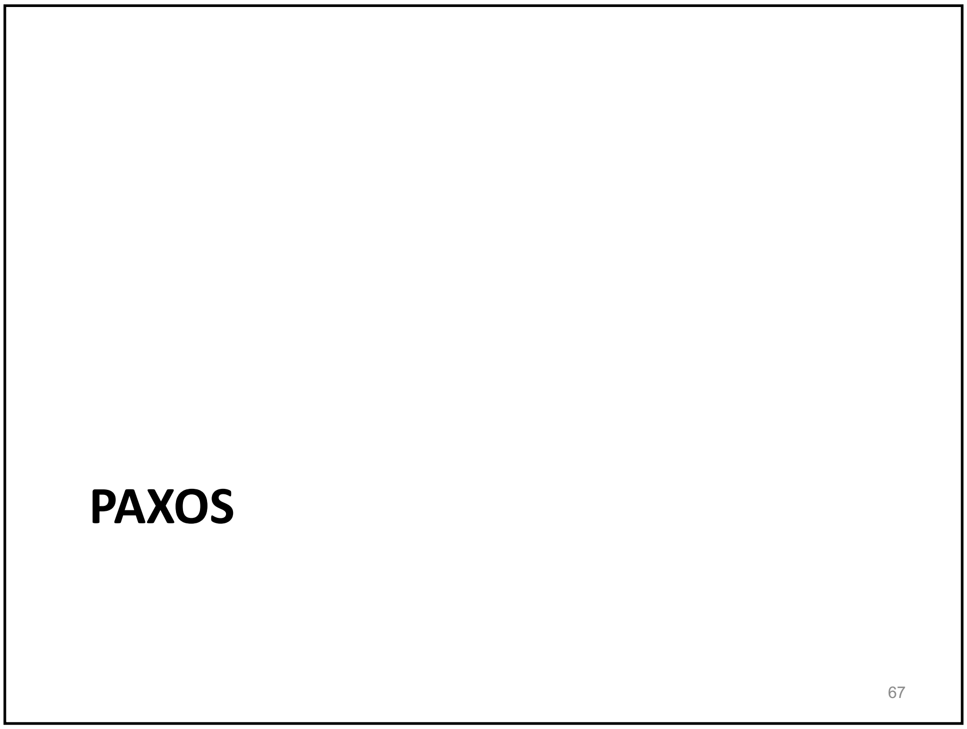
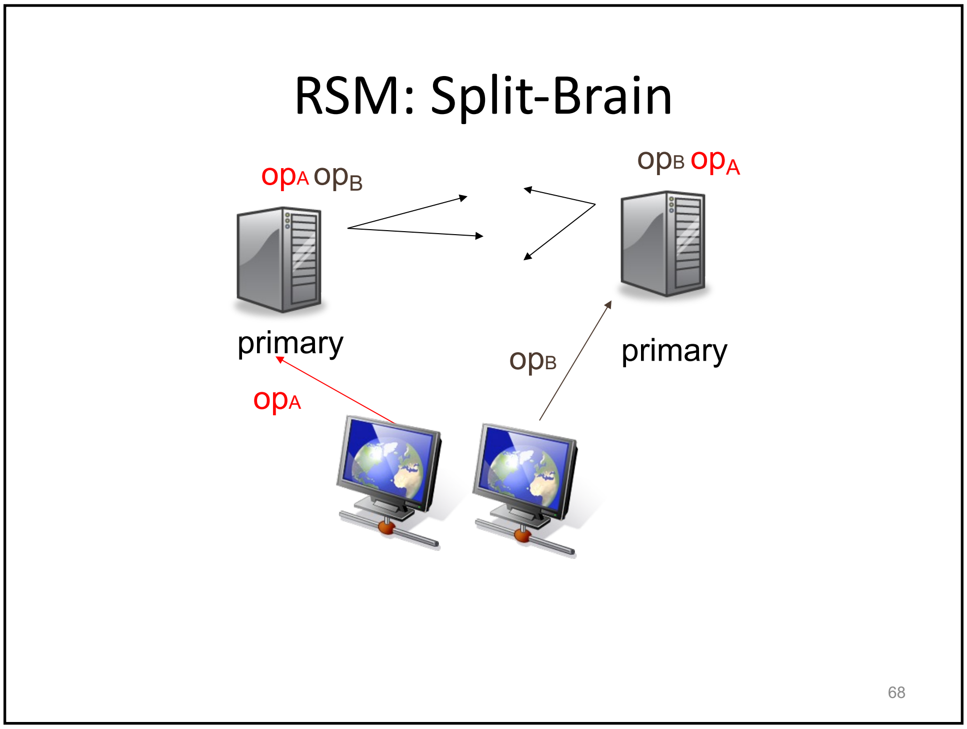


