

Lecturer

Prof. Dr. Christian Tschudin (christian.tschudin@unibas.ch)

Tutor

Jannick Heisch (jannick.heisch@unibas.ch)

Uploaded on

Thu., 16. March 2023

Deadline

Sun., 26. March 2023 (23:55)

Upload to ADAM

Modalities

The exercises have to be done in **groups of 2 students**. Upload only the final version of the exercise to Adam and specify your group partner.

Please upload your solutions (code with detailed comments + short report) in a zip file to ADAM by the deadline at the latest. The report should contain answers to the asked questions as well as instructions on how to run your software.

Your solutions can be realized in C/C++ or Python code.

Task 1 - TCP Chat (3 points)

Implement a chat application based on the reliable protocol TCP.

The goal is to allow a user to exchange text messages (UTF-8 strings) with another user directly via TCP, without an intermediate server. The application has to be fully operable from the command line, i.e. IP address and port of the communication partner are passed as command line arguments and chat input and output are handled via CLI.

User inputs as well as messages of the communication partner should be displayed as fast as possible. To avoid delays, use `select()`^{1 2}.

Task 2 - UDP Chat (4 points)

Now implement an UDP variant of the previous chat application, where a central server mediates direct connections between clients.

As in the previous exercise, the application must be fully operable from the command line.

To establish a chat between two clients, they both have first to log in to the central server using a unique nickname. The server maintains a list containing a mapping between nicknames and the corresponding IP addresses and port numbers. One of the clients then sends the nickname of the desired communication partner, and the server returns the matching IP address and port number. Now both clients are able to establish a P2P UDP connection and can exchange text messages directly (no text messages are sent via the server).

Task 3 - Multicast UDP Chat (5 points)

Extend the UDP version of the chat application to allow direct group chats where clients are able to join chat rooms with multiple users communicating directly over UDP multicast.

As in the previous task, users have to log in with their unique username to the central server. Then a client can request a list of available chat rooms from the server (via unicast UDP) to select and join one of these rooms (UDP multicast directly).

All clients in a chat room are using a UDP multicast domain to exchange messages, without the server being involved.

Question:

Suppose you want to extend your chat application to show the user a list of all the other participants in the chat room without involving the server. How would you implement this? At which layer of the network model ("Internet protocol stack") do you solve this problem?

¹C: <http://man7.org/linux/man-pages/man2/select.2.html>

²Python: <https://docs.python.org/3/library/select.html>