Basic Paxos

Content borrowed from Ousterhout's slides on Paxos

Basic Paxos: General Setup

Failure model: fail-stop / fail-recover

- Crash and restart, messages reordered, dropped, and duplicated, network partitions
- Can't TCP solve some issues?
- What can't Paxos tolerate?

Safety:

- Only one single value must be chosen
- A server learns that a value has been chosen only if it really has been

Availability (as long as majority of servers up and communicate w/ each other):

- Some value is eventually chosen
- Servers eventually learn about a chosen value

Basic Paxos: General Setup

Proposers:

- Active: propose values to be chosen

Acceptors:

- Passive: respond to proposals

To whom do the clients talk?

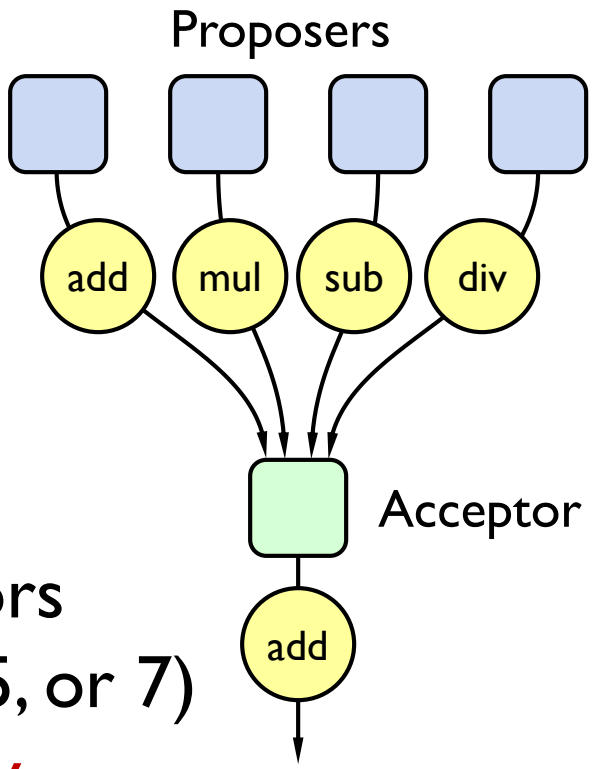
- Proposers – they propose values on behalf of the clients

Naïve Approach

Single acceptor?

Problems?

- Acceptor fails
- Can't choose or learn



Solve by having multiple acceptors

Usually a small odd number (3, 5, or 7)