PAXOS



RSM: Split-Brain

op_A op_B

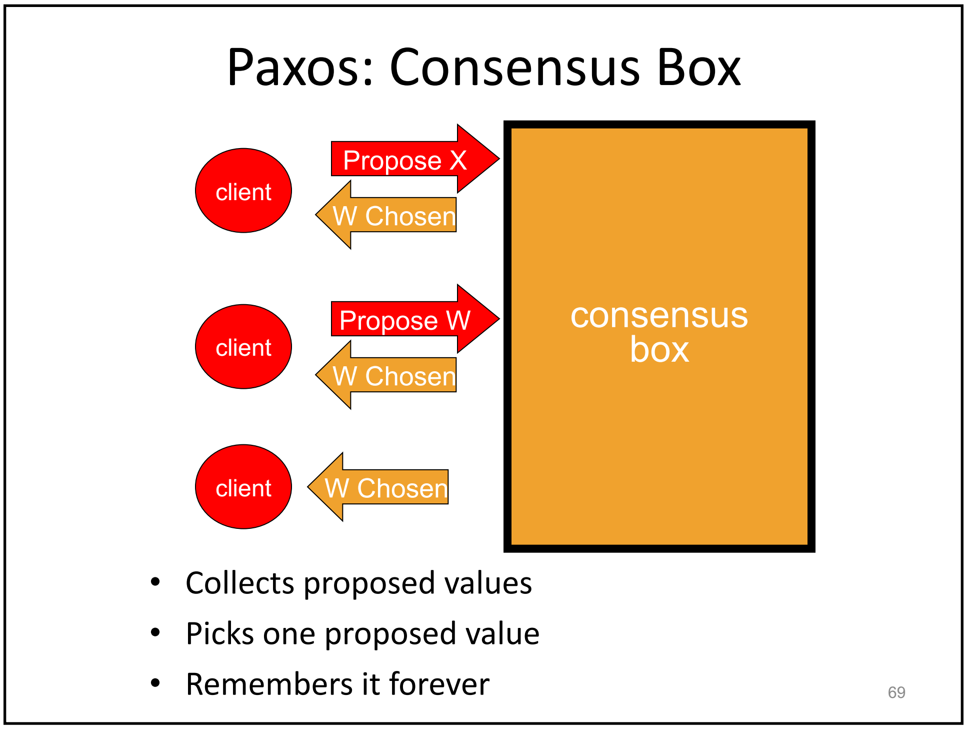
op_B op_A

primary

op_B

primary

op_A

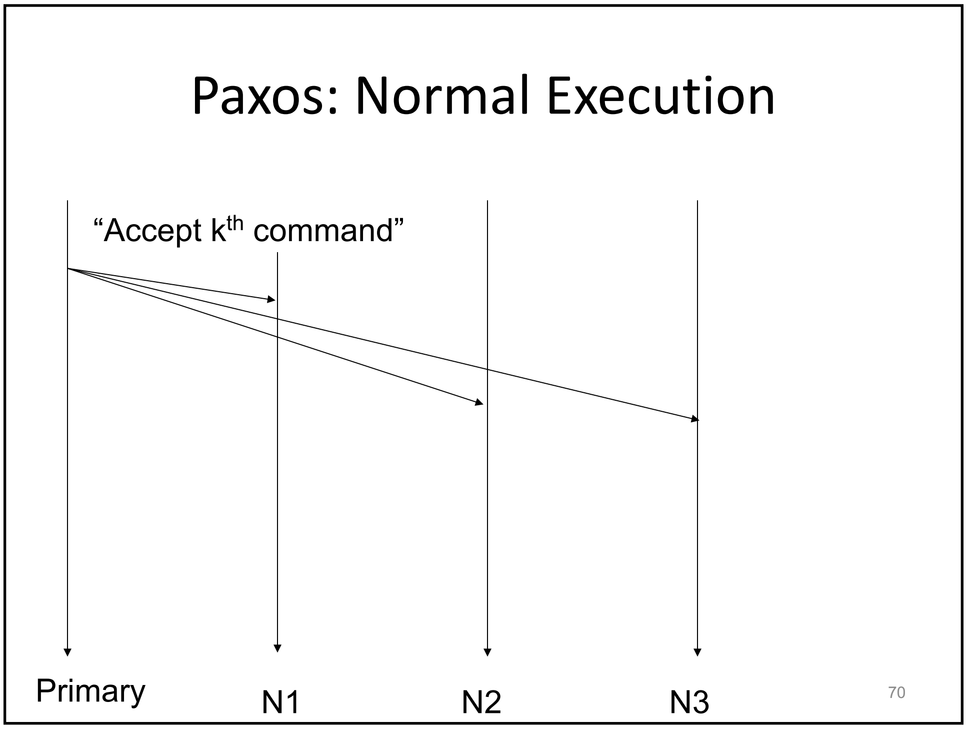


Paxos: Consensus Box

- Collects proposed values
- Picks one proposed value
- Remembers it forever

69

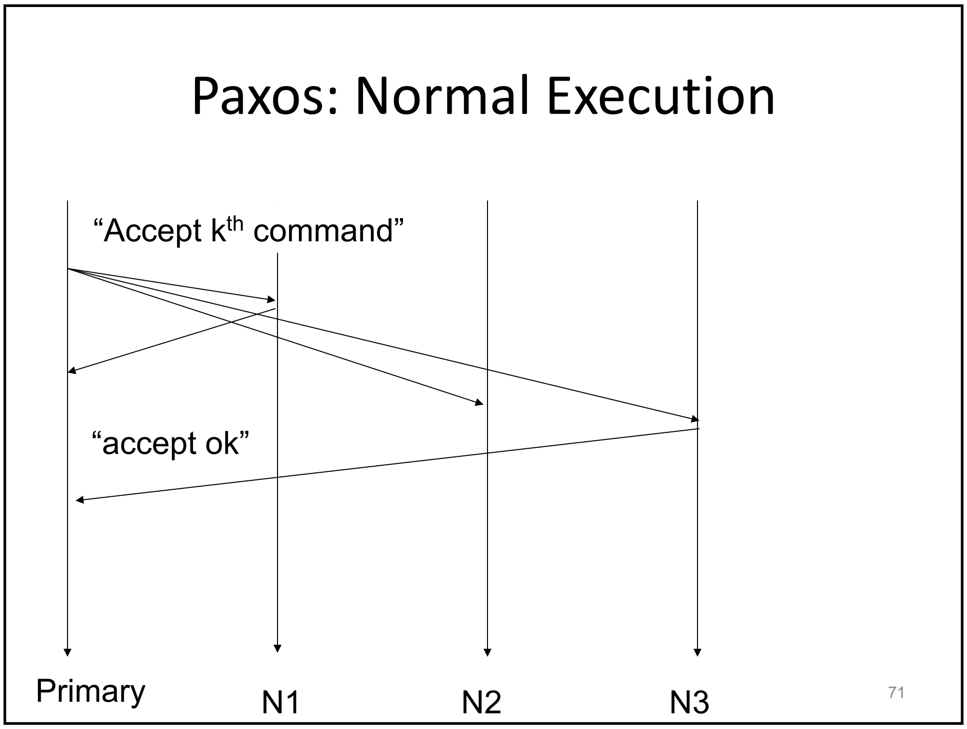
69



Paxos: Normal Execution

70

70



Paxos: Normal Execution

“Accept kth command”

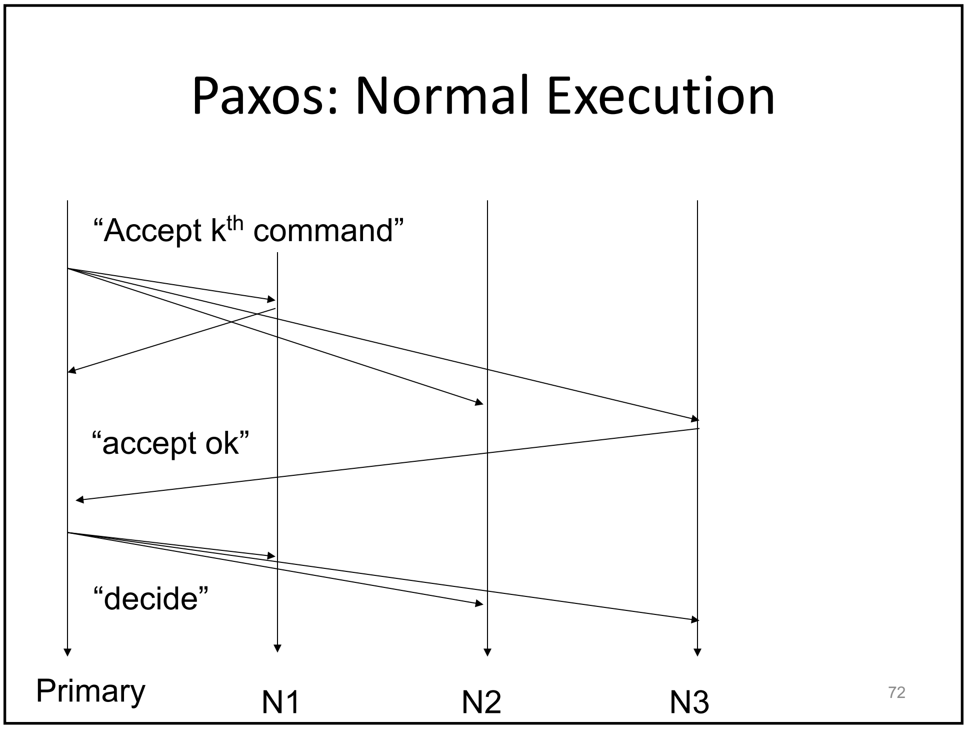
“accept ok”

Primary

N1

N2

N3



Paxos: Normal Execution

“Accept kth command”

“accept ok”

“decide”

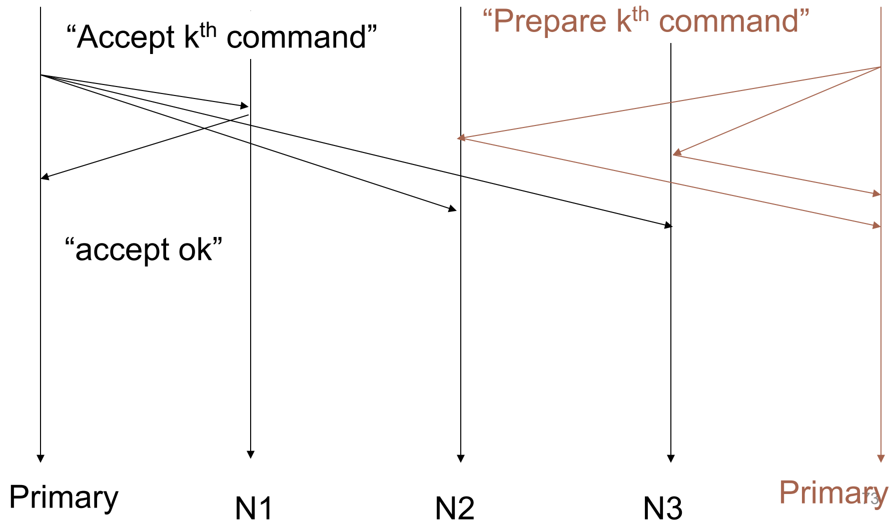
Primary

N1

N2

N3

Paxos: Split Brain



“Accept kth command”

“Prepare kth command”

“accept ok”

Primary

N1

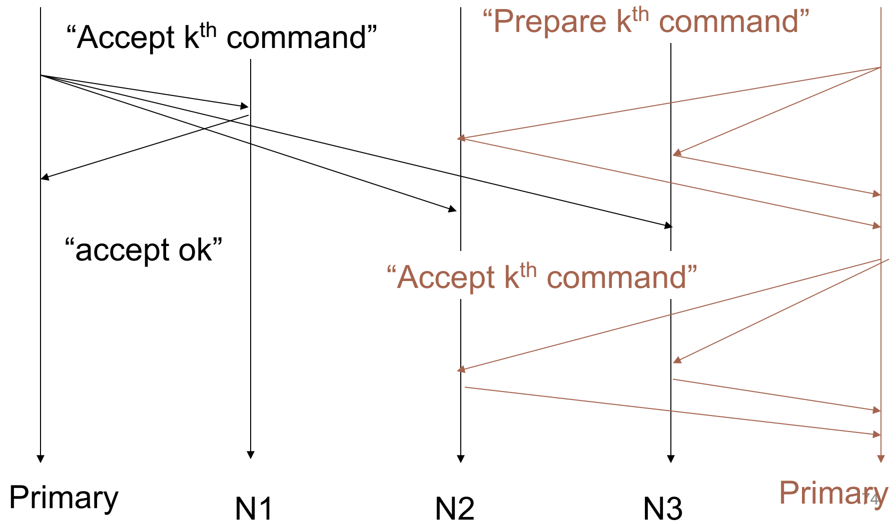
N2

N3

Primary

73

Paxos: Split Brain



“Accept kth command”

“Prepare kth command”

“accept ok”

“Accept kth command”

