Peer-to-Peer-Systems and Security

Chair of Network Architectures and Services TUM

SoSe 2016

Supervisor:
Sree Harsha Totakura
Lukas Schwaighofer
Prof. Dr.-Ing Georg Carle

Students:
Hannes Dorfmann
Dorel Coman

Initial approach report

DE-RO Team working on **RPS** module

General Information

We named our Team DE-RO and we will work on Random Peer Sampling (RPS) module. We are going to use Java because it's inter-platform compatible and runs on almost every operating system. Furthermore, we already have experience in working with Java. Moreover, the java community offers a lot of open source libraries with appropriate licenses for the related field.

As build system we are going to use Gradle, which provides a flexible and highly configurable API for our builds. Also, plugins like Findbugs[1] (static code analysis) and Jacoco[2] (test code coverage) can easily be integrated with gradle along with traditional unit test to ensure code quality.

We really like the idea of reactive programming and observable streams and we believe that such a programming paradigm can help to write parallel running event based code that scales. Hence, we might gonna use RxJava[3] and RxNetty[4]. If cryptologic algorithms are needed we will may use Bouncy Castle[5] and apache commons crypto module[6]. Regarding testing, we are going to use junit and Mockito[7] (mocking library).

We want to publish our work under MIT License[8] because we believe that open source software should be usable for everyone and everywhere and in contrast to Apache2 and others we don't want to worry others about trademarks and patents.

For this project we try to share the workload equally and we will try to respect the interests and strengths of the other. We plan to follow agile development principles and divide workload into sprints and tasks. We may use external tools (like Trello[9]) for task management and issue tracking.

Experience

Dorel: I have already worked with C on a project regarding making UDP reliable using the Selective Repeat algorithm. I also have done many projects in Java so I am quite experienced with this language. I also had the chance in working on projects on asynchronous tasks management which seems very important for this module. Last I have allready done projects in team so I have some experience on how to deal with other teammates.

---

[1] http://findbugs.sourceforge.net/
[2] http://eclemma.org/jacoco/
[3] http://reactivex.io/
[4] https://github.com/ReactiveX/RxNetty
[5] https://www.bouncycastle.org/
[6] https://commons.apache.org/
[7] http://mockito.org/
[8] https://opensource.org/licenses/MIT
[9] https://trello.com/

Hannes: I have experience in implementing network communication and multithreading / synchronization with C and Java.

Issues and complains

We don’t have issues or compliance but there are few things which are not completely clear to us, like the problem of the messages between the RPSs or the other modules. Do can we create custom messages and use Gossip to spread them to other RPS? From the document it seems that yes we can choose to make a communication between RPS directly or use the Gossip instead. But it’s not really clear if we can create custom messages, we guess that’s why there were allocated different message ID for different modules.