Aufgabe 13: Einigungsprotokolle

Literatur:

[1] Singhal, M.; Shivaratri, N. G.
Advanced Concepts in Operating Systems
McGraw-Hill, Inc: New York et al. 1994

[2] Lamport, L.; Shostak, R.; Pease, M.:
The Byzantine Generals Problem
ACM TOPLAS, Vol. 4; No. 3, July 1982, pp 382-401

Anmerkung:

In [1] fehlt im Gegensatz zu [2] der Hinweis auf die mitzuführende Pfadinformation bei dem Algorithmus OralMessage!

a) Welche grundsätzlichen Unlösbarkeitsaussagen sind bei Einigungsprotokollen zu beachten?

b) Beschreiben Sie das Systemmodell!

c) Nach welchen Leistungskriterien werden die Algorithmen verglichen?

d) Geben Sie ein Beispiel, wie beim OM-Algorithmus mit mündlichen Nachrichten und 4 Knoten mit 2 fehlerhaften Knoten keine Einigung nach den vorgegebenen Bedingungen erzielt wird!

e) Veranschaulichen Sie, wie bei **mündlichen** Nachrichten unter der Annahme m=2 Verrätern bei n=7 Knoten eine Einigung erzielt wird! Betrachten Sie dabei die Fälle, daß
   - der Initiator und ein weiterer Knoten sich bösartig verhalten
   - der Initiator sich korrekt und dafür zwei weitere Knoten byzantinisch sind.

f) Veranschaulichen Sie, wie bei **beglaubigten** Nachrichten unter der Annahme m=2 Verrätern bei n=7 Knoten eine Einigung erzielt wird!

Lösungsvorschlag Aufgabe 13: Einigungsprotokolle

a) Welche grundsätzlichen Unlösbarkeitsaussagen sind bei Einigungsprotokollen zu beachten?
   - Das Einigungsproblem kann auf der Grundlage *asynchroner Nachrichten* nicht gelöst werden
   - Das Einigungsproblem ist mit unbeglaubigten (= oral / mündlich) Nachrichten grundsätzlich nur lösbar, wenn die Gesamtzahl der Prozessoren bei *m* fehlerhaften Prozessoren mindestens *3m+1* beträgt. Bestimmte Einigungsprotokolle erfordern deutlich strengere Einschränkungen bezüglich der Zahl fehlerhafter Prozessoren.

b) Beschreiben Sie das Systemmodell!

   Es werden bei allen Untersuchungen im wesentlichen vier Annahmen gemacht:
   - Es sind insgesamt *n* Knoten beteiligt, von denen höchstens *m* fehlerhaft sind.
   - Die Knoten sind vollständig miteinander verbunden.
   - Der Absender einer Nachricht ist immer bekannt.
   - Das Nachrichtensystem ist zuverlässig

   Bei den Verfahren wird unterschieden bezüglich
   - der Fehler (crash / omission / malicious)
   - der Nachrichten (oral = mündlich / signed = beglaubigt)

   Für die weiteren Überlegungen wird angenommen:
   - synchron
   - bösartige Fehler (malicious)
   - die Einigung erfolgt auf Werte aus {0,1}

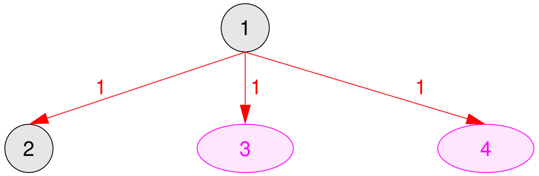
c) Nach welchen Leistungskriterien werden die Algorithmen verglichen?
   - Zeit, gemessen in Anzahl der Runden
   - Nachrichtenaufwand
   - Maximale Zahl tolerierbarer fehlerhafter Prozessoren
   - Speicheraufwand

d) Verständigungsproblem - Unlösbarkeit:
Die zu erfüllenden Bedingungen sind:

1) Alle fehlerfreien Knoten ermitteln den gleichen Wert
2) Wenn der Anführer fehlerfrei ist, ermitteln alle fehlerfreien Knoten seinen Wert.