Primary

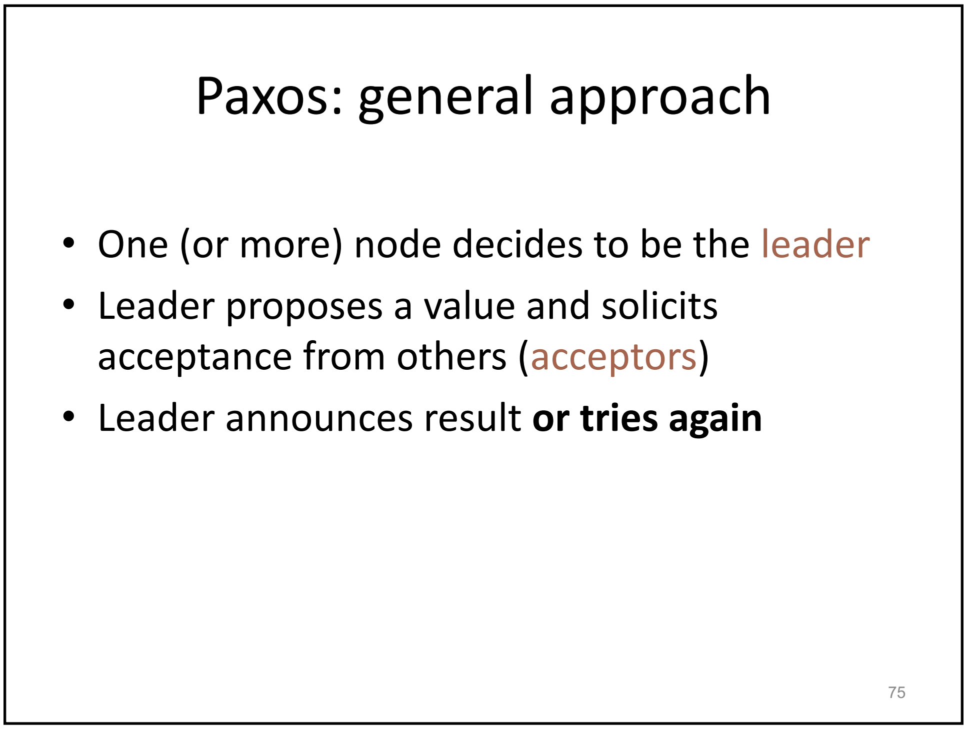
N1

N2

N3

Primary

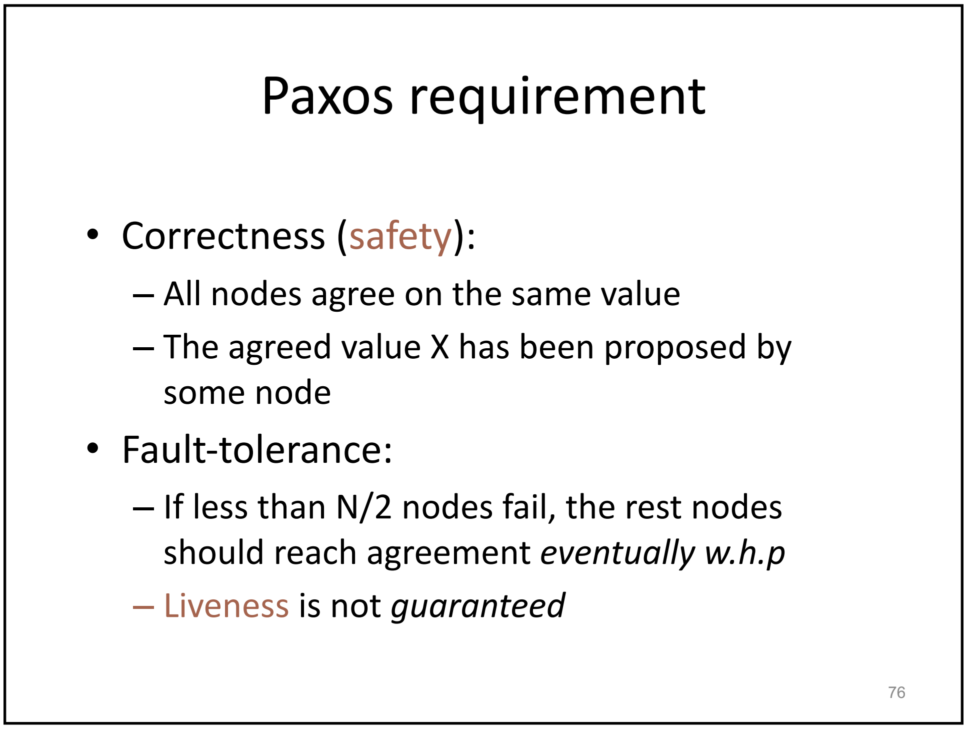
74



Paxos: general approach

- One (or more) node decides to be the leader
- Leader proposes a value and solicits acceptance from others (acceptors)
- Leader announces result **or tries again**

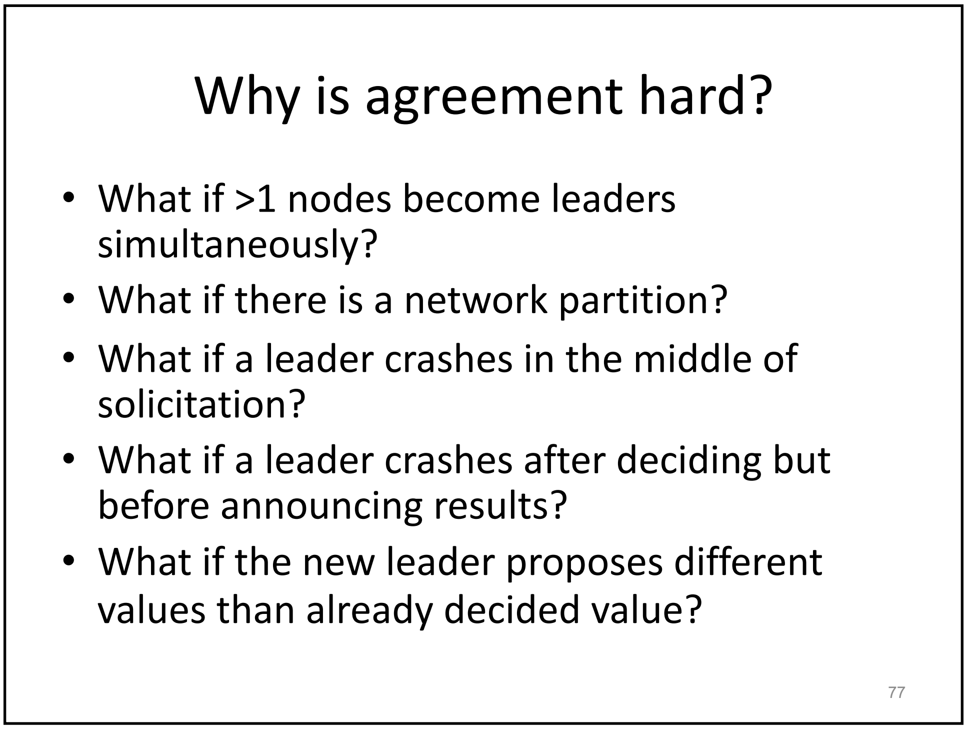
75



Paxos requirement

- Correctness (safety):
  - All nodes agree on the same value
  - The agreed value X has been proposed by some node
- Fault-tolerance:
  - If less than N/2 nodes fail, the rest nodes should reach agreement *eventually w.h.p*
  - Liveness is not *guaranteed*

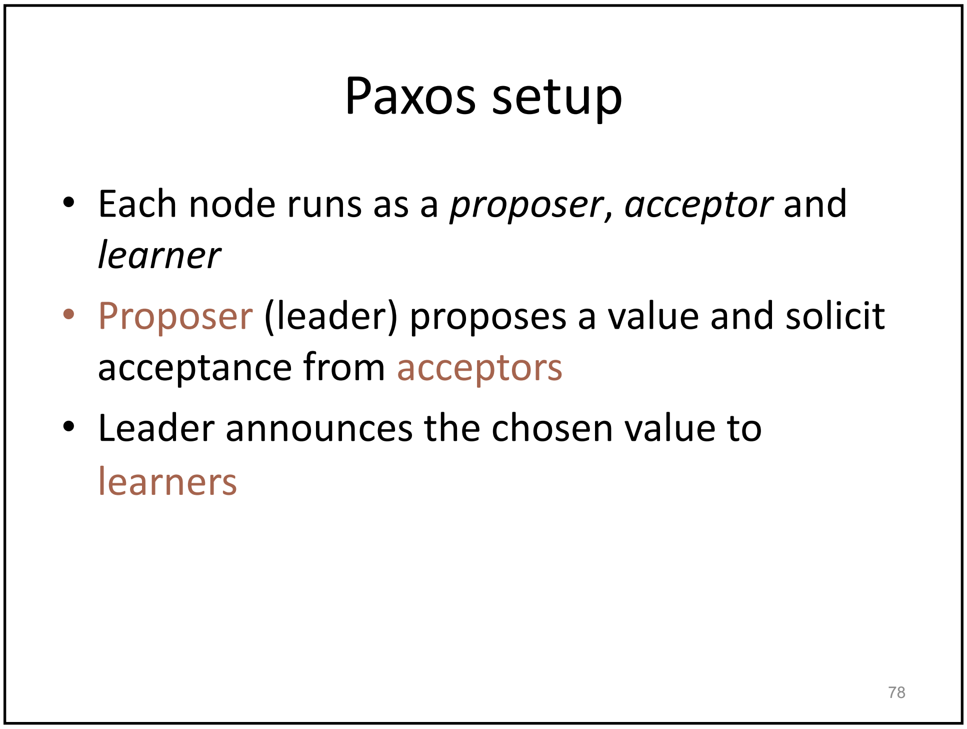
76



Why is agreement hard?

- What if >1 nodes become leaders simultaneously?
- What if there is a network partition?
- What if a leader crashes in the middle of solicitation?
- What if a leader crashes after deciding but before announcing results?
- What if the new leader proposes different values than already decided value?

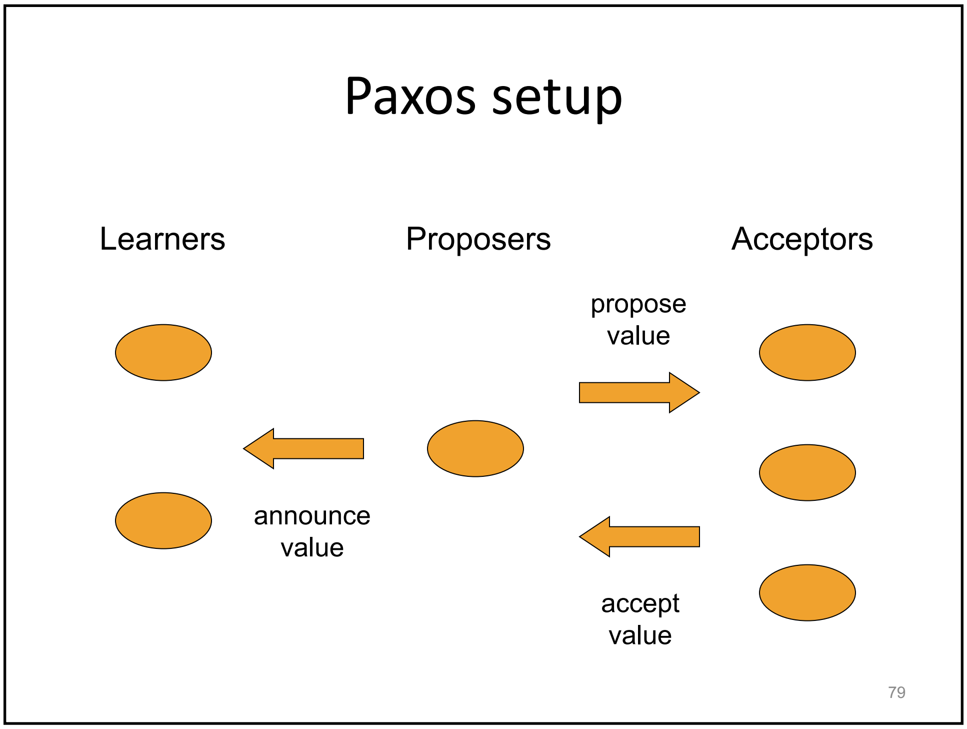
77



Paxos setup

- Each node runs as a *proposer*, *acceptor* and *learner*
- Proposer (leader) proposes a value and solicit acceptance from acceptors
- Leader announces the chosen value to learners

