

Assignment 2: Socket Programming

- This assignment will be introduced in a class meeting on October 25th, 2016 at 11:30 in seminar room 0.124.
- The groups will present the solution of this assignment on November 08th, 2016 at 11:30. We will meet in seminar room 0.124.
- You may work on the assignments in groups of up to three people (if possible, keep the same groups as in the previous assignments).
- All members of a group have to show up together for the grading of the assignment. For the grading, each group will need to have the source code ready for the discussion as well as the running implementation to be presented by the group. Moreover, each group member might be asked questions about the solution.
- If you have questions, send an email to Adnan Tariq to make an appointment.

Task 1 – Implementation of a Very Simple File Transfer Protocol

In this task, a Very Simple File Transfer Protocol (VSFTP) shall be implemented. The purpose of VSFTP is to transfer a file stored on a remote file server (*S*) to a client (*C*) running on a different machine via a communication network. You have to implement suitable client and server processes using sockets for the file transfer and for transferring control information such as the name of the wanted file. The protocol interactions between *C* and *S* are defined as follows (also see Figure 1):

1. *C* opens a stream (TCP) server socket on port 2000 for receiving the file from *S*.
2. *C* sends an UDP datagram to *S* at destination port 3000 with the following information:
 - File name
 - Host name of *C* (e.g. client.de)
 - Port number of server socket from step 1 (2000)
3. *S* receives the datagram from *C* and opens a stream (TCP) connection to *C* (in the example, *S* opens a connections to machine “client.de”, destination port 2000). *S* reads the wanted file from its hard disk and sends it to *C* via the opened connection.
4. *C* receives the file from its stream socket and writes it to the local hard disk.

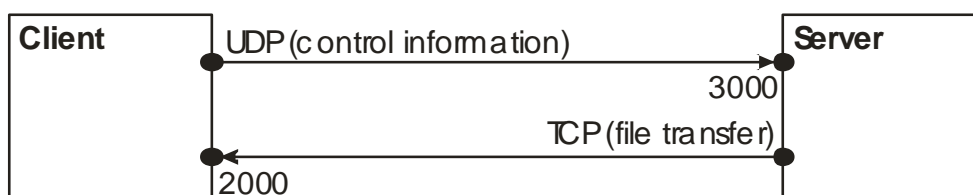


Figure 1

Implement VSFTP using the Java programming language (Implementation using C/C++ is also accepted). Recommended reading: Custom Networking Trail from Oracle's Java Tutorials, available online:

<http://download.oracle.com/javase/tutorial/networking/index.html>

Test your solution with a file stored on the server. Compare the received file to the original file stored at the server (for instance, by using the diff command) to make sure that the file transfer works correctly.

Task 2 – Discuss your Solution

Discuss your solution of Task 1 by answering the following questions:

1. Why is UDP (datagram socket) suitable for sending the control information (Step 1)? Which problems could arise when you use UDP? How could you handle these problems?
2. Why a stream socket (TCP) is better suited for the file transfer (Steps 3, 4) than a datagram socket (UDP)? Which functionality would you have to implement on the application layer (in the client and server processes) that is already provided by TCP?
3. How could the server serve multiple clients in parallel? (You don't have to implement this extension in this task. Just describe a suitable concept for serving multiple clients in parallel.)