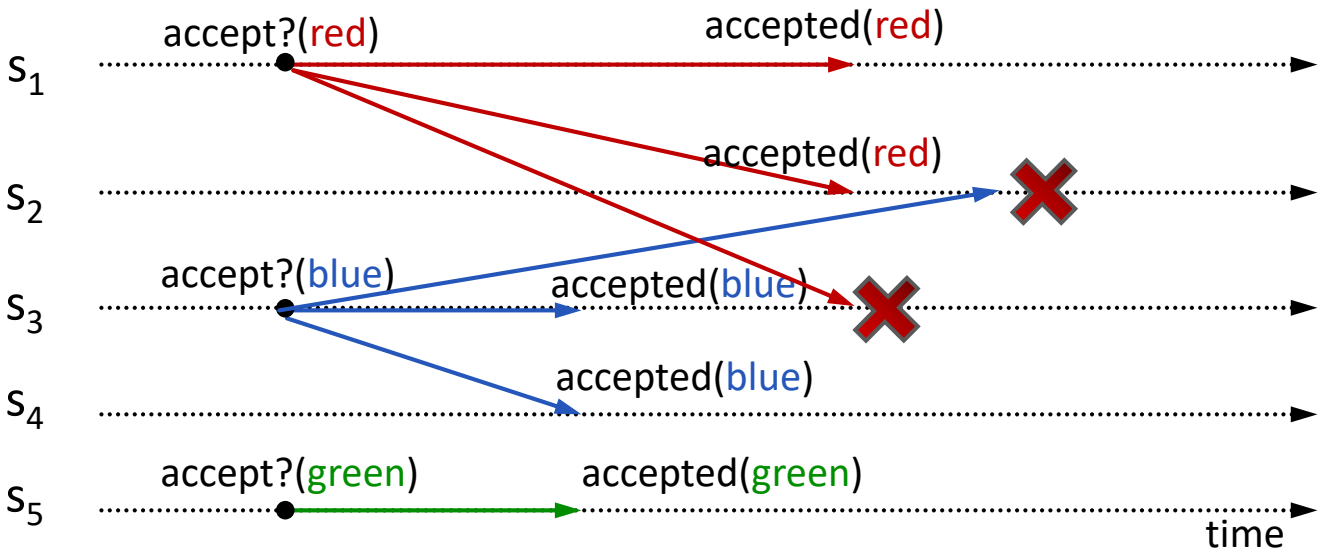
Chosen = accepted at a majority

Choose/learn as long as a majority is up

Accept only first proposal?

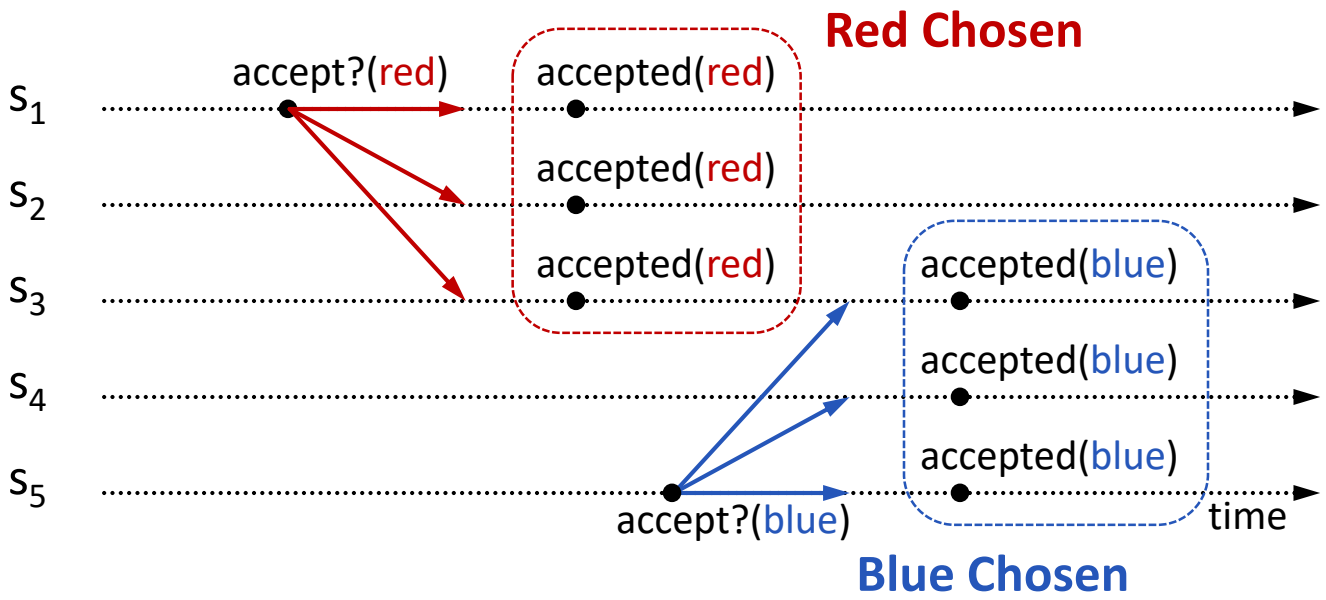
Accept only first received proposal

If simultaneous proposals, no value might be chosen



Acceptors must sometimes accept multiple (different) values

Multiple accepts safe?



Violates fundamental safety!

