Consensus

Ronghui Gu

Fall 2024

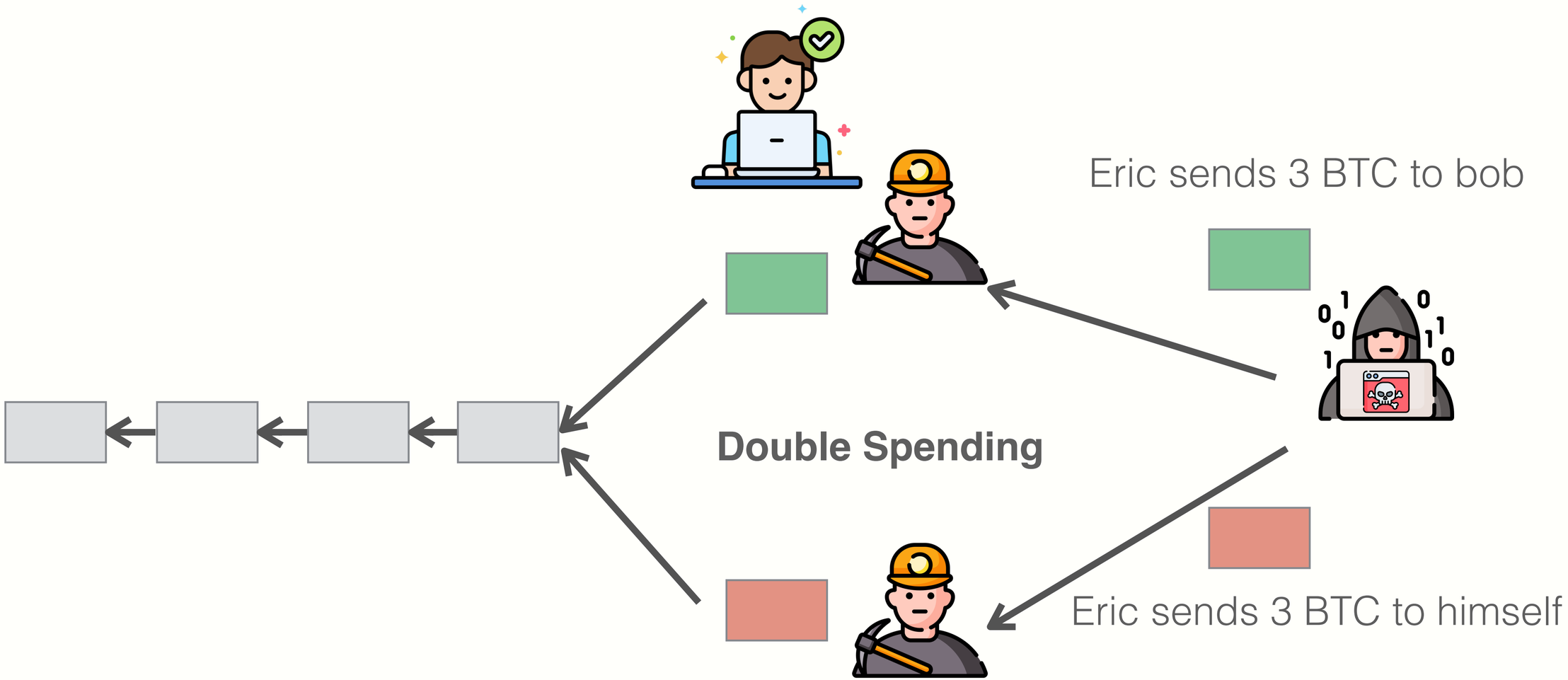
Columbia University

Course website: https://verigu.github.io/6998Fall2024/

Preliminary Project Proposal

- Describe roles of each team member
- Briefly describe the project that you plan to implement
- **1 page**
- Submit via coursework
- Due: Sept 19

Blockchain Forks



Properties

There needs to be a global consensus on the ordering of TX

- Concretely, there needs to be a global consensus on which block extends the blockchain (**Block Choice**).

- Bob sends car only after confirming the right TX on blockchain

Block choice is challenging to solve

Leader

New Block!

Accept!

404

Reject!

How to achieve consensus (informal)

Model: network and adversary

- Network delay or partition
- Composition of committee

Goal: Honest participants follow a consensus protocol to achieve

- **Consistency**: honest nodes agree
- **Liveness**: system makes progress

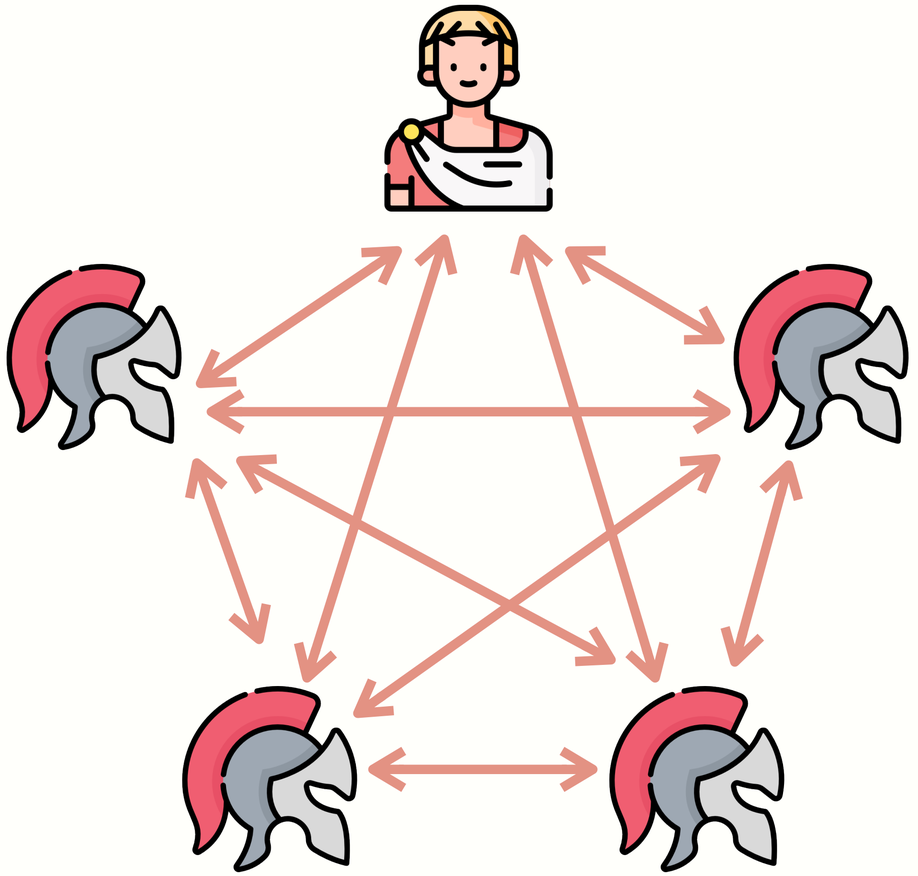
1 Byzantine Fault Tolerant Protocol (BFT)

Byzantine Generals Problem

Leader: gets an input bit 0/1

Every round: each node send messages to every other node.

