Blockchain

Pedro F. Souto (pfs@fe.up.pt)

January 20, 2022

Roadmap

Bitcoin

Bitcoin Blockchain

Bitcoin Proof-of-Work

Bitcoin Forks

Bitcoin Scalability and Energy Consumption

Proof-of-Stake

Permissioned Blockchains

Further Reading

Bitcoin

Problem: How to make direct online payments without going through a trusted 3rd party, e.g. PayPal, Visa, your bank?

Notes:

- Bitcoin is more like a set of accounts rather than a big bucket of digital coins or digital bank-notes
  - Each account resembles a bank account, however
- Bitcoin maintains a record of all transactions ever performed in a distributed fashion
  - Rather than maintaining the balances of all accounts
  - This record is known as the **blockchain**
- A bitcoin transaction corresponds to a payment, i.e. to a transfer of "money" from one account to other accounts.

Bitcoin: Assumptions

1. There is a (large) peer-to-peer network of nodes with some computing resources
   - Peers may join or leave at will,
     - Most nodes are expected to stay once they have joined, and to leave only for a short time
   - The network supports broadcast
     - This is implemented using anti-entropy
     - Even though there are no reliability guarantees, it is very likely that a broadcast-message will be delivered to all nodes
     - There are mechanisms for a node to request missing messages
2. There is a set of accounts each of which has a pair of private and public keys
   - These keys can be generated by the account-owner, i.e. there is no need for a public-key infrastrucure (PKI)
   - An account has an id/number which is the hash of its public-key
   - A user may have more than one account

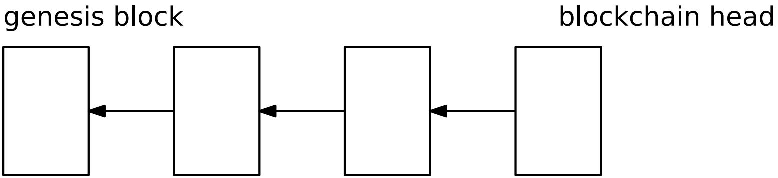
Roadmap

Bitcoin: Blockchain

Blockchain is ... a sequence of blocks

Block contains a set of transactions

- More generally, a block is a set of events (some blockchains may store also state)
- Each block contains a **header** with metadata
  - Including a reference to the previous block in the chain
- The first block in the chain is the **genesis block**
- Blocks are appended to the **blockchain head**, i.e. the most recently added block
- The maximum size of a block is 1 MByte



Bitcoin: Network

- Bitcoin's blockchain is maintained by a peer-to-peer network
  - Peers may join or leave at will,
    - Most nodes are expected to stay once they have joined, and to leave only for a short time
- Peers maintain random connections to other nodes/peers
  - In the reference specification, each node attempts to connect to 8 other nodes
  - But, the node degree can be much larger, if a peer accepts incoming connections
  - The protocol specifies no maximum number of connections
- Peers maintain a copy of the entire blockchain
  - The number of blockchain replicas currently active is about 9'000

Problem How to ensure the consistency of all these replicas?

Bitcoin: Consensus

Consensus is needed to agree on the blocks and on their order

- This together with the public log of transactions, the **blockchain**, prevents double-spending.

Conventional Byzantine Algorithms either Byzantine Quorums or PBFT rely on quorums, i.e. sets of nodes, but in a P2P network:

- It is difficult to know how many nodes there are
- Worse, it is fairly easy to create multiple identities
  - This is known as the Sybil attack

Nakamoto's solution is a protocol based on **proof-of-work**

Roadmap

Bitcoin: Proof-of-Work (PoW)

Idea solve a cryptographic puzzle that takes a random but large time

- Find a **nonce** to include in the block's header such that the header's SHA-256 is smaller than a **target**, known a priori

Rationale SHA-256 is a non-invertible function, thus this puzzle must be solved by brute force

- I.e., by repeatedly trying all possible bit strings

Target can be tuned so as to adjust the difficulty of solving the puzzle

- The expected number of hashes to solve a puzzle is $2^{256}/target$
- Bitcoin is designed to generate blocks at a fixed-rate of 1 block every 10 minutes, independently of the hash-power in Bitcoin's network
- Bitcoin's adjust the target every 2016 blocks, which is expected to occur every 14 days, so that the expected time required to generate a block is 10 minutes

Bitcoin: Miners

- The block header includes, among other metadata:

  The hash of the previous block

  - This is like a link to the previous block, and allows to determine the order of the blocks
  - Bitcoin uses a hash rather than just a sequence number to make changes to a block in the chain unlikely

  The hash of the remaining of the block i.e. the transactions
- To generate the proof-of-work for a block, a node needs not to keep the entire blockchain
  - The size of blockchain is, as of January 2022, 380 GB and increases at a rate of about 60 GB/year
  - Generating the PoW for a block requires the ability to quickly compute hash values
- Thus the PoW is computed by nodes, **miners**, which nowadays use ASIC's specially designed for Bitcoin
- To simplify, we will ignore that nodes with the blockchain and miners are usually different nodes, and assume that nodes keep both the blockchain and (try to) generate the PoW