78



Paxos setup

Learners

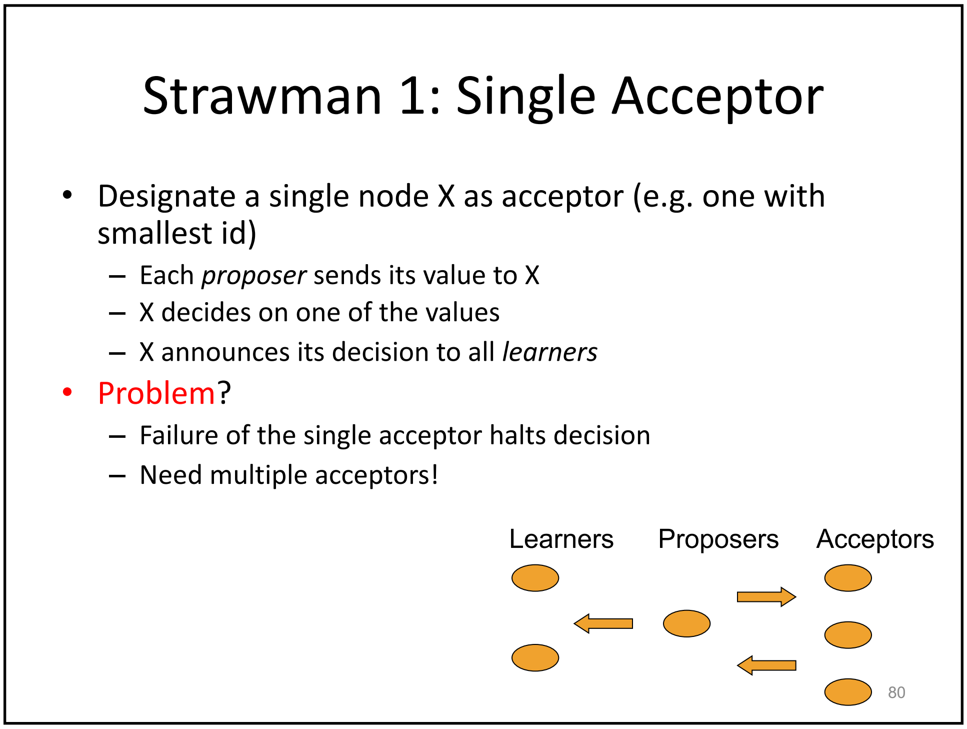
Proposers

Acceptors

propose value

announce value

accept value



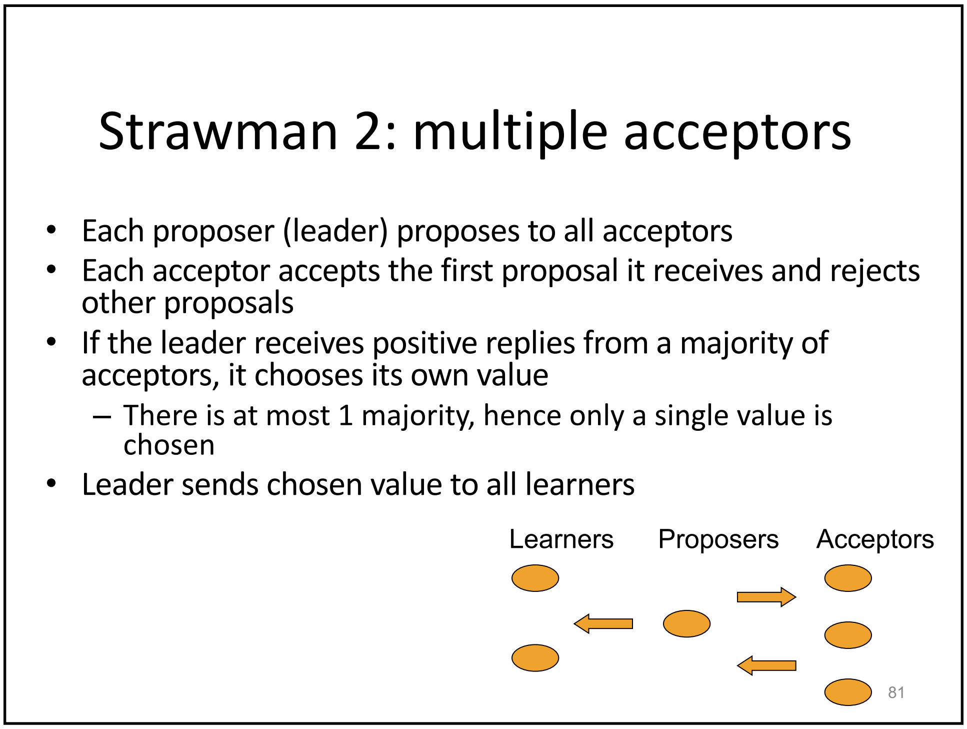
Strawman 1: Single Acceptor

- Designate a single node X as acceptor (e.g. one with smallest id)
  - Each *proposer* sends its value to X
  - X decides on one of the values
  - X announces its decision to all *learners*
- Problem?
  - Failure of the single acceptor halts decision
  - Need multiple acceptors!

Learners

Proposers

Acceptors

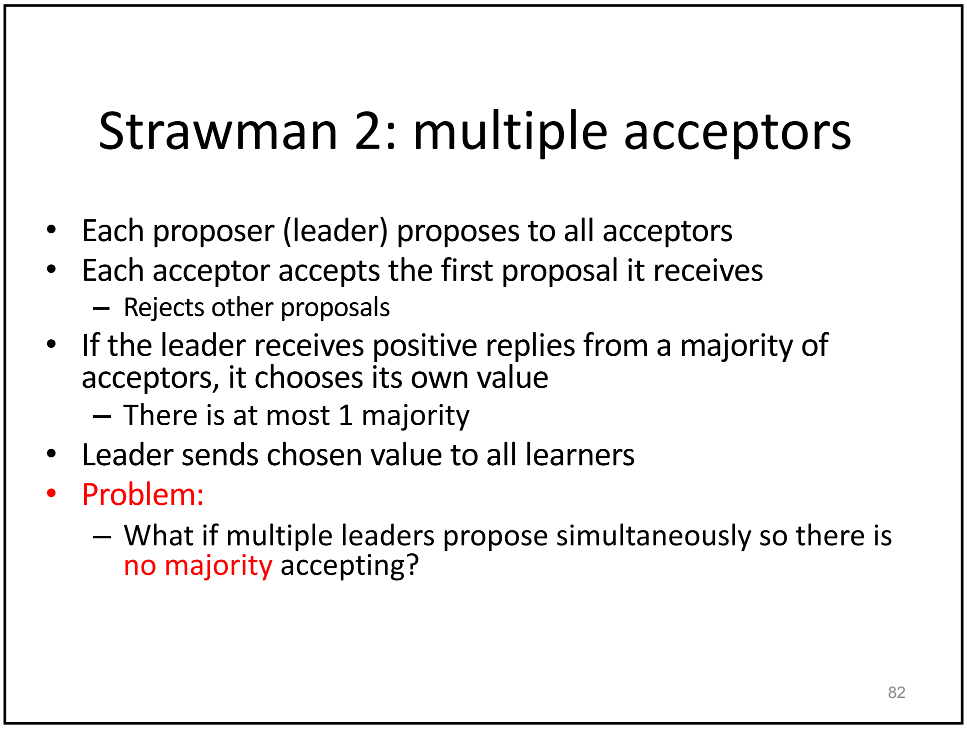


Strawman 2: multiple acceptors

- Each proposer (leader) proposes to all acceptors
- Each acceptor accepts the first proposal it receives and rejects other proposals
- If the leader receives positive replies from a majority of acceptors, it chooses its own value
  - There is at most 1 majority, hence only a single value is chosen
- Leader sends chosen value to all learners

