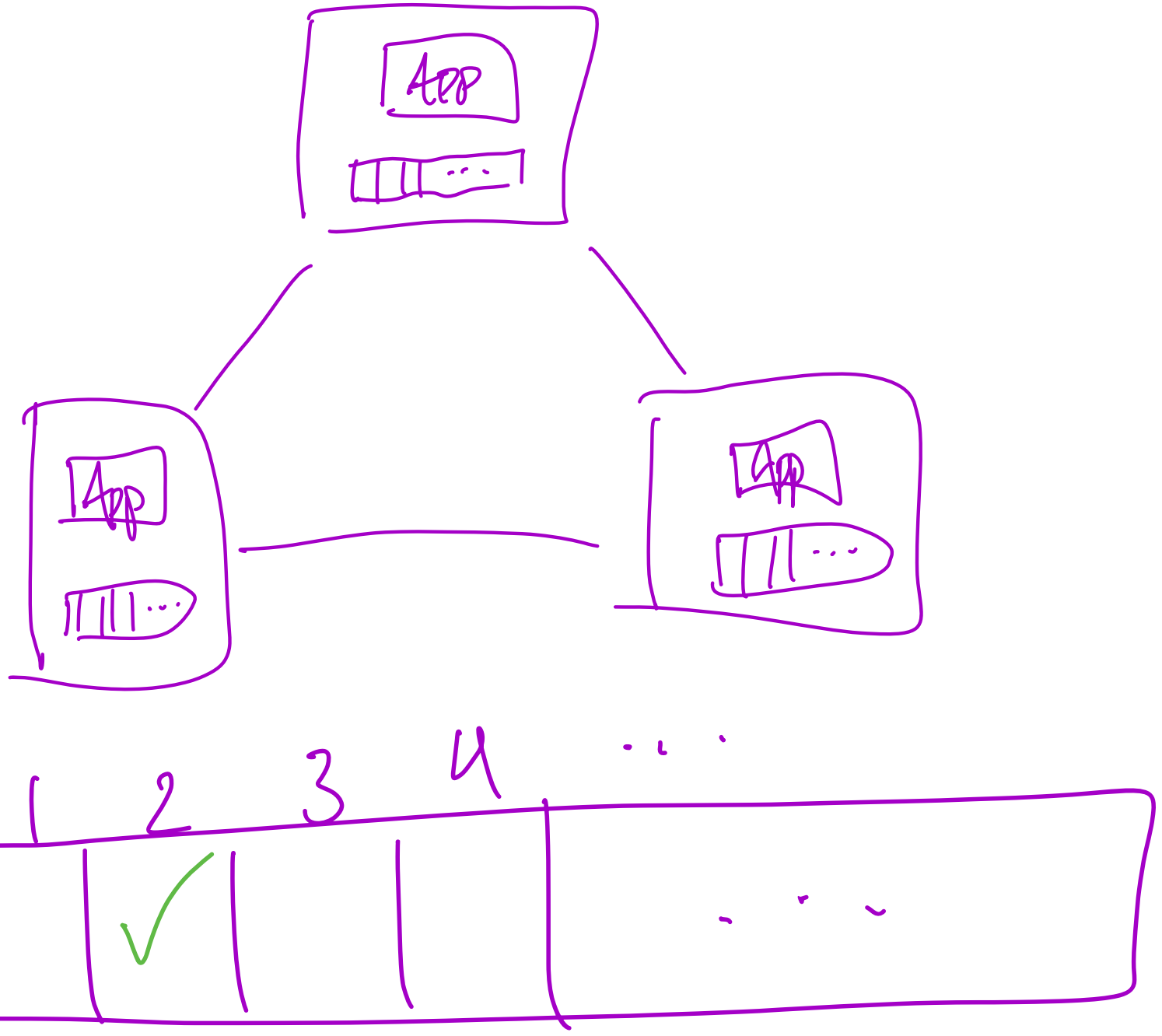
CSE P 552

Distributed Systems

Week 4: MultiPaxos



slot = index into log

Phase 1

1a : Proposer → Acceptor

- ballot : (ballot_num, proposer_id)

1b : Acceptor → Proposer

- ballot
- highest balloted proposal accepted by this acceptor in a smaller ballot

