CS-438 Decentralized Systems Engineering

Week 10

Permissionless consensus, multi-valued history consensus

Bitcoin, distributed ledger tech (DLT):

- Information-based money: a ledger

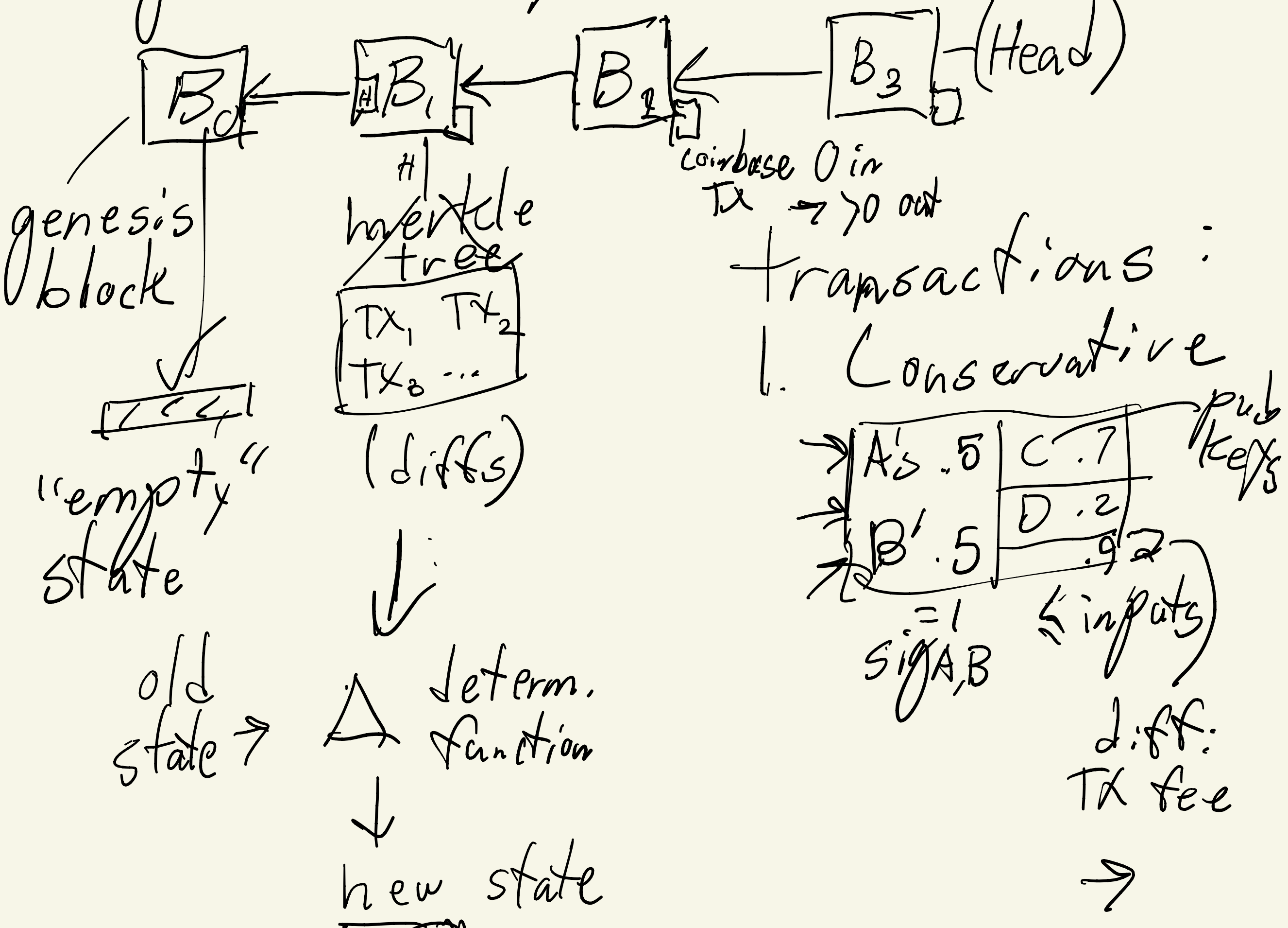
ledger: history of transactions

states:

Alice	1
Bob	2
Ch	1
⋮	

wallet balance

public keys



transactions:

1. Conservative

new state

A	1
B	1
Ch	2

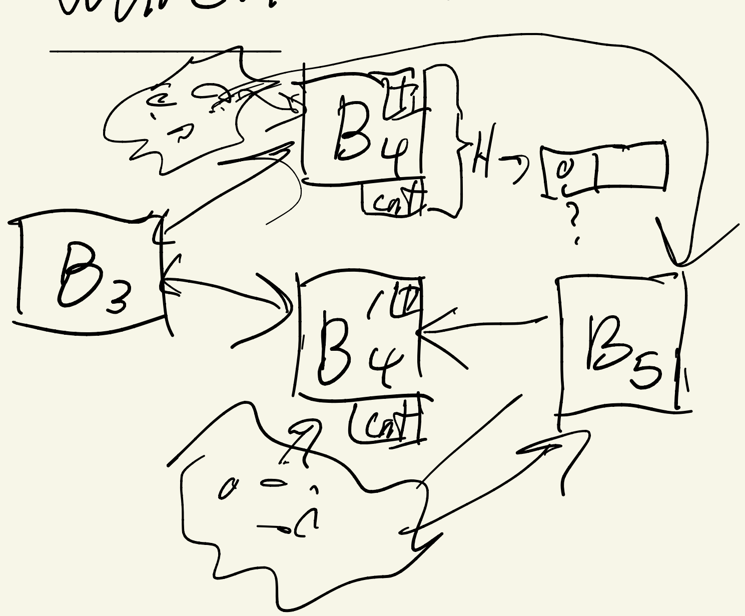
2. "Coinbase" - creates money

Key ideas:

- ledger publicly indicates "who owns what"
- miners rewarded for adding blocks
- hinges on history consensus
  - no double-spending

Bitcoin consensus

- who gets to add a block, on what basis?
  - almost all of real cost is from PoW
  - deliberate, artificial cost to creating/adding blks
- a (honest) miner only accepts valid blocks w/ PoW
  - deterministic agreed-upon validity function
- but which next valid block? (consensus)



preference for "more PoW effort"

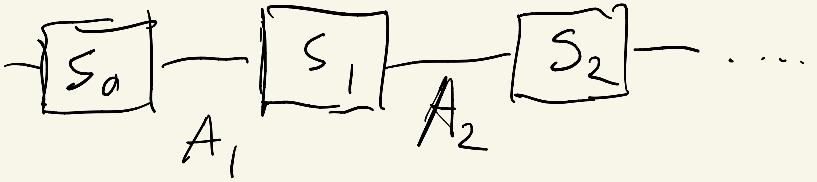
→ temporary inconsistencies

→ history rewriting (in last few blocks)

→ statistically, longest/heaviest always "wins" eventually

"51% attacks"

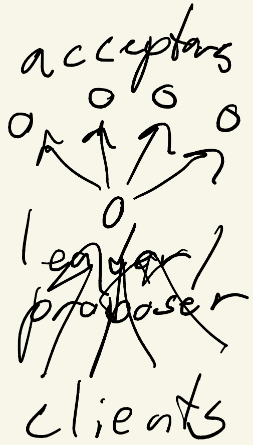
Permissioned consensus, on histories / ledgers



- "Multi-Paxos" – history
- Raft: re-formulation of Paxos
- PBFT: Castro/Liskov

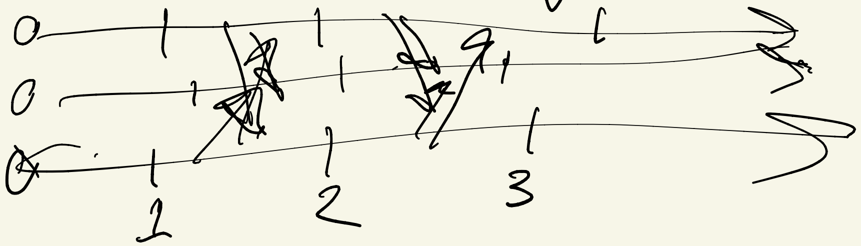
Key challenges:

- Pacing: when does a TX/blk get added?
  - Paxos/Raft/PBFT/...: leader-based
    what if leader fails? → synchrony assumptions
    → complexity of leader changes
    → DoS opportunities
  - Bitcoin: PoW: tuned for target of ~1B-10 mins
  - Asynchronous pacing / consensus:
    can always progress as fast as network communication permits?



