Classical Consensus

Benedikt Bünz
Blockchain Layers

Layer 1: consensus layer

Layer 1.5: compute layer (blockchain computer)

Layer 2: applications (DAPPs, smart contracts)

Layer 3: user facing tools (cloud servers)
Blockchain Forks

\[ \text{TX}_A: \text{Send 3 BTC to Bob} \]

\[ \text{TX}_A: \text{Send 3 BTC to herself} \]
Double Spending

Alice can create two transactions spending the same UTXO!

- One sends money to Bob, the other sends the UTXO to herself.
- Only the ‘first’ transaction should go through
- -> There needs to be a global consensus on the ordering of transactions.
- Concretely, there needs to be an agreement which block extends the blockchain (Fork Choice Problem)
Block choice

New Block

Reject?

Accept

Reject

Block A

404
Byzantine Generals Problem

Block choice is equivalent to BGP
Leader gets an input bit 0/1

Every round each node sends messages to every other general. Messages are received in the next round.

At the end of the protocol honest nodes output a bit or abort.
Honest generals follow the protocol. Malicious generals behave arbitrarily.
Byzantine Fault Tolerant Protocol (BFT)

**Consistency**
If two honest nodes output b and b' respectively, then b = b'.

**Validity**
If the leader is honest and receives input b then all honest nodes output b.
1. Leader sends $b$ to all nodes
2. All nodes forward received bit to all other nodes (Voting)
3. Each node tallies votes (including its own vote) and outputs majority bit

Broken by corrupt leader
Dolev Strong Protocol

Maximum $f$ corrupt nodes, input message $m$

1. Leader sends $m$ to all nodes
2. For $r = 1$ to $f + 1$
   1. If you received an unseen message $m$ signed by $r$ signatures (including leader) sign $m$ and send to all. Set $S \leftarrow S \cup \{m\}$
   2. Otherwise remain silent
3. If $|S| = 1$ output $m \in S$ otherwise output “Confused” (or default message)
Dolev Strong Example

f=2

Brutus

Marc Anthony

Pompeius

Augustus
Dolev Strong Example

f=2
r=1

Brutus

Marc Anthony

Pompeius

Augustus
Dolev Strong Example

- $f=2$
- $r=2$

- Brutus
- Pompeius
- Marc Anthony
- Augustus

$0_{\text{Brutus, Pompeius}}$

Attack $= 1$

- $1_{\text{Caesar, Aug, MA}}$

- $1_{\text{Caesar, MA, Aug}}$

REJECTED
Dolev Strong Example

\[ f = 2 \]
\[ r = 3 \]

- Brutus
- Marc Anthony
- Pompeius
- Augustus

\[ \text{Attack} = 1 \]

\[ 1_{\text{caesar, Aug, MA}} \]

\[ 1_{\text{caesar, MA, Aug}} \]
Dolev Strong Example

f=2
r=3

Brutus

Marc Anthony

Pompeius

Augustus

1_{\text{caesar, Aug, MA}}

1_{\text{caesar, MA, Aug}}
More than f corruptions

\[ f = 2 \]
\[ r = 3 \]
More than $f$ corruptions

$f = 2$
$r = 3$

Confused

Brutus

Marc Anthony

Pompeius

Augustus

0_{Caesar, Brutus, Pompeius}
1_{Caesar, Aug, MA}
1_{Caesar, MA, Aug}
Dolev Strong Analysis

Why $f+1$ rounds?
f corrupt nodes can confuse honest node

Validity?
Honest nodes only update set $S$ if signed by leader

Consistency?
1. If honest node has $m \in S$ at round $r \leq f$ then all other nodes will have $m \in S$ at $r + 1$
2. If honest node receives new $m$ at round $f + 1$ then it must have received it from an honest node
3. -> All honest nodes have identical $S$
In a blockchain we solve a Byzantine General’s Problem for every block.
This is called an iterated BGP
Sybil Resistance

In BC participants are fixed but how are they selected?

Two variants:

*Permissioned*: Nodes are fixed

*Permissionless*: Anyone can participate
Permissioned Consensus
Proof of Stake

Weighted Byzantine Consensus

Assumption 2/3\textsuperscript{rd} of stake with honest nodes

How to initialize?
Incentives?
More in 2 lectures

3 ETH
1 ETH
2 ETH
7 ETH
5 ETH
Permissionless Proof of Work

Recall: $H(x, y) < \frac{2^n}{D}$

- 3 TH/s
- 2 TH/s
- 5 TH/s
- 5 TH/s
- 7 TH/s

Terrible for the environment

Truly permissionless

More next lecture
Network Model

- Dolev Strong assumes messages get delivered by next round
  - Not realistic (honest nodes can have network outages)
  - Protocol broken if messages aren’t delivered in time
Network Model

- **Synchronous**: There is known maximum delay $\Delta$ such that any message sent from one node to another is delivered within $\Delta$ time.