(sending 1b is a promise not to accept any proposals in smaller ballots

ever again)

Phase 2 (Proposer waits for a maj- of 1b msgs)

2a: Proposer → Acceptor

- ballot
- proposed request | value
  - highest balloted proposal on any received 1b
  - or any value if all are None

2b: Acceptor → Learner

- ballot
- proposed value

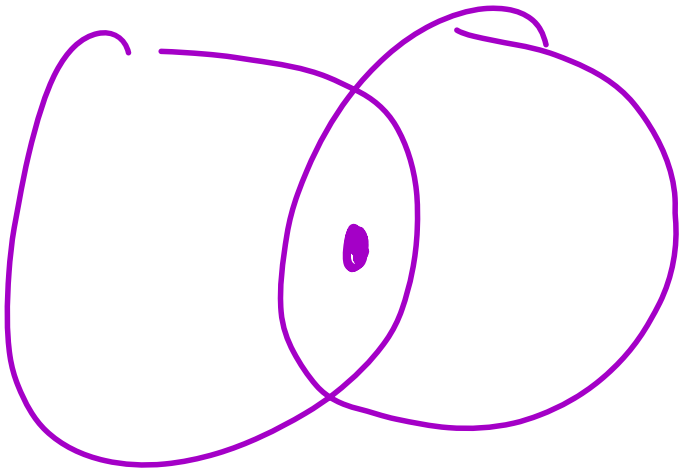
- only send 2b if have not promised not to participate in this ballot

Learner wait for a maj of 2b msgs in the same ballot, then that value

is decided. (execute the request)

Chosen(v) = there is a ballot b and a set of acceptors A s.t. $|A| > \lfloor n/2 \rfloor$ and every acceptor in A sent 2b(b, v)

Suppose Chosen(v) in ballot 50
and then P_2 starts ballot 51
→ waits for msg of 1b responses