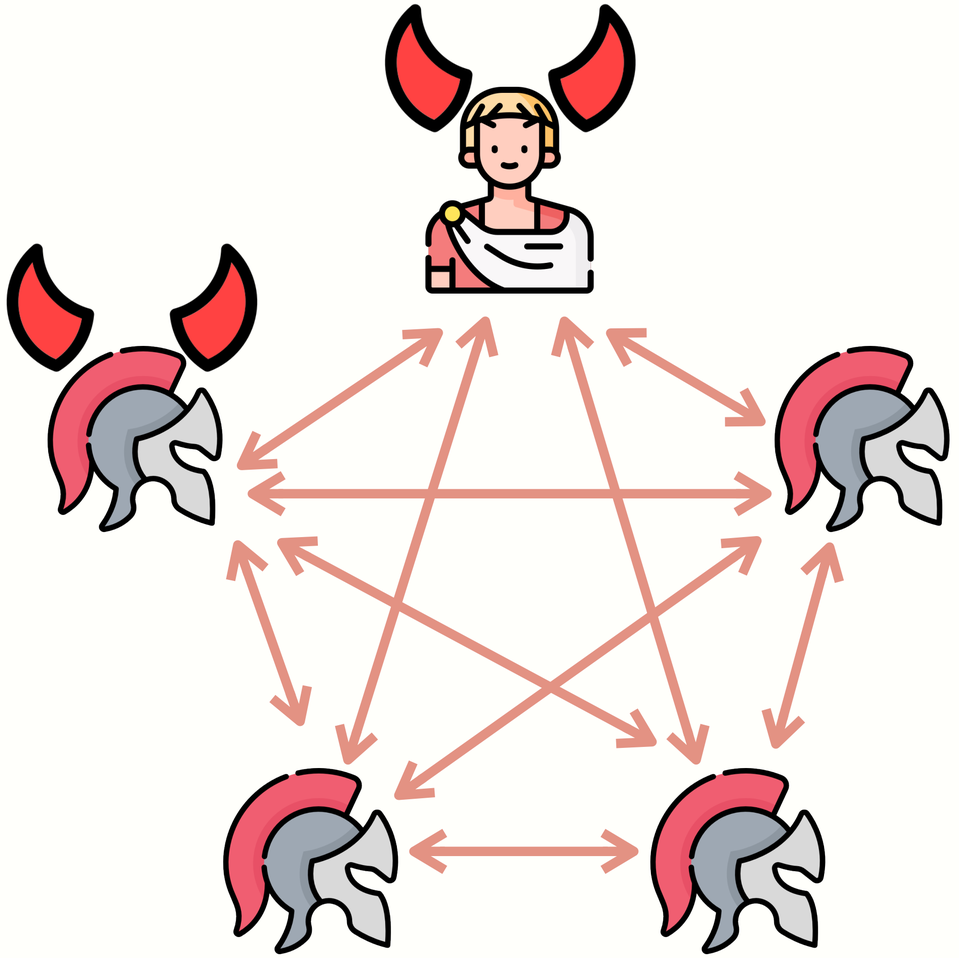
End: honest nodes output a bit or abort



Byzantine Generals Problem

Honest nodes: follow the protocol

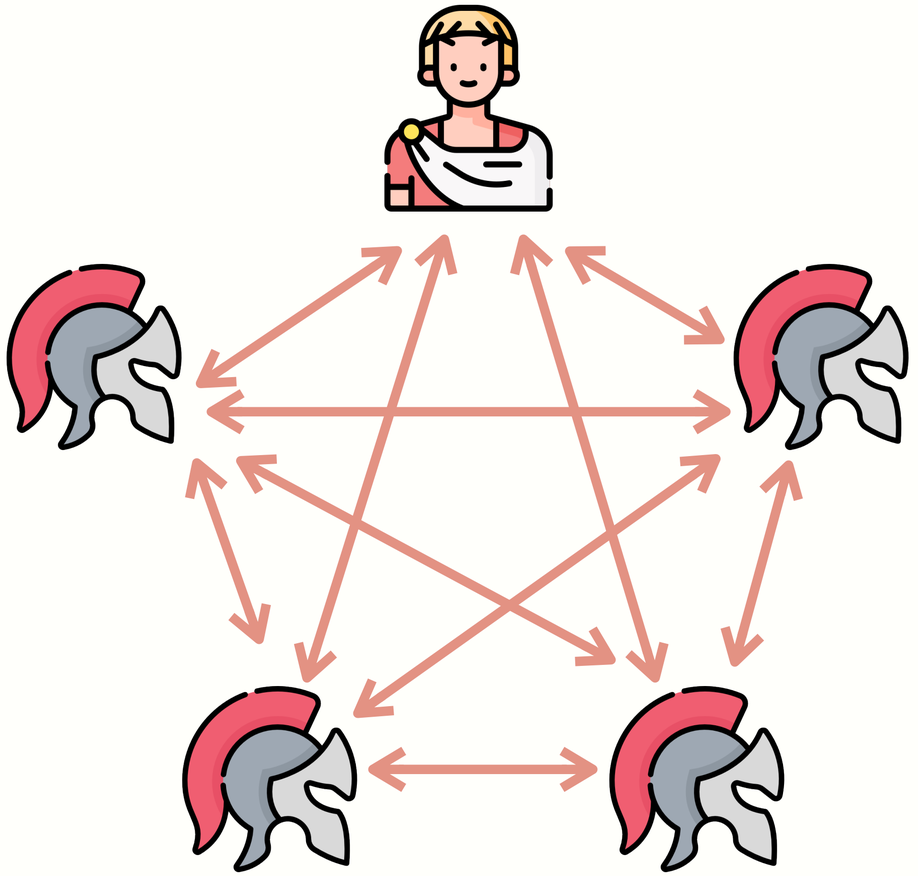
Malicious nodes: behave arbitrarily



Byzantine Fault Tolerant Protocol (BFT)

Consistency: if two honest nodes output b and b', b = b'

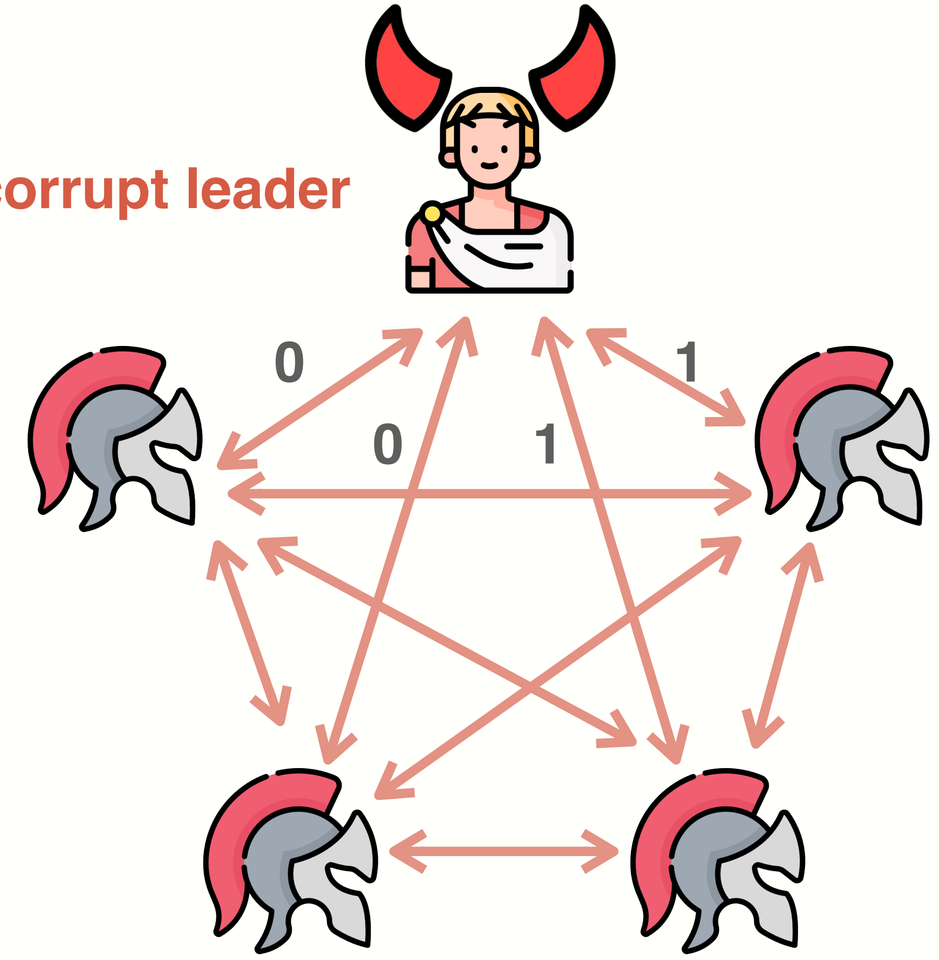
Validity:if the leader is honest and receives input b, all honest nodes output b.



(Flawed) Voting Protocol

- **Leader** sends b to all nodes
- All nodes forward received bit to all other nodes (Voting)
- Each node tallies votes (including its own vote) and outputs **majority** bit

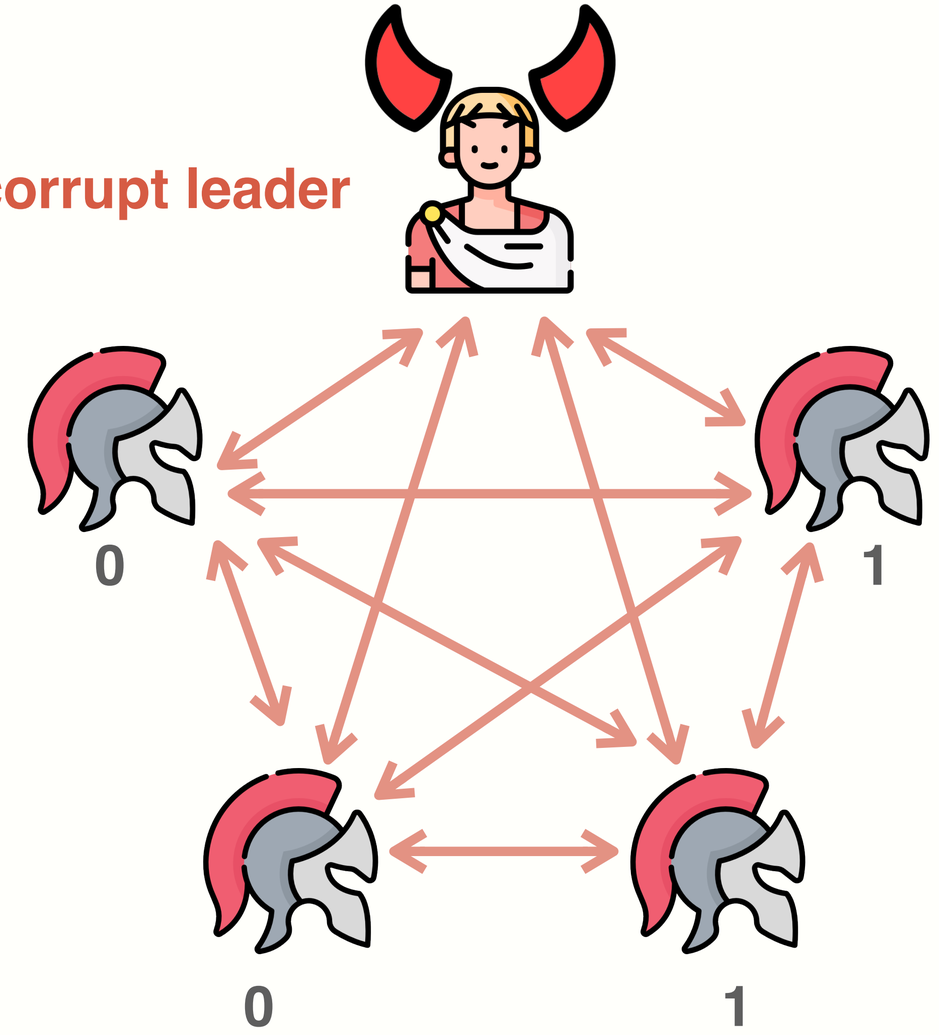
Broken by corrupt leader



(Flawed) Voting Protocol

- **Leader** sends b to all nodes
- All nodes forward received bit to all other nodes (Voting)
- Each node tallies votes (including its own vote) and outputs **majority** bit

