OBG Onboarding Week2

Joe

February 13, 2023

Outline for today:

- Homework Check
- Week1 refresher
- Network topology (Global vs. local view)
- Understanding the blockchain protocol
- The state transition function and when to drop peers
- The social layer (non-protocol forks)
- Tying it all together

Homework Check

Who did the homework

What did you learn from it? Did you find it interesting? What questions do you have?

Figure: Pet3rpan's twitter profile picture

Week1 Refresher

1. Who remembers what we talked about last week?

Week1 Refresher

1. Who remembers what we talked about last week?
2. Cathedral vs. Bazaar style software development

Week1 Refresher

1. Who remembers what we talked about last week?
2. Cathedral vs. Bazaar style software development
3. Bazaar style software development requires guarantees that the stuff you build on won't be shut down or changed

Week1 Refresher

1. Who remembers what we talked about last week?
2. Cathedral vs. Bazaar style software development
3. Bazaar style software development requires guarantees that the stuff you build on won't be shut down or changed
4. Blockchains are perfect for this. Today we are going to look at why!

Week1 Refresher

For collaborative software development to work, we need to overcome the problem of censorship

Think back to the twitter example where they shut down third party apps which discouraged collaboration

Theme for today: What makes blockchains more censorship-resistant than other networks?

What is a blockchain?

Colloquial name for the network: "Blockchain" refers to the type of datastructure used to maintain the history of transactions, says nothing about the network

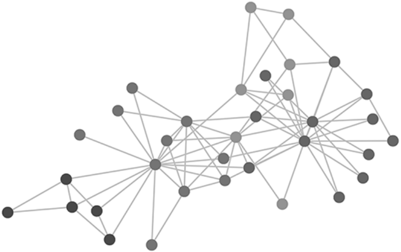
Satoshi's name: "Distributed timestamp server" refers to the idea that multiple people are keeping track of the history of transactions (more accurate imo)

My definition: A protocol that allows for decentralized consensus over the ordering and inclusion of transactions into a database (most accurate imo)

Blockchain: A network of people running clients of the same protocol! $->$ The network is collectively maintaining a timestamped database

Network topology

Figure: Graph Theory



Default settings: Ethereum nodes 25 peers, Bitcoin nodes 125 peers

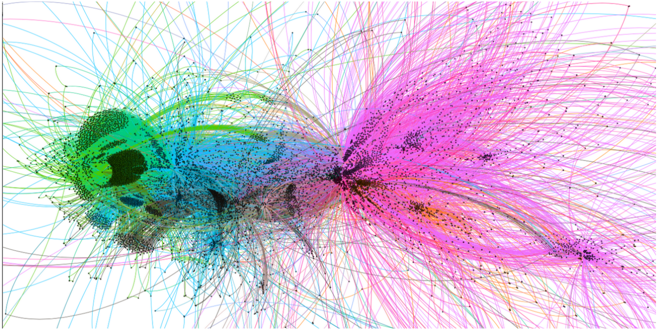
How many peers your node has is bounded by what your hardware/internet can handle

Visualizing Ethereum

Its very hard to visualize or get accurate numbers on nodes on p2p networks

Can make very pretty charts on transaction data! Nodes are wallet addresses, not clients. Edges are proportional to numbers of tokens transferred

Figure: $OMG Txs



Local view of the network

Hopefully that gave you some understanding of where the term nodes came from

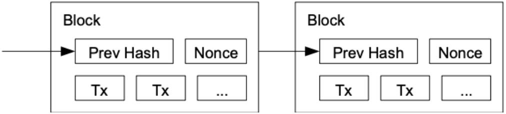
Previous slides were visualizations of looking at the network from a top-down ”global” perspective

Now we are going to look at what goes on with an individual node

Local view of the network

Every node receives new blocks from its peers. Maintains the list of blocks in a chain.

Figure: Blockchain



Nodes run the transactions in the order they received them to find the state of the most recent block

Ex: Alice creates a bitcoin account in block1, Bob sends Alice a bitcoin in block2, Alice's state at block3 is 1 bitcoin

Decisions on what transactions are included to the blockchain are made globally, state is calculated locally. In the future, state will be calculated globally through zk proofs!