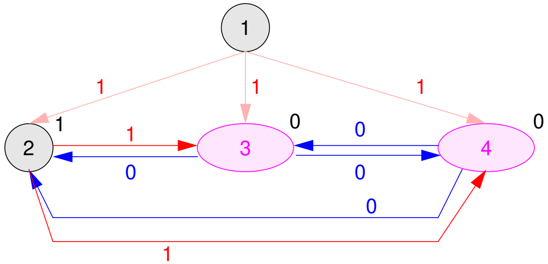
Die Einigung ist nicht gesichert, da von m = 1, also nur einem potentiellen Fehler ausgegangen wird. Folgender Verlauf des OM-Algorithmus verdeutlicht dies:

OM(1)



Nun fungiert jeder der Empfänger als neuer Anführer eines Netzes ohne Knoten 1 und startet jeweils einen OM(0)-Algorithmus:

OM(0)



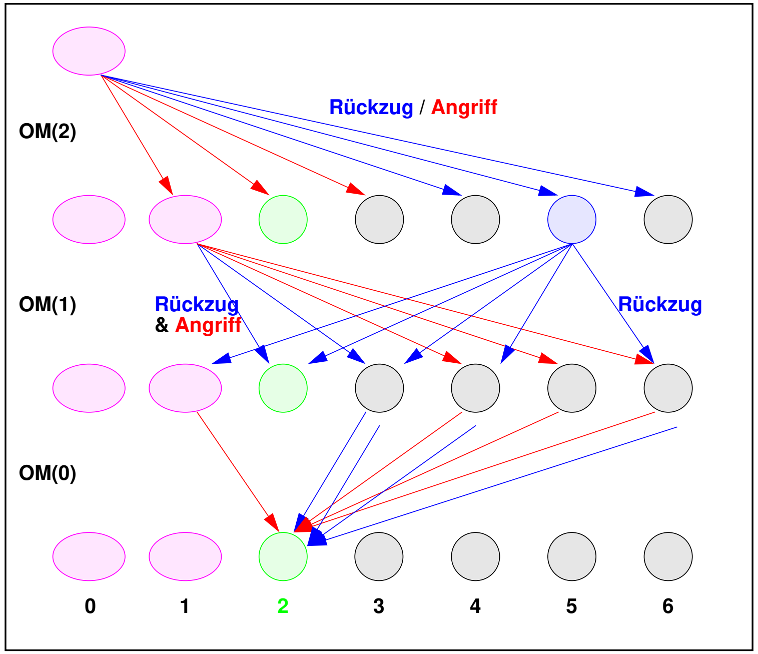
Der fehlerfreie Knoten 2 hat als Vektor v_2=(1, 0, 0) und wählt deshalb als Wert ‘0’. Da jedoch der Initiator fehlerfrei ist, müßte die Bedingung 2) gültig sein, und sich sämtliche korrekten Knoten auf 1 einigen. Das Verständigungsproblem ist also nicht gelöst.

e) Verständigungsproblem: Oral Messages

Veranschaulichen Sie, wie bei mündlichen, unbeglaubigten (verfälschbaren) Nachrichten unter der Annahme m=2 Verrätern bei n=7 Knoten eine Einigung erzielt wird!

Im ersten Beispiel ist der Anführer und ein weiterer Knoten bösartig. Das Ziel des Algorithmus ist demzufolge, daß sich die loyalen Knoten auf den gleichen Wert einigen. Der Algorithmus enthält m+1 (hier also 3) Rekursionsstufen. Zur Unterscheidung müssen die Nachrichten mit einer Pfadinformation versehen sein. Jeder Knoten fällt dann lokal in einem m-stufigen Verfahren eine Entscheidung über seinen Zustand. Es werden keine Rückantworten versandt!