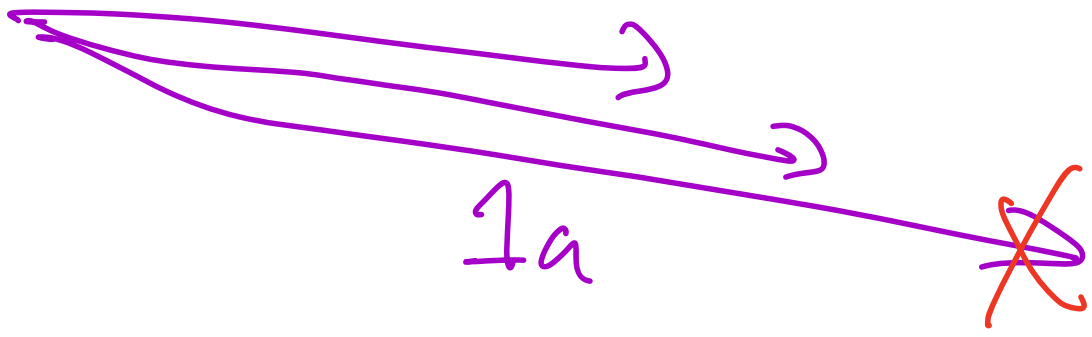


A

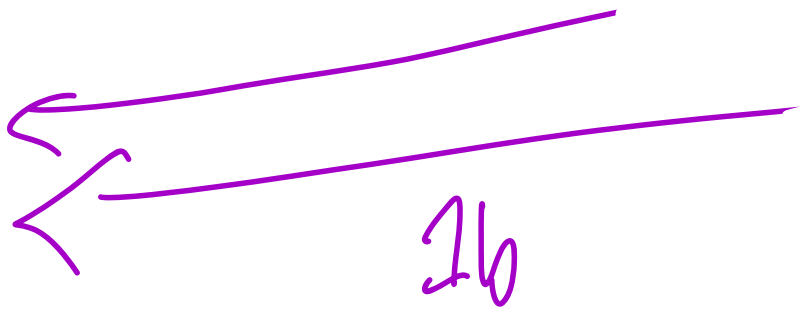
chose V in ballot 50

16 responses in ballot 51

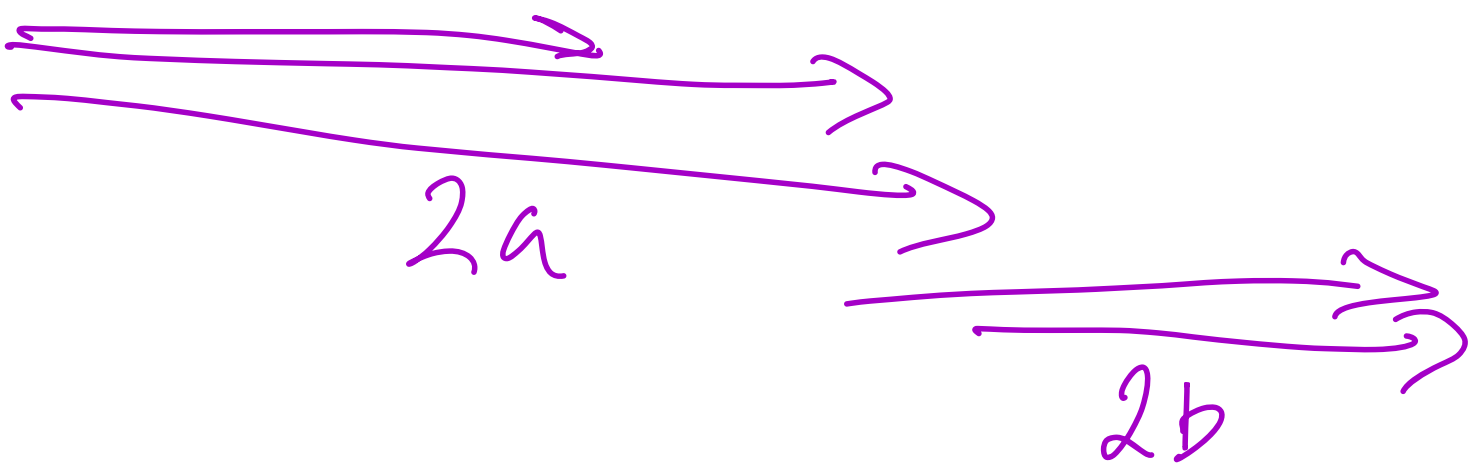
P_1 P_2 A_1 A_2 A_3 L



1a



1b



2a

2b

Replication and Consistency Models

phase1: introduce yourself

say what paper/summarize

initial impressions

phase2 questions:

- when is it safe to tell a client their result?
- how does a replica know it's in charge?
- Chubby/Vmware consistency model?

