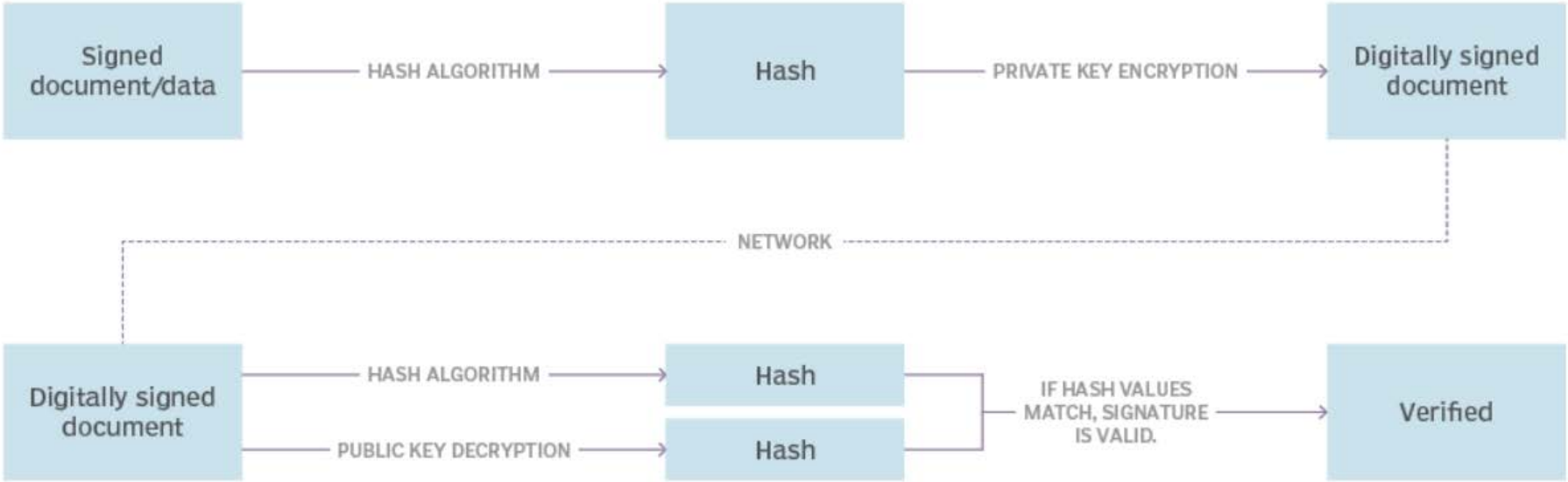
Bitcoin Mechanics

How Bitcoin achieves decentralization

outline

- PoW
- PoS
- DPoS
- BFT
- PBFT

Digital Signature Process



Public Key as Identity

- Public key is used to check the signature, thus can be treated as identity
- Easy to generate a new identity
  - Called address in Blockchain
  - Decentralized identity management
- Anonymity and Privacy
  - Real world identity can be revealed through address hehavior

Why decentralization?

- Centralization
  - Advantage: efficient, less expensive
  - Disadvantage: single-point-failure
- Vs .
- Decentralization
  - Advantage: Robust
  - Disadvantage: inefficient, expensive

Decentralization of Bitcoin

1. Who maintains the ledger of transactions?
2. Who has authority over which transactions are valid?
3. Who creates new bitcoins?
4. Who determines how the rules of the system change?
5. How do bitcoins acquire exchange value?

Distributed consensus

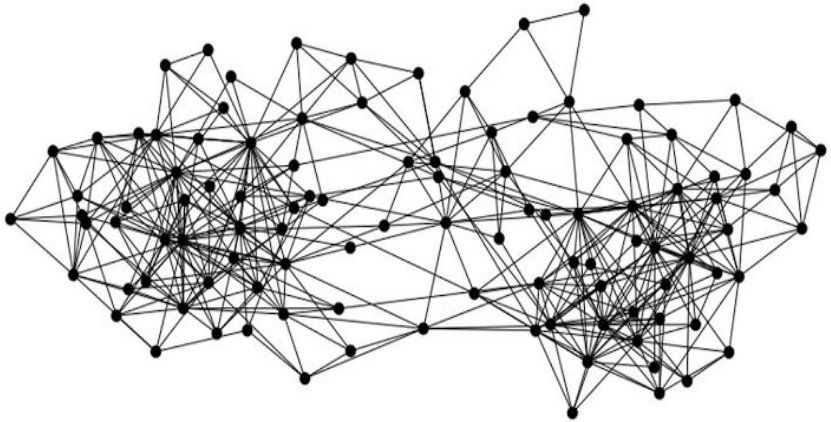
- *Distributed consensus protocol. There are n nodes that each have an input value. Some of these* nodes are faulty or malicious. A distributed consensus protocol has the following two properties:
  - It must terminate with all honest nodes in agreement on the value
  - The value must have been generated by an honest node

Distributed consensus

- When Alice wants to pay Bob, what she actually does is broadcast a transaction to all of the Bitcoin nodes that comprise the peer-to-peer network

signed by Alice

Pay to pk_{Bob} : H( )



Bitcoin consensus algorithm (simplified)

- *This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.*

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

Distributed consensus

- At any given point, every node in the p2p network
  - has a ledger consisting of a sequence of blocks, each containing a list of transactions have reached consensus
  - has a pool of outstanding transactions that it has heard about but have not yet been included on the block chain

Blockchain structure

genesis block

H

BH_1

version	(4 bytes)
prev	(32 bytes)
time	(4 bytes)
bits	(4 bytes)
nonce	(4 bytes)
Tx root	(32 bytes)