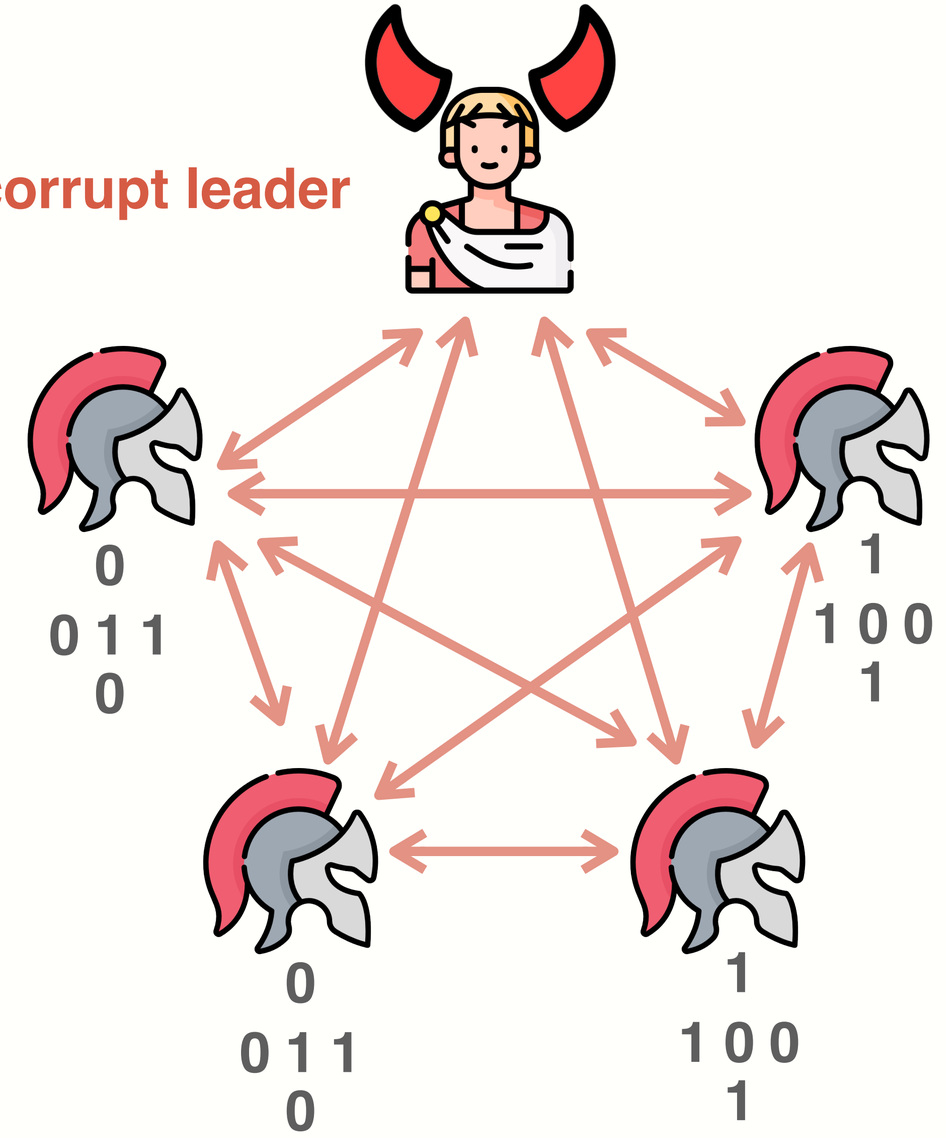
Broken by corrupt leader



(Flawed) Voting Protocol

- **Leader** sends b to all nodes
- All nodes forward received bit to all other nodes (Voting)
- Each node tallies votes (including its own vote) and outputs **majority** bit

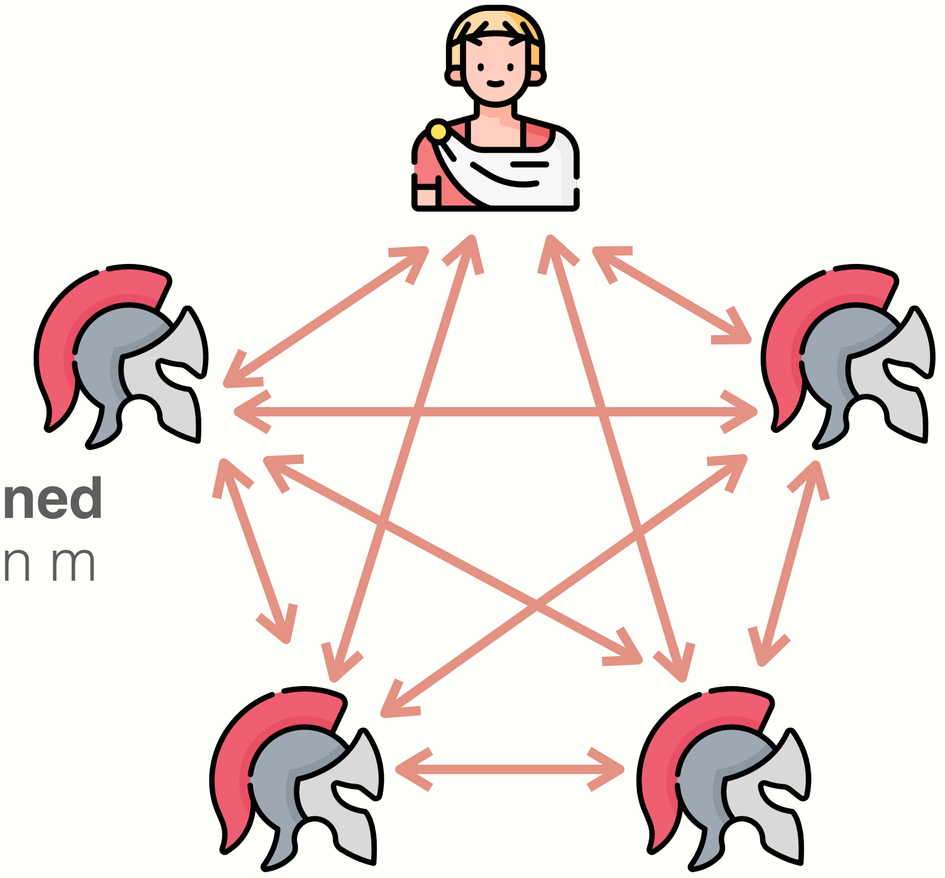
Broken by corrupt leader



Dolev Strong Protocol

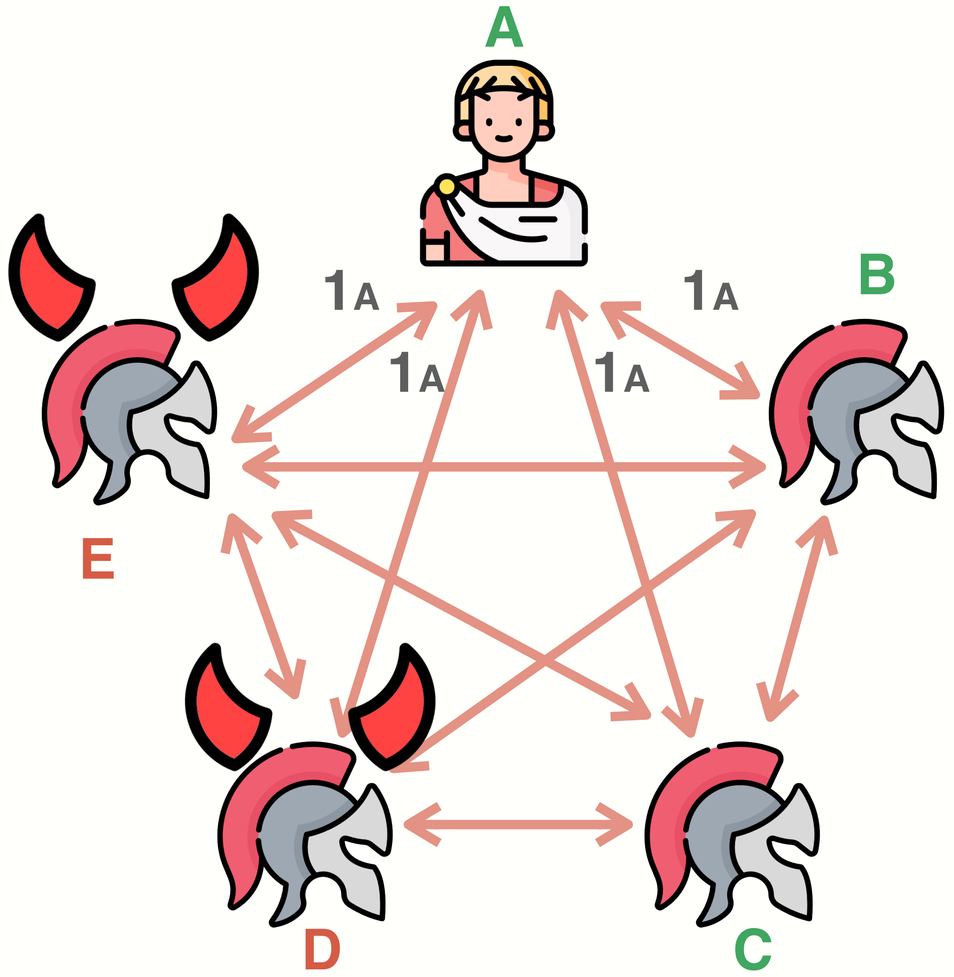
Max F corrupt nodes (need F+1 round)

- **Leader** sends m to all nodes
- For R = 1 to F + 1
  - If you received an unseen message m **signed** by **R** signatures (including the leader), sign m and send to all. S <- S U { m }.
  - Otherwise, remain silent
- If |S| = 1, output m in S; otherwise, output “confused”



