FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA
DEPARTAMENTO DE INFORMÁTICA

Programming with UDP sockets in Python
(Lab-01 – 14/15 September 2023)

1. Introduction

As being discussed in lectures, processes running in distinct machines can communicate with each other using **sockets** which are devices allowing the access to Internet transport protocols like TCP and UDP. In this class, only UDP sockets will be used.

Learning Materials:

- Lecture slides available in CLIP and Kurose & Ross book Ed. 8 section 2.7
- Python basic concepts can be studied in several online texts namely *Python for Everybody: Exploring Data in Python3* by Charles R. Severance et al, available (including a tar file with the code of the examples used in the book) at https://www.py4e.com/book.php
- A Python socket tutorial is http://docs.python.org/howto/sockets.html

2. The Python Program used in the lectures

Please get the source code of *UDPclient.py* and *UDPserver.py* available from CLIP. Study the source code and run it. You can invoke the Python interpreter in a shell or using a IDE like Spyder or Thonny.

3. Adding two numbers[1]

In this assignment, you'll write a client that will use sockets to communicate with a server that you will also write. Here's what your client and server should do:

Your client should first accept an integer between 1 and 100 from the keyboard, create aUDP socket and send a message to your server containing (i) a string containing your name (e.g., "Client of John Q. Smith") and (ii) the entered integer value and then wait for a sever reply.

[1] Assignment available at Kurose and Ross book site

Your server will create a string containing its name (e.g., “Server of John Q. Smith”) and then begin accepting connections from clients. On receipt of a client message, your server should

i. print (display) the client’s name (extracted from the received message) and the server’s name

ii. itself pick an integer between 1 and 100 (it’s fine for the server to use the same number all the time) and display the client’s number, its number, and the sum of those numbers

iii. send its name string and the server-chosen integer value back to the client

iv. if your server receives an integer value that is out of range, it should terminate after releasing any created sockets. You can use this to shut down your server.

Your client should read the message sent by the server and display its name, the server’s name, its integer value, and the server’s integer value, and then compute and the sum. The client then terminates after releasing any created sockets. As an aside (and as a check that you are doing things correctly, you should make sure for yourself that the values and the sums are correct!)

4. File Transfer

You should write a file transfer program where:

- The client is invoked with the name of the file to be transferred
- The client sends a datagram to the server with the file name in the server’s file system
- The server sends the file in 2048 bytes chunks.
- Client receives the blocks and writes them to its local disk. End of file id detected when the size of the chunk is less than 2048