80 bytes

H

BH_2

prev

Tx root

H

BH_3

prev

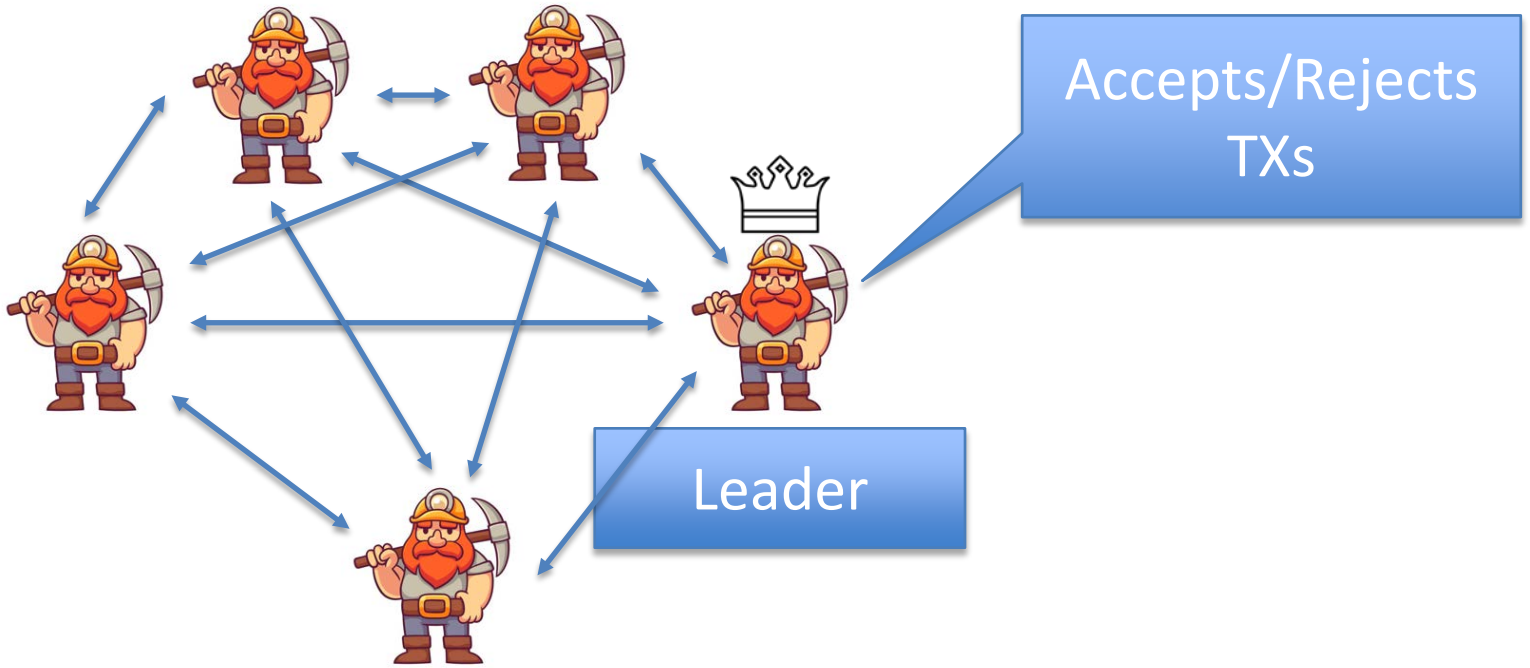
Tx root

...

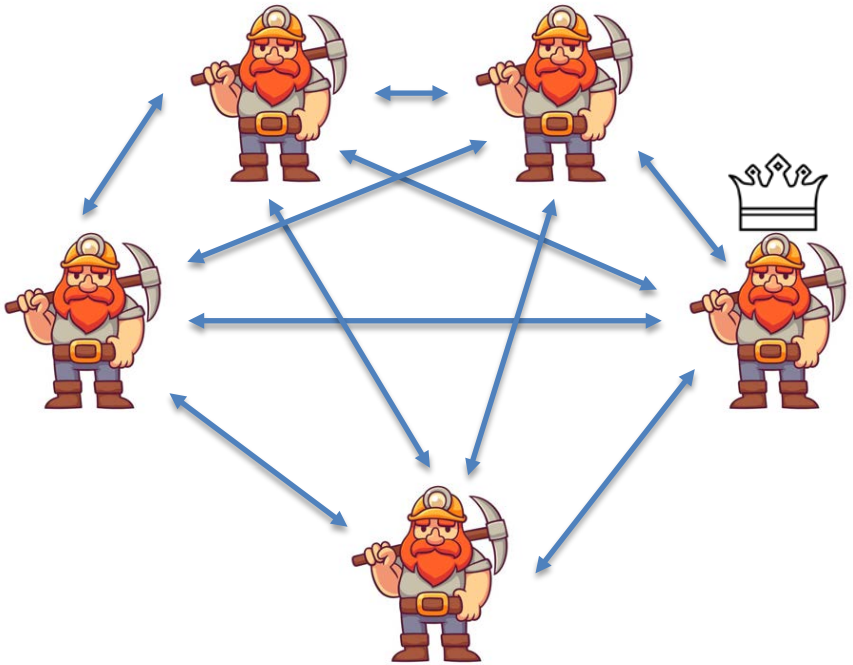
coinbase Tx

coinbase Tx

Distributed Consensus

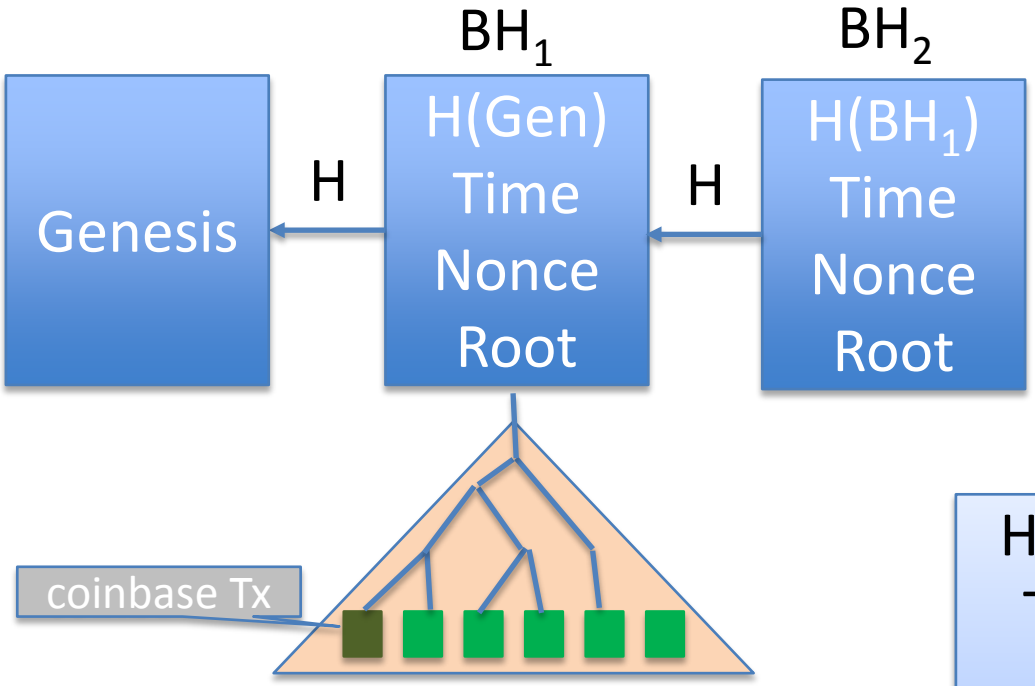


Problems with approach

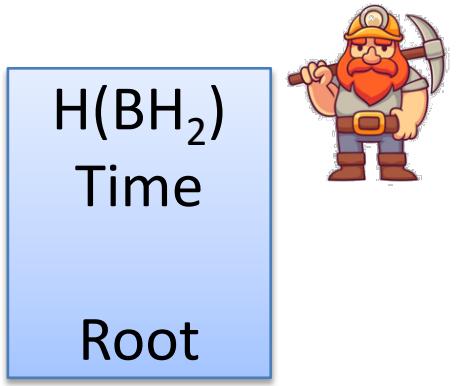


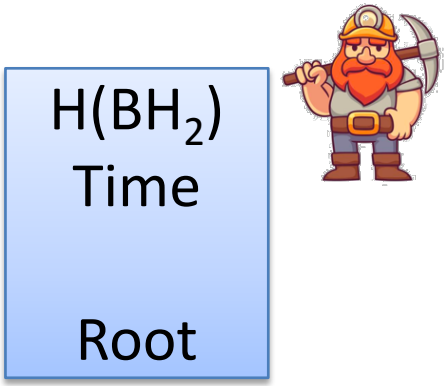
- Known committee
  - (must communicate)
- Large committee
  - Large communication
- Predictable Leader
  - Bribing