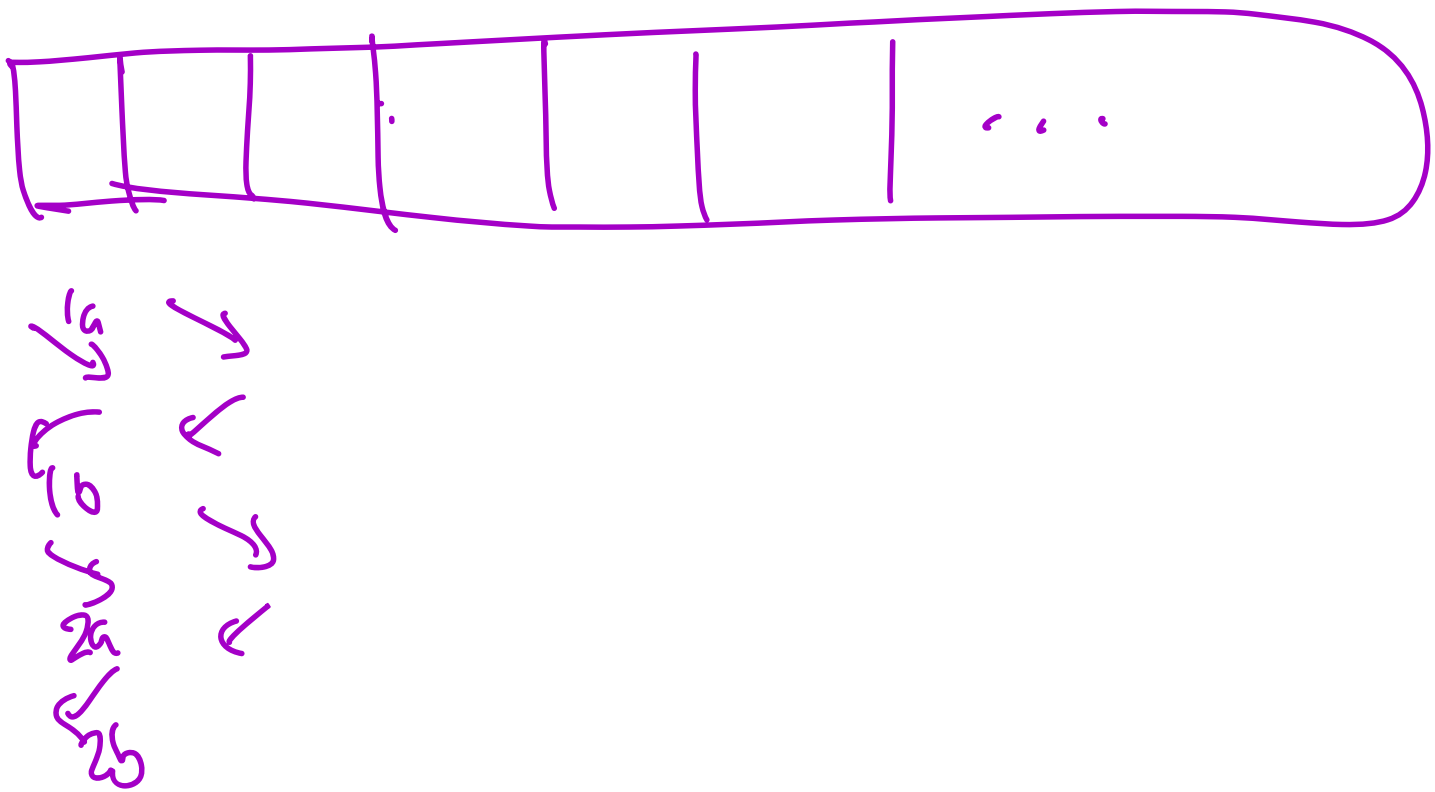
- Different roles need diff consistency models
- tension b/w locks vs application specific consistency
- how to choose consistency level?

