Network programming

dr inż. Radosław Wajman

Final project for TCP and UDP sockets

Goal of the project:
Student should design and implement two algorithms. One using UDP protocol for server discovering and other for communication with the server (including connection initialization, maintaining and resuming).
Final project should consists of two applications (modules): client and server, which functionalities are listed below. Student can choose the development platform / environment.

Client module:

1. Allows to connect to the server module (to only one in the same time),
2. Sends data using TCP protocol,
3. Does not know the server IP address and the port number, therefore sends in the multicast mode (via UDP) the `DISCOVER` message on port 7,
4. After receiving the `OFFER` message retrieves the server IP address and port number and:
    a. writes to the user the IP addresses and ports numbers of all servers that answered with `OFFER`,
    b. writes to the user the message if no server was found after 10 seconds time of waiting,
    c. allows the user to choose the server and connects to it,
5. Stores the servers IP addresses and ports numbers and after next run suggests the connection with the last server,
6. In case of error or when the connection was broken the client module informs the user about it and goes back to the step 3,
7. When the connection on TCP is successfully established:
    a. asks the user about his nick prompting him the last one,
    b. sends to the server the user's nick using i.e. `NICK` command,
    c. using the `VALUE` command starts sending a random value ranged from 0 to 100 with the frequency set by the user between 0 and 10000ms.

Server module:

8. Shows own IP addresses and port numbers it has been listening on,
9. After run starts:
    a. listening for the clients on TCP,
    b. listening on UDP on port 7 for the multicast `DISCOVER` messages and answering with `OFFER` and with port number of the TCP listener,
10. Accepts the clients TCP connections,
11. After accepting the TCP client connection request:
    a. retrieves the client nick,
    b. starts receiving from the client the values with the frequency imposed by the client,
    c. writes in a real time to the user an information in a readable form i.e. *NICK: the current value xxx*,
    d. after client disconnection for 5 seconds announces: *NICK → disconnected*,

Evaluation rules:

1. Originality of the TCP and UDP client – server communication protocol concept as well as its documentation,
2. Code clarity and algorithms efficiency,
    a. CPU consuming, thread counts, algorithms complexity,
    b. hardiness to exceptional situations (connections breaking, wrong commands, etc.)
    c. threads synchronizations.