Nakamoto Consensus

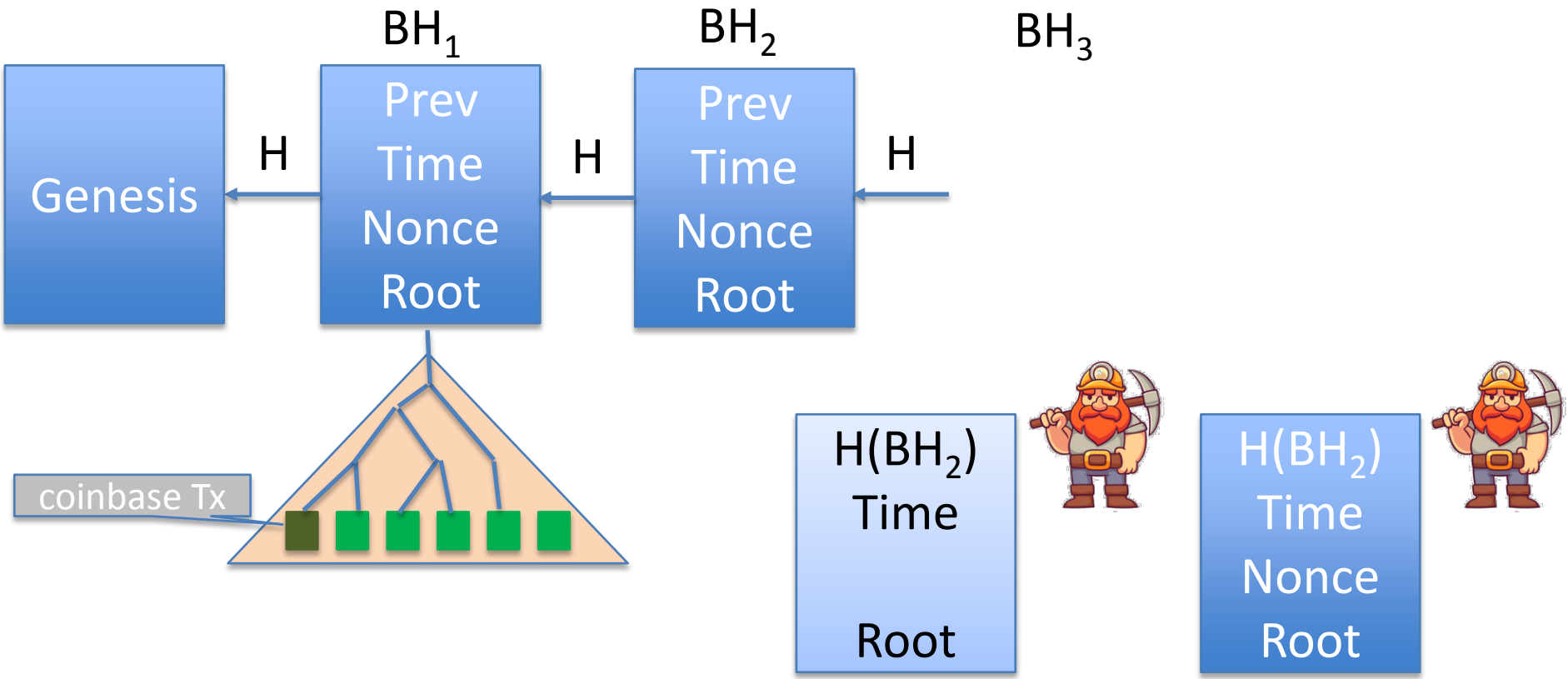


PoW:
Find nonce s.t.
H(Block)<Target
Target s.t. blocks
found every 10 min

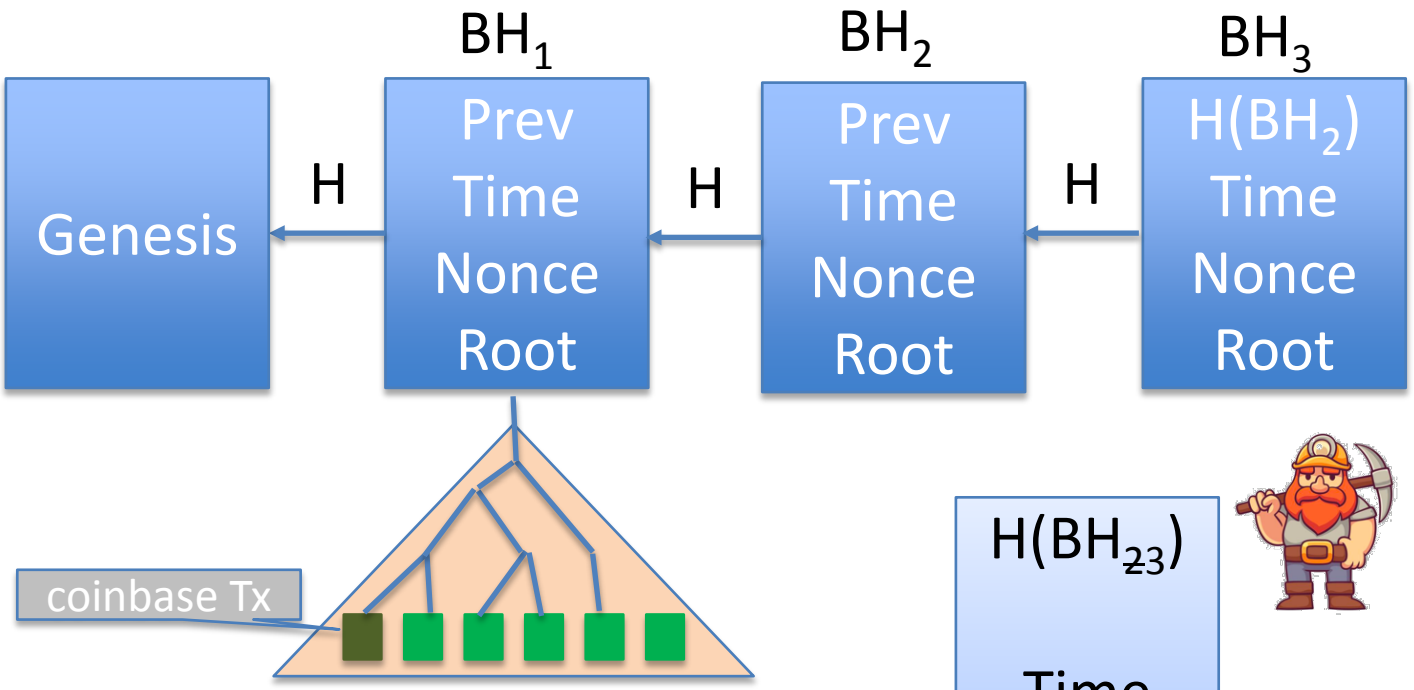




Nakamoto Consensus



Nakamoto Consensus



Nakamoto Consensus

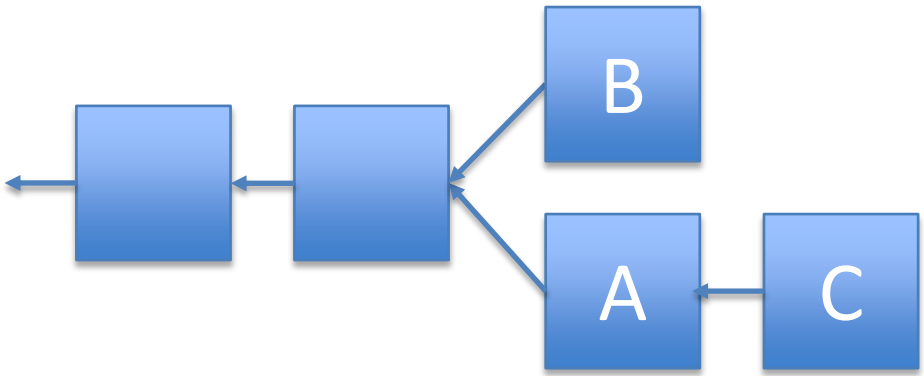
- Miners “race” to add blocks
  - Need to find PoW solution
  - Probability winning ~ Computation power
  - One winner every ~10 min
  - Target adjusted every 2 weeks
- Leader election/race combined with tx adding
- (Honest) miners extend longest chain
- Timestamps must be roughly accurate
- *All transactions must be valid*
- Blocks/Transactions become final after *k* blocks