Combining roles

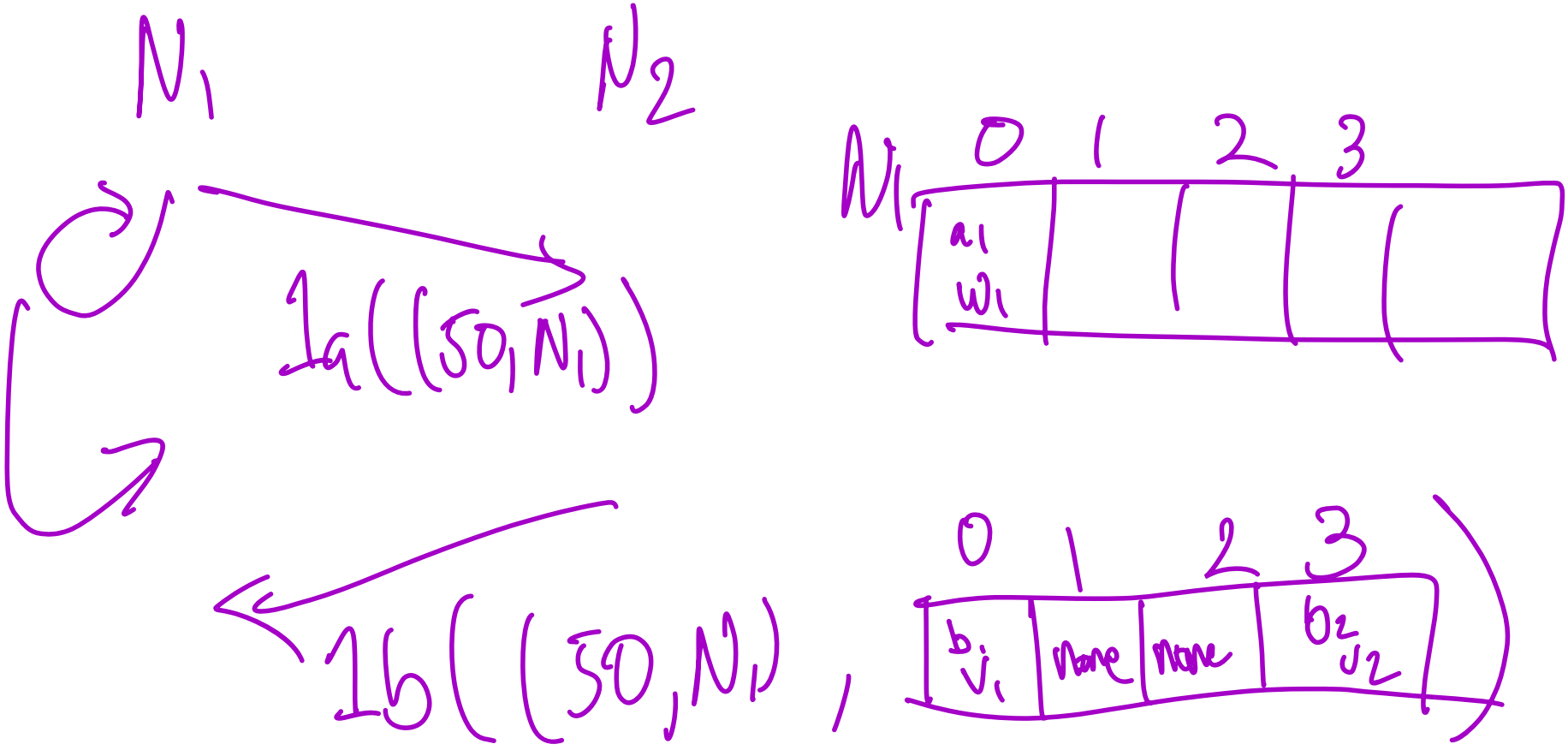
- every node is a proposer, acceptor, and learner
  - proposer conceptually sends 1a/2a msgs to itself (as acceptor)

