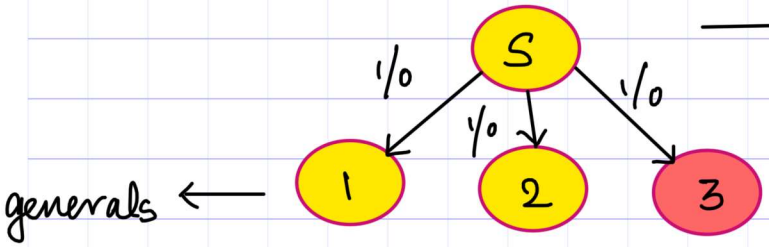
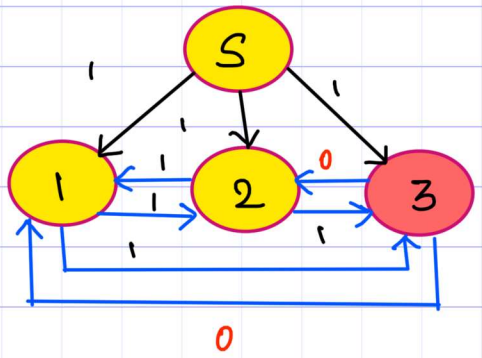
Byzantine Agreement Problem. (Consensus)



→ chief commander

1: attack
0: retreat

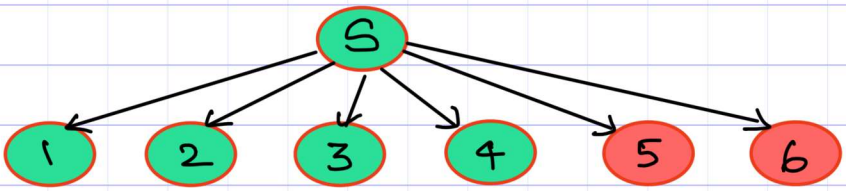
Loyal general will follow the commands.
Malicious generals will sometimes be disloyal.
(red node is malicious)
(yellow node is loyal)



(repeat the order from above among all nodes, 3 tries to be malicious & disrupt)

Identify malicious behaviour:
① value majority is taken in each node.
Nodes 1 & 2 will still execute operation 1.
Node 3 still executes operation 0.

But an issue here: there is a bound on the max. malicious nodes in the distributed environment.



For 7 nodes, max. of 2 malicious nodes are only possible.

no. of nodes : n

no. of malicious nodes : f

$$n \geq 3f + 1.$$

→ relationship for proper behaviour.

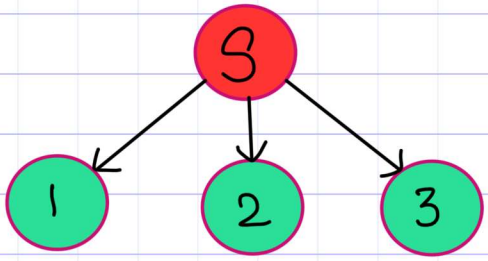
bound condition

Consensus isn't possible with total of 3 nodes.

$$\frac{n-1}{3} \geq f \Rightarrow \frac{2}{3} \geq f$$

↓ round down, 0.

no consensus possible.



if the source is malicious, all the levels below become malicious.

Designed by Leslie Lamport, Shostak & Pease.

With majority consensus, at each level with n nodes, $n(n-1)$ messages (→ overhead) are exchanged.

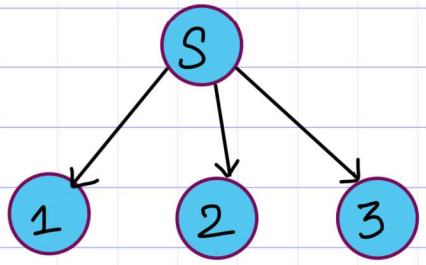
The hierarchy must resemble a tree.

Dolev & Rivest, Shamir & Adleman optimized the algorithm by reducing message overhead.

Birth of public key cryptography & digital signatures!

"Why should I exchange messages?" in the Byzantine Agreement Problem.

Nancy Lynch wrote a book on Distributed Algorithms.



Order : 1 0 1

	t_1	t_2	t_3	t_4
(1)		1	0	1
(2)	1	0	1	
(3)	1	0	1	

→ some unavoidable non-malicious issue.

↓

but this is still a problem (here majority consensus algorithm we avoid any conflict)

With asynchronous systems, even a single bit changes ⇒ the whole agreement will collapse.

Conclusion: Byzantine Agreement is unusable in Asynchronous Distributed systems.

The Byzantine Agreement Problem has far reaching practical consequences.