Byzantinische Fehler

Matthias Reumann

7. April 2020

Inhaltsverzeichnis

1 Problem der byzantinischen Generäle

Mehrere Truppen mit jeweils einem General umzingeln eine feindliche Stadt und kommunizieren direkt über Boten miteinander. Jeder der Generäle beobachtet den Feind und schlussendlich müssen die Generäle eine gemeinsame Entscheidung treffen. Jedoch kann es unter den Generälen Verräter geben, deren Ziel es ist eine gemeinsame Entscheidung der loyalen Generäle zu unterbinden.

Die Generäle müssen einen Algorithmus finden der garantiert, dass

(a) Alle loyalen Generäle die gleiche Entscheidung treffen

(b) Eine geringe Anzahl an Verrätern die loyalen Generäle nicht zu einer Fehlentscheidung führt

Eine Entscheidung unter den Generälen wird gefällt indem jeder General den Feind beobachtet und seine Informationen an die anderen Generäle weitergibt. *v(i)* ist die Nachricht gesendet vom i.ten General. Jeder General erhält die Nachrichten *v(1)* bis *v(n)*, wobei n die Anzahl der Generäle darstellt. Die finale Entscheidung wird durch die absolute Mehrheit der Werte dieser Nachrichten bestimmt. Allerdings könnte es sein, dass loyale Generäle unterschiedliche *v(i)* erhalten, da ein Verräter unterschiedliche Werte zu unterschiedlichen Generälen sendet, was widerum Punkt A widerspricht.

Deshalb müssen sogennante *interactive consistency*-Bedingungen gelten. Ein führender General, auch Commander gennant, sendet einen Befehl an seine $n-1$ Leutnant Generäle so, dass:

IC1 Alle loyalen Leutnants den gleichen Befehl ausführen
IC2 Wenn der Commander loyal ist, dann befolgt jeder loyale Leutnant seinen Befehl

1.1 Beispiel mit drei Generälen

In diesem Abschnitt wird ein Bespiel beschrieben in dem drei Generäle miteinander kommunzieren. Wichtig dabei ist, dass niemand der loyalen Generäle weiß wer der Verräter ist.

In Abbildung 1 ist Leutnant 2 (L2) der Verräter. Der loyale Commander sendet den Befehl *ANGRIFF* an beide Leutnants L1 und L2. Zur Verifizierung sendet L2 L1 die Nachricht, er habe den Befehl *RÜCKZUG* erhalten. Um die IC2 zu erfüllen, muss L1 jedoch den Befehl des Commanders ausführen.

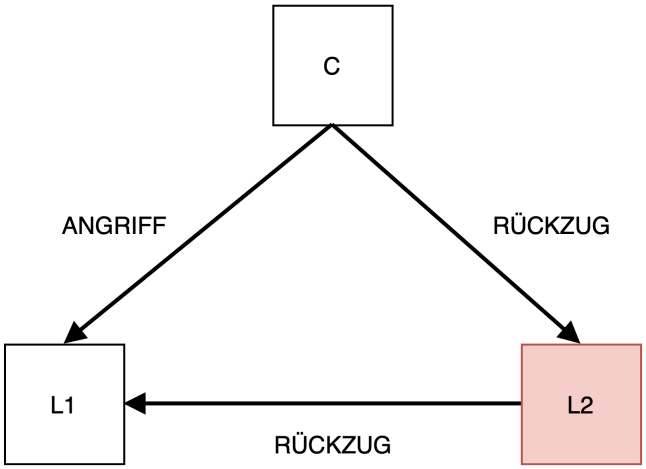

Abbildung 1: L2 ist der Verräter. L1 muss den Befehl des Commanders auführen um die IC2 einzuhalten.

Im nächsten Szenario (Abbildung 2) ist der Commander der Verräter. Dieser sendet unterschiedliche Befehle an L1 und L2. Für L1 ist es die exakt selbe Situation wie in Abbildung 1: Um die IC2 zu erfüllen muss er den Befehl *ANGRIFF* ausführen.

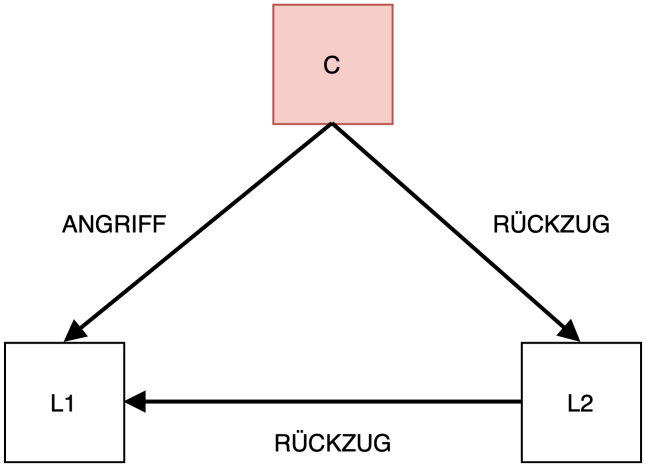

Abbildung 2: Der Commander ist der Verräter. Für L1 ist es die selbe Situtation wie in Abbildung 1.

Hierbei liegt aber das Problem. Die gleiche Argumentation muss demnach für L2 verwendet werden, der den Befehl *RÜCKZUG* ausführt. Dies widerspricht jedoch IC1. Es existiert daher keine Lösung für drei Generäle mit einem Verräter. Um einen byzantinischen Fehler zu erkennen muss daher folgende Eigenschaft

gelten: $n \geq 3m + 1$, wobei n die Anzahl der Generäle und m die Anzahl der Verräter in diesem System ist.

1.2 Lösung

Eine mögliche Lösung dieses Problems bietet der Oral-Message-Algorithmus, kurz OM-Algorithmus. Der Algorithmus funkioniert für Systeme die die Eigenschaft $n \geq 3m + 1$ erfüllen. Außerdem werden folgende Annahmen getroffen:

- Jede Nachricht wird korrekt empfangen
- Der Empfänger kann sicherstellen von wem die Nachricht versendet wurde
- Fehlende Nachrichten können erkannt werden