Paxos

- or how to decide the price of olive oil

Johan Montelius
KTH

The problem

- How do we reach a consensus when:
  - nodes can crash
  - messages get lost
  - we have no failure detectors

The environment

- Nodes can crash
  - but are restarted and
  - will remember where in the protocol they were.
- Messages can take arbitrary long time to be delivered, get lost or duplicate
  - but not corrupted.

Actors

- Proposers
  - drive the execution and wants to find a consensus
  - will inform the learners if consensus is found
- Acceptors
  - vote for proposals
- Learners
  - wait for a consensus to be reached

Outline

- Proposers:
  - send request with sequence number
- Acceptors:
  - promise not to vote for a proposal with lower sequence number
- Proposer:
  - collect promises and cast a ballot with a proposal
- Acceptors
  - vote for the proposal if they have not promised not to vote in the sequence number

Proposer

- Operates in rounds, each round using a unique sequence number.
- In a round
  - send a request to all acceptors
  - collect a quorum of promises
  - keep the proposal with highest sequence number
  - request votes for the proposal
  - if a quorum vote for the proposal, we have reached consensus

Acceptor

- Keeps track of:
  - a sequence number below which it has promised not to vote
  - the accepted value with the highest sequence number that it has voted for
- If requested to promise:
  - promise and
  - return accepted value and sequence number
- If requested to vote for a proposal:
  - vote, if not promised otherwise

Messages

- Request:
  - Please do not vote in any sequence number less than #23:1
- Promise:
  - I promise not to vote in any sequence number less than #23:1, but I have voted for 8 euros in sequence number #12:4
- Ballot:
  - Please vote for 12 euros in sequence number #23:1
- Vote
  - I vote for 12 euros in sequence number #23:1

Failures

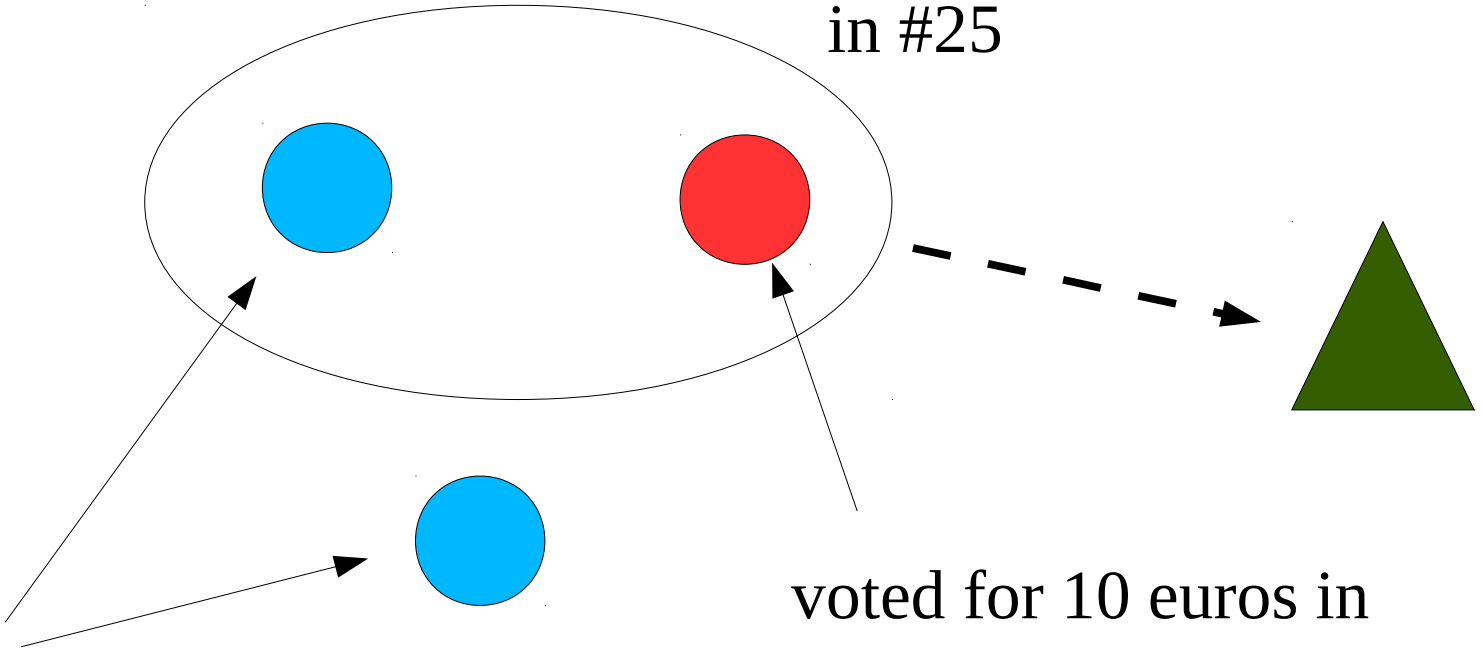
- Acceptors need never reply on anything
  - the protocol will never end in more than one value being selected by a quorum
- A proposer can abort and restart anytime
  - must select unique sequence numbers
- Progress is not guaranteed
  - two proposers can fight forever over a quorum

