

Aufgabenblatt 8

(zum 16. Januar 2013)

Gruppe 08

Björn Stabel 222128

Tim Strehlow 316594

Friedrich Maiwald 350570

Aufgabe 7.1: Byzantinische Generäle

Die Anzahl der Generäle und der Verräter kann in unserem Algorithmus komplett über die Config-Datei konfiguriert werden. Für die in der Aufgabenstellung beschriebenen Setups sind die drei Config-Dateien mit angehängen, hier als Beispiel blatt8-aufgabe1b.txt:

```
[graph.random]
teachnet.util.FullyMeshedGraphGenerator
n=java.lang.Integer:7

[network]
teachnet.network.LinearDelayNetwork
delay.min=java.lang.Double:1.0
delay.max=java.lang.Double:2.0

[stack.algorithms]
algorithms.ByzantineGenerals

[stack.configs]
0.node.id=java.lang.String:$node.id
0.leadingGeneral=java.lang.Integer:0
0.disloyalGenerals=java.lang.String:3,4
0.network.size=java.lang.String:$network.size

[initis]
0,0,0.0
0,0,10.0
```

Wir benutzen den FullyMeshedGraphGenerator. Die ID des Anführers kann unter [stack.configs] über den Parameter 0.leadingGeneral gesetzt werden. Über disloyalGenerals gibt man in Kommata getrennt (ohne Leerzeichen) die Ids der illoyalen Generäle an.

Wichtig ist, dass die Id des Anführers auch initiieren muss, und das zwei mal (siehe letzte beiden Zeilen).

Bei der ersten Initiierung schickt der Knoten zu allen anderen eine InitiateMessage, welche die Id des sendenden Knoten enthält. Jeder Knoten der eine InitiateMessage erhält, speichert sich über welches Interface der Knoten mit der ID aus der Message erreichbar ist und schickt ebenso über alle Interfaces eine InitiateMessage mit seiner eigenen Id raus. Nach dem Senden der InitiateMessages setzt der Knoten das isInitiated-Flag auf true, damit die Initiierung terminiert. Nun weiß jeder Knoten zu jedem seiner Interfaces, welchen Knoten er darüber erreichen kann.

Bei der zweiten Initiierung wird der eigentliche Algorithmus gestartet. Der Anführer schickt an alle Knoten eine ByzantineMessage, die seine Id im IdPath und wenn er loyal ist den Wert true

enthält, ansonsten wird an jeden Knoten ein unterschiedlicher zufälliger boolean-Wert geschickt.

Empfängt ein Knoten eine ByzantineMessage, so wird sie in einer Liste zwischengespeichert (receivedMsgs).

Ist die Länge ihres IdPaths kleiner als die Anzahl der Runden (Anzahl an Verrätern + 1), so wird eine neue Nachricht konstruiert, die den IdPath der alten plus der des aktuellen Knoten enthält. Ihr Wert wird durchgereicht, wenn es sich um einen loyalen Knoten handelt, ansonsten wird er umgekehrt und der isCorrupt-Flag der Nachricht wird auf true gesetzt. Anschließend wird die Nachricht an alle Knoten geschickt, deren Id nicht in ihrem IdPath vorkommt.

Ist die Länge des IdPaths der empfangenen Nachricht nicht kleiner als die Anzahl der Runden, so wird gecheckt, ob der Knoten schon alle Nachrichten empfangen hat. Wenn ja, wird die Mehrheit gebildet (buildMajority()).

Die empfangenen Nachrichten werden anhand ihres IdPaths in einer Baumstruktur aufgebaut (buildParents()). Anschließend wird rekursiv die Mehrheit gebildet, aus dem Wert jedes Knoten und den direkt unter ihm liegenden Knoten.