

## Assignment 2: Simulation of a Simplified TCP

*Due date: Friday, May 8, 2020 at 22:00*

*This is an individual assignment. You may discuss it with others, but your code and documentation must be written on your own.*

Write a simulation of the TCP transport protocol that implements multiplexing/de-multiplexing, and a basic reliable stream. In essence, you need only implement a *simplex* part of a socket without congestion control.

Use the `basic_tcp_sim.tar.gz` simulation package available on-line<sup>1</sup> as a specification for what you have to implement.

This package is based on the simulation framework and examples seen in class. The package contains a basic simulation framework and, on top of that, a simulation of a simple constant-rate, fixed-latency communication link with a fixed transmission queue. The package also contains a very basic implementation of a TCP transport, specifically in the classes/files *TCPPacket*, *TCPReceiver*, *TCPReceiverStack*, and *TCPSender*. That implementation is then used by a simulation scenario coded in the main class *TCPSimulation*.

The basic TCP already implements multiplexing/de-multiplexing. You must also implement the basic stream reliability feature of TCP, including the use and management of buffers, acknowledgments, and retransmissions.

As a simulation configuration, use the same example simulation coded in *TCPSimulation* that runs two TCP flows over the same basic link.

### Submission Instructions

You may write your solution in C, C++, Java, or Python. However, you are strongly encouraged to start from and then extend the Java implementation provided in the example and specification package.

Package all the source files plus a README file in a single zip or tar archive. Make sure that you include all the necessary components to build and run your solution on a standard installation of a C, C++, Java, or Python environment. In particular, make sure your solution works with the most basic command-line tool, outside of any integrated development environment.

Add comments to your code to explain sections of the code that might not be clear. Use the README file to add general comments to properly acknowledge any and all external sources of information you may have used, including code, suggestions, and comments from other students. If your implementation has limitations and errors you are aware of (and were unable to fix), then list those as well in the README file.

Submit your solution package through the iCorsi system.

---

<sup>1</sup>[https://www.inf.usi.ch/carzaniga/edu/adv-ntw/basic\\_tcp\\_sim.tar.gz](https://www.inf.usi.ch/carzaniga/edu/adv-ntw/basic_tcp_sim.tar.gz)