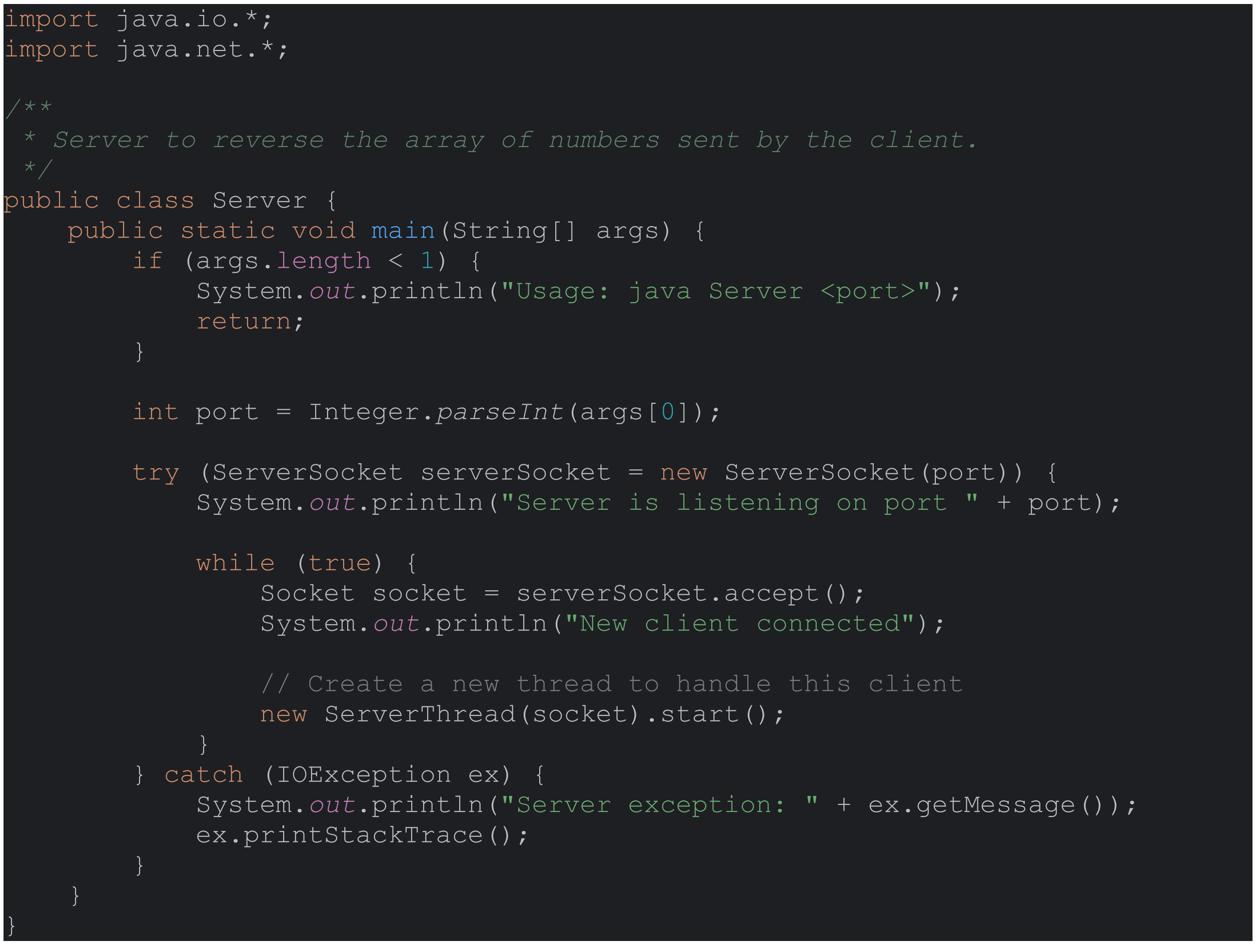
G: Consider the subsequent TCP **FIN** packet sent by the server to end communication. What is its packet no and flag on other than **FIN** also in response to this **FIN** what is the Ack no?

(1.5 Marks)

Q3. Build a simple **client-server Multi-threaded system**. The protocol between the client and server is as follows.

1. The **server** is first started on a known port.
2. The **client** program is started (**server IP** and **port** are provided on the command line).
3. The **client** connects to the server and then asks the user for a comma-separated list of numbers (e.g., 37, 2, 11, 9, 21, 62).
4. The **user's input** is sent to the server via the connected socket.
5. The **server** reads the user's input from the client socket, **reverses the order of the array**, and sends the reversed array back to the client.
6. The **client** displays the server's reply to the user and prompts the user for the next input until the user terminates the client program by typing **"Exit"** or using **Ctrl+C**.

[CLO: 5; Bloom Taxonomy Level < Creating >] **(12 Marks)**



```java
import java.io.*;
import java.net.*;

/**
 * Server to reverse the array of numbers sent by the client.
 */
public class Server {
    public static void main(String[] args) {
        if (args.length < 1) {
            System.out.println("Usage: java Server <port>");
            return;
        }

        int port = Integer.parseInt(args[0]);

        try (ServerSocket serverSocket = new ServerSocket(port)) {
            System.out.println("Server is listening on port " + port);

            while (true) {
                Socket socket = serverSocket.accept();
                System.out.println("New client connected");

                // Create a new thread to handle this client
                new ServerThread(socket).start();
            }
        } catch (IOException ex) {
            System.out.println("Server exception: " + ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```