

# BTN415 Lab 4 – UCP/IP

---

In this lab, you will work with UDP/IP client and server applications. You will be asked to update provided source codes in order to fulfill specific requirements.

## LEARNING OUTCOMES

Upon successful completion of this lab, you will have demonstrated the ability to:

- Create a UDP client application
- Create a UDP server application
- Implement a reliable data communication using the TCP/IP standards and protocols

For this lab, you should use as a starting point the codes that were discussed during our lecture on UDP/IP. These source codes can be downloaded from our course's Github repository using these two links: [https://github.com/marceljar/BTN415\\_Labs/blob/main/lab4/udp\\_client.cpp](https://github.com/marceljar/BTN415_Labs/blob/main/lab4/udp_client.cpp) for the client source code and [https://github.com/marceljar/BTN415\\_Labs/blob/main/lab4/udp\\_server.cpp](https://github.com/marceljar/BTN415_Labs/blob/main/lab4/udp_server.cpp) for the server source code. After downloading the source codes, you should modify them in order to achieve what is asked in what follows.

### PART A – [1.5 marks]

Update the **client's** source code in order for the client to be able to keep transmitting messages to the server (and receiving feedback messages) until the client sends a message containing only: "[q]". (i.e., the sequence of characters [, q, and ]).

### PART B – [1.5 marks]

Update the **server's** source code in order for server to be able to keep receiving messages from the clients, as well as sending replies, with the message "Thanks for your message!", after each received message, until a client sends a message containing only: "[q]".

### PART C – [2.0 marks]

Update the both the **client's** source code, as well as the **server's** source code from parts A and B, so that when a client sends a message [q], the client quits and the server sends a message containing the string "Goodbye", while remaining ready to receive messages from other clients. Moreover, if the client sends a message [x], both the client as well as the server quit. In this scenario, the server sends a "Goodbye" message to the client, and prints in its own terminal the message "Terminated based on client's request".

## SUBMISSION INSTRUCTIONS

*You only need to submit two source codes, one for the client, and one for the server. Note that these instructions are in an increasing order of difficulty. Therefore, you only need to submit the **cpp** files containing the source codes that solve the most difficult part you have had success or got the closest to*

*the solution for both the client and the server. For example, if you finished part C, simply submit your source codes for the client and server for this part. If you got stuck on part B, then submit your client and server source codes for this part.*