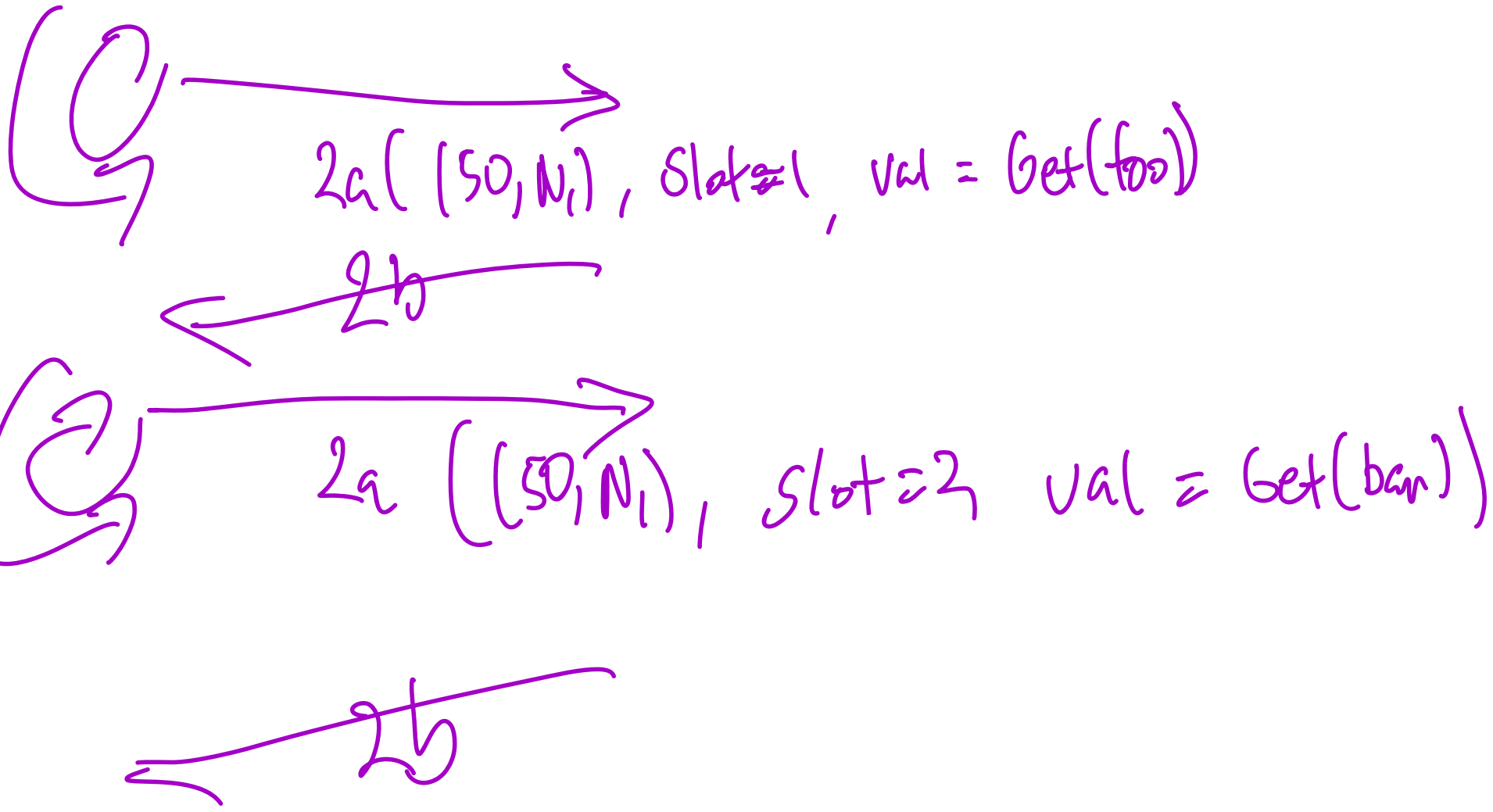
MultiPaxos Changes + optimizations

- put the slot # on msgs
- combine phase 1 across all slots



- phase 1 msgs don't have slot #
  - conceptually execute in all slots



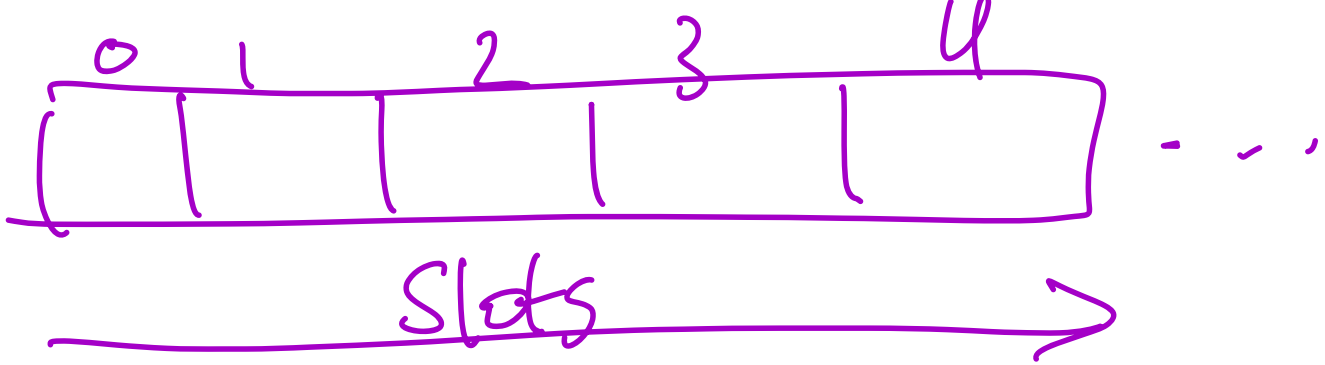

2a((SO, N1), slot=1, val = Get(foo))
2b
2a ((SO, N1), slot=2 val = Get(bar))
2b

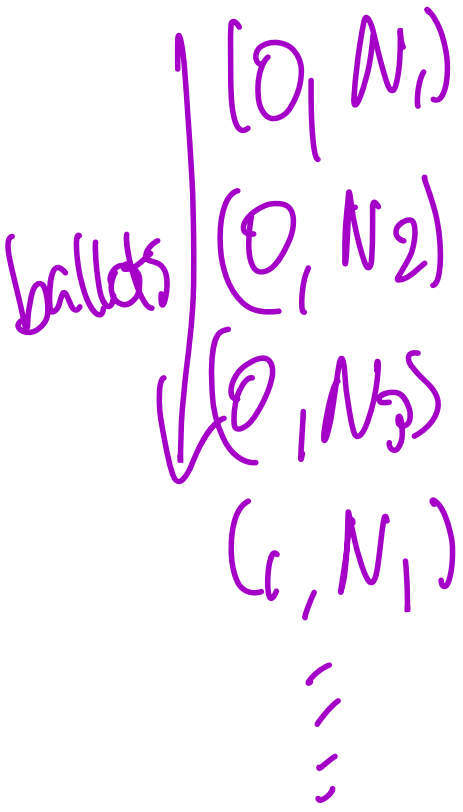
Leader / Distinguished Proposer

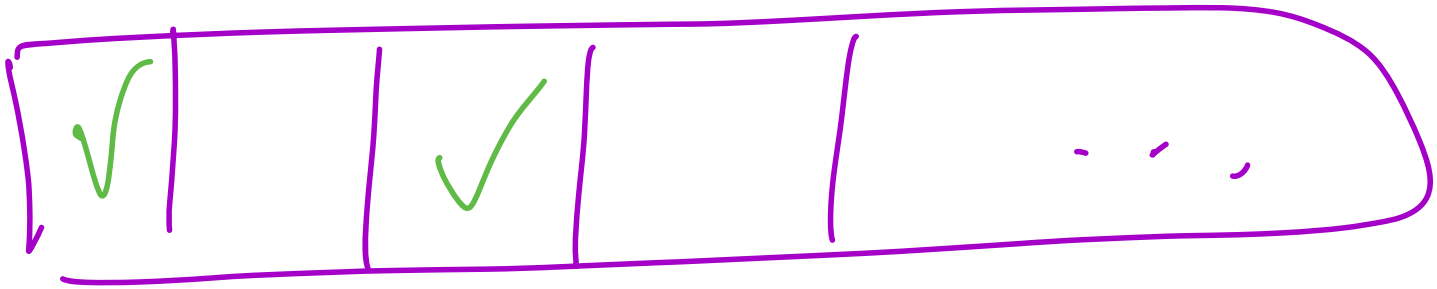
- phase 1 completion means other nodes won't accept proposals in smaller ballots
- we have freedom to propose in all empty slots
