Lab session 2

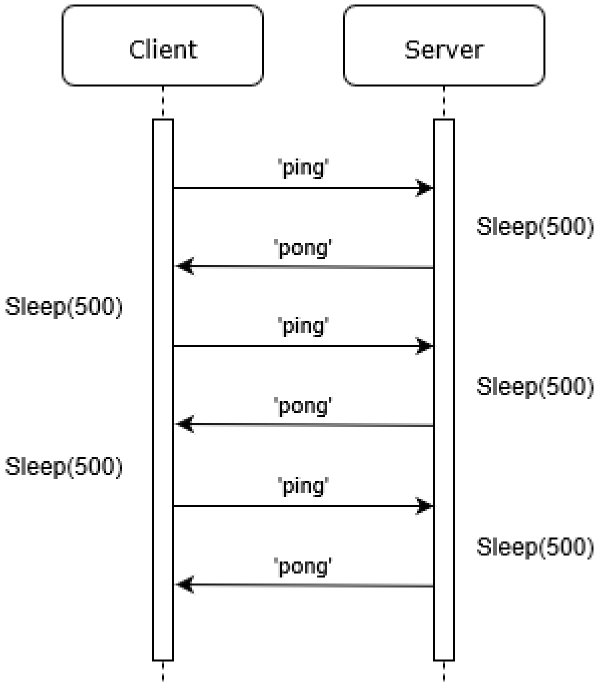
Exercises

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.

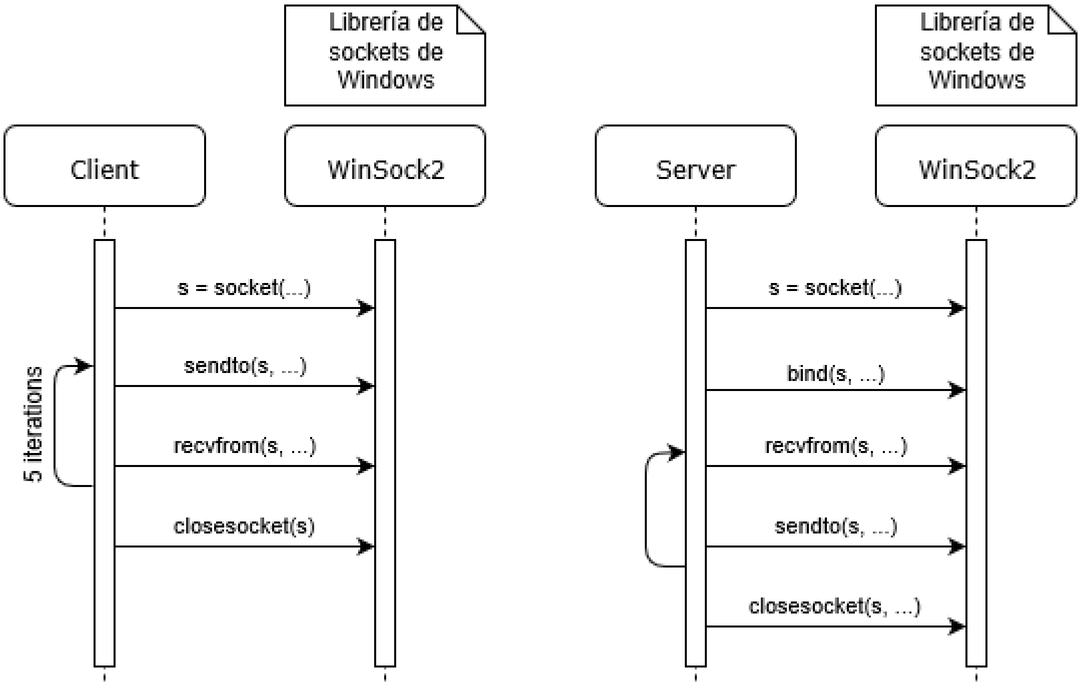
The sequence of messages between both processes will be the following:



