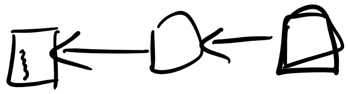
Permissionlessness:

\# Byzantine parties < 50%.

Fraction of Byzantine computatation power < 50%.



In expectation, next block is produced in 10 mins

$\Longrightarrow$ 20 CPUs. $\rightarrow$ 10 mins.

40 CPUs $\rightarrow$ 5 mins.

$\sim 2 \times 7 \times 24 \times 6$.

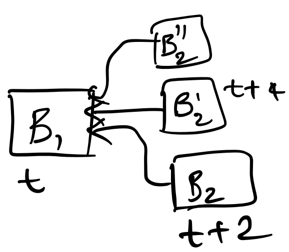
$\underbrace{000000}_{x+1}$ $\longmapsto$

Dynamic availability:

- 1 block in expectation every 10 mins $\Longleftarrow$ 12 secs, 2 secs
- The time to propagate a block is ~ 10 secs.

10 mins $\gg$ 10 secs.

Δ



Q: forking rate observed in Bitcoin vs. Ethereum.

Network assumption:

synchrony: any message sent by a party will arrive within some bounded delay Δ. (10secs).

Economic incentives: Block rewards: 6.25 BTC

$19K ≈ $120K

Transaction fees:

Network connectivity. Peer-to-peer network.

Communication complexity: every party is connected to every other party.

Adding block : $O(n^2)$.

Committing on block : $O(n^2 k)$

latency of a commit: k blocks (k confirmations)

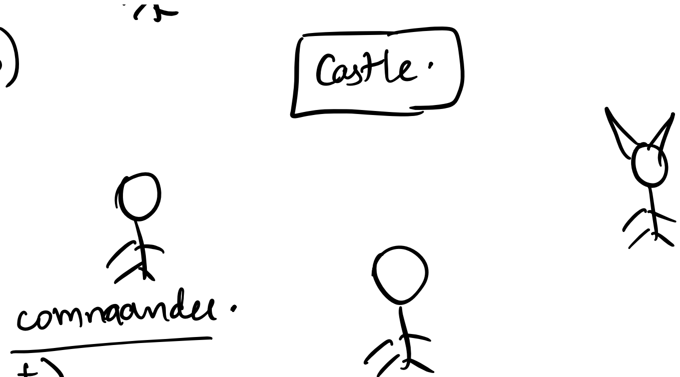
---

n parties, t are corrupt.

Lamport, Shostak, Pease 1980: Byzantine General Pease.

Byzantine

Broadcast (BB)



(Attack, retreat).

Agreement: No two honest generals take different actions.

Termination: Eventually every honest general to attack or to retreat.

Validity: If commander is honest, then all honest generals follow commander's input.

Byzantine agreement: every party$_i$ has an input v_i

Agreement: same as BB

Termination: same as BB.

Validity: if all honest parties have input $v_i = v$, then all honest parties output v.

Bitcoin: - general changes.
- validity:

BB	Bitcoin
- Only parties interested in	convincing an external

the result.

- Validity:
- Single-shot.

world.

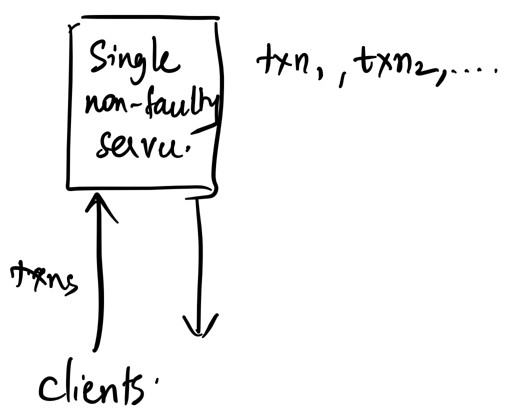
External validity.

Sequence/log of transaction

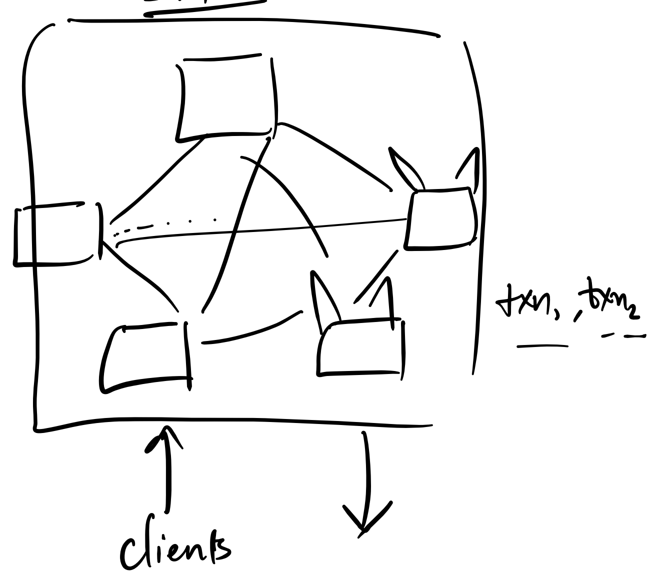
State machine replication:

Multiple server replicas (parties), some of which may be faulty, and they provide you with an interface of a single non-faulty server.