The different type of blockchain nodes

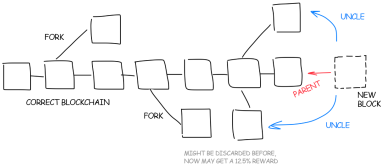
There are multiple types of blockchain nodes but only two have power over the network

The blockproducers (miners in PoW, validators in PoS) act as the "servers" of the network and make updates to the network. These types of nodes cost a lot to operate but are also paid with transaction fees and newly issued currency.

Fullnodes are what the average person would run on their computer. Not paid but are the only way to access the network without using someone else's full node. Going to look at why relying on someone else's full node defeats the purpose of a blockchain in week4!

What about latency and network partitions?

Figure: Network Partition



Due to latency or delays in network communication, nodes might receive blocks at different times. Each protocol has a way of handling this although the "longest chain rule" is most popular (in PoW the chain with the most blocks and in PoS the chain with the most weight)

Important point is that blockchains need to be able to handle asynchrony in the network!

The blockchain protocol

What are the different parts of the protocol?

P2P networking layer, fork choice rule, issuance/reward schedule for those running the network, many other things!

Where can I see the protocol?

When a blockchain is first made the protocol is typically outlined in a whitepaper. Sometimes they make a yellowpaper that updates the whitepaper with changes, othertimes they stop maintaining a paper altogether. Most people usually reference an individual client as the official protocol.

In ethereum, people typically reference the go implementation of the protocol (geth)

If one client makes a change to their protocol and the other clients do not, the network will split. Ethereum has a very specific governance process for making changes so all client teams come to consensus on changes to the protocol.

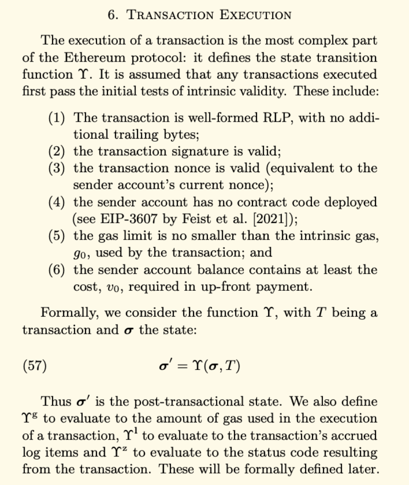
How do blockchains prevent censorship?

Now that we know what a blockchain is, we can look at what makes it what makes it different from other networks.

This comes down to the state transition function and whether the majority of people are running full nodes.

State transition function

Figure: State Transition Function



6. Transaction Execution

The execution of a transaction is the most complex part of the Ethereum protocol: it defines the state transition function Υ. It is assumed that any transactions executed first pass the initial tests of intrinsic validity. These include:

(1) The transaction is well-formed RLP, with no additional trailing bytes;
(2) the transaction signature is valid;
(3) the transaction nonce is valid (equivalent to the sender account's current nonce);
(4) the sender account has no contract code deployed (see EIP-3607 by Feist et al. [2021]);
(5) the gas limit is no smaller than the intrinsic gas, g_0, used by the transaction; and
(6) the sender account balance contains at least the cost, v_0, required in up-front payment.

Formally, we consider the function Υ, with T being a transaction and $\boldsymbol{\sigma}$ the state:

$$\boldsymbol{\sigma}' = \Upsilon(\boldsymbol{\sigma}, T) \tag{57}$$

Thus $\boldsymbol{\sigma}'$ is the post-transactional state. We also define Υ^{g} to evaluate to the amount of gas used in the execution of a transaction, Υ^{l} to evaluate to the transaction's accrued log items and Υ^{z} to evaluate to the status code resulting from the transaction. These will be formally defined later.

State transition function

The state transition function makes sure that block producers are following the protocol correctly!

This prevents block producers from publishing whatever update they want. This is VERY IMPORTANT because it is what makes blockchains different from other networks

This is the basis for why nothing in the collective database can be changed without the consent of the entire community! Power dynamic between the users and operators of the network is symmetric.