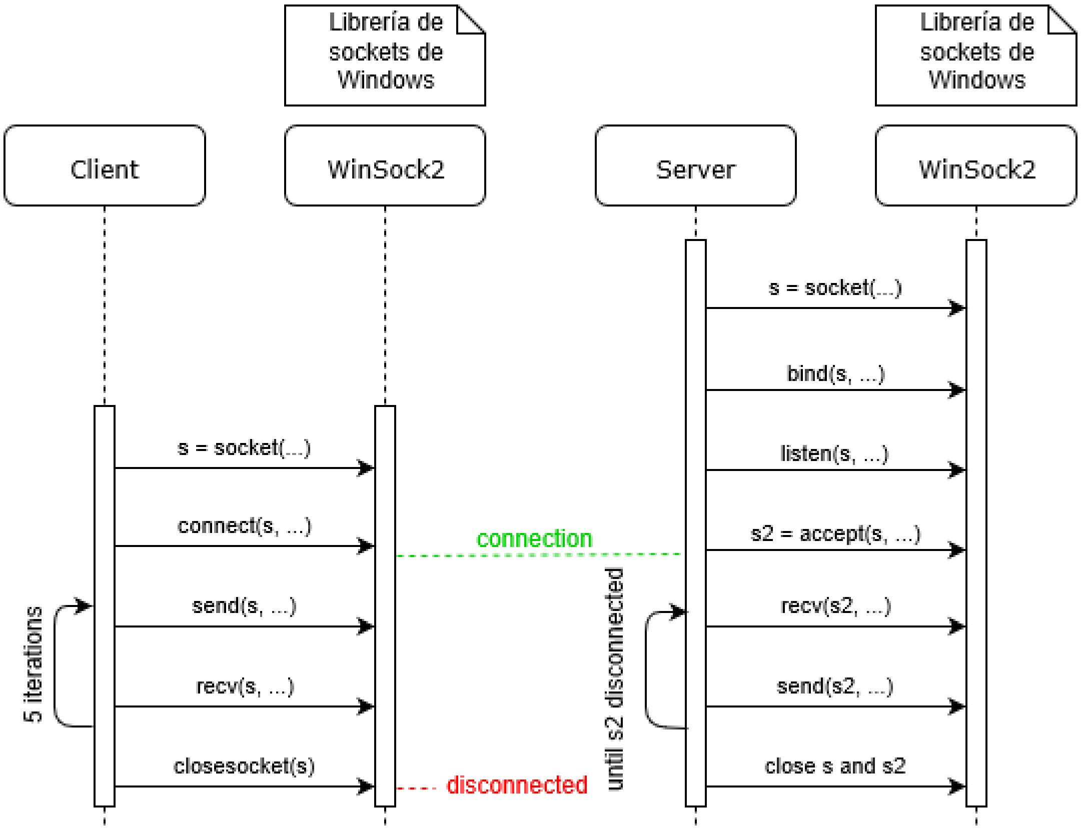
Statement

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)



Working locally

We can work executing both processes in the same machine, referring to itself with the IP address 127.0.0.1 (localhost). Of course, you will have to launch two instances of the application (the client, and the server) both in visual studio in order to make the test.

Working remotely

If you want to connect with a remote machine (a classmate’s machine), you will have to copy your generated binaries to the other machine, or share and compile the same visual studio solution, and test the application through the network.

If you need to know your public IP address (your router’s IP address publicly visible in the internet), just type “my ip address” in google.

Of course you will not be interested in sending any data packets to your classmate’s router, but his/her machine in the local network. You will have to configure your router to forward packets coming into a certain port number to the desired machine. (of your choice, but some high number such as 9999 for instance). Different routers have different interfaces, you will have to investigate on how to do that. Here’s some reference:

https://www.noip.com/support/knowledgebase/general-port-forwarding-guide/

Also, to know your IP address in the local network there are several methods, but an easy one is opening a command line console and typing “ipconfig”. Your IP address will likely be something such as 192.168.1.*???*, 172.16.1.*???*, or some similar pattern.