PoW:
Find nonce s.t.
H(Block)<Target

Prev
Time
Root

Prev
Time
Root

Forks and Orphans

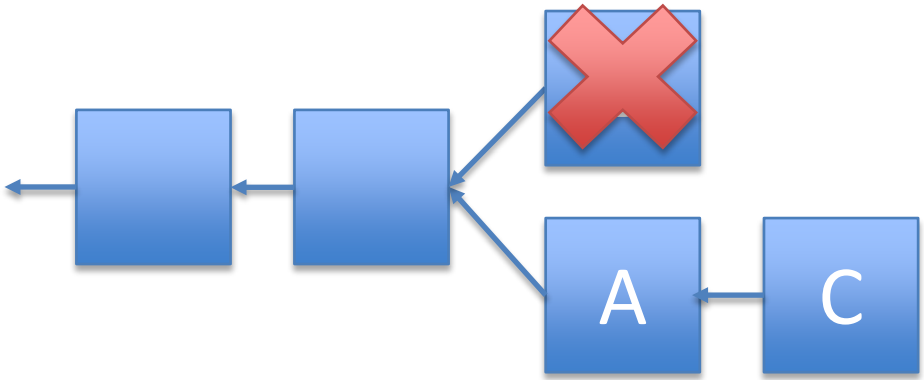


Working on B

Working on A

Forks and Orphans

Orphaned block



Working on ~~B~~ C

Working on ~~A~~ C

Stealing bitcoin

- Can Alice simply steal bitcoins belonging to another user at an address she doesn’t control?

Stealing bitcoin

- Can Alice simply steal bitcoins belonging to another user at an address she doesn’t control?
  - To steal other’s bitcoin means to forge other’s signature
    - the private key

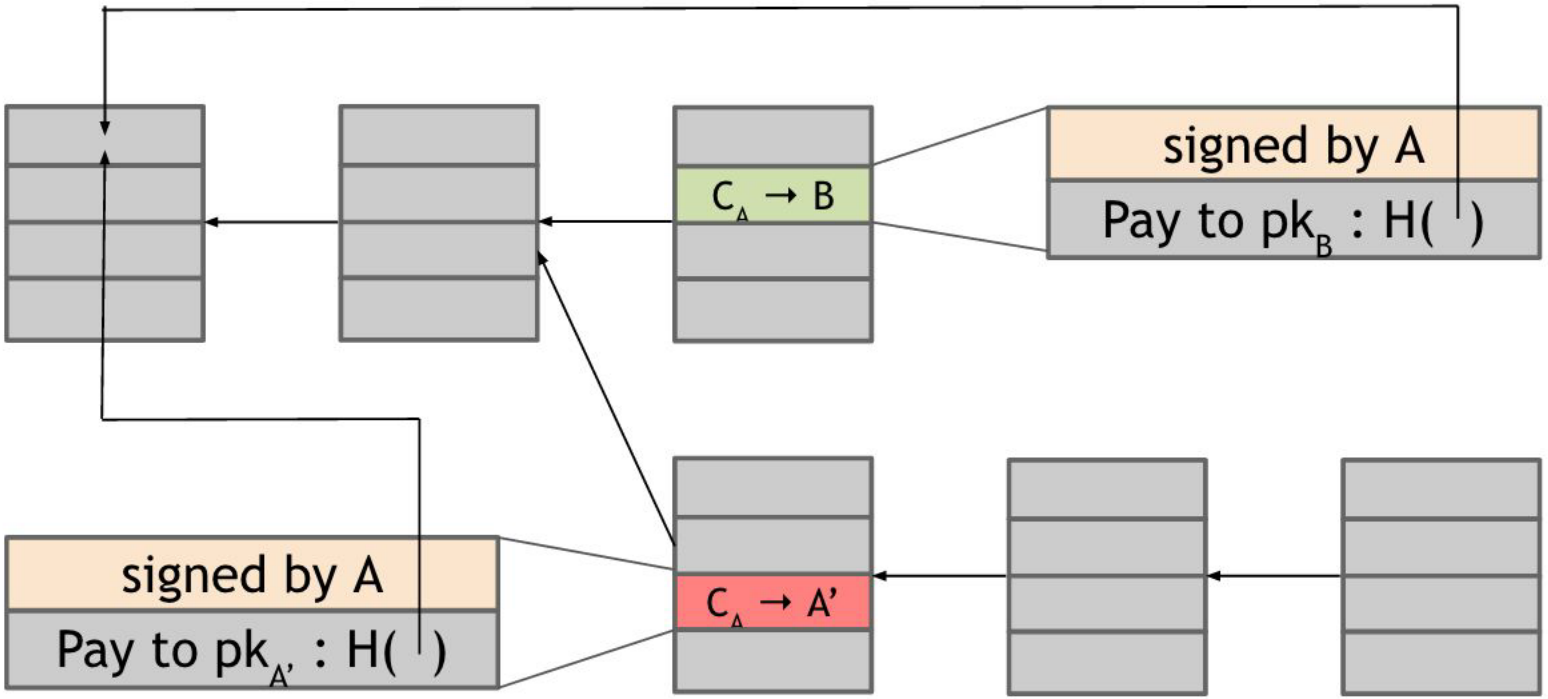
Denial of Service Attack

- Alice hates Bob, she will not include any transaction of Bob
  - As long as the majority of the network is honest, Bob’s transaction will be included in one block