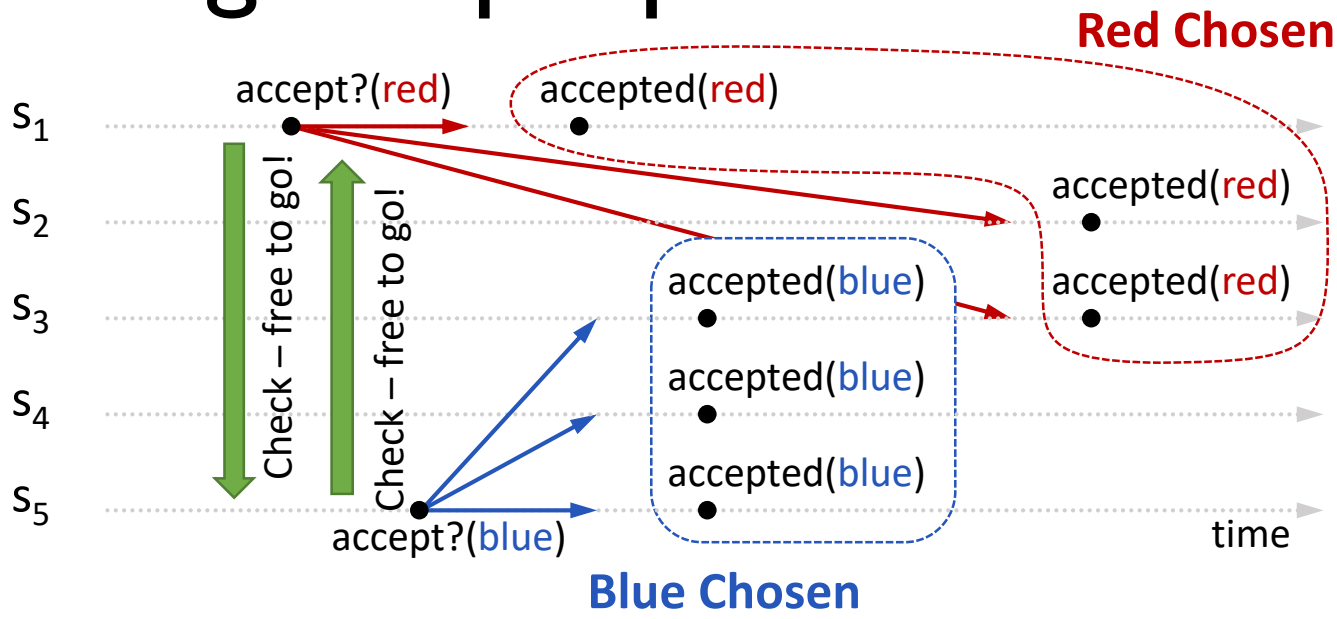
Once a value has been chosen, future proposals must propose/choose that same value

Need two phases:

first, find any (potentially) chosen

then, ask for acceptance

Rejecting old proposals



Violates fundamental safety!

s_5 need not propose red (it doesn't discover red)

s_1's proposal must be aborted (s_3 must reject it)