Algorithmus **OralMessage** (Lamport-Shostak-Pease)



An dem Diagramm fällt zuerst auf, daß es keine senkrechten Pfeile gibt. Initiatoren nehmen an Runden nicht mehr teil, die sie selbst angestoßen haben. Das heißt, daß ein Knoten nur höchstens einmal in einem Pfad vorkommt.

Am Beispiel des **Knoten 2** ergibt sich die folgende Situation. Es sind die Nachrichten mit den dargestellten Pfadinformationen eingetroffen:

OM(0) 4-er Pfade	OM(1) 3-er Pfade	OM(1) Ergebnis	OM(2) 2-er Pfade	OM(2) Ergebnis
	0-1-2: R			
0-1-3-2:R				
0-1-4-2:A		0-1: Angriff		
0-1-5-2:A				
0-1-6-2:A				
			0-2: A	
	0-3-2: A			
0-3-1-2:R				
0-3-4-2:A		0-3: Angriff		
0-3-5-2:A				
0-3-6-2:A				
	0-4-2: R			
0-4-1-2:A				Rückzug
0-4-3-2:R		0-4: Rückzug		
0-4-5-2:R				
0-4-6-2:R				
	0-5-2: R			
0-5-1-2:A				
0-5-3-2:R		0-5: Rückzug		
0-5-4-2:R				
0-5-6-2:R				
	0-6-2: R			
0-6-1-2:A				
0-6-3-2:R		0-6: Rückzug		
0-6-4-2:R				
0-6-5-2:R				

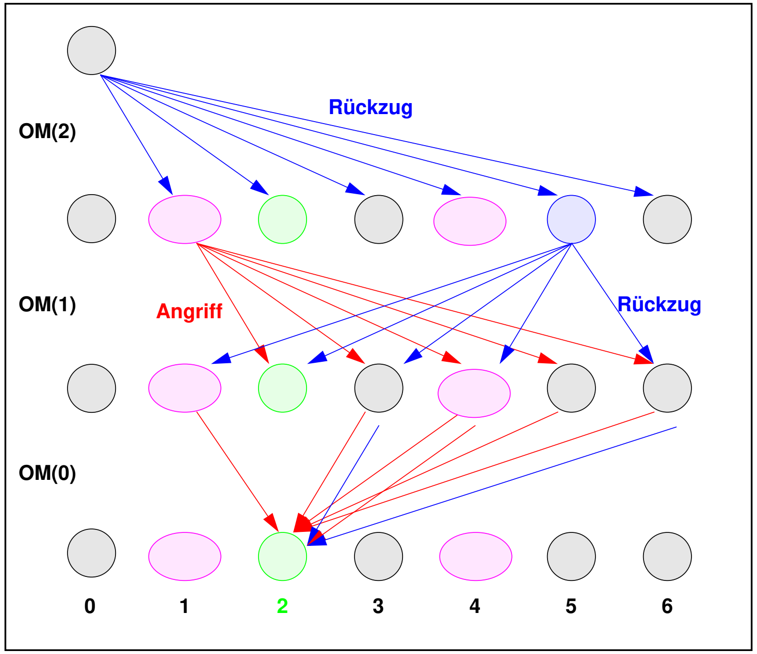
Das Diagramm zeigt die Auflösung der Rekursion und den damit zusammenhängenden Entscheidungsprozeß. Es ist in die jeweiligen Rekursionsstufen aufgeteilt. In der jeweiligen Spalte “Ergebnis” wird das Ergebnis der zugrundeliegenden majority-Funktion aufgezeigt. In der dritten Spalte steht das, was der jeweilige Knoten meint bzw. vorgibt, von dem Initiator 0 gehört zu haben.