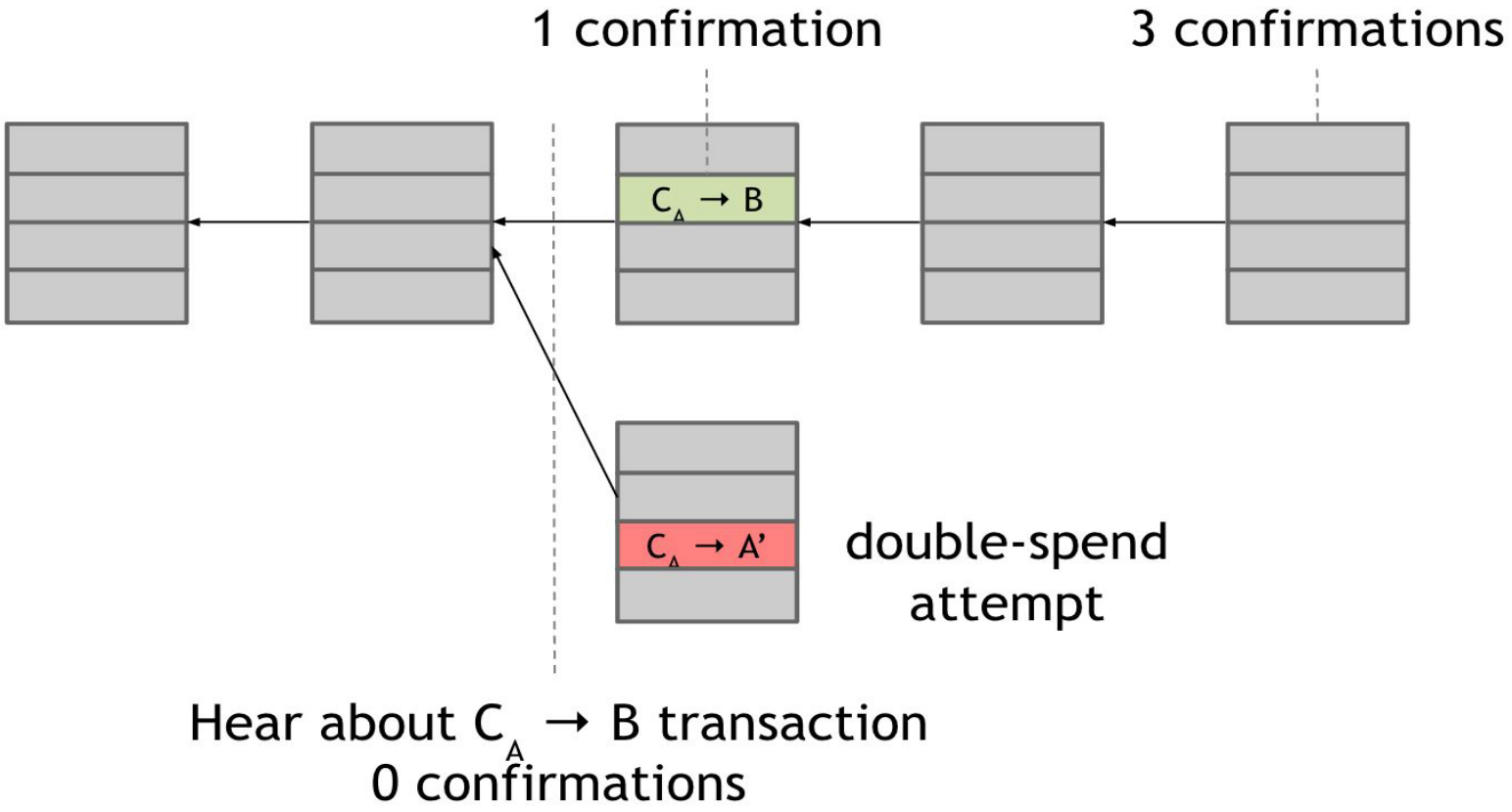
Double Spending

- Alice creates two transactions: one in which she sends Bob Bitcoins, and a second in which she double spends those Bitcoins by sending them to a different address that she controls.
- only one of these transactions can be included in the block chain
- For p2p nodes, the two transactions are indistinguishable

Double-spending



Prevent Double spending



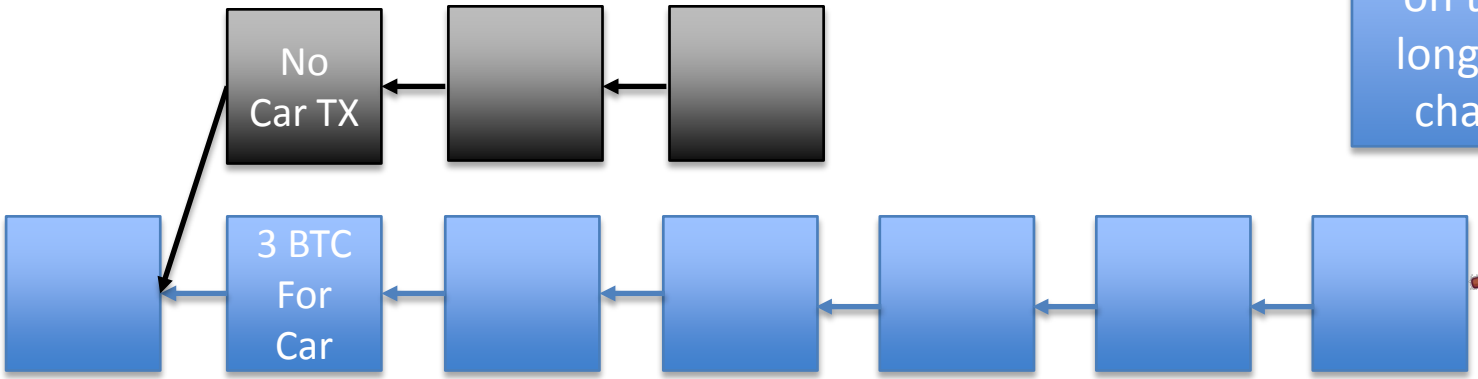
Preventing double spends

I'll wait k blocks

Here are the keys

I'll just produce a different chain

We'll be working on the longest chain

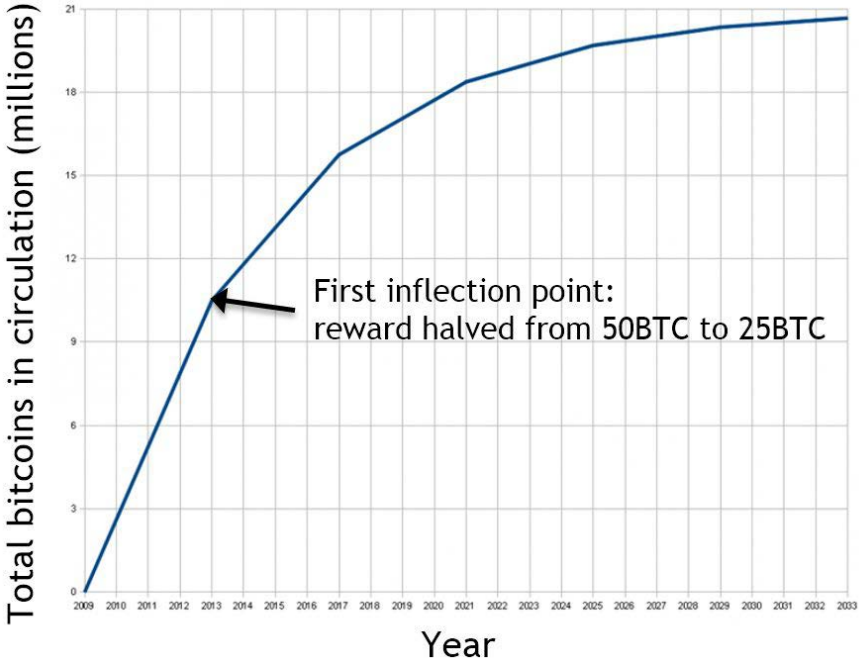


Proof of work

- Consensus in Bitcoin works better in practice than in theory
  - partly a technical mechanism and partly clever incentive engineering
  - Hard to punish malicious node; incentivize nods to behave honestly by paying them
  - Block reward and transaction fee
  -

Block reward

- the block reward is cut in half every four years limiting the total supply of bitcoins to 21 million



Block reward

- For a specific miner

$$\textit{mean time to next block} = \frac{10\ \textit{minutes}}{\textit{fraction of hash power}}$$

- Cost of mining

If

mining reward > mining cost

then miner profits

where

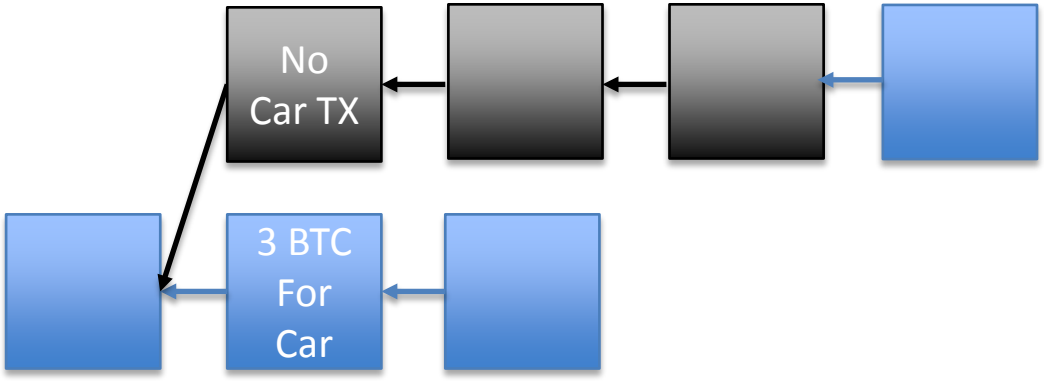
mining reward = block reward + tx fees

mining cost = hardware cost + operating costs (electricity, cooling, etc.)