Ideal world



SMR



Safety: (Agreement) No two non-faulty server replicas output different values at the same position of the log.

Liveness: (Termination, validity). Non-faulty server replicas keep committing new client transactions (valid).

Dolev-Strong Protocol for Byzantine Broadcast (1982).

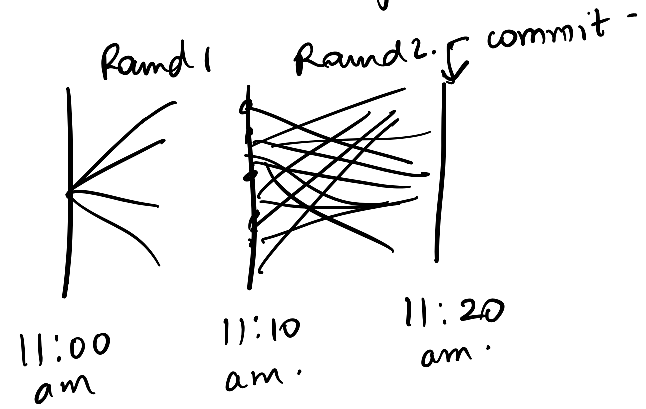
<u>Intuition:</u> If at any point, a party receives a value/txn, then that party will share it with everyone.

If I know something in round r, all other parties will know it in r+1.

Round 1: Commander (sender) sends some <u>value</u> v to all parties.

Rand 2: If I receive a value v in round 1, ~~share~~ send that value to all other parties.

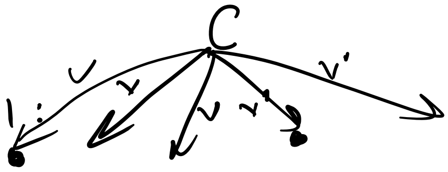
Commit: (Output) If I receive exactly one value v', then output v'. O.W. output $\perp$.



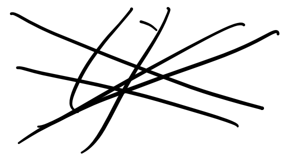
<u>Attack 1:</u> Commander does not send anything. ✗

Attack 2: Commander sends 2 different values

Round 1:



Round 2:



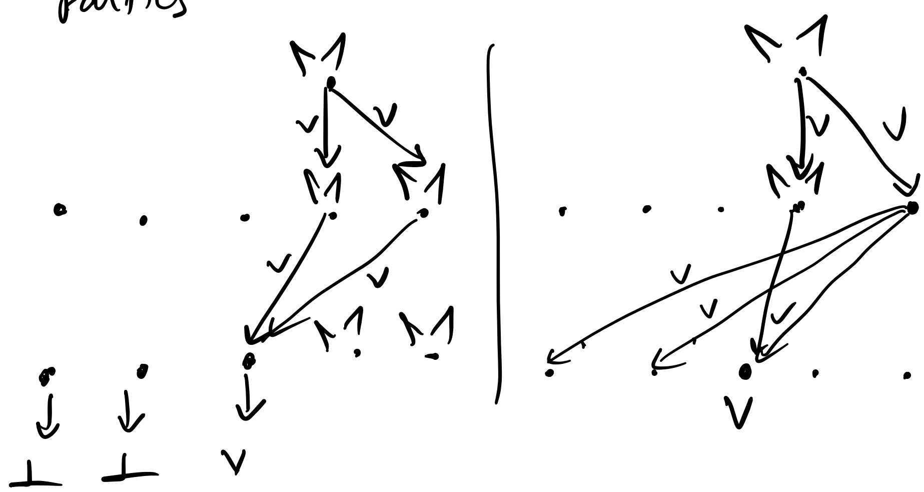
All honest see v & v'; output $\perp$.

Attack 3: Commander sends v to half the parties.

Attack 3': Commander sends nothing to honest parties; sends ~~me~~ value v to only malicious parties.