Bitcoin: Block Broadcasting

- Upon solving the PoW, a node broadcasts the new block
- Upon receiving a new block, a node:
  - Checks its validity, by:
    1. Verifying its PoW, i.e. computing the hash of its header
    2. Checking all transactions in the block
  - If the new block is valid
    - The node stops working on the PoW for a block extending the current head
    - Adds the new block at the head of the blocchain
    - Forwards the new block
- In both cases, a node starts working on the next block, which will follow the one just added
- When a node receives a new block, its chain may be missing some of its ancestors
  - The node will have to fetch and validate the missing blocks
  - The protocol is designed to efficiently synchronize nodes that were disconnected for some time

Bitcoin: Block Broadcasting with Anti-Entropy

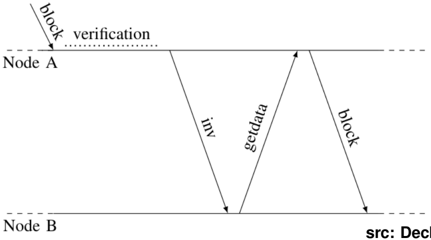
Upon validation of a new block a node sends to its neighbors **inv(entory)** messages with a set of hashes of blocks it has

Upon receiving an inv message with hashes of blocks it does not have in its blockchain, a node sends a **getdata** message with a list of the hashes of blocks it wants

Upon receiving a getdata message a node sends each block in getdata's block list in its own **block** message

Each block is inserted into the network by a miner using an unsolicited block message to one or more peers

- The block has just been generated



src: Decker13

Bitcoin: Block Propagation Delay

- Block validation can add a significant delay
  - It may require access to blockchain blocks that are on disk
  - In the case of **orphan** blocks, it may even require fetching the missing blocks over the network (and validating them)
- Block validation is repeated at every hop
  - Validation time adds to block transmission time
- **Block propagation delay** has a long tail distribution
  - As of January 2022, whereas most nodes have propagation delay below one second, a few may have delays of almost 10 seconds
  - This is almost a one order of magnitude improvement over the values 2013 paper by Decker and Wattenhofer
  - May be because of the improvements suggested in that paper (in order of relevance):
    1. reduction in the diameter of the overlay network, by increasing the connectivity
    2. faster validation, thanks to faster HW (note that hash-power is not that significant for validation)
    3. minimize verification delay, by advertising a new block after the PoW verification, but before validating transactions (a lot more expensive)

Roadmap

Bitcoin: Forks

Fork occurs when two or more nodes add a different block at the head of an otherwise identical blockchain at more or less the same time

- The blockchain in each node is totally ordered
- But different nodes have different blockchains

Resolution is based on the expected amount of work (usually the length) of competing blockchains

- A node switches to "a longer" chain, when it learns about it

No finality there is no 100% guarantee that a block will persist

- However, the likelihood of a bock to be removed decreases with each added block (**confirmation**)
  - Blocks with 6 confirmations are often considered final
- At some point, Nakamoto added "code-based checkpointing"
  - The hash of a block that cannot be replaced (and all those that precede it) is hardcoded in the software

Eventual consistency with high probability

- Assuming that the hash-power of an adversary is limited

Bitcoin: Fork Analysis

- Accidental forks depend mainly on two factors
  1. The expected time to generate the PoW
  2. The block propagation delay
- However, selfish mining strategies may exacerbate the problem
  - A node that finds a PoW for a block may withhold pushing that block to the network until it learns of a competing block
- This strategy can be used by an adversary to replace blocks
  - It does not need more than 50% of the network's hash-power for the success probability to be non negligible
- Network partitions can also lead to forks
  - Network partitions may lead to a significant drop in the PoW generation rate
  - Conversely, forks lead to a "partition" of network nodes
    - Usually, competing blocks are ignored

Roadmap

Bitcoin: (Design) Scalability Issues

Broadcasting :

- Tx
- Blocks (can be 1 MB long)

Proof-of-work is computationally intensive

- and increases with the system's hash-power:
  - The target is one block per 10 minutes

Blocks cannot be larger than 1 MB long

- Together with the target block-rate the maximum number of Tx per second becomes much smaller than that of Visa

Storage of the whole blockchain kept by all (full-)nodes

- The restrictions above limit the rate of growth to about 144 MB/day $\sim$ 50 G/year (actually 60GB in 2021)

Bitcoin: Transaction Rate Bound

Statistics the maximum 30-day average number of transactions confirmed per second by Bitcoin over the last 5 years, was 4.3

Theoretical limit of less than 8 transactions per second

- Assuming an average transaction size of 208 bytes

Visa processes, on average, 1700 transactions per second

- Claims a capacity of more than 65'000 transactions per second, as of Aug. 2017

Bitcoin parameter tunning cannot make for this difference of more than 3 orders of magnitude (assuming capacity of 10K),

Block size if we increase it by an order of magnitude

- Block propagation will increase, but may be this is OK, as we would get back to the numbers of 10 years ago
- But block chain size will increase at a rate of 500 GB/year