Example

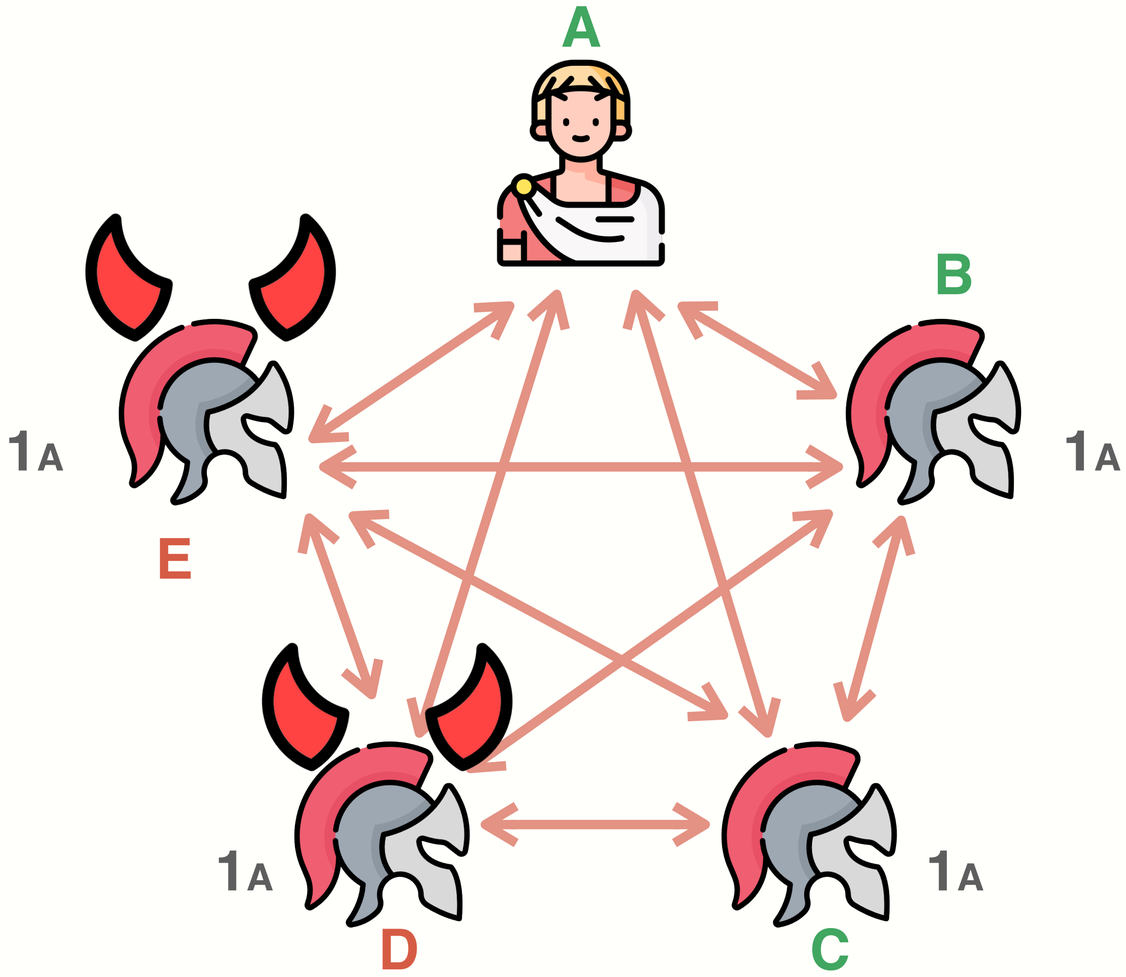
F = 2



Example

F = 2

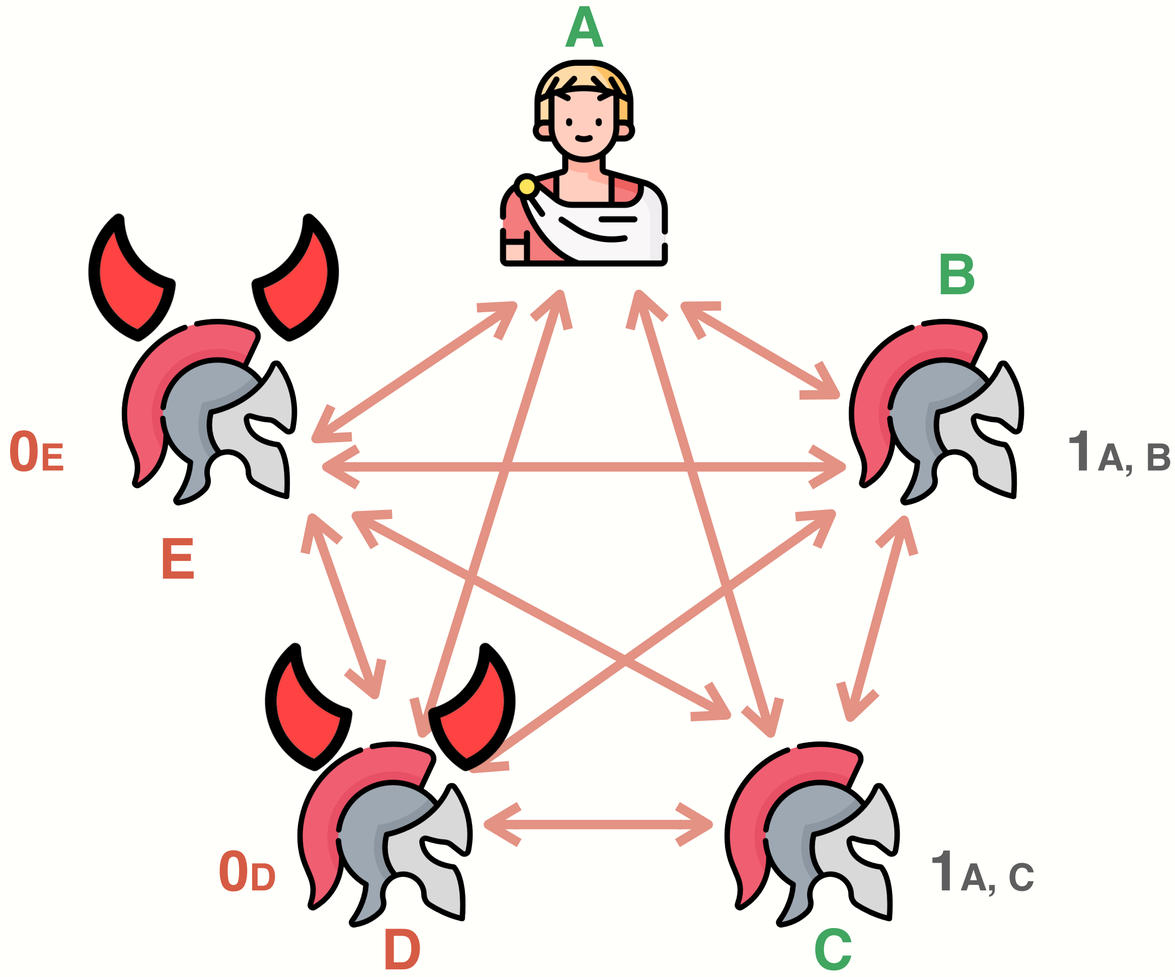
R = 1



Example

F = 2

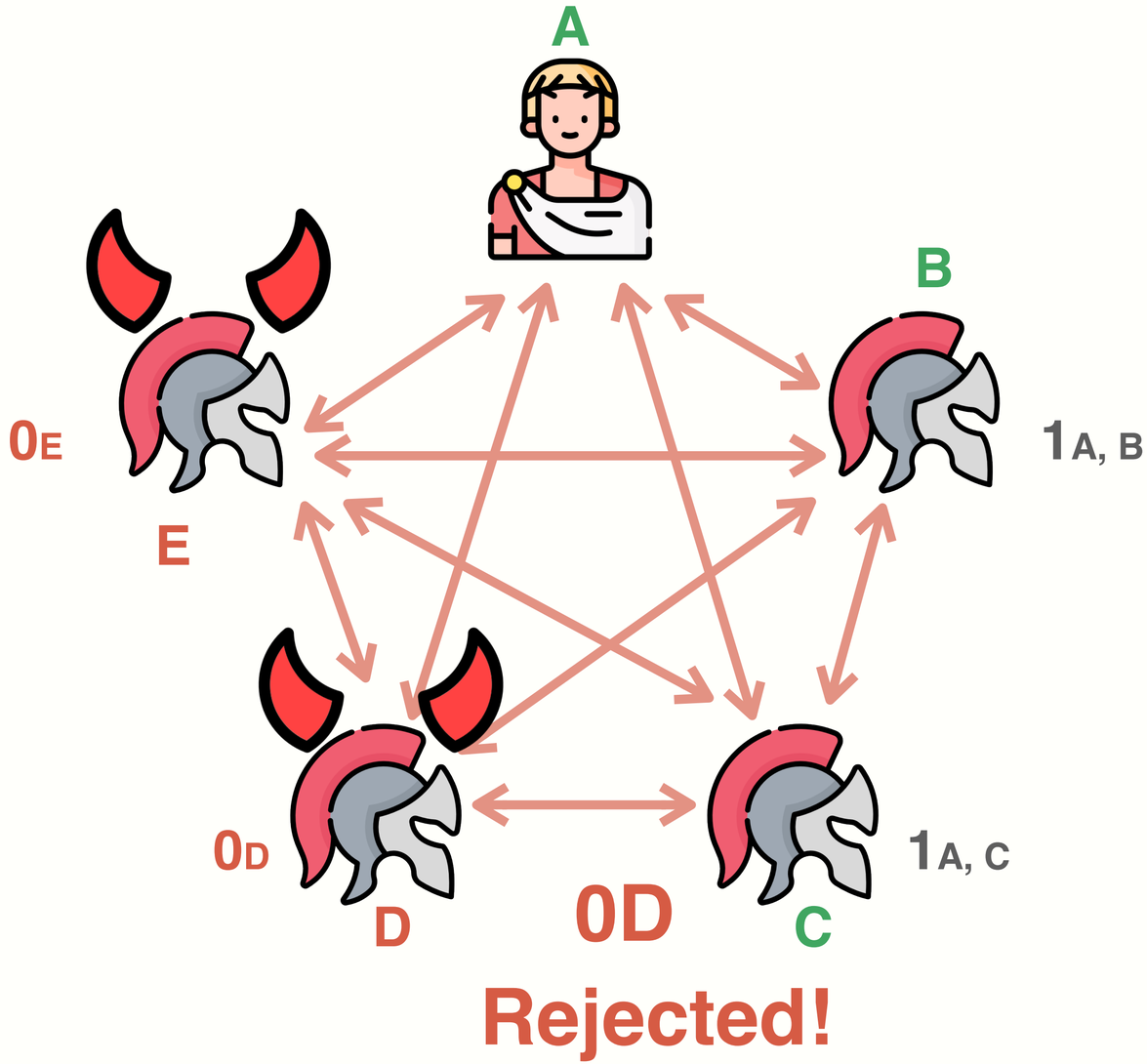
R = 1



Example

F = 2

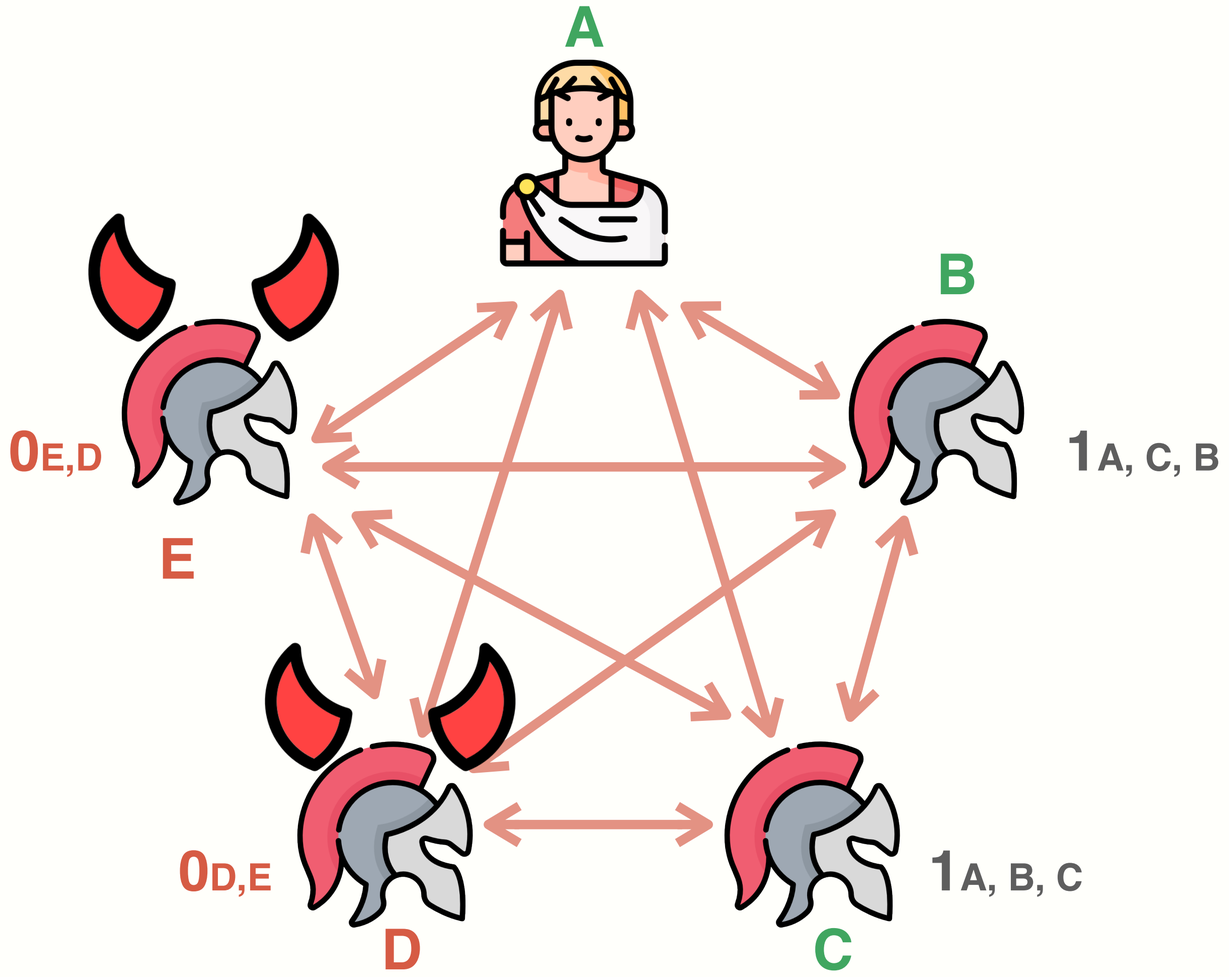
R = 2



Example

F = 2

R = 3

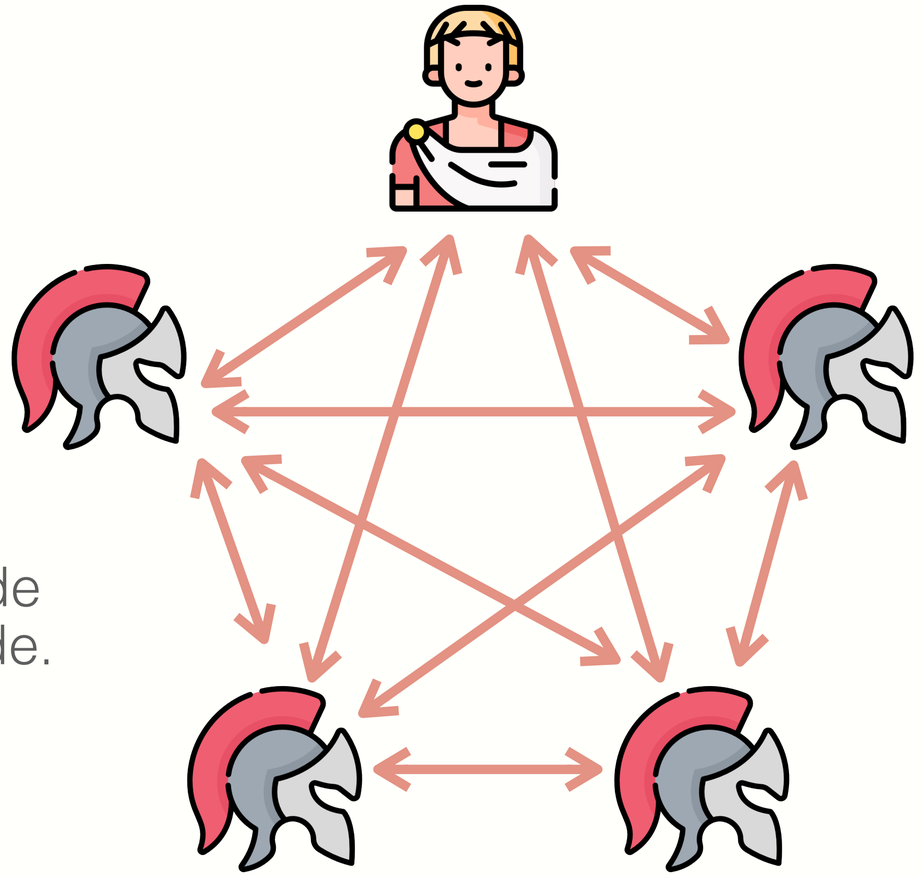


Properties

Consistency?

- If an honest node has m at R <= F, all other nodes will have m at R+1
- If an honest node receives m at R = F+1
  - Then m has F+1 signatures
  - Then m has been received by an honest node before, since there are at most F corrupt node.