Learners
Proposers
Acceptors

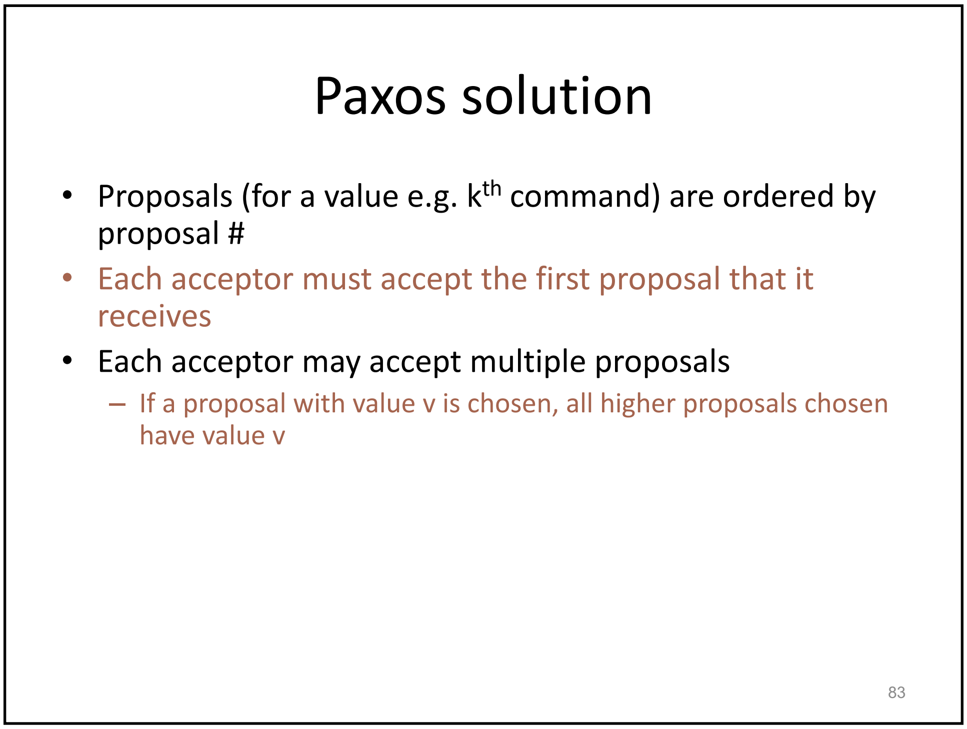
81



Strawman 2: multiple acceptors

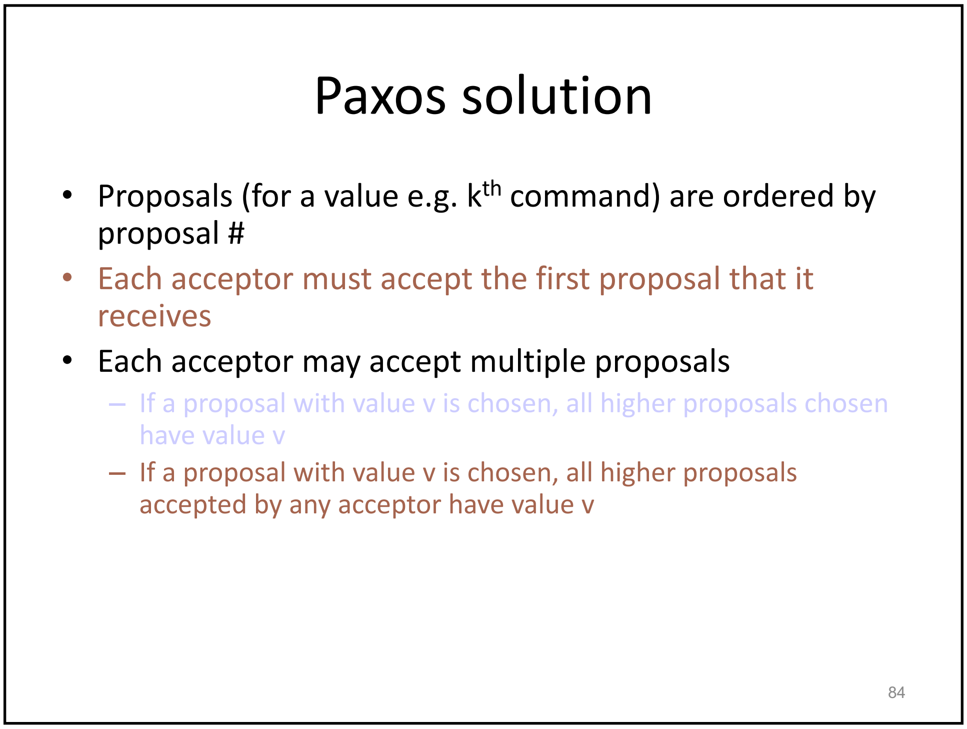
- Each proposer (leader) proposes to all acceptors
- Each acceptor accepts the first proposal it receives
  - Rejects other proposals
- If the leader receives positive replies from a majority of acceptors, it chooses its own value
  - There is at most 1 majority
- Leader sends chosen value to all learners
- Problem:
  - What if multiple leaders propose simultaneously so there is no majority accepting?

82



Paxos solution

- Proposals (for a value e.g. k^{th} command) are ordered by proposal #
- Each acceptor must accept the first proposal that it receives
- Each acceptor may accept multiple proposals
  - If a proposal with value v is chosen, all higher proposals chosen have value v



Paxos solution

- Proposals (for a value e.g. k^{th} command) are ordered by proposal #
- Each acceptor must accept the first proposal that it receives
- Each acceptor may accept multiple proposals
  - If a proposal with value v is chosen, all higher proposals chosen have value v
  - If a proposal with value v is chosen, all higher proposals accepted by any acceptor have value v

Paxos solution

- Proposals (for a value e.g. kth command) are ordered by proposal #
- Each acceptor must accept the first proposal that it receives
- Each acceptor may accept multiple proposals
  - If a proposal with value v is chosen, all higher proposals chosen have value v
  - If a proposal with value v is chosen, all higher proposals accepted by any acceptor have value v
  - If a proposal with value v is chosen, all higher proposals issued by any proposer have value v

85

85

Paxos solution

- Proposals (for a value e.g. kth command) are ordered by proposal #
- Each acceptor must accept the first proposal that it receives
- Each acc
  - If a pro sen have va
  - If a pro accept
  - If a pro ed by any proposer have value v

Before proposing value v for proposal n, proposer will poll acceptors for
- Promise that they will not accept any future proposals < n
- What value if any that they accepted for highest numbered proposal < n

86

86

Paxos operation: node state

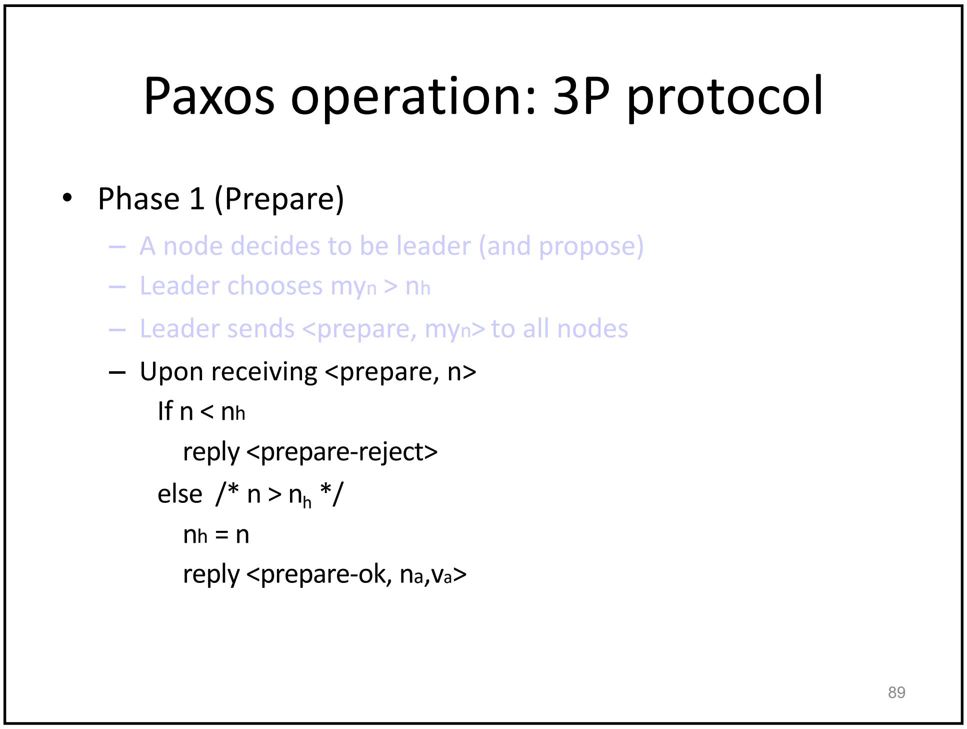
- Each node maintains:
  - – n_a, v_a: highest proposal # and its corresponding accepted value
    - initially null
  - – n_h: highest proposal # seen
  - – $my_{n:}$ my proposal # in current Paxos

87

Paxos operation: 3P protocol

- Phase 1 (Prepare)
  - – A node decides to be leader (and propose)
  - – Leader chooses $my_n > n_h$
  - – Leader sends <prepare, my_n> to all nodes

88

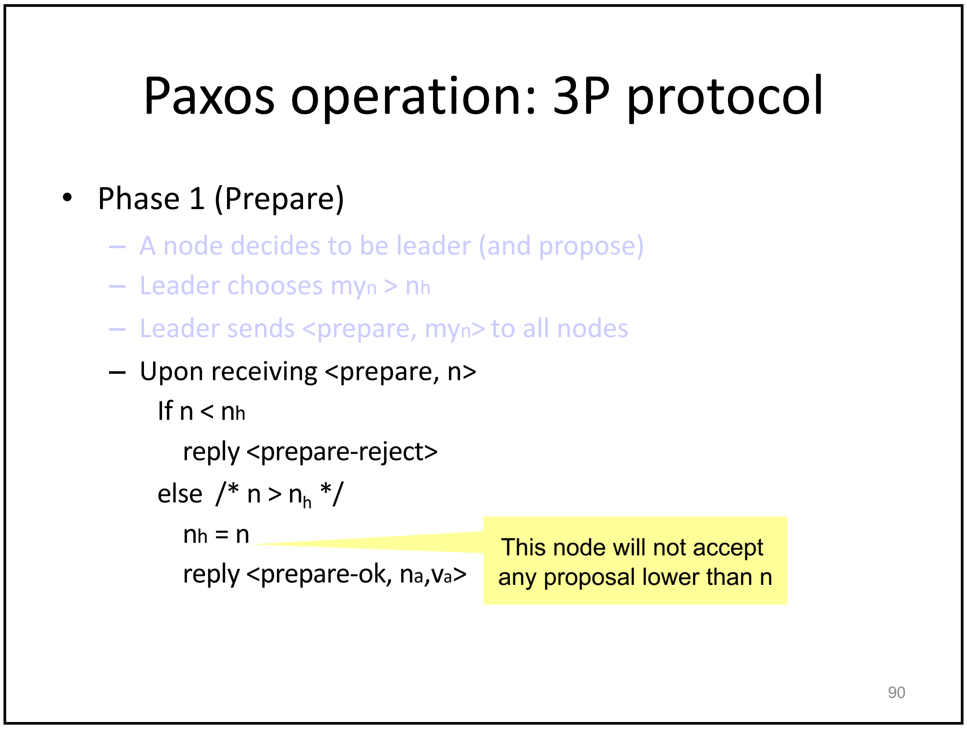


Paxos operation: 3P protocol

- Phase 1 (Prepare)
  - A node decides to be leader (and propose)
  - Leader chooses $my_n > n_h$
  - Leader sends <prepare, my_n> to all nodes
  - Upon receiving <prepare, n>

```
If $n < n_h$
   reply <prepare-reject>
else  /* $n > n_h$ */
   $n_h = n$
   reply <prepare-ok, $n_a$,$v_a$>
```

89



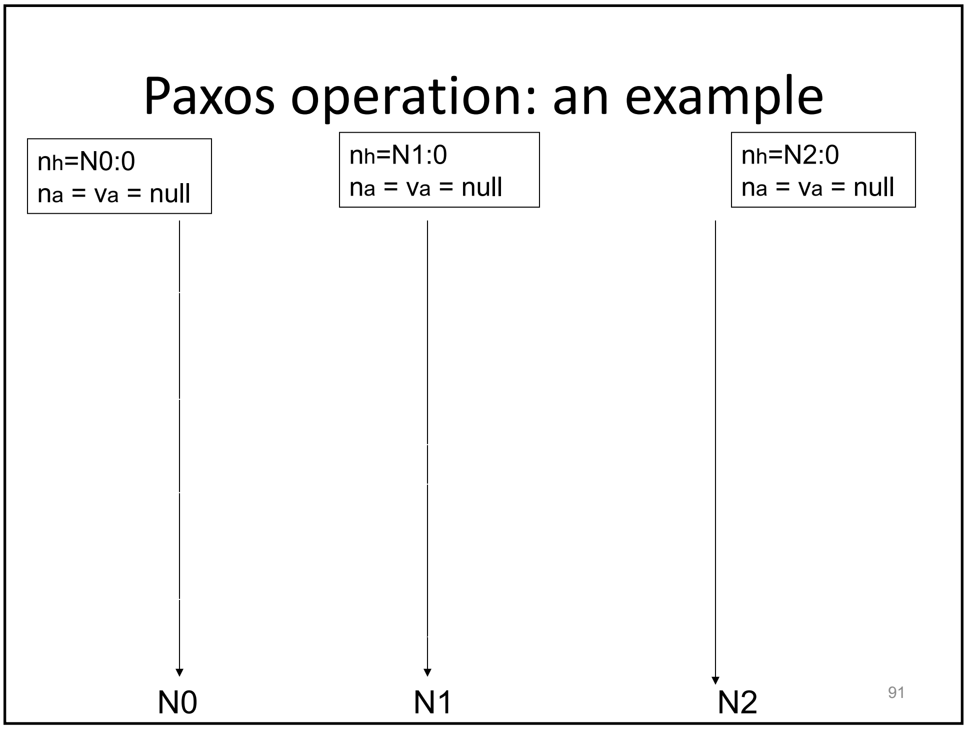
Paxos operation: 3P protocol

- Phase 1 (Prepare)
  - A node decides to be leader (and propose)
  - Leader chooses $my_n > n_h$
  - Leader sends <prepare, my_n> to all nodes
  - Upon receiving <prepare, n>

```
If $n < n_h$
   reply <prepare-reject>
else  /* $n > n_h$ */
   $n_h = n$
   reply <prepare-ok, $n_a$,$v_a$>
```

