Projects Demo

This document outlines the official expectations for the Project 1 and Project 2 demos. Please review all requirements carefully and prepare your demo accordingly. Each group will have 10 minutes total to complete the demo.

Project 1 Demo

We might ask you to show a few of the below mentioned tasks during the live demo. Please do all these tasks on your own and take screenshots of the outputs showing the correct working of your project so that we can see that during the demo as well.

1. Explain within a minute what you did in this project
2. Show your Makefile
3. Run the server and show how it is listening on port 7734
   3.1. There should not be any errors in this
4. The server must print logs when clients connect to make the demo faster
   4.1. You should print something like this:
   4.2. [Server] Connection from host peerA at 192.168.1.5:5678
5. Next, start Peer A and Peer B, each with a few RFCs
   5.1. Peer A should show output like this
      5.1.1. [Peer] Connected to server at port 7734
      5.1.2. [Peer] Registering RFC 123: TCP/IP Illustrated
      5.1.3. P2P-CI/1.0 200 OK
      5.1.4. RFC 123 TCP/IP Illustrated peerA 5678
   5.2. Server should show output like this
      5.2.1. [Server] Added peerA:5678
      5.2.2. [Server] Added RFC 123 from peerA
6. Show a list of peers connected and RFCs added
7. From Peer A, show a LIST ALL and LOOKUP for an RFC owned by Peer B
   7.1. e.g., peer A runs
      7.1.1. LIST ALL P2P-CI/1.0
      7.1.2. Host: peerA
      7.1.3. Port: 5678
   7.2. Expected response is
      7.2.1. P2P-CI/1.0 200 OK
      7.2.2. RFC 123 TCP/IP Illustrated peerA 5678
      7.2.3. RFC 2345 Routing Protocols peerB 6789
   7.3. Then peer A runs
      7.3.1. LOOKUP RFC 2345 P2P-CI/1.0
      7.3.2. Host: peerA
      7.3.3. Port: 5678
      7.3.4. Title: Routing Protocols

7.4. Expected response is

7.4.1. P2P-CI/1.0 200 OK

7.4.2. RFC 2345 Routing Protocols peerB 6789

8. Have Peer A download RFC 2345 from Peer B

8.1. Peer A sends to Peer B (via upload port 6789):

8.1.1. GET RFC 2345 P2P-CI/1.0

8.1.2. Host: peerA

8.1.3. OS: Windows 11

8.2. Peer B should respond like this

8.2.1. P2P-CI/1.0 200 OK

8.2.2. Date: Thu, 27 Nov 2025 18:10:00 GMT

8.2.3. OS: Windows 11

8.2.4. Last-Modified: Tue, 26 Nov 2025 10:00:00 GMT

8.2.5. Content-Length: 9456

8.2.6. Content-Type: text/plain

8.2.7. <CONTENT OF RFC FILE>

9. Close one peer and show the server log update.

Project 2 Demo

1. Explain within a minute what you did in this project
2. Preparation before demo:
   a. Please ensure:
   b. You prepared a file of reasonable size for all tasks.
   c. No parameters are hard-coded. All must be tunable at runtime.
      i. Window size
      ii. MSS
      iii. Packet loss probability
3. Functional Demonstration (Go-Back-N implementation)
   a. Show that server starts correctly
      i. Run Simple_ftp_server 7735 output.txt p (e.g., p = 0.05)
      ii. Print “Packet loss, sequence number = X” occasionally → every 10 seconds
   b. Show that client starts correctly
      i. Run Simple_ftp_client serverHost 7735 input.txt N MSS (e.g., N = 4, MSS = 500)
      ii. Show that it reads from the file and sends segments.
   c. After running server and client, the sequence numbers in the header should be increasing properly.
   d. And the output file at the server matches the original input file (use diff to show that there is no difference).

4. Error Handling
    a. Simulate high packet loss (e.g., p = 0.3) → client should retransmit after timeout.
    b. Show that out-of-sequence packets are ignored (server prints nothing for skipped packets).
5. Performance Differences
    a. Present performance differences across different MSS values by showing precomputed graphs or logs
    b. Present logs or graphs showing loss and timeout behavior