Why does this work

- The only thing we need to prove is that:
  - if a quorum is reached then it will not go away

Danger!!

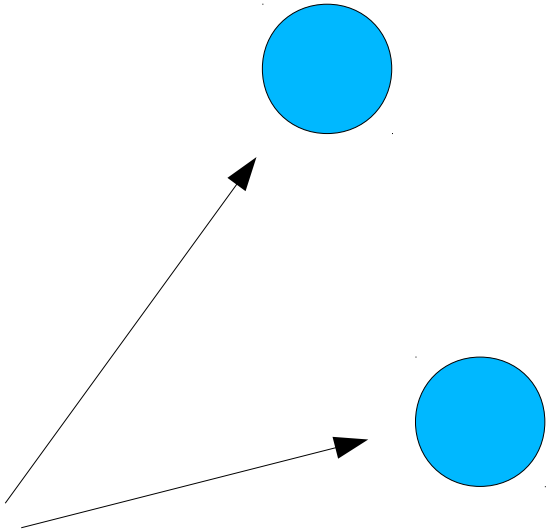
promise not to vote in any sequence number less than #26 but I have voted for 10 euros in #25



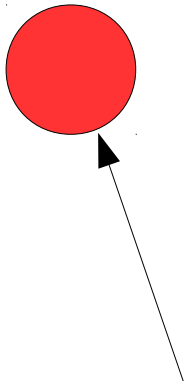
voted for 8 euros in #24

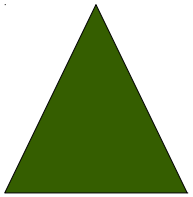
voted for 10 euros in sequence number #25

but how .....



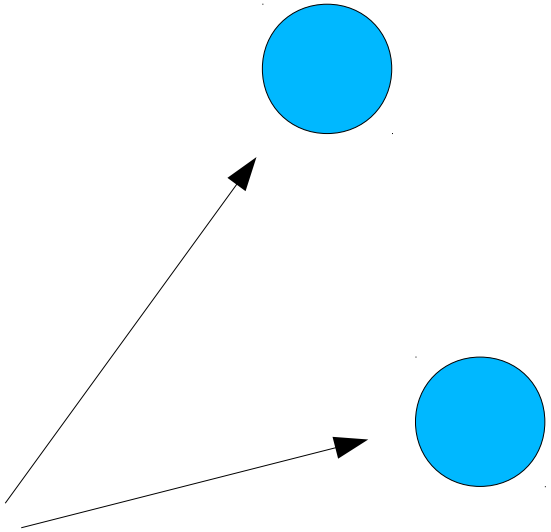
then one of these nodes promised not to vote in any sequence number less than #25

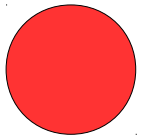


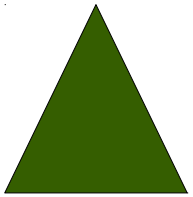


If this node voted for 10 euros in sequence number #25

impossible







but they did vote in #24 and thus must have reported “I voted for 8 euros in #24”

It works (sort of)

- If a quorum is formed
  - then proposers will be informed about the value
  - they will only cast a ballot for this value
  - and acceptors can thus only vote for this value.
- But do we know that a quorum will be formed?

Summary

- Paxos is a quorum based consensus protocol.
- Will agree on one unique value even if:
  - nodes fail and restart
  - messages are lost
- Used when we do not want to rely on eventually perfect failure detectors.