LAB EXERCISES

Distributed Algorithms (IN4150)

Distributed Systems Group (DS)
Department Software Technology
Faculty EEMCS

Assignment 2A

Detecting Global States

Ed. 2022-2023

Assignment

Implement Chandy and Lamport's algorithm for detecting global states [1]. Please ensure that the solution adheres to the following requirements:

1. The algorithm is implemented using either the Java or the Python template (available on Brightspace).
2. The implementation runs across multiple Docker containers.
3. The network structure and the node behaviour are adjustable (e.g. network size and messages sent).
4. Each node incorporates random delays before sending a message in order to emulate network conditions (e.g. through traffic control).
5. Each node logs events separately (e.g. separate log files or terminals). Ensure that these logs are **human-readable**.
6. The solution contains a **minor** report (e.g. a README file) highlighting executed test cases. This report documents the steps taken to validate the workings of the algorithm.

The assignment can be split up into three parts:

Part A

Implement the functionality for sending and receiving messages, and for maintaining the local state.

Part B

Extend the functionality to record the global state. It can be assumed that a single designated process starts the algorithm.

Part C

Design a few test cases for the algorithm and document these in your report. Ensure that the correct functioning can be derived from the output of your processes.

Hints

- Simulate a complete network.
- Number the messages, that nodes send to each other, sequentially.

- Let n be the number of nodes. Then, let the state of a process consist of $2(n-1)$ integers, indicating the numbers of the last message sent to and received from every other process.
- Let the state of a channel be the sequence of the numbers of the messages sent but not yet received along the channel.

References

[1] K. M. Chandy and L. Lamport, "Distributed snapshots: Determining global states of distributed systems," *ACM Transactions on Computer Systems (TOCS)*, vol. 3, no. 1, pp. 63–75, 1985.