This node will not accept any proposal lower than n

90



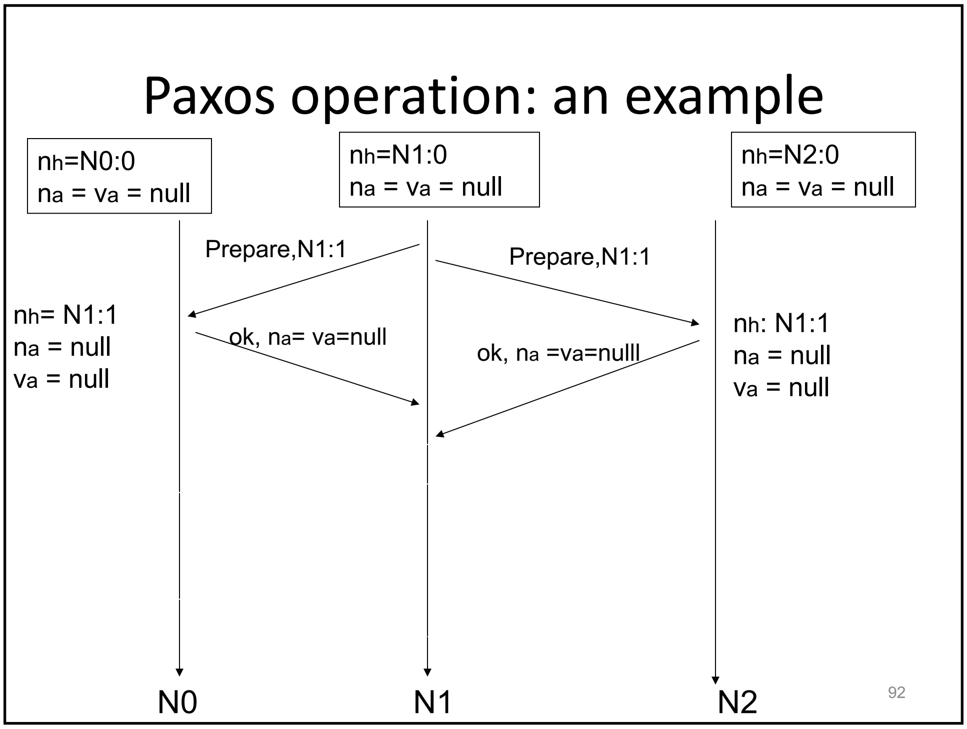
Paxos operation: an example

n_h=N0:0
n_a = v_a = null

n_h=N1:0
n_a = v_a = null

n_h=N2:0
n_a = v_a = null

N0 N1 N2



Paxos operation: an example

n_h=N0:0
n_a = v_a = null

n_h=N1:0
n_a = v_a = null

n_h=N2:0
n_a = v_a = null

Prepare,N1:1

Prepare,N1:1

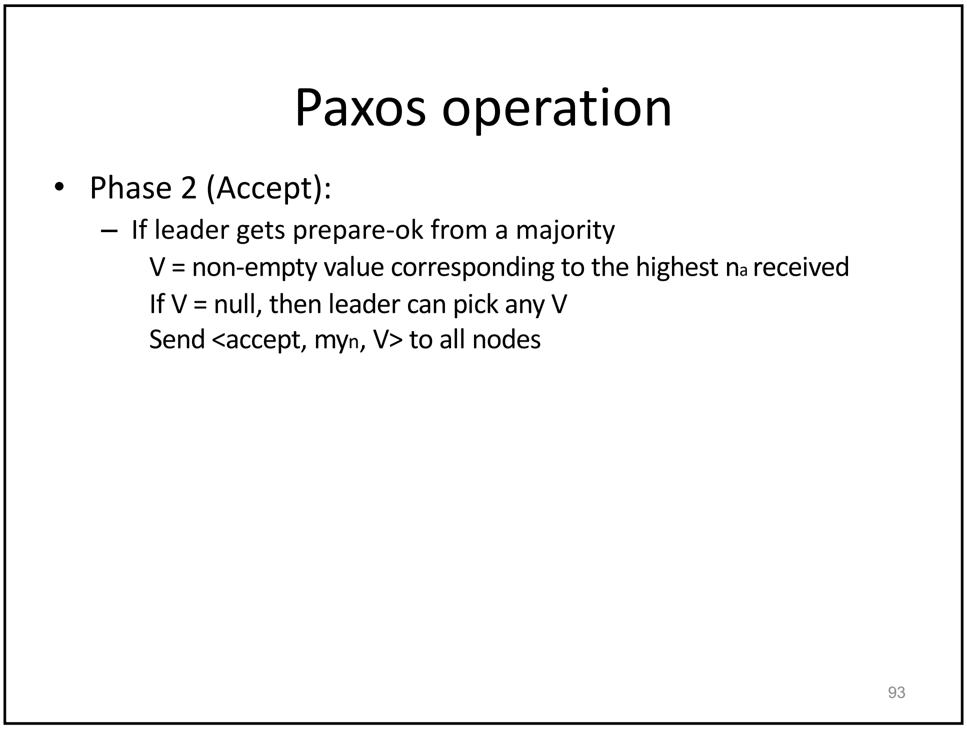
n_h= N1:1
n_a = null
v_a = null

ok, n_a= v_a=null

ok, n_a =v_a=nulll

n_h: N1:1
n_a = null
v_a = null

N0 N1 N2



Paxos operation

- Phase 2 (Accept):
  - If leader gets prepare-ok from a majority

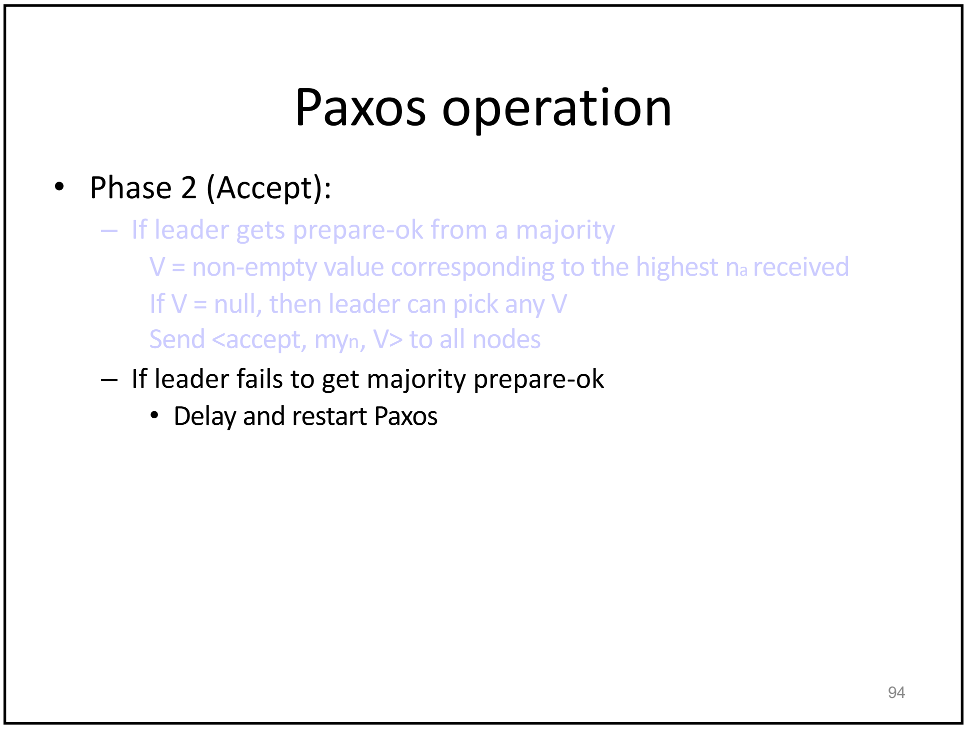
    V = non-empty value corresponding to the highest n_a received