Must order proposals, reject old ones

Proposal numbers

A server should always use a new/unique proposal number

Larger proposal number denotes later proposal

To break ties, use server id – general form: round.serverId

Proposal numbers must be maintained on disk to survive crashes

Basic Paxos phases

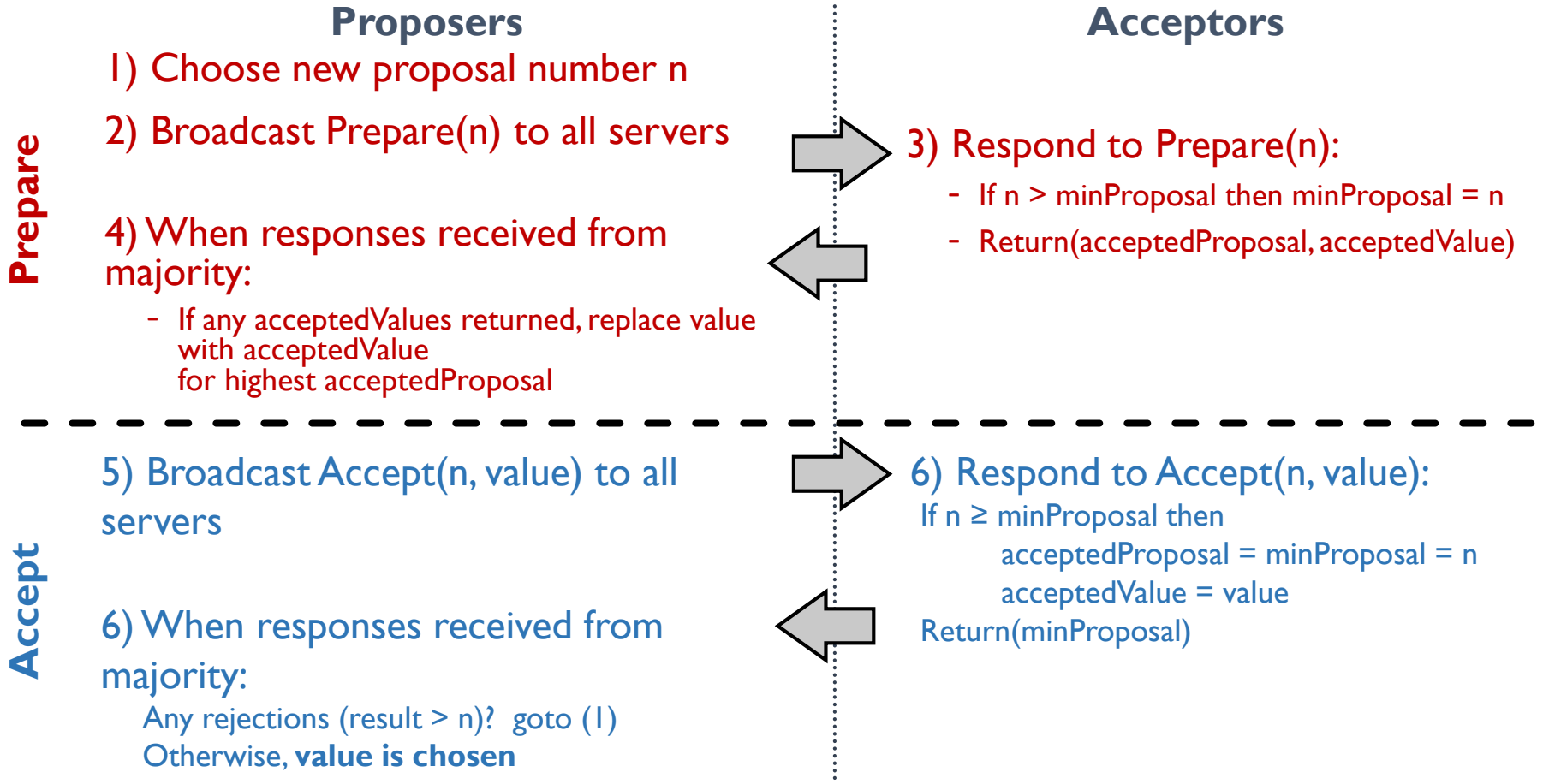
Phase 1: Prepare

- Find out about any (potentially) chosen values
- Block older proposals that have not yet completed

Phase 2: Accept

- Ask acceptors to accept a specific value

Basic Paxos - Summary



acceptors: minProposal, acceptedProposal, and acceptedValue on disk
proposers: latest proposal number on disk