- other nodes should avoid running phase1 unless they suspect a problem

Heartbeat Protocol

- periodic "I'm alive" Leader → others
- piggy back bunch of stuff here
  - informing other nodes about new chosen values
    - including "blank" slots and overwriting existing (non chosen) values


0
1
2
3
4
Slots


Ballots
$(0, N_1)$
$(0, N_2)$
$(0, N_3)$
(c, N_1)



✓ = leader has chosen value for this slot

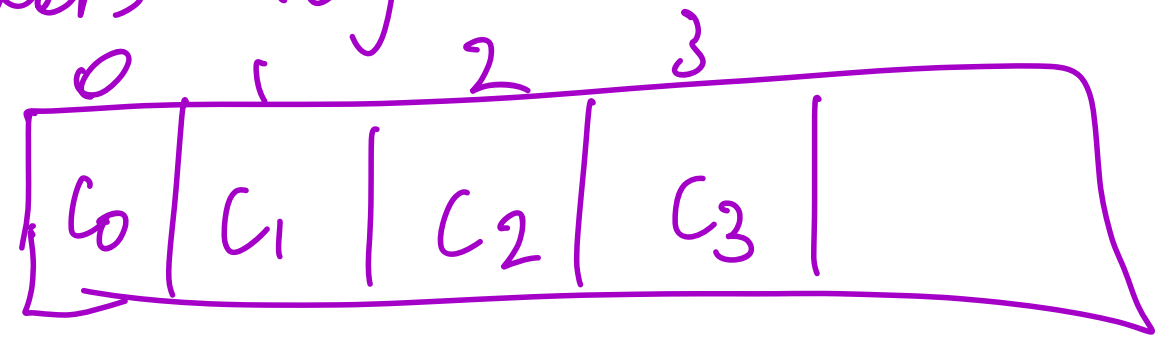
= there is a chosen value

Leader should attempt to fill gaps

- if any IS has a value for the slot, then propose it (by SDP)