Auch der **Knoten 4** entscheidet sich für Rückzug:

OM(0) 4-er Pfade	OM(1) 3-er Pfade	OM(1) Ergebnis	OM(2) 2-er Pfade	OM(2) Ergebnis
	0-1-4: A			
0-1-2-4:R				
0-1-3-4:R		0-1: Angriff		
0-1-5-4:A				
0-1-6-4:A				
	0-2-4: A			
0-2-1-4:R				
0-2-3-4:A		0-2: Angriff		
0-2-5-4:A				
0-2-6-4:A				
	0-3-4: A			
0-3-1-4:R				
0-3-2-4:A		0-3: Angriff		Rückzug
0-3-5-4:A				
0-3-6-4:A				
			0-4: R	
	0-5-4: R			
0-5-1-4:A				
0-5-2-4:R		0-5: Rückzug		
0-5-3-4:R				
0-5-6-4:R				
	0-6-4: R			
0-6-1-4:A				
0-6-2-4:R		0-6: Rückzug		
0-6-3-4:R				
0-6-5-4:R				

Alle weiteren loyalen Knoten entscheiden sich äquivalent. Man sieht also, daß das Verständigungsproblem durch den Algorithmus in diesem Fall gelöst wird.
Der Aufwand an Nachrichten und der benötigte Speicherplatz sind jedoch schon bei diesem kleinen Beispiel sehr groß, so daß nach besseren Algorithmen gesucht werden muß, um diese Probleme zu lösen. Hier bieten sich Verfahren mit nicht fälschbaren, beglaubigten (signed) Nachrichten an.

Im zweiten Beispiel ist der Anführer korrekt, und zwei weitere Knoten bösartig. Das Verständigungsproblem ist demzufolge gelöst, wenn sich sämtliche korrekten Knoten auf den Wert des Anführers einigen:



Der Entscheidungsprozeß aus Sicht des **Knotens 2**:

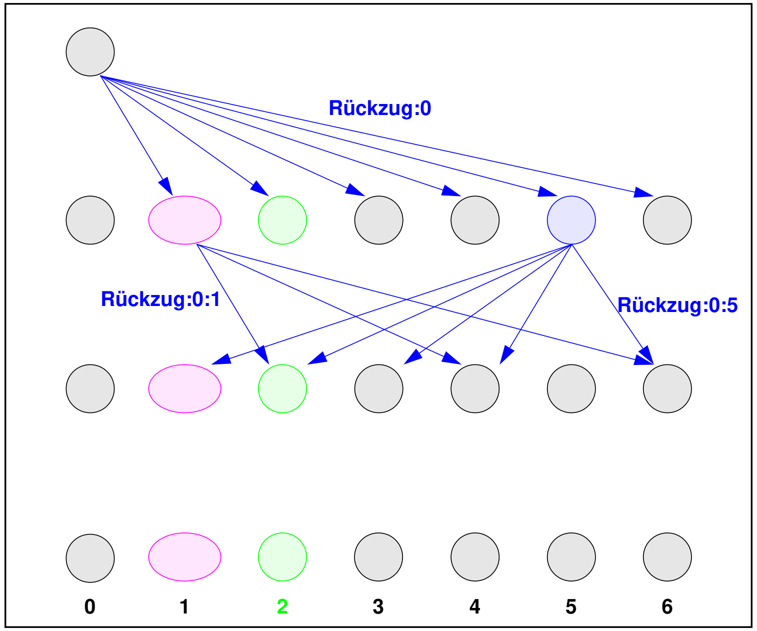
OM(0) 4-er Pfade	OM(1) 3-er Pfade	OM(1) Ergebnis	OM(2) 2-er Pfade	OM(2) Ergebnis
0-1-3-2:A 0-1-4-2:A 0-1-5-2:A 0-1-6-2:A	0-1-2: A	0-1: Angriff		
			0-2: R	
0-3-1-2:A 0-3-4-2:A 0-3-5-2:R 0-3-6-2:R	0-3-2: R	0-3: Rückzug		
0-4-1-2:A 0-4-3-2:A 0-4-5-2:A 0-4-6-2:A	0-4-2: A	0-4: Angriff		Rückzug
0-5-1-2:A 0-5-3-2:R 0-5-4-2:A 0-5-6-2:R	0-5-2: R	0-5: Rückzug		
0-6-1-2:A 0-6-3-2:R 0-6-4-2:A 0-6-5-2:R	0-6-2: R	0-6: Rückzug		

Der Knoten 2 entscheidet sich, wie auch alle weiteren korrekten Knoten, für den Wert des Anführers - das Verständigungsproblem wird also korrekt gelöst.

