Distributed Trust and Blockchains	Date: *September 26, 2019*
Instructor: *Sujit Prakash Gujar*	Scribes: Mehul Gupta, Nikhil Vemuri

Lecture 7: Bitcoin Network, Nodes and Fork

1 Recap

In the following subsections, we provide a brief recap of the topics discussed in the previous lecture:

1.1 Other Applications of Bitcoin Scripts

1.1.1 Escrow Transactions

- It is a transaction which is used when there is a trade between two parties, a third party is introduced in the middle whose job is to ensure that both parties involved in the trade uphold their parts of the deal.
- It is implemented using MULTISIG.
- For example, if you are buying some goods from a merchant, you don't want to send the money directly to the merchant before receiving the goods and the merchant does not want to send the goods before getting paid, so you instead of sending money directly to the merchant create a MULTISIG transaction that requires two of the three people to sign for it to be complete. Those three people are you, merchant and the third party which comes into play in case of a dispute. If both you and the merchant are honest then these two will sign and complete the transaction. In case of a dispute the third party decides who gets the money.

1.1.2 Green addresses

- Green addresses are used when for some reason a user cannot check the blockchain for a transaction. For a transaction to be part of the blockchain it has to be confirmed by six blocks, which takes up to an hour. For some merchandise such as food the merchant cannot wait for an hour before delivering.
- To solve this problem, we have to introduce a third party(financial intermediary) such as a bank. The user will ask the bank to pay to the merchant on their behalf. The bank will pay the merchant from one of its green addresses. The merchant should trust this address, that the bank won't double spend and provide the service.
- Here we put a lot of trust in the bank and there is no Bitcoin-enforced guarantee. If the bank ever double spends then people will stop trusting its green addresses.

1.1.3 Micro Payments

- Bitcoin scripts is a way to to do efficient micro payments.
- Micro payments are used when a user has to continually pay a merchant for a service that the merchant provides.
- In this we start with a MULTISIG transaction that pays the maximum amount the user would ever need to spend to an output requiring both the user and the merchant to sign to release the coins. Every time the user uses the service he will sign a transaction spending those coins that were sent to the MULTISIG address, sending the amount to be paid to the merchant and the rest to himself. Whenever the user uses the service they sign such a transaction. Once the

user is done using the service, the merchant will sign the last transaction signed by the user to receive the payment.

- Here all the initial transactions are double spends but only the last transaction will make it to the blockchain the one signed by the merchant as it is the only transaction that is valid.

1.2 Bitcoin Block Structure

- Blockchain is a clever combination of two hash based data structure.
- Each block in the blockchain has a hash pointer to the previous block in the sequence, block header and a hash pointer to some transaction data(Merkle tree root). Block header contains time-stamp, bits and nonce. Each block has a Merkle tree which stores the transaction data in an efficient way.
- A block contains many transactions.

1.3 Coinbase Transaction

- It is a transaction in which the miner rewards himself.
- The value of coinbase transaction is the block reward plus all the transaction fees.
- New coins are created in this transaction(block reward).
- The input for this transaction is NULL and the output is a script that the miner can use later to redeem these bitcoins.

1.4 Joining Bitcoin Network

- You can join bitcoin network as a thin client or a full node/miner.
- Thin or SPV(Simplified Payments Verification) clients only stores block headers for payments, does not have to store the whole blockchain.
- Full nodes/miners store the whole blockchain. Full nodes not necessarily mine blocks.
- The protocol used to join the network is call Gossip protocol(TCP port 8333). You should know a node. Ask this node for its neighbours. Send the same request to these neighbours. When you see a lot of repetition stop sending this request as you have enough network.
- Eclipse attack
- Role of a Bitcoin (full) node
  - Maintain peer-to-peer network. Answer the gossip protocol query.
  - Check the transaction inputs are in UTXO set.
  - Ensure not a double spent transaction.
  - Execute the output script of input transactions along with scriptSig. If true, add it to the transaction pool.
  - If true in previous step, broadcast the transaction. If the transaction is a repeat do not broadcast it again to avoid flooding.
- UTXO are unspent transaction outputs.

2 Overview

The following topics were covered in this lecture

1. Exploring Blockchain.com
2. Incentives to join as a full node
3. Things that cannot be undone without consensus
4. Effects of block size on block propagation time
5. Changing the protocol/Forks

3 Exploring Blockchain.com

- On this website we can see the latest block mined and get all the information about it.
- We can see what all transactions are present in a block.
- We can also get information about hashrate, difficulty, transaction fees etc.

4 Incentives to Join as a Full Node

As Bitcoin is a currency it can itself be used to maintain its ledger.

4.1 Block Reward

- The node that mines a block includes a coinbase transaction in that block.
- This coinbase transaction is basically the node that mined that block paying itself for mining that block.
- In this coinbase transaction there is no input. New coins are minted which are received by the miner.
- Current block reward is 12.5 BTC per block. Initially the block reward was 50 BTC per block. The block reward becomes half after 210,000 blocks.
- Roughly the block reward becomes half every 4 years. Only 21 million bitcoins will be ever mined, so the block reward is expected to hit zero by 2140.
- Only those nodes receive block reward whose blocks get added to the main chain.