Asynchronous pacing, consensus

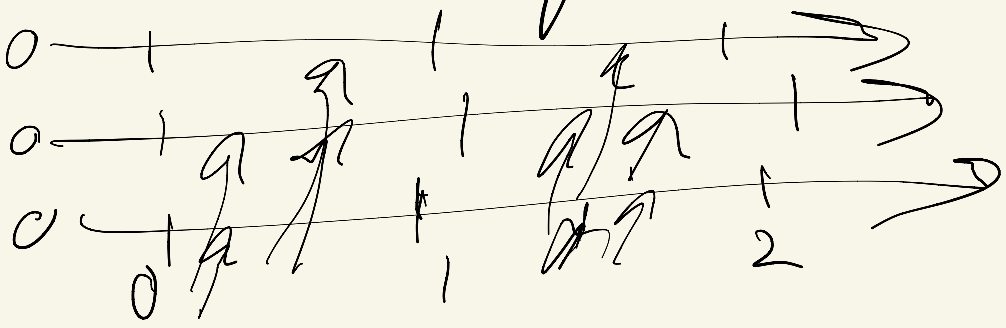
- Threshold Logical Clock (TLC)



- a single (integer) notion of "time-steps" across group
- paced: no node "get ahead" of (majority) of the rest

(vs Lampart clocks)

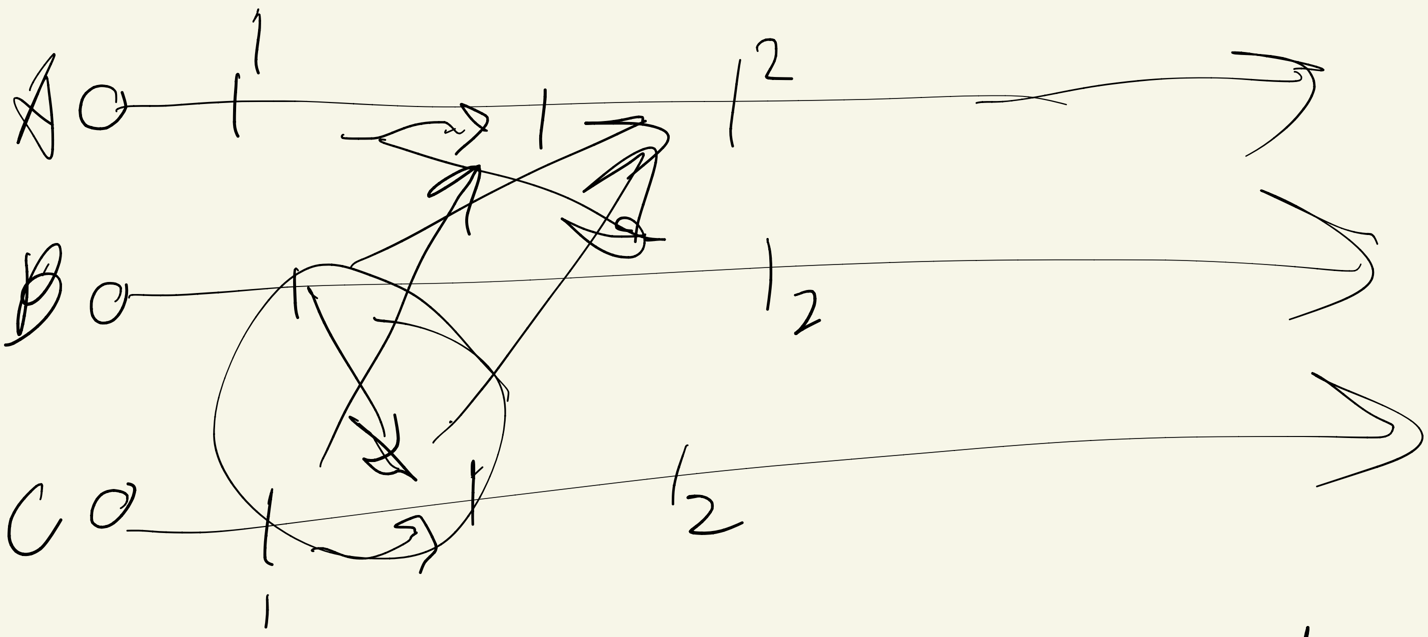
- HW3: using Paxos (example)



paxos - needs sychrony to achieve liveness

praper(s)
- prepare / reserve
- propose / commit

Asynchronous TLCs, "Que Sera Consensus" (QSC)



each node at step s waits to proceed to s+1:

- received updates from threshold t of other nodes
- at least t nodes have all received updates from at least t nodes

= I know a set $|M| \geq t$ that have been received by at least t nodes