  - Protocol *can* use $\Delta$ as parameter

- **Partially Synchronous**: $\Delta$ exists but is unknown

  - Same protocol must work for any $\Delta$
  - Equivalent definition: There exists periods of synchrony in which delay is $\Delta$. Protocol does not know when these begin

- **Asynchronous**: Network experiences arbitrary failures

  - Consensus problem unsolvable
Blockchain Consensus

• ”State Machine Replication” on n nodes (or servers)
• Stream of transactions $tx_1, tx_2, \ldots$
• For $i = 1, \ldots, n$: $L_i(t)$ is a list of confirmed Tx by node $i$ at time $t$

• **Goal:** Protocol that satisfies two properties:
  ✓ Nodes confirmed transactions are consistent with each other
  ✓ Transactions will eventually get confirmed
Blockchain Consensus

**Consistency**

For all honest nodes $i, j \in [n]$ and times $t, t'$:
Either list $L_i(t)$ is a prefix of $L_j(t')$ or vice versa

**$\Delta$ – Liveness**

There exists function $T$ such that:
If any honest node receives $tx$ at time $t$ then $\forall i \ tx \in L_i(t + T(\Delta, n))$. At time $t + T(\Delta, n)$ $tx$ is finalized

$\Delta = \text{maximum network delay}$
Epoch t

\[ \mathcal{L}_1(t) \]

\[ \mathcal{L}_2(t) \]

\[ \mathcal{L}_3(t) \]

\[ \mathcal{L}_4(t) \]

\[ \mathcal{L}_5(t) \]

\[ S = \{ t_{x_k}, \ldots, t_{x_l} \} \]

s.t. \( t_{x_k}, \ldots, t_{x_l} \notin \mathcal{L}_1(t) \)

“S” is a new block
Block chain from Byzantine Consensus

Epoch $t+1$

$L_1(t + 1) = L_1(t) || S$

$L_2(t + 1) = L_2(t) \cup S$

$L_5(t + 1) = L_5(t) \cup S$

$L_3(t + 1) = L_3(t) \cup S$

$L_4(t + 1) = L_4(t) \cup S$
*Blockchain from Byzantine Consensus*

**Epoch t+1**

\[ L_1(t + 1) \]

\[ L_2(t + 1) \]

\[ L_3(t + 1) \]

\[ L_4(t + 1) \]

\[ L_5(t + 1) \]

Dolev Strong can take f+1 rounds
Dolev Strong is synchronous
Can we built something better?

Rotating leader
Streamlet: A simple Blockchain protocol

Assumptions:
- $n$ nodes (permissioned)
- Less than $1/3$ corrupt
- Partially synchronous network
- Proceed in epochs

Random rotating leader:
Leader id = $H($epoch$)$ mod $n$

Each node stores locally notarized chain
Streamlet [Chan,Shi20]

**Propose Vote** In every epoch:

1. Leader creates block of TXs extending *longest* local *notarized* chain
2. Nodes sign off on first block from leader iff it extends one of their longest local *notarized* chain
3. If *any* Block has signatures from 2n/3 nodes it becomes *notarized* (Can be from a prior epoch)

**Finalize**

1. If a chain has 3 notarized blocks from consecutive epochs, chop off the final block and *finalize* the chain
Streamlet: A simple Blockchain protocol

Assumptions:
Less than 1/3 corrupt
Partially synchronous network
Proceed in epochs

Random rotating leader:
Leader id= H(epoch) mod n

Each node stores locally notarized chain
Streamlet: A simple Blockchain protocol

Assumptions:
Less than 1/3 corrupt
Partially synchronous network
Proceed in epochs

Random rotating leader:
Leader id = H(epoch) mod n

Each node stores locally notarized chain
Streamlet: A simple Blockchain protocol

Assumptions:
Less than 1/3 corrupt
Partially synchronous network
Proceed in epochs

2n/3 sigs. -> notarized

Random rotating leader:
Leader id = H(epoch) mod n

Each node stores locally notarized chain
Streamlet: A simple Blockchain protocol

Assumptions:
- Less than 1/3 corrupt
- Partially synchronous network
- Proceed in epochs

Random rotating leader:
Leader id = H(epoch) mod n

Each node stores locally notarized chain
Streamlet: A simple Blockchain protocol

Assumptions:
Less than 1/3 corrupt
Partially synchronous network
Proceed in epochs

Random rotating leader:
Leader id = H(epoch) mod n

No other block on level 6 can be notarized

Each node stores locally notarized chain
Streamlet: Consistency Analysis

1. No two blocks with same epoch can be notarized (2/3 majority)
2. If X<5 then more than 1/3 honest nodes voted on 3. These nodes would never notarize 5 (because 5 doesn’t extend 3). Without these 1/3+1 nodes 5 can’t get notarized (Contradiction)
3. If X>7 more than 1/3 honest nodes have notarized 6. They won’t notarize X because it doesn’t extend 6

No other block on level 6 can be notarized.

Consistency holds irrespective of network
Next lecture: Nakamoto Consensus, Incentives, Large Scale Consensus