Round 1

Round 2



Signature chains: Parties: $P_1, \ldots P_n$

$\downarrow$

P_1 is the sender.

$$\langle \underbrace{\langle \underbrace{\langle v, 1 \rangle_{P_i}}_{\text{message,}}, 2 \rangle_{P_j}}_{\text{message.}}, 3 \rangle_{P_k} \ldots$$

$\langle x \rangle_{P_i}$: $\text{sign}_{P_i}(x)$.

What is a valid signature chain:

- in round i, the signature chain that I receive should have length i.
- the signers in this chain should be distinct.
- the signatures should be valid.

<u>Protocol: for party P_i</u>

Round 1: Commander P_1 sends $\langle v, 1 \rangle_{P_1}$ to all parties.

Round i: if I receive a valid round-i signature chain (m), $\langle m, i \rangle_{P_j}$ and send it to everyone.

$\langle\langle\langle \ldots \langle v, 1 \rangle \ldots \rangle\rangle\rangle$

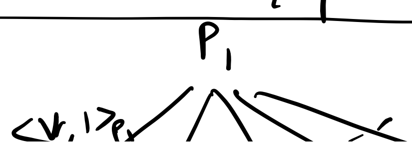
signature chain for value v.

for a signature chain for some value v, I will sign it at most once.

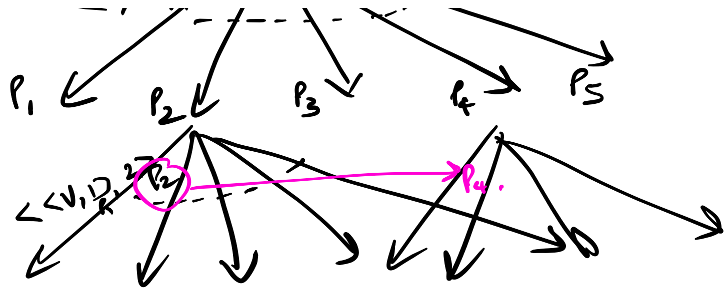
At the end of Round $t+1$: if a party receives exactly one valid signature chain for value v, output v.

o.w. o/p $\perp$/ ϕ.

All honest execution:



Round 1



Round 2

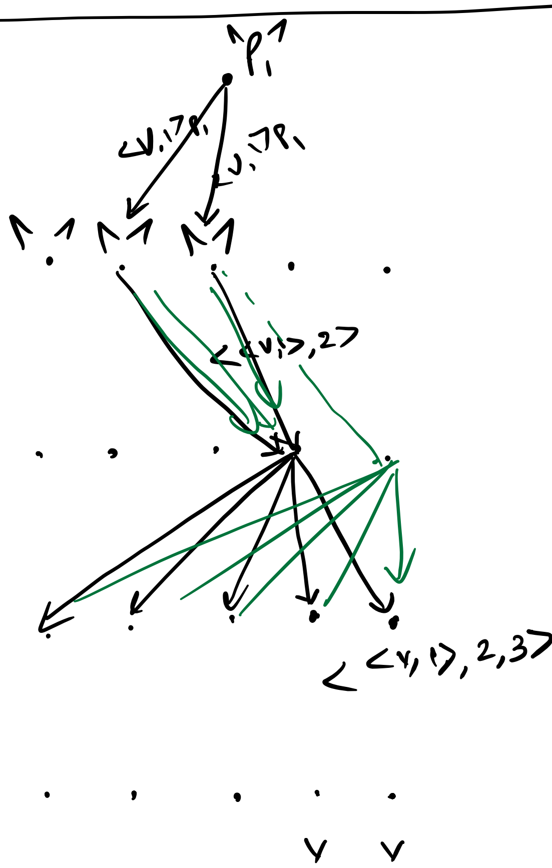
Round 3

Round 4:

---

all parties o/p v.

$t = n-2$



Round 1

Round 2

Round 3

Round 4

Proof:

Termination: t+1 rounds.

Validity :

Agreement: ~~For~~ every msg I receive :

Rounds $1 \dots t$: If a party P_j receives a valid signature chain for value v, then all parties receive a valid signature chain for v. in the next round.

Round $t+1$: the chain has length $t+1$.
$\exists$ some honest party P_h & P_h would have sent the message to everyone in an earlier round.

Latency: $t+1$ rounds.

Communication: For each value, (2).

Every party sends a signature chain to every party.
$O(n)$. $O(t)$. $O(n)$.

$O(n^2 t)$. signatures.

$O(n^2 k)$ ... (Bitcoin).

---

1. Are $O(t)$ rounds necessary? $\Longleftarrow$ UB/LB
2. Is $O(n^2 t)$ communication necessary? $\Longleftarrow$ UB/LB
$O(n^2)$

3. What happens if messages do not arrive in time (at the end of the round)? ⟸ ~~LB~~ UB/LB.