51% Attack


Cloud


I'll just produce a different chain


No Car TX
3 BTC For Car


New longest chain

51% attack

- If there were 51% attack, the developers will notice and react to it
- If the attacker destroy confidence in Bitcoin, honest users will abandon the cryptocurrency

The story of BCH

- In 2017, bitmain, as the largest mining machine producer, owned the largest mining pool
- Their computing power accounts 62.8% of all

The Story of BCH

最近3天算力分布

The story of BCH

- Bitmain Vs. BitCoin Core
  - Change the block size from 1M to 2M
  - 07.23. 2017, hard fork
  - At its highest price, 1 BCH = 0.4 bitcoin

May 01, 2019	267.99	478.53	266.46	445.07	445.07	80,232,305,155
Apr 01, 2019	168.90	342.87	167.09	267.76	267.76	47,456,448,228
Mar 01, 2019	132.08	177.73	123.25	168.94	168.94	12,156,626,485
Feb 01, 2019	114.80	157.65	111.84	132.19	132.19	8,773,675,717
Jan 01, 2019	150.90	175.46	108.09	114.81	114.81	7,086,368,841
Dec 01, 2018	172.52	229.01	75.08	151.05	151.05	10,007,033,624
Nov 01, 2018	423.07	638.55	154.28	173.07	173.07	15,210,219,241
Oct 01, 2018	531.64	547.49	411.52	422.86	422.86	10,436,746,000
Sep 01, 2018	543.96	653.69	413.04	529.92	529.92	13,279,607,000
Aug 01, 2018	779.95	779.95	481.43	543.08	543.08	11,034,762,000
Jul 01, 2018	749.18	880.98	669.52	777.15	777.15	15,957,662,984
Jun 01, 2018	995.66	1,206.53	651.21	748.71	748.71	15,393,792,032
May 01, 2018	1,348.64	1,838.49	880.89	995.34	995.34	29,661,600,832
Apr 01, 2018	688.01	1,560.94	603.71	1,350.05	1,350.05	19,907,360,032
Mar 01, 2018	1,204.84	1,303.66	677.53	685.25	685.25	12,342,156,128
Feb 01, 2018	1,491.12	1,641.40	764.02	1,204.16	1,204.16	19,146,145,984
Jan 01, 2018	2,534.82	3,071.16	1,361.02	1,486.89	1,486.89	42,244,036,064
Dec 01, 2017	1,381.81	4,355.62	1,226.21	2,533.01	2,533.01	64,494,638,272
Nov 01, 2017	438.30	2,477.65	437.91	1,389.78	1,389.78	65,023,965,216
Oct 01, 2017	433.38	526.12	289.04	439.05	439.05	10,862,082,928
Sep 01, 2017	588.40	707.98	301.69	432.63	432.63	11,313,133,064
Aug 01, 2017	294.60	1,091.97	200.98	588.17	588.17	17,120,742,664

Bitcoin and Elon Musk (Tesla)

- On February 8, 2021 Tesla bought $1.5 billion worth of bitcoin. Elon Musk said a Tesla could be bought for bitcoins
  - Bitcoin price increased 16%
- On May 12, 2021 Elon Musk announced Tesla would not accept bitcoin because proof of work by fossil fuel is bad for the environment
  - Bitcoin price dropped 15%

Bitcoin and renewable energy

- Iceland is a bitcoin mining center because of geothermal
- China has been a bitcoin mining center because of inexpensive hydroelectrical power in the wet season
  - Chinese miners used electricity from coal during the dry season
  - China is now driving bitcoin miners out of the country
  - Chinese miners are relocating to Kazakhstan, Canada, etc where hydroelectric energy is readily available

Proof of Stake

- Take a look at the example PoS of Peercoin

- Ethereum is said to replace PoW with PoS at then end of 2021

Delegate Proof of Stake

- EOS and Daniel Larimer
  - Bitshare & steem
  - Bebore EOS was released, it attracted about investment about 4.2 billion
  - However….

Delegate Proof of Stake

- In DPoS, the holders of EOS Token, unlike in Proof Of Stake, do not validate but delegate the work by electing a node to validate on their behalf , called as the validator
- The number of validators in the network is limited, to 21
- The validator, once appointed as a Supernode, helps operate the network

Delegate Proof of Stake

- IF a validator continually missed their block creation times or publishes invalid transactions, the token holders using their stake can vote them out and replace them with a better validator

Delegate Proof of Stake

- DPoS pros:
- The network processes more transactions in a given period.
- DPoS blockchains are more scalable than networks with PoW or PoS since they don’t require high computational power.
- The digital democracy gives more token holders the chance to decide the block producers, compared to PoW where miners with more capital produce more blocks.
- The system is energy efficient, which means it’s also environmentally friendly.

Delegate Proof of Stake

- DPoS cons:
- Decentralization is often hard to maintain, as the decision lays in the hands of a limited number of holders, leaving room for conspiracy and censorship.
- Low participation in the voting process can generate centralization, by placing the power in the hands of a limited number of token holders

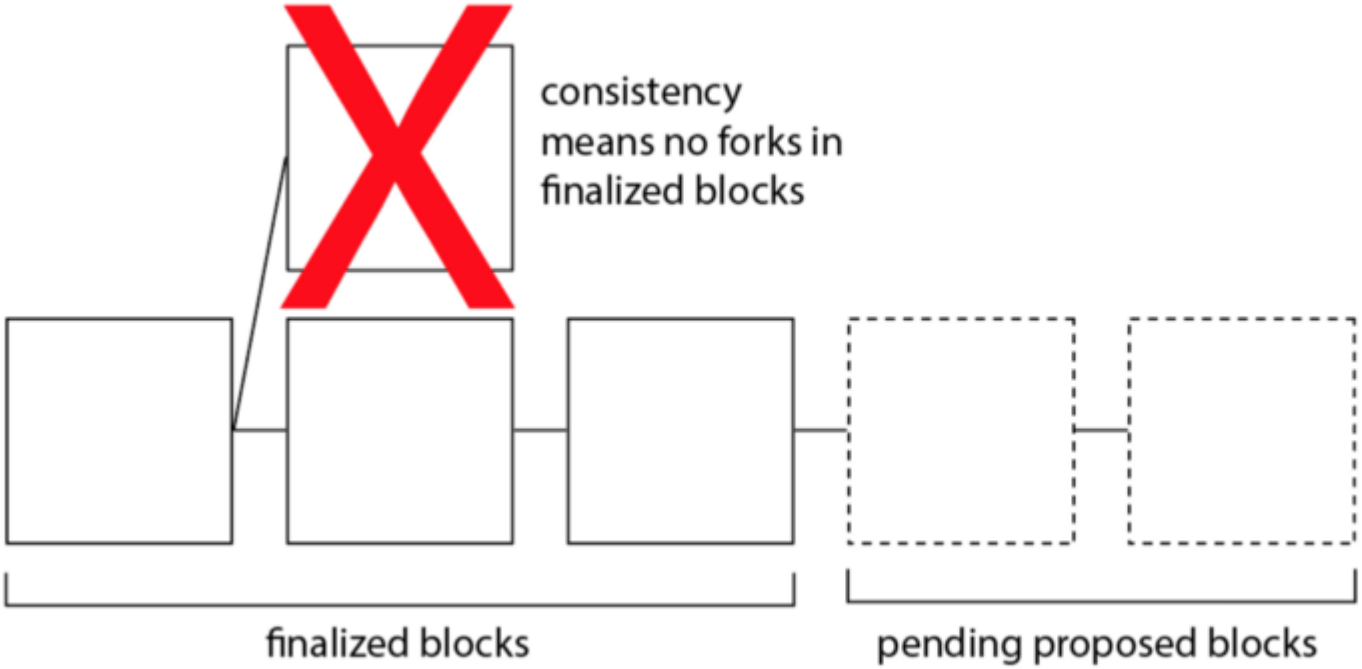
Recap

- Consensus
  - where a set of nodes seek to agree on an ever-growing linearly-ordered log, such that two important properties are satisfied:
    - ***Consistency***: all honest nodes' logs agree with each other
    - ***Liveness***: all honest nodes are able to make progress on their logs

