Project

Network Programming, ID1212

1 Goal

- To pass, you can apply what you have learned from homeworks, without detailed instructions on how to proceed.
- To get a higher score, you can also extend that knowledge and apply it on a more complex problem.

2 Grading

Your project must be accepted, but that alone does not give any points for higher grades. If you pass, you get points for the following qualities of your work. The points can be combined in any way. **Note that, just as was the case for the homeworks, your Canvas score will be one point higher than the achieved number of points for higher grades**. Also note that the exam questions are presented at the written exam page on the course web site. Please read those questions before starting the project work, they help to understand what is needed to pass the project and to get the higher grade points explained below.

1 point for higher grades is given if your project has been accepted before or on due date.

1 point for higher grades is given if your project has complete error handling of communication related errors. You can get this point only if your work is accepted before or on due date.

2 points for higher grades are given if your project includes a communication paradigm not covered in any homework. This could be for example WebSockets, a message queue, or RESTful web services. To get these points, the paradigm must be used throughout the project, it is not sufficient to provide a minimal working example. Also, you do not get these points just for using another programming language than Java (but you are allowed to use another programming language if you wish). You can get this point only if your work is accepted before or on due date.

1 point for higher grades is given if your project solves a difficulty not encountered in any homework. This could be for example one of the difficulties mentioned in the

list below. You must solve the difficulty completely, it is not sufficient to provide a minimal working example. You can get this point only if your work is accepted before or on due date.

- Notably more complex communication, e.g., a distributed algorithm like the chord protocol.
- Unit testing classes involved in the communication.
- Performance evaluation, for example analyze how the time required to perform a unit of work changes when the number of nodes change.

3 Auto-Generated Code and Copying

You must be able to explain and motivate every single part of your code. You are *not* allowed to copy entire files or classes from the example programs on the course web, even if you understand it and/or change it. However, you are allowed to write code which is very similar to the example programs on the course web. You are also allowed to use GUI builders and other tools that generate code.

4 Task

You are free to define your own task. The minimum requirement is that it shall not be smaller than the homeworks. For example, if you are interested only in passing and not in getting a high score, you can choose the task from homework one that you did not report for that homework. A more complex task gives a higher score as described above, in the section Grading. Just as in the homeworks, your solution must have an acceptable layered architecture and be well designed. This means it must follow the guidelines of the lecture on architecture, and of the programming examples on the course web. Your are, however, not required to use exactly the same layers as in those examples. You can, but are not required to, hand in a project proposal in Canvas if you want more detailed information about which score you would get for a particular program.