  - Then all honest nodes had m

Validity?



Blockchain from Byzantine Consensus

- **For each block:** use BFT to reach consensus
- **For the next block:** select a new leader (how?)
- **Cost**: F+1 rounds to reach consensus for each block

2 Nakamoto Consensus

Nakamoto Consensus

One person/ID, one vote

- **Trusted** identity is hard in a decentralized system
- **Sybil Attack**

Nakamoto Consensus

One CPU, one vote

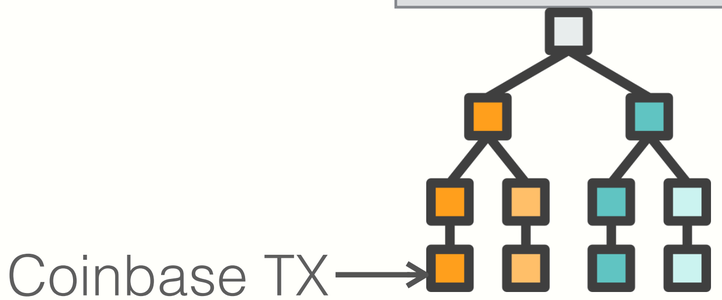
- **Sleepy mode:** (honest) nodes may come online or offline at any point
- **Sybil resistance:** costs computing resources to become leader