    If V = null, then leader can pick any V

    Send <accept, my_n, V> to all nodes

93

93



Paxos operation

- Phase 2 (Accept):
  - If leader gets prepare-ok from a majority

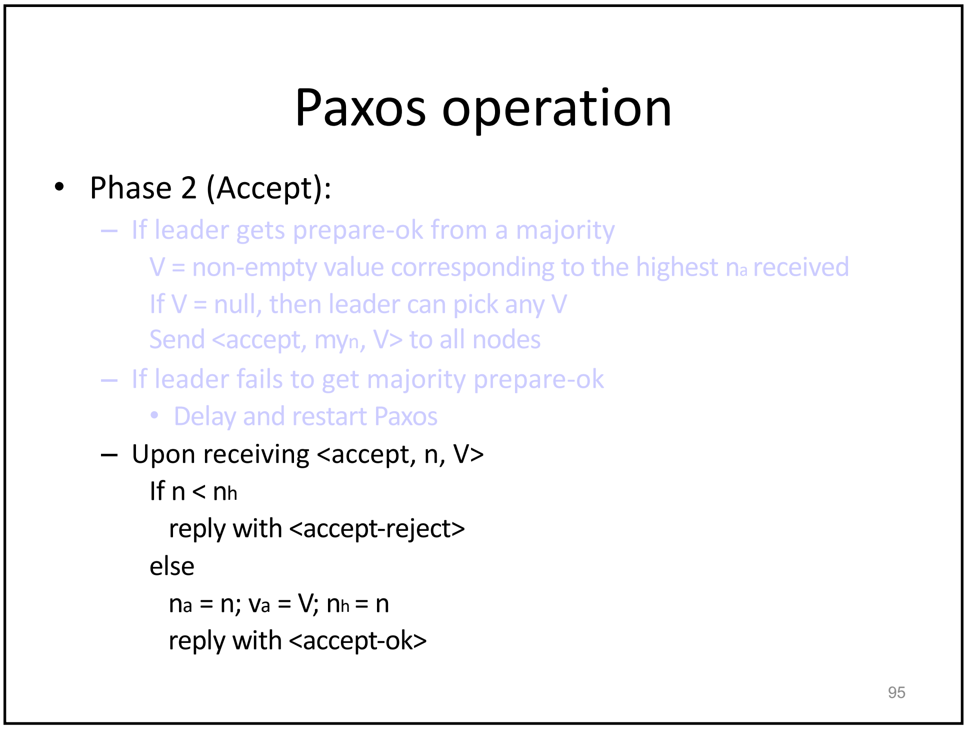
    V = non-empty value corresponding to the highest n_a received

    If V = null, then leader can pick any V

    Send <accept, my_n, V> to all nodes
  - If leader fails to get majority prepare-ok
    - Delay and restart Paxos

94

94



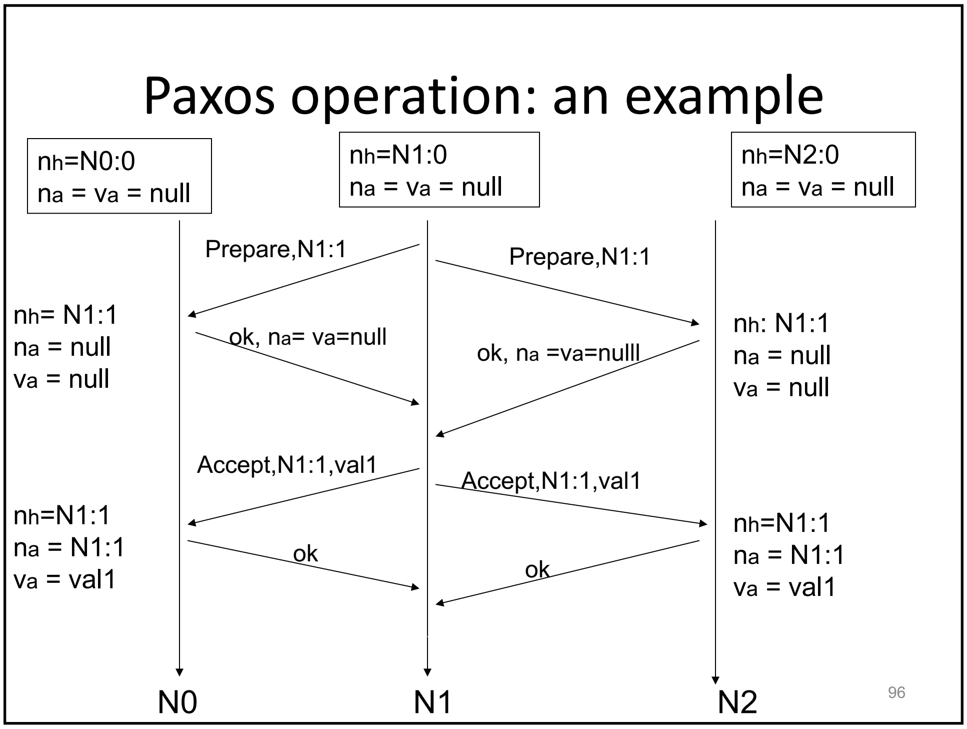
Paxos operation

- Phase 2 (Accept):

```
– If leader gets prepare-ok from a majority
    V = non-empty value corresponding to the highest $n_a$ received
    If V = null, then leader can pick any V
    Send <accept, $my_n$, V> to all nodes
– If leader fails to get majority prepare-ok
    • Delay and restart Paxos
– Upon receiving <accept, n, V>
    If $n < n_h$
      reply with <accept-reject>
    else
      $n_a = n$; $v_a = V$; $n_h = n$
      reply with <accept-ok>
```

95

95



Paxos operation: an example

n_h=N0:0
n_a = v_a = null

n_h=N1:0
n_a = v_a = null

n_h=N2:0
n_a = v_a = null

Prepare,N1:1

Prepare,N1:1

n_h= N1:1
n_a = null
v_a = null

ok, n_a= v_a=null

ok, n_a =v_a=nulll

n_h: N1:1
n_a = null
v_a = null

Accept,N1:1,val1

Accept,N1:1,val1

n_h=N1:1
n_a = N1:1
v_a = val1

ok

ok

n_h=N1:1
n_a = N1:1
v_a = val1

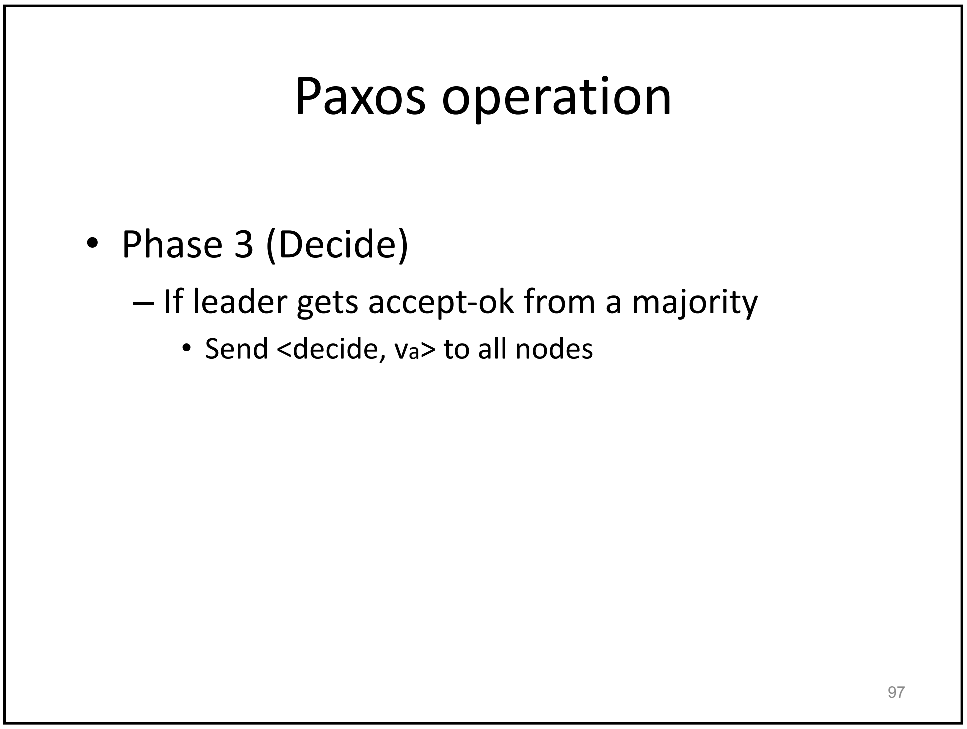
N0

N1

N2

96

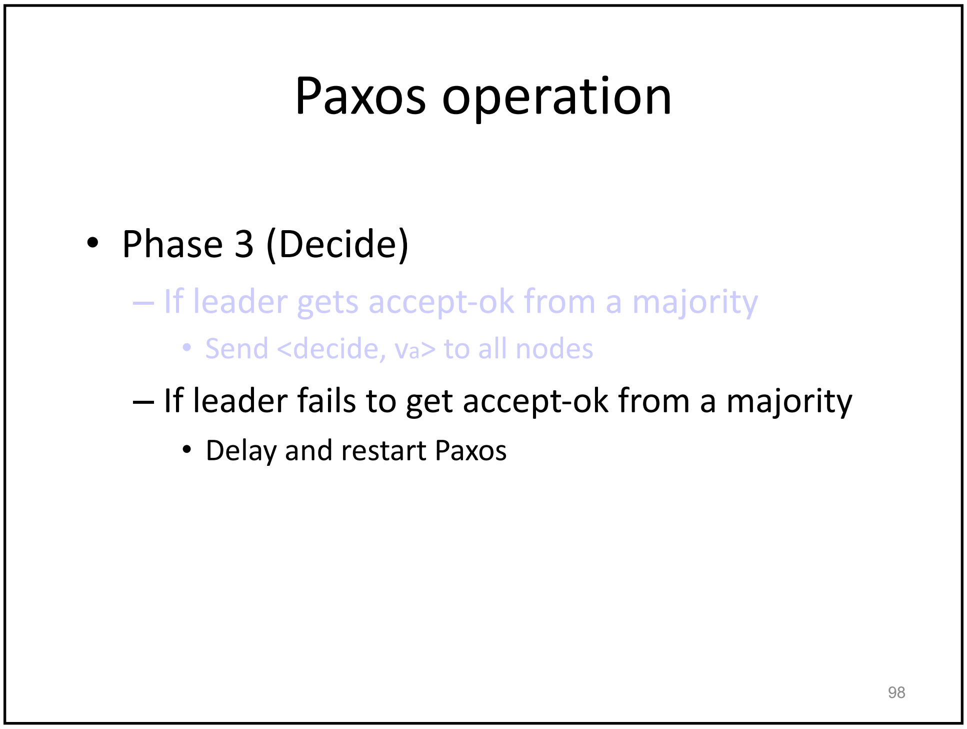
96



Paxos operation

- Phase 3 (Decide)
  - If leader gets accept-ok from a majority
    - Send <decide, v_a> to all nodes

