Assignment 2 Socket Programming - Implementation of an instant messaging service

In this assignment, you will implement the core functionalities of a messaging application like WhatsApp or Telegram. No need to submit any written report. Instead, you must demonstrate your programs to our course assistants and submit your source code.

Deadline: 18.02.2022

Mandatory Requirements

1) **Message delivery:** Please implement a service that allows any two clients to send messages to each other.
    - One client should be able to send messages to multiple clients and to receive messages from multiple clients. For example, if there are 3 clients, A, B and C, Client A can send a message to either B or C, and can receive messages from either B or C. Make sure the messages can be delivered to the right receivers, and correctly displayed on both the sender and receiver sides.
    - The sender will be notified if the messages have been sent successfully or not.
    - The clients can be connected to different networks. (In some cases turning off the firewalls is a good idea…).
    - Accept requests from both IPv4 and IPv6 clients.
    - It is **not** required to implement graphical user interface. You can choose to implement the service using either TCP or UDP socket.

    There are two ways to implement the service. You can choose to implement the server in either way described below.

    a) Every message is sent to the server and then forwarded to the receiver. This is recommended as it makes totally asynchronous communication (in which 2 communicating clients are never online at the same time) easy to implement and to maintain.
    b) Every message is sent directly from one client to another. In this case, the sender needs to query the network address of the receiver.

2) **Group Chat**: The client which creates a group is the owner of the group and has the rights to rename the group and to manage the member list. Any client within a group can send one message simultaneously to the whole group. The receivers within a group can see the sender and the group name of any received message. See an example of broadcast in https://www.youtube.com/watch?v=3UOyky9sEQY .

3) **Offline messages:** The sender is informed if the receiver is offline. The sender can see when a receiver last was online. The server can buffer and forward the messages to the receiver when the receiver gets connected again. The sender can see if the receiver has read his/her message.

You will get max 2 bonus points if you manage to implement file transfer. In that case, a client can send a file to another one. The receiver can decide whether it wants to receive the file or not.

ELEC-C7420 Basic Principles in Networking Spring 2022

Please demonstrate your programs to our course assistants following the instructions below.

1) Open at least 3 clients and 1 server. Briefly explain the communication mechanisms between clients and between client and server during a short code review.
2) Send messages from one client to any of the other clients. Show the sent/received messages on both senders and receivers.
3) Exchange messages between several pairs of clients simultaneously.
4) Send a message from one client to a group of clients simultaneously.
5) Send a message to a receiver which is offline. When the receiver comes online again, show the messages sent to it when it was offline.
6) Send a message to a group of clients and show which of the clients in that group have received that message.
7) Send a text file from one client to another.

Assessment Criteria (**max 20 points + 2 bonus points)**:

	Unacceptable (0)	**Marginal (1)**	**Acceptable(2)**	**Very Good(3)**	**Exceptional(4)**
Message delivery when the receivers are online(3)	Messages cannot be sent successfully.	Messages can be sent from one client to another one successfully (half-duplex). Sent/received messages are displayed on both sender and receiver sides.	Two clients can send messages to each other simultaneously (Full-duplex). Sent/received messages, sender information, and timestamps are displayed on both sender and receiver sides.	Messages can be exchanged between different clients successfully. Support multiple pairs of clients to exchange messages simultaneously. Sent/received messages, sender information, and timestamps are displayed on both sender and receiver sides.	Message exchange satisfies the previous properties and the sender can check if the receiver could have read its message. The clients can connect to different networks. The service works on both IPv4 and IPv6.
Group chat (1)	Messages cannot be sent from one client to multiple clients.	Messages can be sent from one client to multiple clients at the same time. The client IDs or IPs are indicated as command line parameters.	Support group management. Allow messages to be broadcasted within a group.	A client can attend several groups and can receive messages from different groups and individual clients at the same time.	Message exchange satisfies the previous properties and a sender can see which of the group members have read his/her message. The clients in one group can connect to different networks. The service works on both IPv4 and IPv6.
Offline messages (2)	Does not support message delivery when the receiver is offline.	A sender is informed if the receiver is offline and when the receiver was online last time.	A sender is informed if the receiver is offline and when the receiver was online last time. Messages can be buffered when the receivers are offline, and forwarded to the		

			receivers when they get connected again.		
Files(1 bonus)	Files cannot be sent.	Files can be sent and received.	Files can be sent and receiver decides if he wants to receive them.		