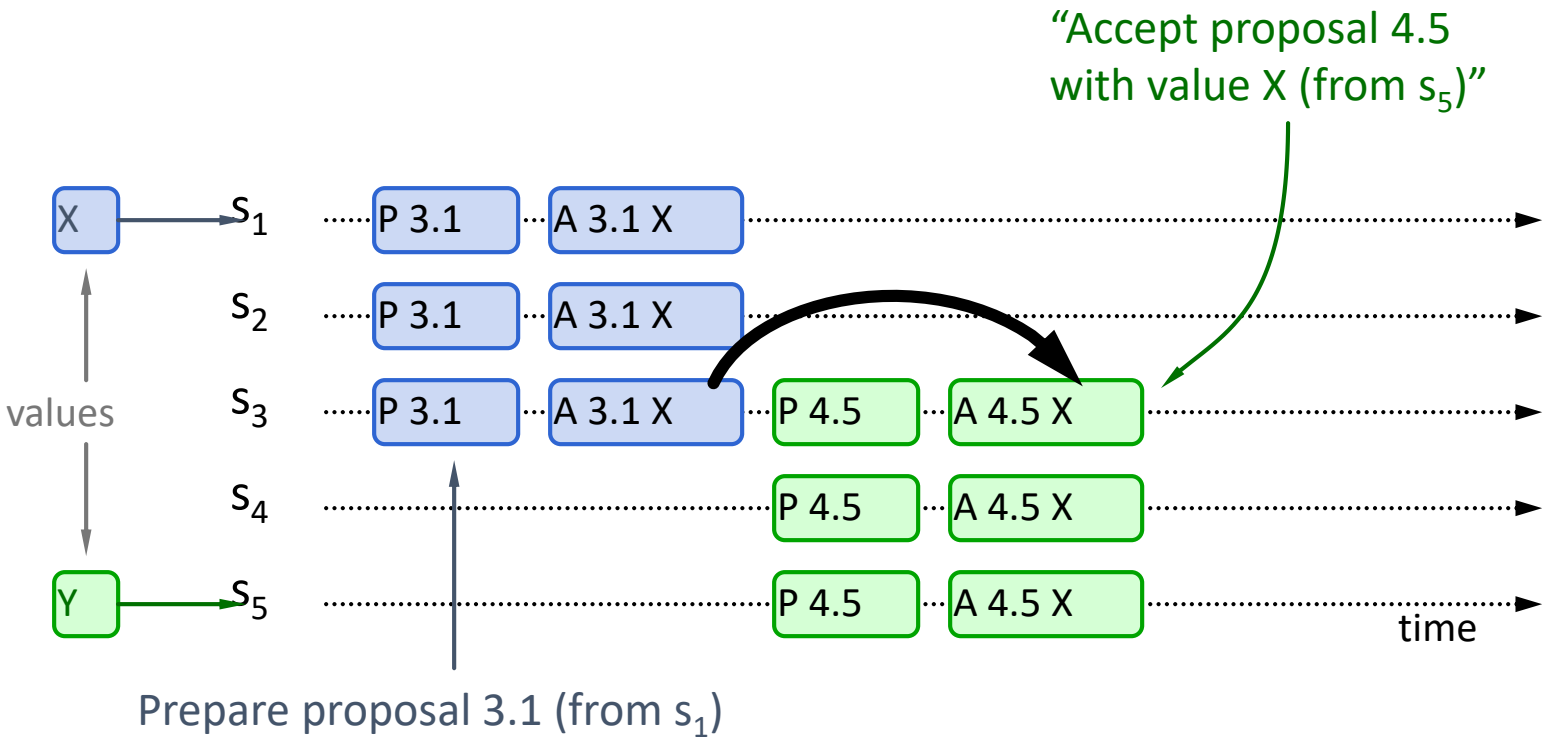
!! It is often tempting to come up with flawed optimizations !!

Example case 1

Three possibilities when later proposal prepares

Previous value already chosen:

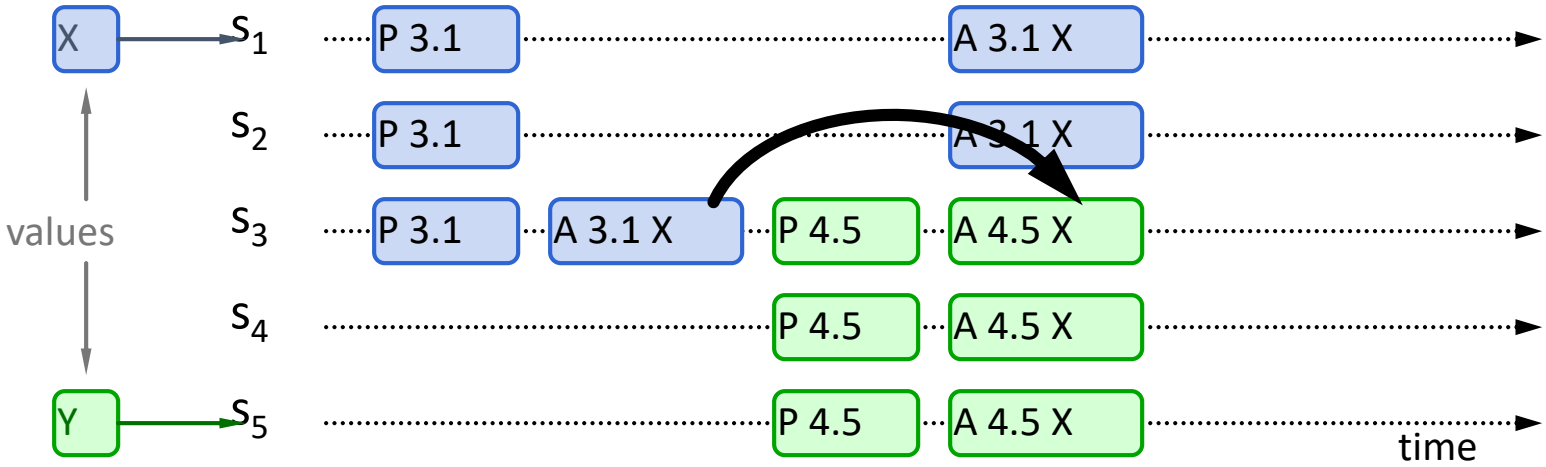
- ➡ New proposer will find it and use it



Example case 2

Previous value not chosen but later proposal sees it:

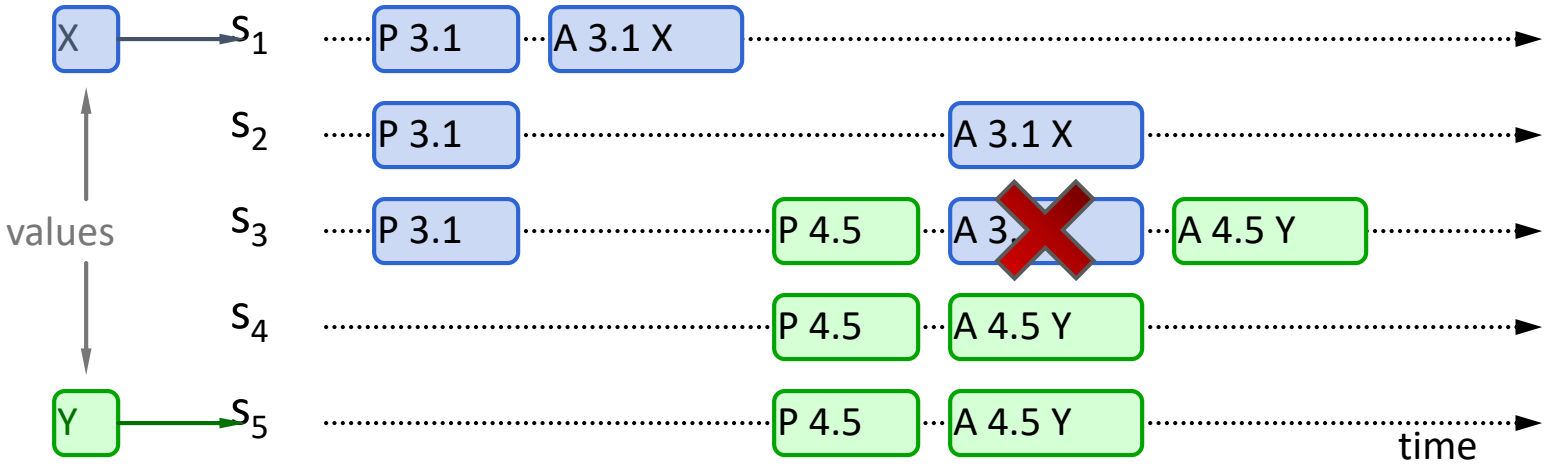
- New proposer will use existing value
- Both proposers can succeed