PoW difficulty if we increase block rate to 1 per minute (one order)

- Forking will be much more frequent
- This is made worse if we try to tune both block size and rate

Bitcoin: Energy Consumption

Extremely low energy-efficiency The PoW is tuned so as to ensure a constant block-rate
- At best, Bitcoin takes advantage of new technology
  - ASICs are more efficient than GPUs
- But, to be relevant and secure it requires a huge hash-power

Climate change The impact is difficult to assess
- The energy consumption estimates are not that tight
  - The ratio between the worst and the best scenarios is about 7
- It is not clear how this maps to CO_2 emissions
  - It depends on the energy mix
- Anyway there is an **opportunity cost**
  - May be we could use that energy to produce hydrogen
  - Electrolysis installed capacity is 8 GW (not sure this is input or output, efficiency is currently between 70%-80%), and accounts for 4% of total hydrogen production
  - Bitcoin's power consumption estimate is almost 40 GW in the worst case scenario and $\sim$16 GW in the best-guess scenario

Roadmap

Proof-of-Stake (PoS) (1/2)

PoS is an alternative to PoW

- Ethereum plans to replace PoW with PoS

Idea run a lottery to decide which user adds the next block to the chain

- As in a lottery users that "buy" more tickets have a higher chance to win

Coinage (from "coin" + "age") is the product of the amount of coins by the time that amount is held

- Consider the amount of coins as a function of time
- The coinage is the integral of that function

Lottery is run by requiring the hash of the block header to be below a given target

- This target depends on the coinage the block generator is willing to pay, if it wins the lottery
- The hash-rate is fixed to 1 hash-per-second
  - PoS uses a timestamp instead of a nonce

Proof-of-Stake (PoS) (2/2)

Clock synchronization is needed to validate blocks

- Given the propagation delay, NTP is more than enough

Ties are broken using the block's coinage

- PoS chooses the block with higher coinage, not the first block
- The main chain is the chain with highest-total coinage

Coinage consumption occurs when a block is added to the chain

- Users can use different strategies
- Coins transferred in recent blocks cannot be used, given that such blocks may not be final

Advantages PoS is more energy efficient and has higher block-rate

Disadvantages PoS appears to:

- Be harder to get right: The replacement of PoW by PoS in Ethereum has been postponed several times
- Have some undesirable properties to implement a decentralized crypto-currencies: check alternative PoX

Roadmap

Permissioned

Blockchain has nice features

- It allows to store unforgeable data in a persistent and transparent way
- It can be used to implement **smart contracts**

Smart contract is just code that may be executed upon some event added to the blockchain

- Ethereum is a smart-contract platform that relies on blockchain

Some applications need not to be open to the Internet at large

- Some businesses are subject to laws that require confidentiality

Problem is mainly of ensure consensus on the contents of each block an on their order

- Prior to Bitcoin and blockchain, there was a significant work on consensus and its many faces
- Byzantine Quorums and PBFT pre-date Bitcoin for about 10 years

PBFT and Bitcoin

PBFT can be used to maintain a replicated log, i.e. a blockchain, but

- It was designed for 4 or 7 replicas, i.e. for $f = 1$ or $f = 2$
- Its complexity in terms of messages is $O(n^2)$

Bitcoin showed that PoW scales (kind of) to thousands of nodes

- But this may not be needed in permissioned networks

Taxonomy regarding authorization to maintain, grow and access a **blockchain**

Permissionless/Public anyone can both read and grow the blockchain. Example: Bitcoin

Consortium maintained by the members of a consortium

- Each member may run a few nodes
- These nodes are responsible for persistence but also for growing the blockchain
- The blockchain may use different read policies, i.e. from open access to consortium-only

Private/Permissioned owned by a single organization, which controls which blocks are added

PBFT vs PoW

Feature	PoW	PBFT
Node ids	open	known a priori
Cons. finality	No	Yes
Node scalability	$\sim$ 1000's	unknown
Client scalability	$\sim$ 1000's	$\sim$ 1000's
Throughput (tx/s)	$\sim$ 10	$\sim$ 1000's
Latency	confirmation score	close to network's
Power consumption	high	low
Adversary power	theoretically 1/2	1/3 of the votes
Synchrony	for block validation	for liveness
Correctness proofs	kind of	yes

src:Vukolic15

- In many businesses, participants must be authenticated/identified, but they do not trust each other

Roadmap

Further Reading

- Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*
- C. Decker and R. Wattenhofer, *Information Propagation in the Bitcoin Network*, in IEEE Conference on Peer-to-Peer Computing, 2013
- Developer@Bitcoin.org
- M. Nielsen, *How the bitcoin protocol actually works*, 2013
- William Stallings, *A blockchain tutorial*, in The Internet Protocol Journal, 20(3), 2017
- A. Zohar, *Bitcoin: Under the Hood*, in CACM, 58(9), 2015
- M. Vukolić, *The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication*, In International Workshop on Open Problems in Network Security