Nakamoto Consensus

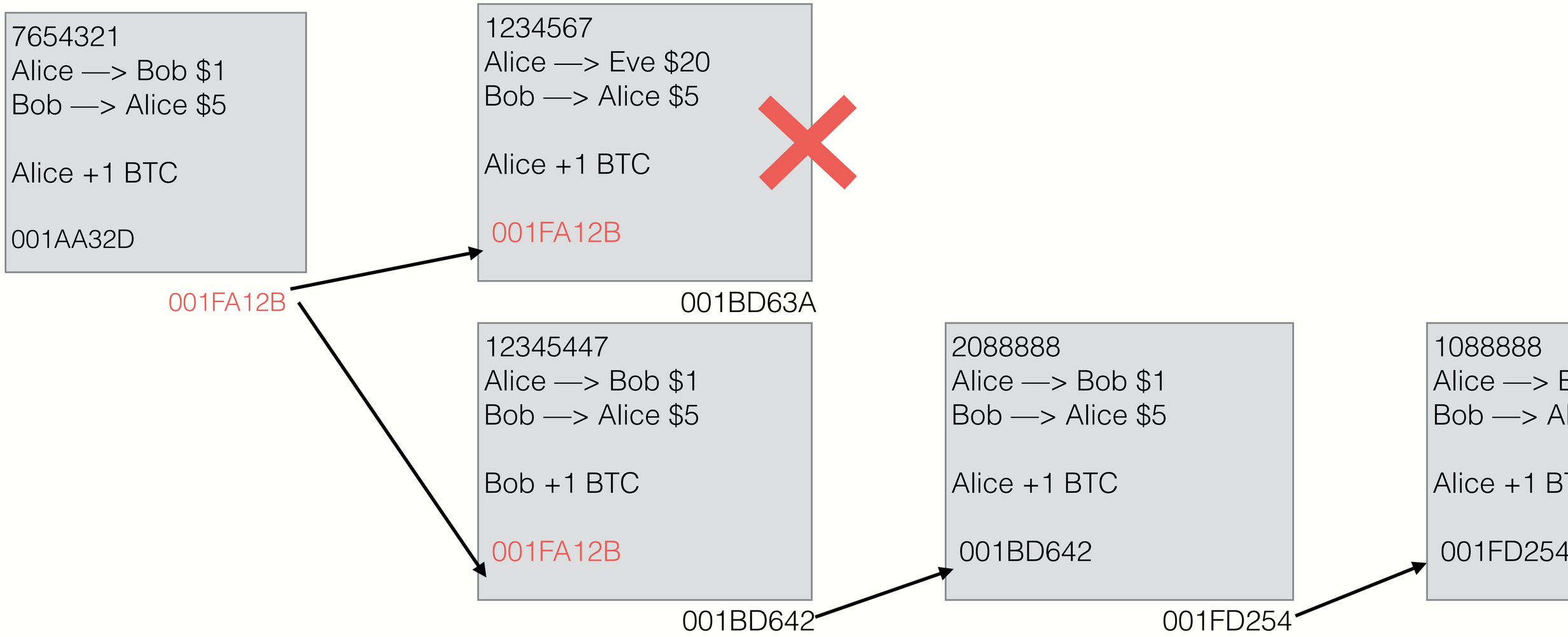
Hash(version, prev, time, **bits**, **nonce**, Tx root) = 001BD63A

< 001FFFFF

Indicated by "**bits**"



Nakamoto Consensus



Nakamoto Consensus

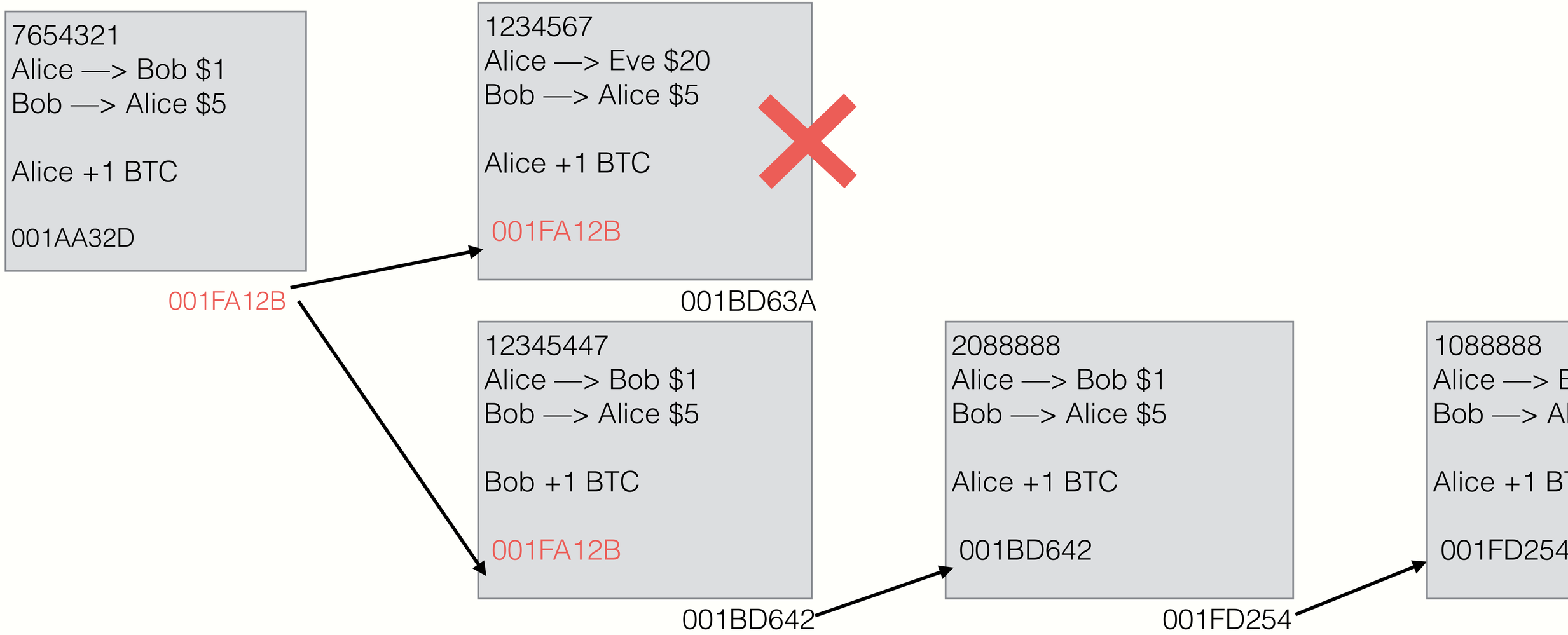
Miners “race” to add blocks

- Prepare Block Template
- Find **nonce** (PoW solution)
- One winner every ~**10 min**
- Target adjusted every **2016 blocks** (**2 weeks)**
- Probability winning ~ Computation power

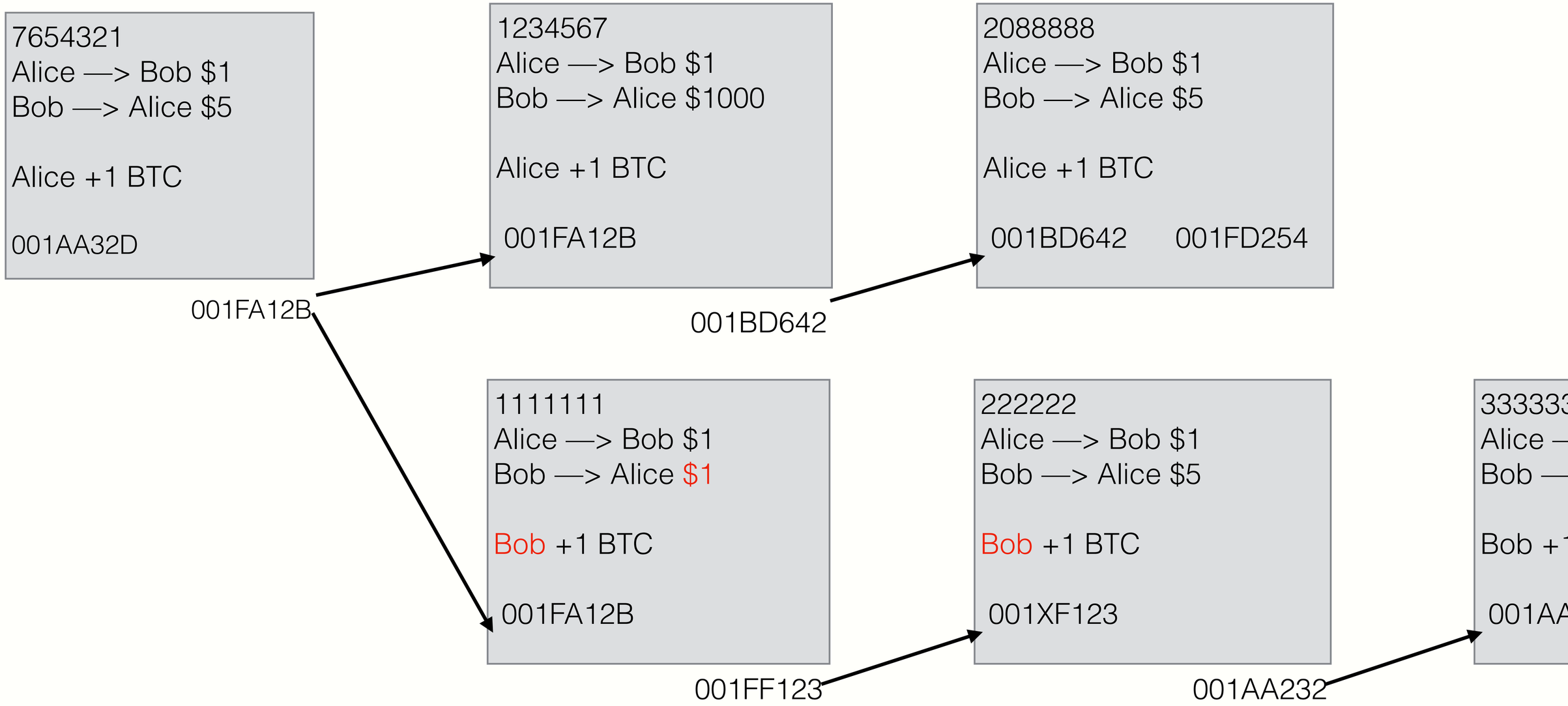
(Honest) miners extend longest chain

Blocks/Transactions become “final" after K blocks

Preventing Double Spends



51% Attack


7654321
Alice —> Bob $1
Bob —> Alice $5
Alice +1 BTC
001AA32D
001FA12B
1234567
Alice —> Bob $1
Bob —> Alice $1000
Alice +1 BTC
001FA12B
001BD642
2088888
Alice —> Bob $1
Bob —> Alice $5
Alice +1 BTC
001BD642 001FD254
1111111
Alice —> Bob $1
Bob —> Alice $1
Bob +1 BTC
001FA12B
001FF123
222222
Alice —> Bob $1
Bob —> Alice $5
Bob +1 BTC
001XF123
001AA232