4.2 Transaction Fees

- The protocol specifies the transaction fees.
- Transaction fees is the difference between input amount and output amount of bitcoins. Input is always greater than or equal to the output.
- A miner must add a transaction meeting all the following 3 conditions
  1. Size of transaction < 1000 bytes.
  2. All the output values must be 0.01 BTC or higher.
  3. High priority.
- $Priority = (\sum input\ age\ *\ input\ value)/(transaction\ size)$
  Priority is included so that no transaction remains in the transaction pool forever.
- Practically users pay 0.7% to 1% of input value as transaction fees.

5 Things that cannot be undone without consensus

- Average duration between two consecutive blocks is ten minutes. If this time is less then the blocks will not be properly propagated.
- Maximum possible block size is 1MB.
- There are 10^8 Satoshi per Bitcoin. Satoshi is the smallest bitcoin denomination.
- Number of bitcoins that would be ever minted are 21 million.
- Bitcoin reward structure.
- Crypto primitives: Only ECDSA, Hash functions(SHA256, RIPEMD160)

6 Effect of Block Size on Block Propagation Time

- Block propagation time is the average time new block takes to propagate to every node of the network.
- It is directly proportional to the block size because of the network bottleneck.

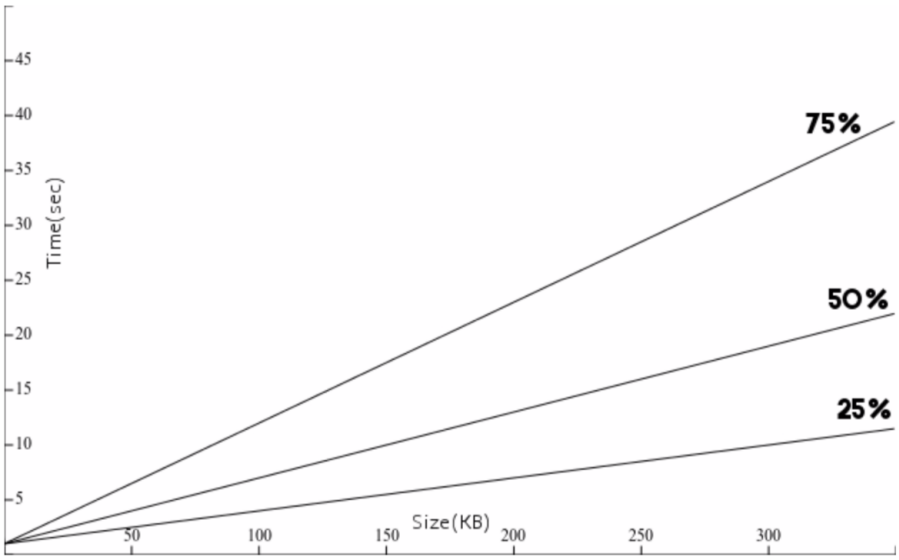

Figure 1: Shows the average time that it takes a block to reach various percentages of the nodes in the network. [1]

7 Forks

- Forks are usually done to add new features to a blockchain. There is a change in the rules.
- Fork between majority of miners and remaining minors when they don't agree on something.
- There are two types of forks.

7.1 Hard Fork

- New rules support more flexibility.
- Nodes running old software(old clients) will reject blocks mined by nodes running new software(new clients).
- New clients will support old blocks.
- The chain will split as old clients will only add to old blocks where as the new clients can add to both old and new blocks. Initially the chain of old blocks can be longer but eventually the chain of newer blocks will become longer as they are in majority. There will be two versions of the chain which will never merge.

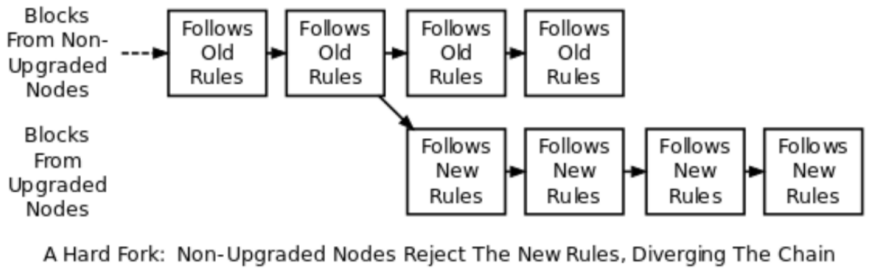

Figure 2: Hard Fork [1]

- Common UTXO set continue in both the chains, so the older coins can be spent in both the chains.
- Examples
  - Bitcoin Cash (1 August 2017) from Bitcoin
  - Bitcoin Gold (24 October 2017) from Bitcoin
  - Bitcoin SV (15 November 2018) from Bitcoin Cash

7.2 Soft Fork

- New rules are more strict.
- The blocks added by old clients will be rejected by newer clients, so every node is encouraged to update to the new software.
- Temporarily there can be two version but eventually the newer will be the longest chain.
- When a soft fork change is made, all nodes (whether upgraded or not) will continue to recognize new blocks and maintain consensus on the blockchain.

References

[1] Arvind Narayanan. Bitcoin and Cryptocurrency Technologies. *A Comprehensive Introduction.* Princeton University Press, 2017, pp. 93–94.

[1] https://bitcoin.org/img/dev/en-hard-fork.svg