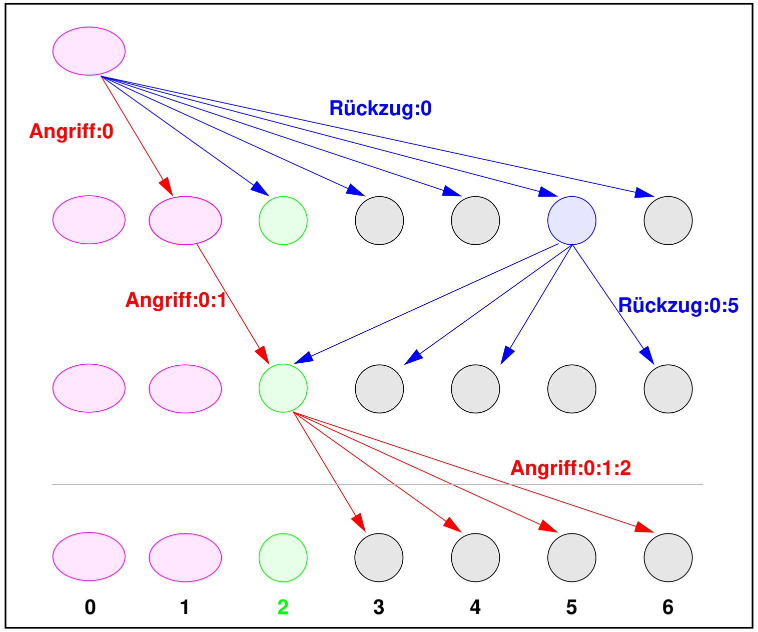
f) Verständigungsproblem: Signed Messages

Der SM-Algorithmus läuft in m+1 Runden ab. Es wird zwischen Nachrichten und Informationen/Befehlen unterschieden. Alle Information werden in Form einer Befehlsmenge V gesammelt. Es werden nur weitere Nachrichten versandt, wenn neue Befehle in die Befehlsmenge eingetragen werden. Zur Unterscheidung müssen die Nachrichten mit einer Unterschrift versehen sein. Jeder Knoten fällt dann lokal eine Entscheidung über seinen Zustand. Steht dabei Angriff und Rückzug in der Befehlsmenge, wird auf einen Default-Wert (zum Beispiel Rückzug) entschieden. Es werden keine Rückantworten versandt!

Algorithmus **SignedMessage(2)** (Lamport-Shostak-Pease)



Die dritte Welle an Nachrichten muß nicht gesendet werden, da der Initiator kein Verräter ist und alle Befehle gleich sind. Die Befehlsmenge V besteht bei jedem Knoten nur aus einem Element V= {Rückzug}. Trotzdem muß noch der Timeout einer weiteren Runde abgewartet werden, bis sich die Knoten endgültig auf den Wert entscheiden! Folgendes Beispiel soll dies verdeutlichen:



Obwohl Knoten 3-6 nach 2 Runden keine weiteren Informationen in V erhalten haben, bekommen Sie von Knoten 2 in der 3. Runde eine neue Information. Knoten 0 schickt in der ersten Runde nur an Knoten 1 einen Angriff, den Knoten 1 in der zweiten Runde nur an Knoten 2 weiterleitet. Ein bösartiger Knoten kann die Nachricht zwar nicht verfälschen, er kann aber das Senden der Nachricht unterbinden. Im worst case senden bösartige Knoten immer nur an einen bösartigen Knoten die fehlende Information, was jedoch nur m-1 mal möglich ist, da ansonsten die Pfad-Informationen nicht mehr stimmen (0:1:0:2 wäre z.B. nicht möglich). Die Konsequenz ist die Notwendigkeit von m+1 Runden, damit alle fehlerfreien Knoten die gleichen Befehlsmengen aufweisen.