Recap

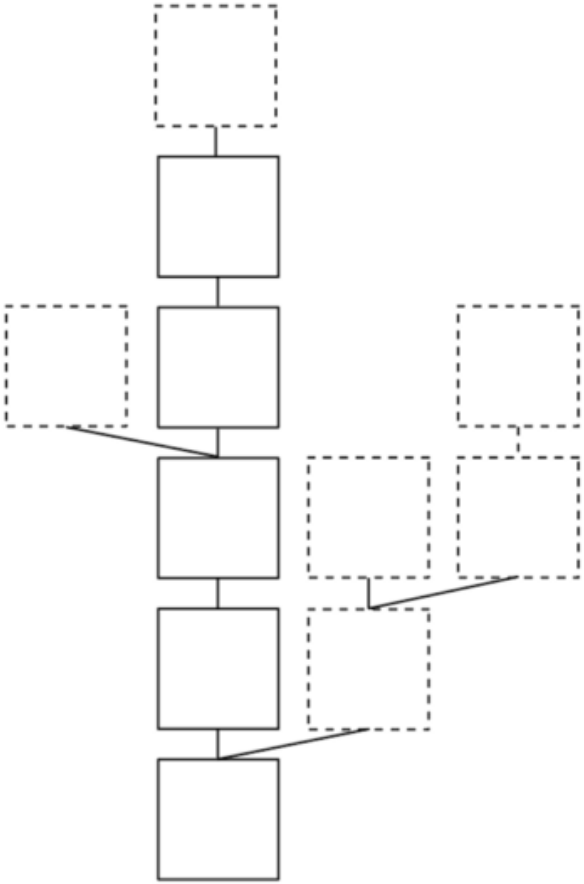
- In context of blockchain,
  - consistency (sometimes called “safety”) means that there is a single canonical chain (no forks) that all honest nodes will agree on.
  - Liveness means that honest nodes on the network are always able to add new blocks to the blockchain.
  - When all nodes agree on a block, we say the block is *finalized*.

Recap



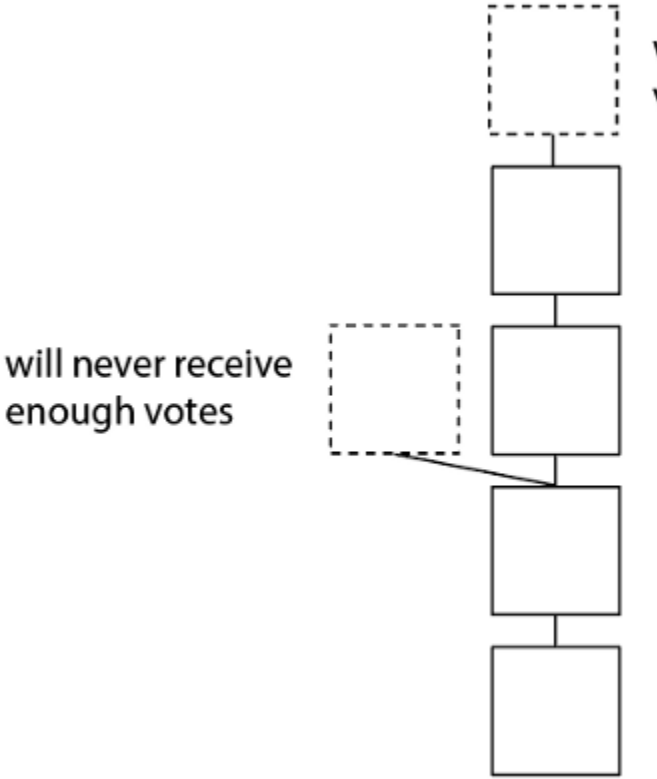
liveness means we can always add new blocks to the blockchain

the longest chain is also called
the "canonical chain"



"longest chain rule" example
solid lines are finalized blocks
dotted lines are "orphaned" or pending blocks

Classical Consensus



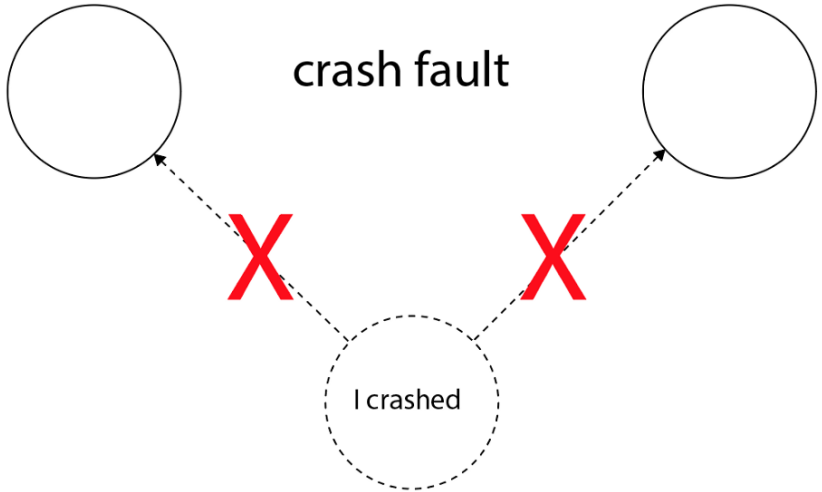
voting example
solid lines are finalized blocks
dotted lines are blocks that have not received enough votes

Classical Consensus

- Classical consensus protocols reach deterministic consensus through voting. These protocols confirm transactions fast relative to Nakamoto consensus as the consensus network size is fixed and progress can be made as soon as the required votes are seen.

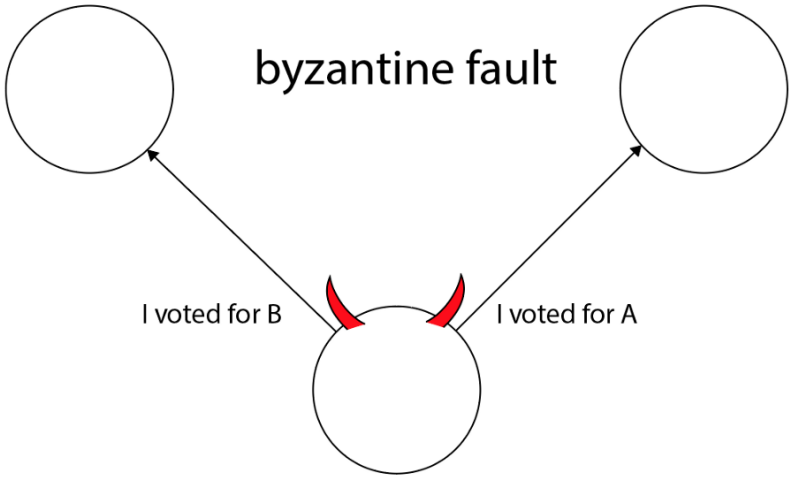
Byzantine Faults and Crash Faults

- Crash fault
  - crash fault tolerant consensus protocols
    - Raft & Zookeeper



Byzantine Faults and Crash Faults

- A byzantine node may behave arbitrarily
  - not sending messages
  - sending deliberately misleading messages
    - Double spending



BFT

- Byzantine Fault Tolerance(BFT) is the feature of a distributed network to reach consensus(agreement on the same value) even when some of the nodes in the network fail to respond or respond with incorrect information

QUESTION

- Is PoW BFT?
- Is raft BFT?

