IN2194
Peer-to-Peer Systems and Security
Summer Semester 2023

Initial Report

Team Gossiphers (Gossip-1):
Yannis Matezki and Dominik Ritzenhoff

Date Submitted: May. 23, 2023

Module and Implementation

Module: Gossip, using the Brahms Algorithm for resilience against malicious behavior

- Push flooding is prevented by imposing a computational puzzle
  and dropping messages over a certain limit
- Push and pulls both contribute to a nodes view in a controlled, balanced way
- Some of the nodes view contains previously observed other nodes (history samples)
- Occasionally verify node status using a probe request

(Bortnikov et al., 2008, Brahms: Byzantine Resilient Random Membership Sampling, PODC '08)

Chosen Programming Language and Environment

Programming Language: Go

- Standard library with support for networking, cryptography, message encoding and decoding.
  No additional libraries required as the standard library is sufficient.
- Platform agnostic.
- Go is opinionated and enforces one style, which reduces friction when developers are working together.
- Very convenient Toolchain containing all necessary tools for formatting, compilation and testing.
- Go is a compiled language and known to have an acceptable performance for real time networking tasks.

Operating System: Linux-based

- Our final objective is to run the program in a Docker container, but the code should be platform agnostic.

Development Environment:

- Go Toolchain.
- GitLab CI/CD for automated testing and building of the finished Docker container.
- VSCode (Dominik) and GoLand (Yannis).

Code Quality and Testing

Intended Measures to Guarantee Quality of Software

- Integration tests with the given test suite in the CI/CD on Gitlab.
- Above 90% unit test coverage with personally written tests.
- Peer review code before merging into the Main branch.
- Go enforces a standardized style.

Miscellaneous

License: MIT

- Very permissive, high compatibility with other licenses
- Simple, well-known and understood

Previous programming experience of team members

- Dominik → Year of experience with Go with side projects.
- Yannis → Year of experience with Go with side projects.

Planned Workload Distribution

- Reports are worked on together, protocol and important implementation details are also discussed
- Implementation, testing and documentation is split into small tasks and ideally each team member will contribute roughly half of the finished code and documentation