- if no prev value,
  - if have a client req, propose it
  - otherwise, propose a "no-op"

Leader's log



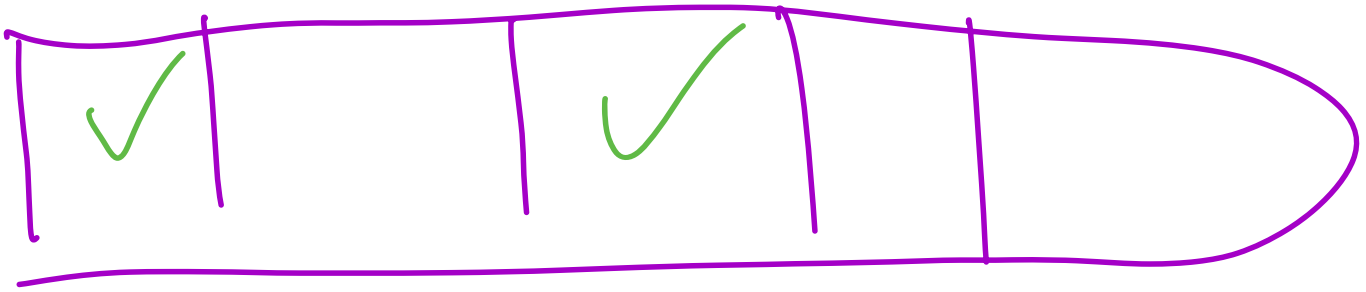
Slots 0 and 2

- N_2 receives 2a and sends 2b

Slots 1 and 3

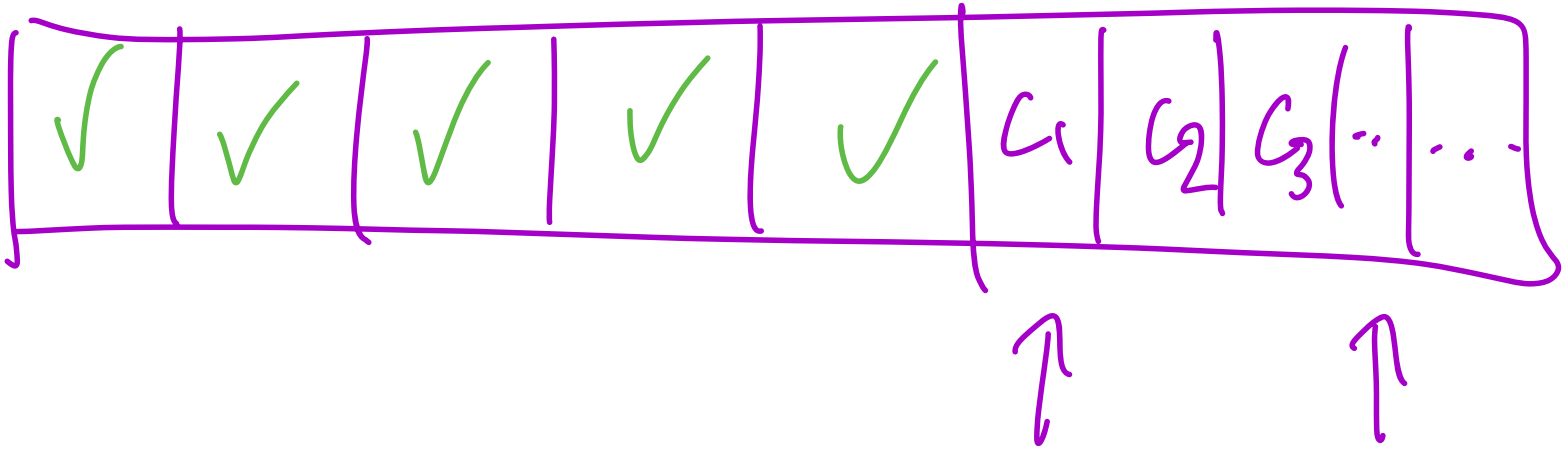
- all 2a's dropped

new leader log



Garbage Collection / Log Compaction

- when is it safe to delete
  what we know about a slot?



- Set every node to

Acknowledge that they know that slot is chosen

slot to execute

slot_out

empty slot

slot_in