PBFT

- Widely used in permissioned blockchain
  - Public chain
    - anyone and everyone to join and contribute to the network
  - Private chain
    - only selected entry of verified participants
  - Permissioned chain
    - Between public and private

PBFT

- Permissioned blockchain
  - Allow anyone to join the permissioned network after suitable verification of their identity
  - Allocate select and designated permissions to perform only certain activities on the network
  - grant special permissions to each participant

PBFT

- Application
  - used for managing dealings in farm produce from its origin (the farm) to the end customer (the market) across the globe [different prices in different markets]
  - multiple entities
    - The farmer
    - The customs (海关)
    - The shipping company
    - The warehouse operators
    - The customers
    - …..

PBFT

- Replica

- Primary
- Backup
- Quorum : 法定人数
  - The minimum number of members of an assembly or society that must be present at any of its meetings to make the proceedings of that meeting valid.

PBFT

- Suppose you have N replicas, f of which might crash (non-Byzantine failure)
- What quorum size Q do you need to guarantee liveness and safety?
- * Liveness: (or pseudo-liveness, i.e., avoiding stuck states)
- There must be a non-failed quorum (*quorum availability*)
- Hence: Q <= N - f
- * Safety: Any two quorums must intersect at one or more nodes
  Otherwise, two quorums could independently accept operations, diverge
  This property is often known as the *quorum intersection* property
- Hence: 2Q - N > 0
- So: N < 2Q <= 2(N - f)
- Note highest possible f: N < 2N-2f; f < N/2
- And if N = 2f + 1, smallest Q is 2Q > 2f + 1; Q = f + 1

PBFT

- What quorum size Q do we need in Byzantine setting?
- * Liveness: Q <= N - f As in non-Byzantine case, failed nodes might not reply
- * Safety: Quorum intersection must contain one non-faulty node
- Idea: out of f+1 nodes, at most one can be faulty [at least one is no-faulty]
- Hence: 2Q - N > f (since f could be malicious)
- So: N + f < 2Q <= 2(N - f)
- Highest f: N+f < 2N-2f; 3f < N; f < N/3 And if N = 3f + 1, the smallest Q is: N + f < 2Q; 3f + 1 + f < 2Q; 2f + 1/2 < Q; Q_min = 2f + 1

The Algorithm

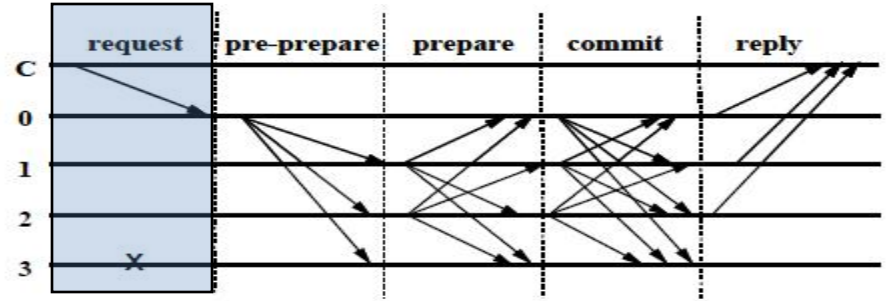


Figure 1: Normal Case Operation

- 1. A client sends a request to invoke a service operation to the primary

$$\langle \text{REQUEST}, o, t, c \rangle_{\sigma_c}$$

o= requested operation
t= timestamp
c= client
σ= signature

The Algorithm

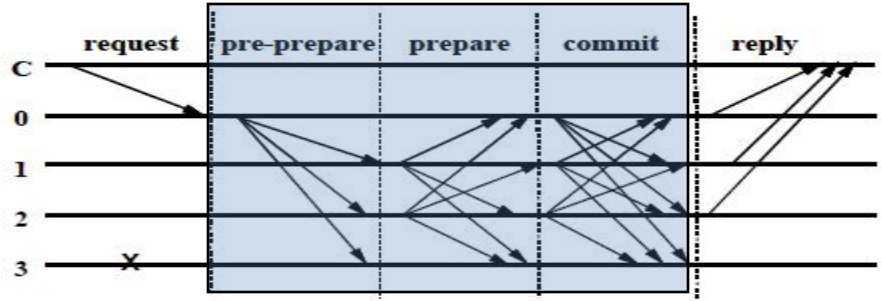


Figure 1: Normal Case Operation

- 2. The primary multicasts the request to the backups (three-phase protocol)

The Algorithm

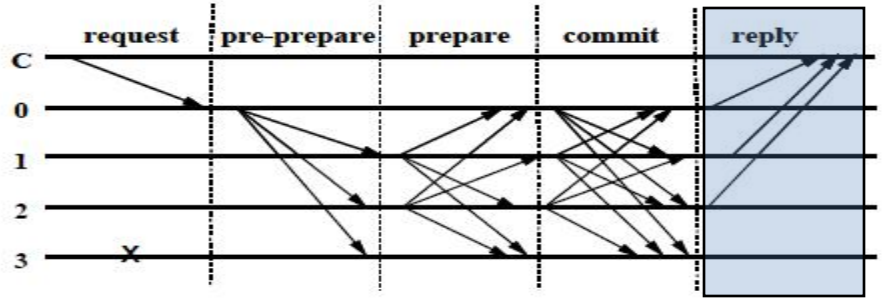


Figure 1: Normal Case Operation

- 3. Replicas execute the request and send a reply to the client $\langle \text{REPLY}, v, t, c, i, r \rangle_{\sigma_i}$

o= requested operation
t= timestamp
c= client
ϭ= signature

v= view
i= replica
r= result

The Algorithm

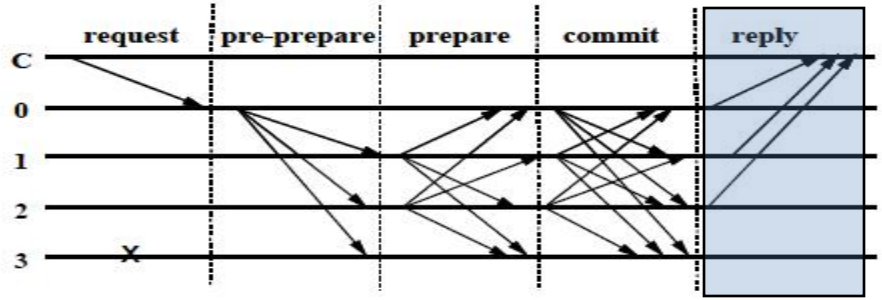


Figure 1: Normal Case Operation

- 4. The client waits for f+1 replies from different replicas with the same result; this is the result of the operation

Three-phase protocol

- Primary sends pre-prepare message to all
- Pre-prepare contains <v#,seq#,op>
  - Records operation in log as pre-prepared
  - Keep in mind that primary might be malicious
    - Send different seq# for the same op to different replicas
    - Use a duplicate seq# for op

Three-phase protocol

- Replicas check the pre-prepare and if it is ok:
  - Record operation in log as pre-prepared
  - Send prepare messages to all
  - Prepare contains <i,v#,seq#,op>

- All to all communication

Three-phase protocol

- Replicas wait for 2f+1 matching prepares
  - Record operation in log as prepared
  - Send commit message to all
  - Commit contains <i,v#,seq#,op>
- What does this stage achieve:
  - All honest nodes that are prepared prepare the same value

Three-phase protocol

- Replicas wait for 2f+1 matching commits
  - Record operation in log as committed
  - Execute the operation
  - Send result to the client