Distributed algorithms for fault-tolerance

Synchronous algorithms,
Byzantine agreement

Simin Nadjm-Tehrani

So far...

- Asynchronous models
- Crash or partition failures

- This time:
  - What is meant by synchrony in algorithms?
  - How to deal with byzantine failures?

Synchronous algorithms

- Proceed in rounds initiated by pulses
- Pulses can be implemented using local physical clocks, based on assumed bounded message delays

Can this help to solve difficult problems?

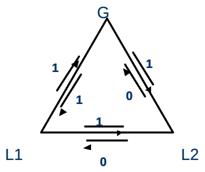
Byzantine generals

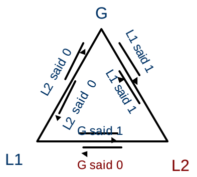
- A difficult agreement problem
- Solved in 1980 by Pease, Shostak and Lamport

- There is an upperbound t for the number of byzantine failures compared to the size of the network: $N \geq 3t+1$

Scenario 1

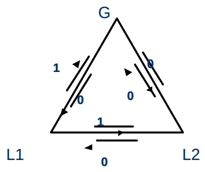
- G and L1 are correct, L2 is faulty

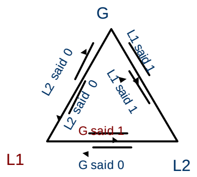




Scenario 2

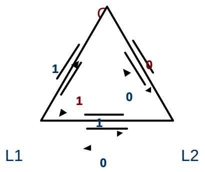
- G and L2 are correct, L1 is faulty

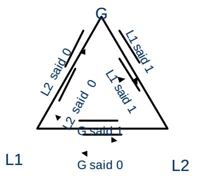




Scenario 3

- The general is faulty!





2-round algorithm does not work with t=1!

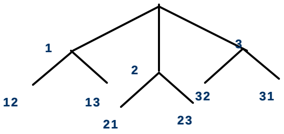
- Seen from L1, scenario 1 and 3 are identical, so if L1 decides 1 in scenario 1 it will decide 1 in scenario 3
- Similarly for L2, if it decides 0 in scenario 2 it decides 0 in scenario 3
- L1 and L2 do not agree in scenario 3 !

Idea of [PSL80] algorithm

- Algorithm proceeds in rounds
  - At round 1 each process sends its value to all others
  - At next round the recieved messages are relayed and the algorithm recursively applied with (N-1, t-1)
- Each process maintains a t+2 level tree, in which the nodes at each level k are decorated with values received in round k-1

Illustration

- V[xy]= v after round 2 means: y said that x has value v



Untrusted values are denoted by ⊥

Decision procedure

- After the t+1 rounds, the tree for each process is evaluated bottom-up
- At each level 1 ≤ k ≤ t+1 the value of each node is computed as the majority of the values of its children. If a majority doesnot exist, the value is ^

Correctness

- Agreement: if all nodes have the same initial value the computed value for each non-faulty node is the same
- Termination: based on decreasing chain of recursive calls

Effects of faults

- Transfer of incorrect own state
- Incorrect relay of another process' message

- Authentication:
  - avoids the latter
  - With t+1 rounds can tolerate t < N failures

Reading material

- Lynch, Chapters 6.3 and 6.4
- Tel, Chapter 12.1 and 15