If there are more fullnodes than blockproducers, the power over the network actually belongs to the fullnodes

State transition function

Figure: Traditional Client/Server model

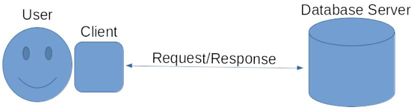
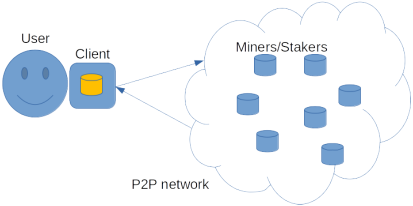


Figure: Blockchain



In traditional database model, clients trust the database is operated correctly. In blockchains, clients verify that the database operators are following the agreed upon protocol

Full node majority

Blockchains gives everyone the same power over the network even if they do not have the hardware requirements to be a database operator

This security model only works if there are more full nodes than block producers. Without full nodes to keep them in check block producers might be tempted to cheat the protocol!

Ex: everyone accesses the blockchain through a remote connection to a block producing node. Since they aren't running a node themselves, no way to check that the block producing node is following the protocol!

When to drop a peer

Rules for when to drop a peer from the network are typically not in-protocol, behavior is decided by client developers

For bitcoin (only 1 client), Bitcoin Core client keeps track of a "ban score" module which keeps track of peer behavior. An invalid block or transaction automatically drops the peer for 24 hours.

For ethereum each client handles it differently. Geth will disconnect peers if it receives invalid responses but can always reconnect (has no memory of past behavior)

As long as your node can connect to a single honest peer it will be connected to the network

Full node majority

If you are not verifying the chain or that the protocol has been followed correctly, you lose the security model that makes blockchains different from other networks

Not problematic if only a handful are doing this, but big problem if everyone is

Turns out that if the full nodes make up the network majority, they can collectively force block producers to change to whatever protocol they want

social forks

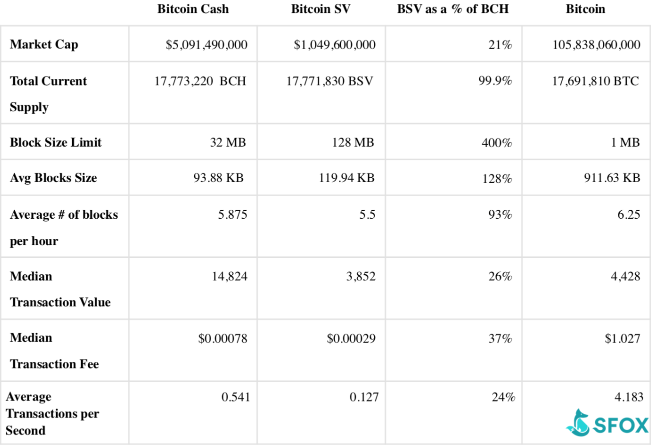
A social fork is when the community wants to make an out-of-protocol change to the network

Can happen if they want to upgrade the protocol ie: ”the merge”

Can also happen if they want to make a change to the state of the network if something bad has happened. ie: Ethereum vs. Ethereum Classic

Requires someone rewriting the client and then getting everyone else to upgrade to the new version. If this does not happen the network splits based on the portion that upgrade! Even though it may be a small change, the market can price these two networks very differently

Figure: Blockchain



	Bitcoin Cash	Bitcoin SV	BSV as a % of BCH	Bitcoin
Market Cap	$5,091,490,000	$1,049,600,000	21%	105,838,060,000
Total Current Supply	17,773,220 BCH	17,771,830 BSV	99.9%	17,691,810 BTC
Block Size Limit	32 MB	128 MB	400%	1 MB
Avg Blocks Size	93.88 KB	119.94 KB	128%	911.63 KB
Average # of blocks per hour	5.875	5.5	93%	6.25
Median Transaction Value	14,824	3,852	26%	4,428
Median Transaction Fee	$0.00078	$0.00029	37%	$1.027
Average Transactions per Second	0.541	0.127	24%	4.183

Field trip time!

Lets go look at the OBG nodes in the entrepreneurship center down the hall!

We operate a bitcoin full node and ethereum full node, bobby runs a cosmos block producer node (validator) at his home

Besides being fun, it's a good way to support the security of the network and also lets us do some cool projects ie: bitcoin ordinal nfts

The cost of running a node

Running a node is expensive!

Next week going to look at research and theories that makes blockchains possible. Will help us understand why running a node is so hard

Homework: Read Bitcoin Independence Day, real world situation that proved that the blockchain security model works (will link in telegram)