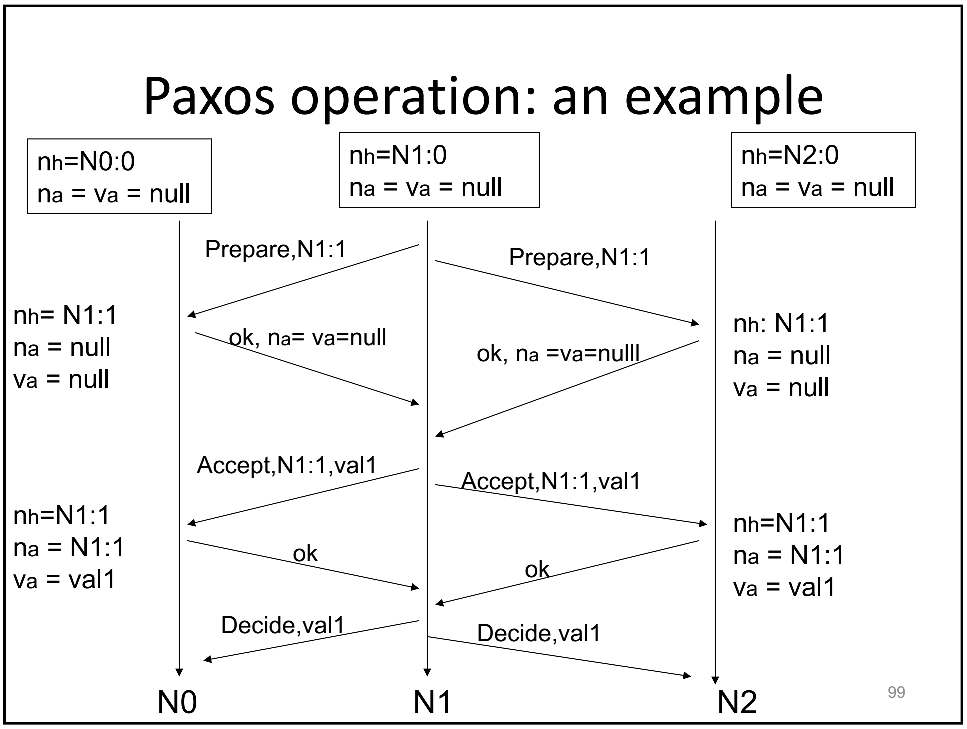
97



Paxos operation

- Phase 3 (Decide)
  - If leader gets accept-ok from a majority
    - Send <decide, v_a> to all nodes
  - If leader fails to get accept-ok from a majority
    - Delay and restart Paxos

98



Paxos operation: an example

n_h=N0:0
n_a = v_a = null

n_h=N1:0
n_a = v_a = null

n_h=N2:0
n_a = v_a = null

Prepare,N1:1

Prepare,N1:1

n_h= N1:1
n_a = null
v_a = null

ok, n_a= v_a=null

ok, n_a =v_a=nulll

n_h: N1:1
n_a = null
v_a = null

Accept,N1:1,val1

Accept,N1:1,val1

n_h=N1:1
n_a = N1:1
v_a = val1

ok

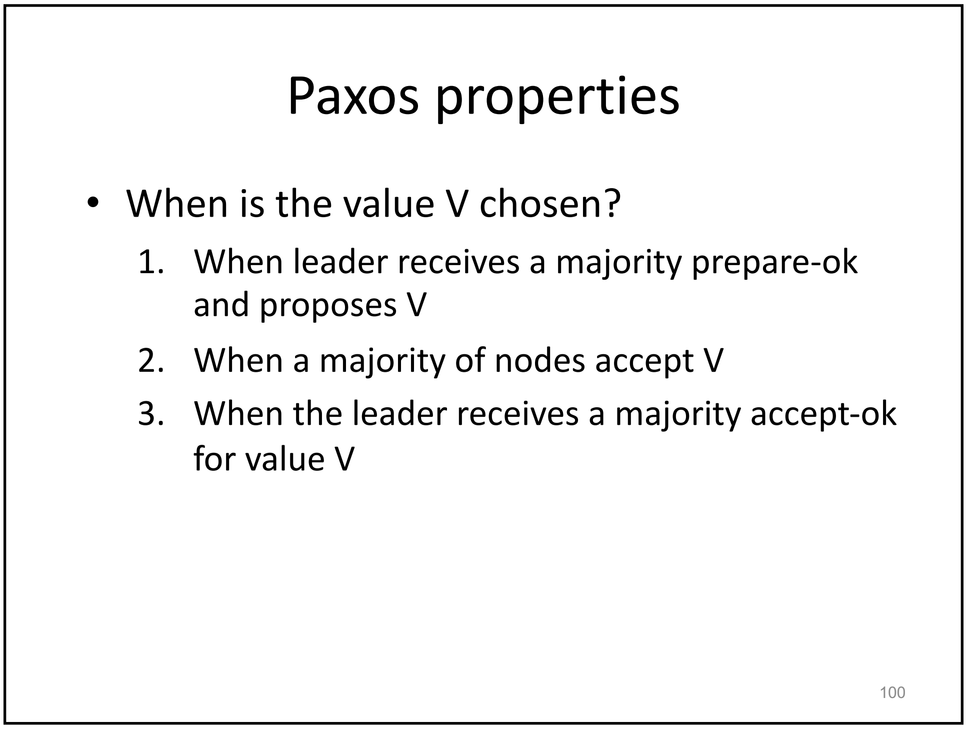
ok

n_h=N1:1
n_a = N1:1
v_a = val1

Decide,val1

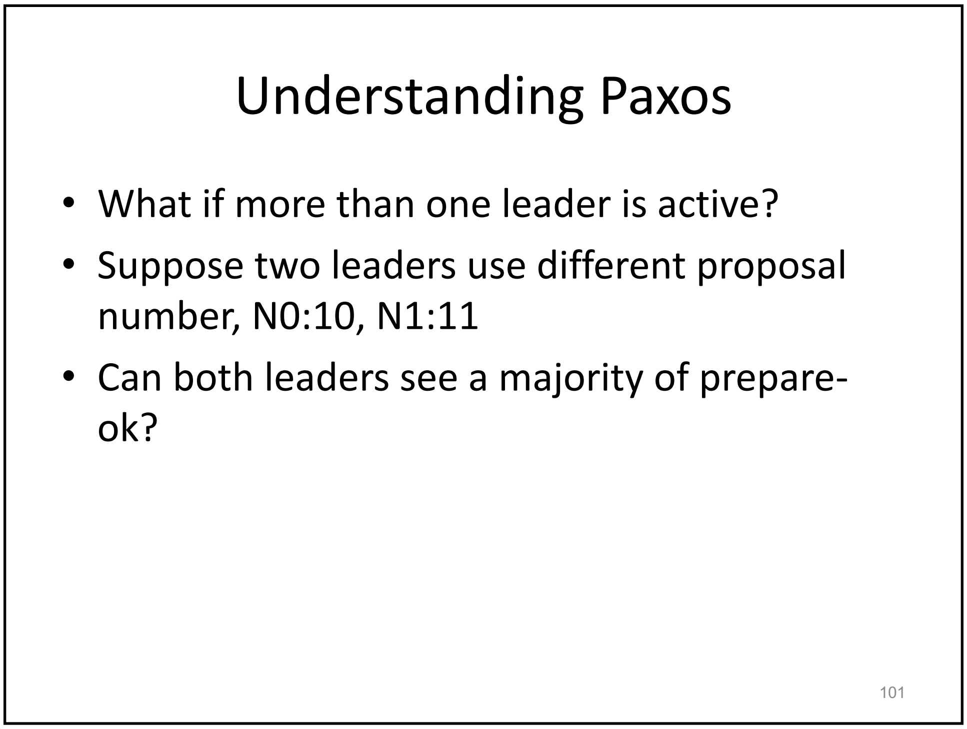
Decide,val1

N0 N1 N2



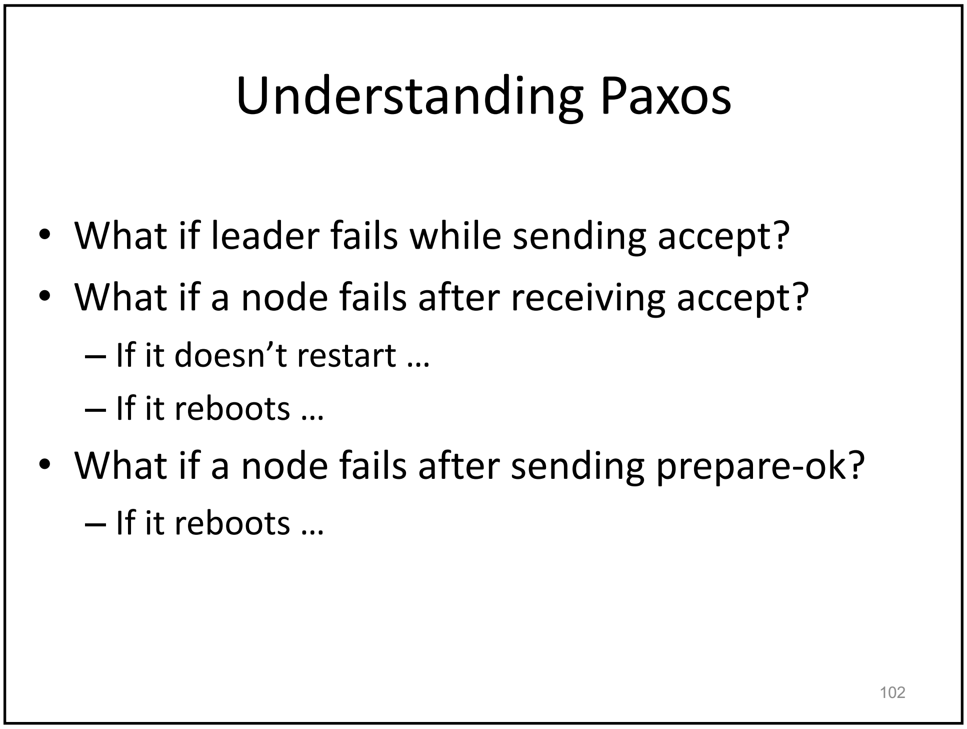
Paxos properties

- When is the value V chosen?
  1. When leader receives a majority prepare-ok and proposes V
  2. When a majority of nodes accept V
  3. When the leader receives a majority accept-ok for value V



Understanding Paxos

- What if more than one leader is active?
- Suppose two leaders use different proposal number, N0:10, N1:11
- Can both leaders see a majority of prepare-ok?



Understanding Paxos

- What if leader fails while sending accept?
- What if a node fails after receiving accept?
  - If it doesn’t restart …
  - If it reboots …
- What if a node fails after sending prepare-ok?
  - If it reboots …