Nakamoto Consensus Properties

Consistency: honest nodes agree on all but last **k** blocks

Chain quality: miners controlling p fraction of power should roughly mine p fraction of blocks

Chain growth: chain grows at a steady rate

Nakamoto Consensus Properties

Pros:

- Anonymous participation
- Nodes can join/leave (very scalable)
- Leader not known beforehand
- Up to ½ corruptions

Cons:

- Slow
  - Even when everyone is honest
- Resource intensive
- No finality

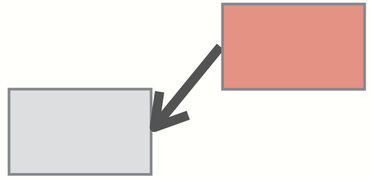
Incentives

- Mining (solving PoW puzzles) is very expensive
  - Honest majority does not seem realistic
- Satoshi's genius idea: combine issuance and rewards
- Block reward only paid if block part of longest chain (High Variance -> Mining Pools)
  - Large opportunity cost for unsuccessful attacks

Block rewards

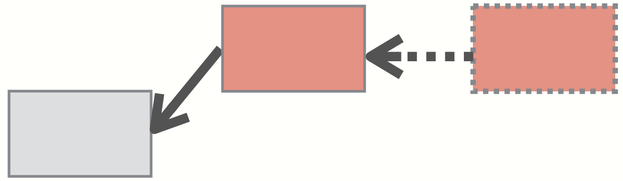
Selfish Mining Attack

- Attacker has 1/3 of mining power.
  - Honest reward = 1/3
- Selfish mining
  - When attacker finds a new block, keep it private



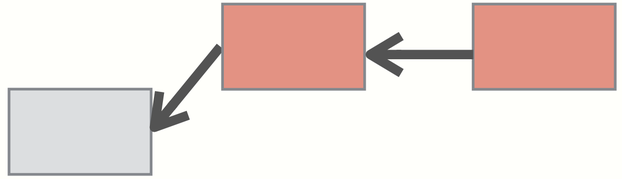
Selfish Mining Attack

- Attacker has 1/3 of mining power.
  - Honest reward = 1/3
- Selfish mining
  - When attacker finds a new block, keep it private
  - Tries to find the next block



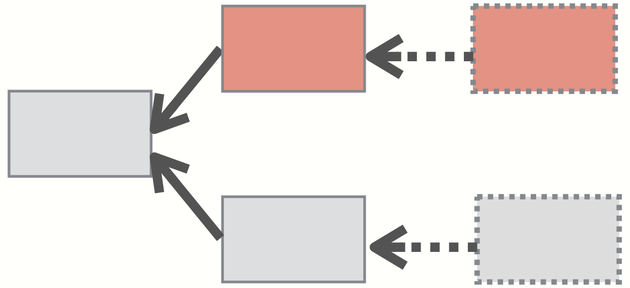
Selfish Mining Attack

- Attacker has 1/3 of mining power.
  - Honest reward = 1/3
- Selfish mining
  - When attacker finds a new block, keep it private
  - Tries to find the next block
    - If succeeds, get 2 out of 3 block rewards



Selfish Mining Attack

- Attacker has 1/3 of mining power.
  - Honest reward = 1/3
- Selfish mining
  - When attacker finds a new block, keep it private
  - Tries to find the next block
    - If succeeds, get 2 out of 3 block rewards
    - If honest miner finds a new block
      attacker publishes its block, it is a block race



Selfish Mining Attack

- Attacker has 1/3 of mining power.
  - Honest reward = 1/3
- Selfish mining
  - When attacker finds a new block, keep it private
  - Tries to find the next block
    - If succeeds, get 2 out of 3 block rewards
    - If honest miner finds a new block
      attacker publishes its block, it is a block race

1/3 * 2/3

Selfish Mining Attack

- Attacker has 1/3 of mining power.
  - Honest reward = 1/3
- Selfish mining
  - When attacker finds a new block, keep it private
  - Tries to find the next block
    - If succeeds, get 2 out of 3 block rewards
    - If honest miner finds a new block
      attacker publishes its block, it is a block race

1/3 * 2/3

+ 2/3 * 1/3 * 2/3 = **10/27**

Attacks in Practice

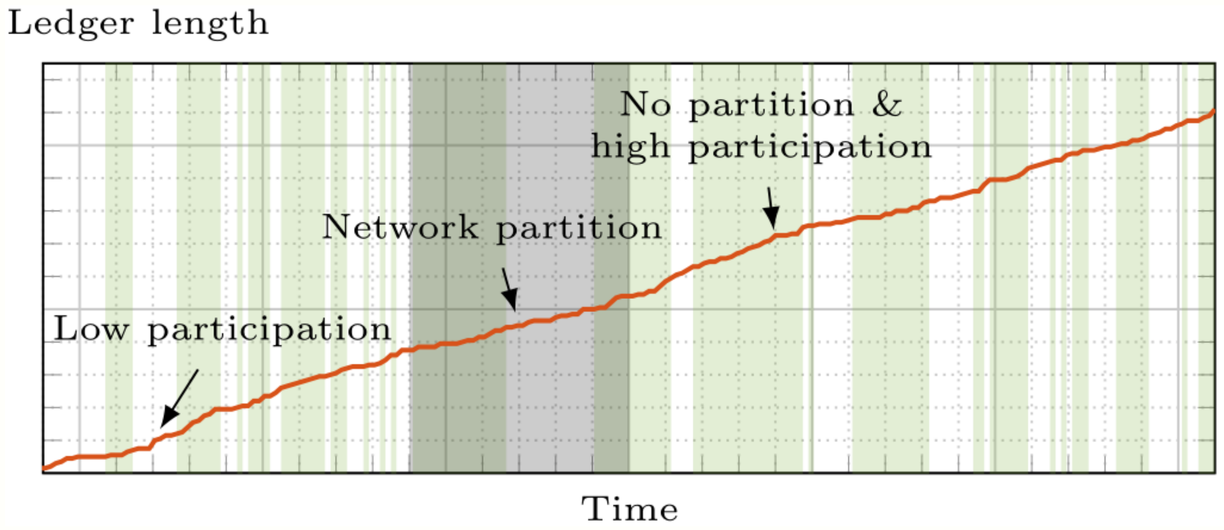
51% Attacks possible: but not seen

No Selfish mining attacks:

- Miners care about BTC price
- Not rational

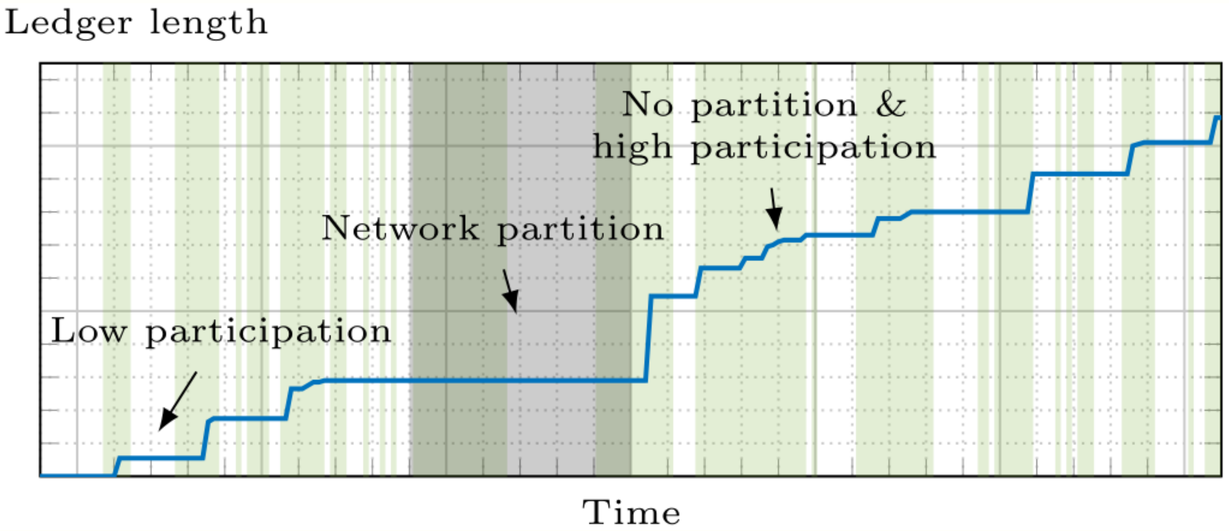
3 Ethereum 2.0 PoS Consensus

Nakamoto vs BFT under network outage



Nakamoto Consensus:

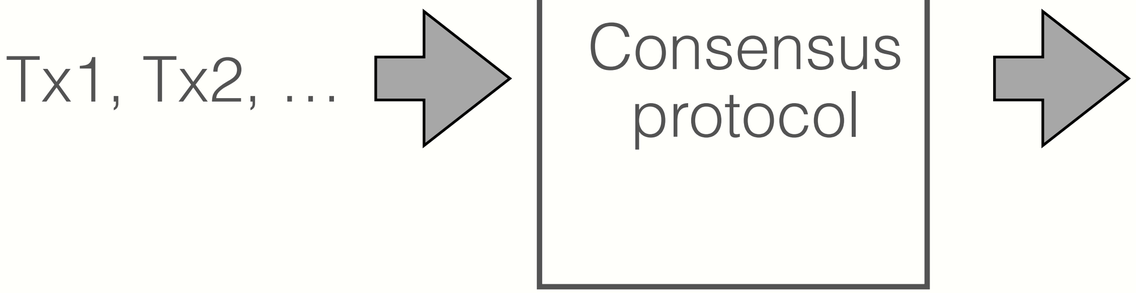
Dynamic availability



BFT protocol

Finality

Resolving the dilemma



LedgerF

Provides

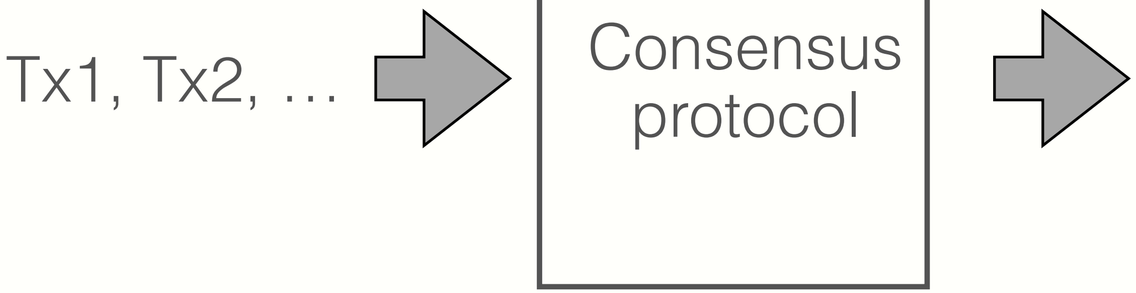
finality

LedgerA

Has

availability

Resolving the dilemma



LedgerF

Provides

finality

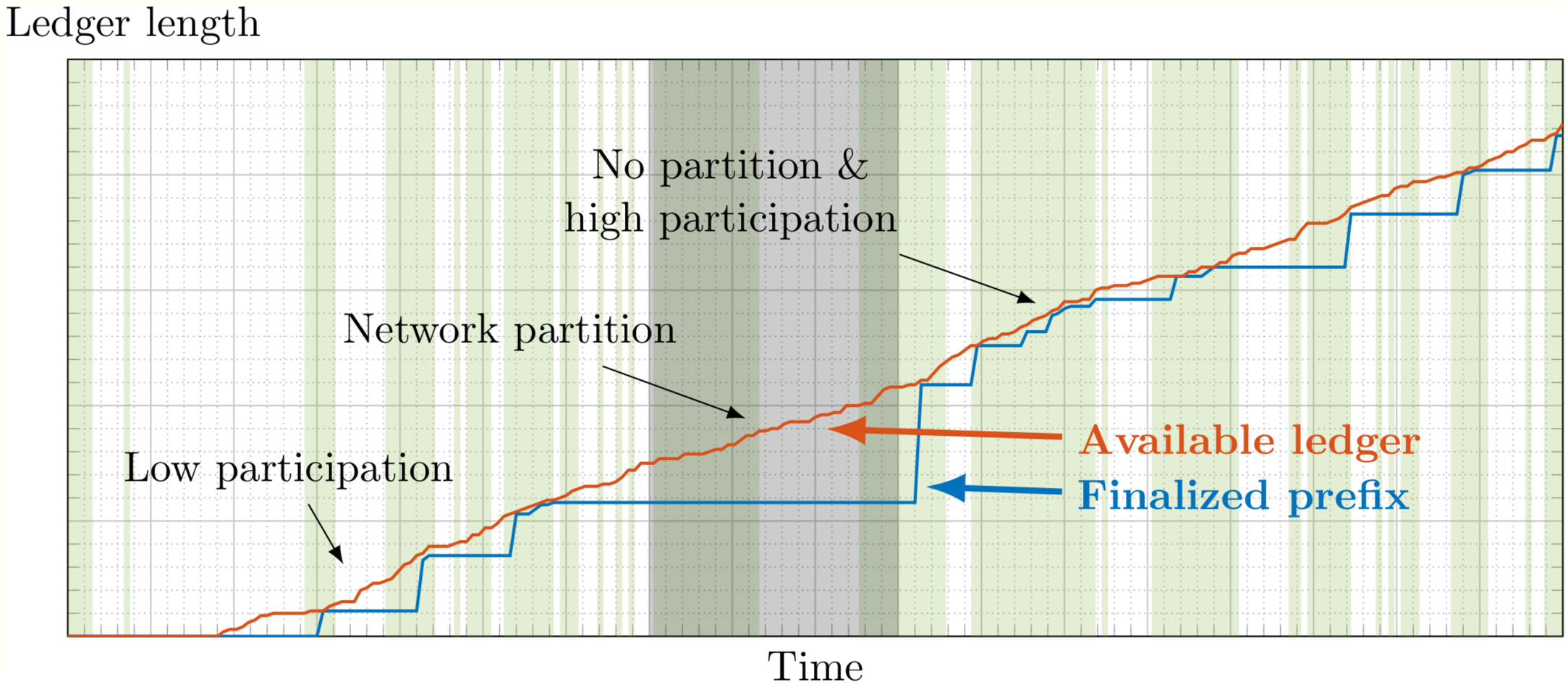
LedgerA

Has

availability

Prefix condition: LedgerF <= LedgerA

Resolving the dilemma



Ethereum 2.0

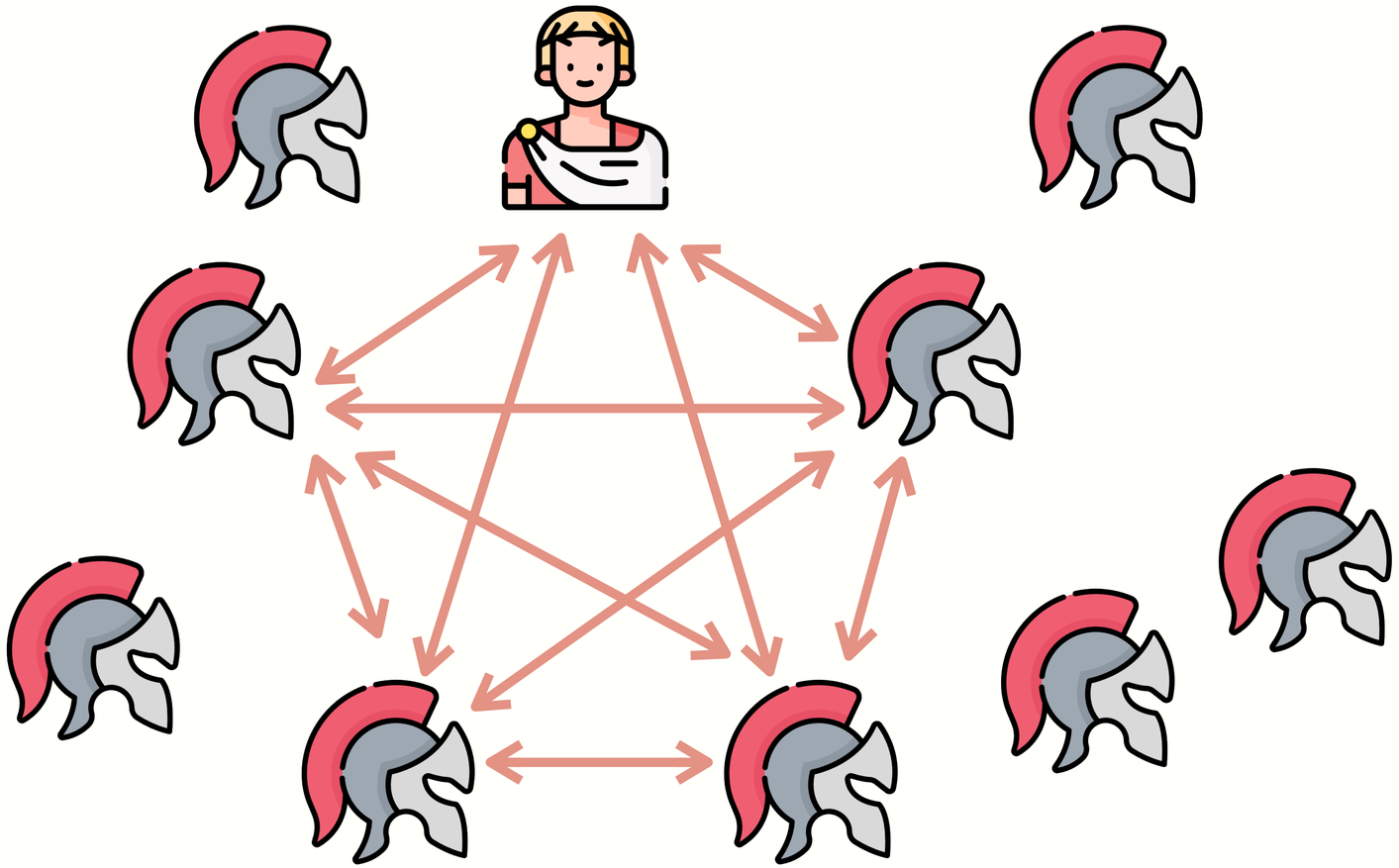
- Ethereum previously used PoW Nakamoto Consensus
- A separate PoS chain has been there since 2020
- The two chains merged in 9/2022 and PoW was deactivated
- PoS chain uses a snap and chat style protocol
  - 12s block time
  - 1 epoch is 32 blocks (6.4 minutes)
  - Finalization in 2 epochs (~13 minutes)

Proof of Stake

- Replace Sybill resistance of PoW with money
  - Stakes coins (through transaction)
  - Can’t use staked coins for anything else!
- **Incentives**: Get’s rewards/fees. Can use punishments/slashing
- **Voting Power**: Proportional to relative stake

Scaling Byzantine Consensus

Sub select a set of participants to run BC



How to select sub committee?

- Each staker computes H(block number, PK)
  - If H(block number,PK) < target
    - Become part of committee for round

How to select sub committee?

- Random Selection with Beacon
- Each Block wait for beacon randomness
- Each staker computes H(beacon, PK)
  - If H(beacon,PK) < target
    - Become part of committee for round

4

Discussion Session

How to start a startup?