Lab session 2

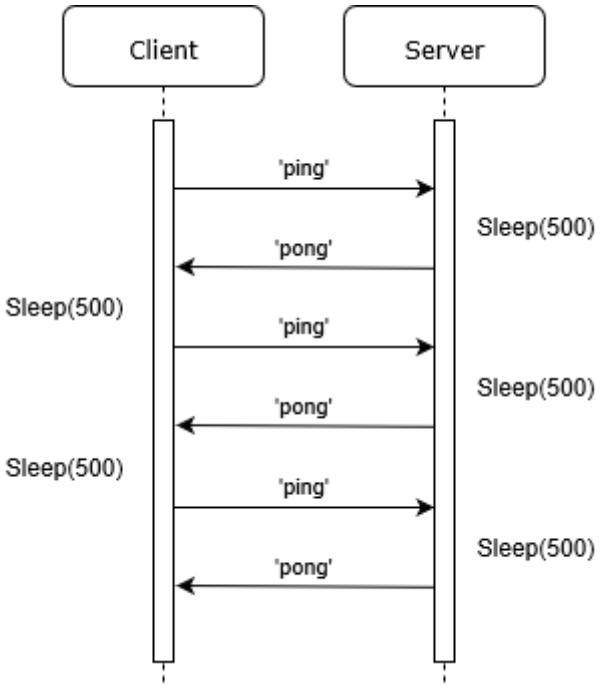
Exercises

Program an application that transmits the messages “ping” and “pong” between two hosts.

The goal of this exercise is to start learning the basics of the sockets library in Windows and get used to the sequence of calls used for both UDP sockets and TCP sockets.

Create a solution in Visual Studio (Visual C++) with two projects in it. One will be called Server, which will remain active waiting for “ping” messages, and will send “pong” as a response. The other project will be called Client and will send a “ping” message in the first place, then it will wait to receive a response “pong” from the server before sending “ping” again.

The sequence of messages between both processes will be the following:

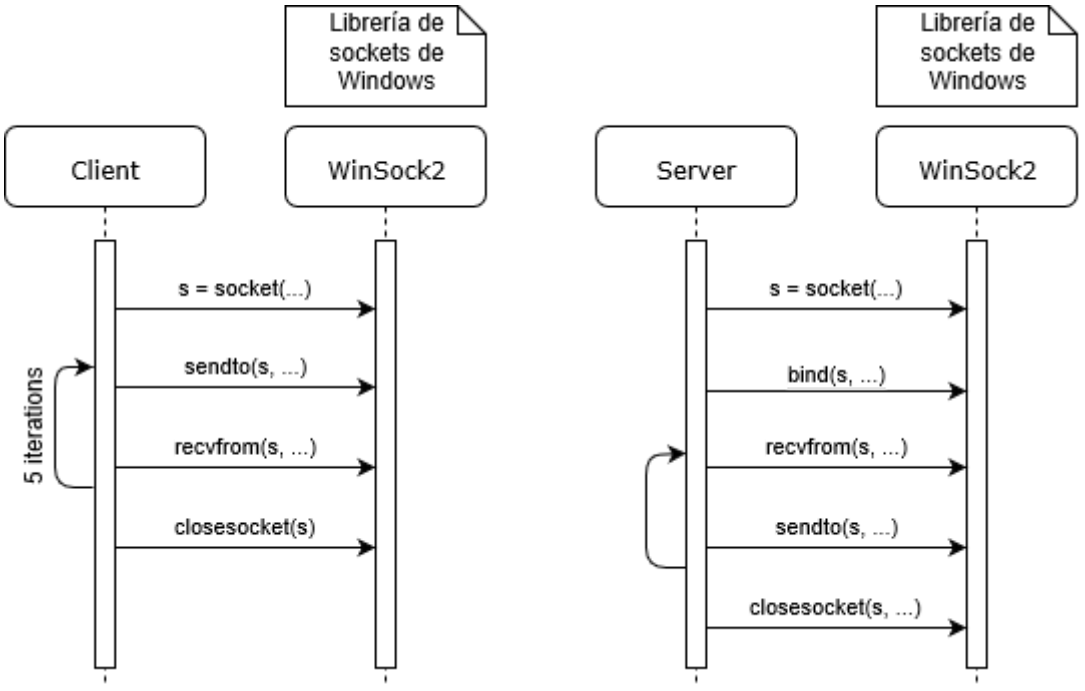


NOTE: We can work executing both processes in the same machine, referring to itself with the IP address 127.0.0.1 (localhost). However, you are free to copy your generated binaries to another machine (in which case you will need to provide the proper IP address instead of localhost) and testing the application through the local area network.

Statement

1. Implement the exercise using UDP sockets.
2. Implement the exercise using TCP sockets in two ways:
    a. The server will launch in the first place and will wait for a single client connection. After that, a client will start and will connect to the server, sending “ping” and waiting to receive “pong” 5 times. After the 5 repetitions, the client will disconnect and the server, after being notified from the client disconnection, will also finish its execution.
    b. The server will receive several client connections, but will only attend one at a time. In this case, the client code is the same, but the server will accept one connection, then it will exchange “ping”/”pong” messages, and after noticing the client disconnection, it will loop back to accept a new client that wants to initiate the “ping”/”pong” process.

Schematic overview for UDP sockets



Schematic overview for TCP sockets (case A)