TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr. Max Mühlhäuser

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

TK1: Distributed Systems - Programming & Algorithms

4. Programming Assignment
Submission Date: 27.01.2021

By handing in a solution you confirm that you are the exclusive author(s) of all the materials. Additional information can be found here: https://www.informatik.tu-darmstadt.de/de/sonstiges/plagiarismus/

P4. Time synchronization using NTP (20P)

The TK airport maintains hundreds of clocks to ensure on-time departures for aircrafts, as well for informing the passengers when it is time to proceed to the TSA and to the gate. At the moment, all these clocks are set to the correct time manually which causes a lot of problems. The solution to this issue is to automatically synchronize clocks of all the different devices using NTP.

Develop a Java-based client/server console application that calculates the time difference between the system clocks of both client and server. Use the NTP time synchronization protocol (cf. lecture slides) over a TCP connection. Create a *timeserver* and a *timeclient*. Use the skeleton code provided in *TK1-EX4-Programming-Template.zip*. Consider the following points:

- The client automatically starts the measurements after it got started.
- The server uses threads (1 thread per connection).
- In each measurement, calculate and display the values o_i and d_i, as shown in the lecture.
- The measurement is to be carried out **ten** times. Wait 350ms between two measurements.
- Finally, the client prints the selected time difference and the corresponding accuracy of the approximation.
- Assume an artificial offset of 1100ms for the server. In addition, implement a random delay between 10ms and 100ms on server and client side to simulate the communication more realistically (this is the case, when both client and server are started on the same computer).
- You do not need to provide a graphical user interface. Command line output is sufficient.
- Please generate a working Gradle project, which contains two run-tasks (server start and client start).