Example case 3

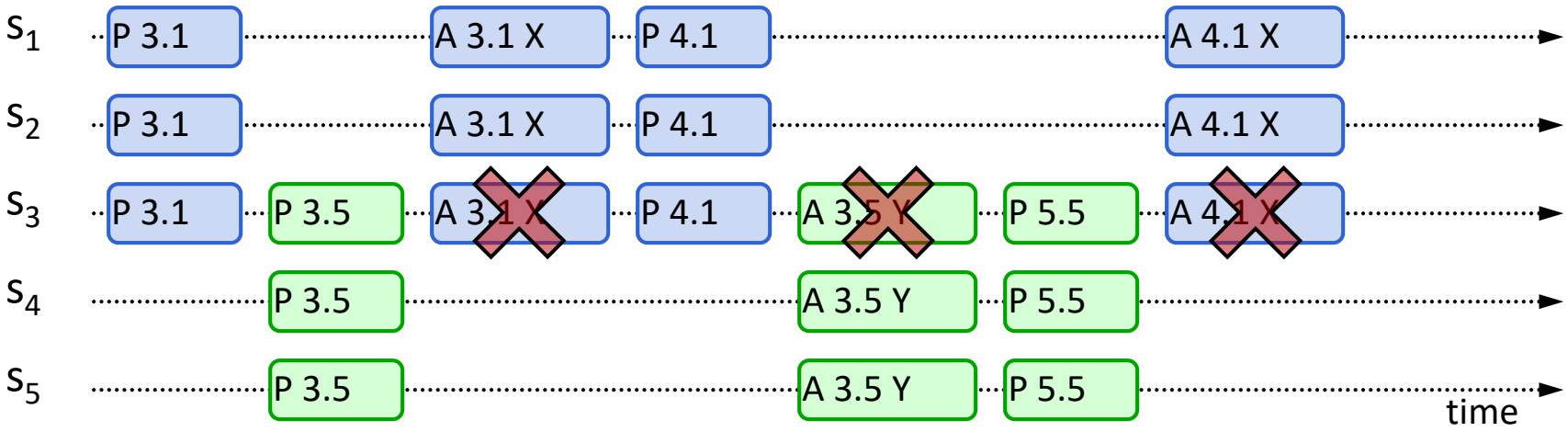
Previous value not chosen and later proposal doesn't see it:

- New proposer chooses its own value
- Older proposal blocked



Liveness

Does basic Paxos guarantee liveness always?



Solutions?

Proposers back-off with randomized delays

Only one proposer at a time (leader)

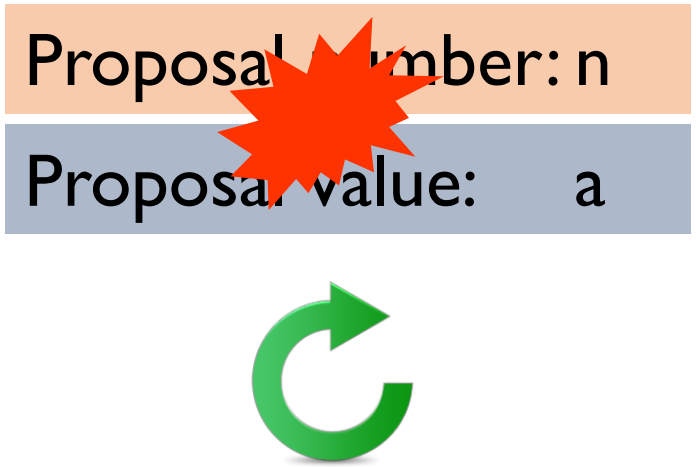
Q0

Proposal 5.1 with value X has been accepted on 3 servers (in a 5-node cluster)

After this, is it possible that any server could accept a different value Y?

Q1

Time



Could have crashed any (unknown) point in protocol: during prepare, during accept, or even after a successful accept

Re-execute from beginning with the same proposal number but different value, b

Is this safe?

Q2

Respond to Prepare(n):
If n > minProposal then
minProposal = n
Return(acceptedProposal, acceptedValue)

Respond to Accept(n, value):
If n ≥ minProposal then
acceptedProposal = n
minProposal = n
acceptedValue = value
Return(minProposal)

Respond to Prepare(n):
If n > minProposal then
minProposal = n
Return(acceptedProposal, acceptedValue)

Respond to Accept(n, value):
If n ≥ minProposal then
acceptedProposal = n
acceptedValue = value
Return(minProposal)

Is this safe?