Final project

Computer Networks Design Laboratory

Socket programming Client-Server Protocol

Published date:	13/05/25
Deadline for the submission:	17/06/25

BEN-GURION UNIVERSITY OF THE NEGEV - COMMUNICATION SYSTEMS ENGINEERING

Final Project Plan

READ BOTH THIS PDF AND THE PRESENTATION ON MOODLE!

In our final project, you will design and write a Client-Server protocol.

You may choose a subject from the following list or suggest a subject yourself. You must email the lab instructor to approve your project's subject.

The following list is only meant to give your ideas for projects; you may think about a completely different subject:

1. Implement a "reliable" TCP with congestion control transport protocol over UDP.
2. Implement a "reliable" multicast streaming protocol over multicast and UDP.
3. Implement a DNS or a DHCP server.
4. Implement a command-and-control botnet with backup. Be able to attack a network.
5. Implement a multiplayer game.
    a. Bingo
    b. Online Trivia
    c. Chat room
    d. Racing game
6. Distributed streaming radio station with user-selected music. (NOT like previous years!)
7. Invent your project

Note that any new offer for a project must present something substantial.

Bonus points may be given for creative and out-of-the-box projects.

1. General guidelines for the project:

In general, all programs and projects must include the following points:

- The programs must be written in *C*, including *socket programming* and *multi-threaded* (using select() threads or processes).
- The programs must work over a network inside *GNS3*, and communication should be presented over several hops (not on the same LAN and not on the same machine).
- The programs must include a client and a server program, which will use a client-server protocol.
    a. The client-server protocol must be documented and defined in your project's design.
    b. The server must be able to run concurrently with several different clients.
    c. The server must hold and use a *multicast* socket (in addition to *UDP\TCP*)

- When applicable, if the protocols you implement are based on real protocols (e.g., DHCP server), they will have to work with the real network software found on the routers or the Linux machines in GNS3.
- Think about network protocols you learn in the Lab to gain an idea about what you need to design:
    a. FSM
    b. Authentication
    c. Keep-alive
    d. Different message types
    e. And more…
- Design your own protocol messages; consider their structures, uses, cycles (if any), and how to design messages with variable lengths.
- **Bonus-** make Wireshark identify the messages of your new protocol!
- Use timers where needed (the select() function provides this function for free).
- Provide your project with some "QA." Consider how you can test your protocol and programs in various real-time scenarios.
    a. Note that we are interested in errors related to communication, such as malformed packets, missing packets, timeouts, etc. We are not interested in user input errors, programming errors, etc.

2. Design Paper:

Each project will include a design paper approved by one of the lab instructors. Each design paper will consist of the following points:

- What is my project supposed to do?
- How will I manage several clients simultaneously (threads, select(), process)?
- What is the communication protocol between the client and server? What timeouts are part of this protocol?
- Present a **state machine** for the client and server's protocols.
- Which tests will I run to test my programs (QA)? **At least 4!**

3. Project Defense:

During your project defense, you will have 15-20 minutes to present your project, and we will have 5 minutes for questions.

Presentation: Please provide a few slides that will present your project.
For example: 1 introduction slide. 1-2 slides to explain what has been done and how you implemented the project. 1-2 slides to show your protocol, the FSM, and the message types. 1-2 slides to show a demonstration of QA.
Project: Present your project. **First**, show it works as expected under normal conditions. **Second**. Show how it behaves under non-optimal conditions when errors appear.

4. Submission:

- Please submit the design paper as a PDF file in Moodle on 27/05. With the file name: design_paper_ ID1_ID2.pdf
- For the final project, please submit:
  - Your source code and instructions on how to compile and run it.
  - Your presentation.
  - Submit a single zip file named: “Project_ID1_ID2.zip”.

.