Permissionless:

- Byzantine parties < 50%.
- Fraction of Byzantine computation power < 50%.

\[
\begin{array}{c}
\square \rightarrow \bigcirc \rightarrow \square
\end{array}
\]

In expectation, next block is produced in 10 mins:

\[
\begin{array}{c}
20 \text{ CPUs} \rightarrow 10 \text{ mins}, \\
40 \text{ CPUs} \rightarrow 5 \text{ mins}, \\
\sim 2 \times 7 \times 24 \times 6
\end{array}
\]

\[
\frac{0.000001}{a+1}
\]

Dynamic availability:

- 1 block in expectation every 10 mins \(12 \text{ secs} / 2 \text{ sec}\)
- The time to propagate a block is \(\sim 10 \text{ secs}\).

Q: Forking rate observed in Bitcoin vs. Ethereum.
Network assumption:
- Synchrony: any message sent by a party will arrive within some bounded delay $\Delta$ (10sec).

Economic incentives:
- Block rewards: 6.25 BTC $\approx$ $19K \approx $120K

Transaction fees:

Network connectivity: Peer-to-peer network.

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

\[
\begin{align*}
\text{Adding block:} & \quad O(n^2) \\
\text{Committing a block:} & \quad O(n^2-k) \\
\text{Latency of a commit:} & \quad k \text{ blocks} \quad (k \text{ confirmation})
\end{align*}
\]

$n$ parties, $t$ are corrupt.

Lamport, Shostak, Pease 1980: Byzantine General Problem.
Broadcast \((BB)\) → Castle

(commander)

(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:

<table>
<thead>
<tr>
<th>BB</th>
<th>Bitcoin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only parties interested in convincing an external real world.</td>
<td></td>
</tr>
</tbody>
</table>
the result:

- Validity:
- Single-shot

State machine replication:

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

Ideal world:

```
| Single non-faulty server |
```

transactions:

```
| clients |
```

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

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

K wουνa:

External validity.

Sequence/log of transactions.
Dolev–Strong Protocol for Byzantine Broadcast
(1982).

Intentions: If at any point, a party receives a value \( v \), then that party will share it with everyone.

If I know something in round 9, all other parties will know it in \( r+1 \).

- **Round 1:** Commander (sender) sends some value \( v \) to all parties.

- **Round 2:** If I receive a value \( v \) in round 1, send that value to all other parties.

- **Commit:** If I receive exactly one value \( v' \), then output \( v' \).
  Otherwise, output 1.

**Dashboard Diagram:**

```
11:00 am  11:10 am  11:20 am
```

- **Attack 1:** Commander does not send anything.
Attack 2: Commander sends 2 different values:

Round 1:

All honest see v & v'; output 1.

Attack 3: Commander sends v to half the parties.

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

Round 1

Round 2

\[ v \to M \quad M \to v \quad M \to \quad M \]

Signature chains:

Parties: \[ P_1 \ldots P_n \]

\[ \langle \langle v, 1 >, P_j, 2 >, P_j, 3 >, P_k \ldots \]

message;

message.
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.

Protocol: for party $P_i$

Round 1: Commander $P_i$ sends $<v, i>_{P_i}$ to all parties.

Round i: if I receive a valid round i signature chain $(m)$, $<m, i>_{P_j}$ and send it to everyone:

$<<<<<v, i>...>>>$

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. 0\slash p + 1/0.

All honest execution: $P_i$
Proof:
Termination: $t+1$ rounds.
Validity:

Agreement: For every msg I receive:

Rounds 1...t: If a party pj receives a valid signature chain for value v, then all parties receive a valid signature chain for v on the next round.

Round t+1: the chain has length t+1.
If some honest party ph & ph 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.

\( O(n^2 t) \) - (Bitcoin).

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

\( O(n^2) \)
3. What happens if messages do not arrive in time (at the end of the round)? \( \leq UB/LB \).