Blockchain and Cryptocurrencies

Week 3 — Chapter 2: How Bitcoin Achieves Decentralization

Prof. Dr. Peter Thiemann

Albert-Ludwigs-Universität Freiburg, Germany

SS 2020

Towards Decentralization

Problem with Scroogecoin

Dependency on central agent Scrooge

Towards Decentralization

Problem with Scroogecoin

Dependency on central agent Scrooge

Questions

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?

Towards Decentralization

Problem with Scroogecoin

Dependency on central agent Scrooge

Questions

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?

Bitcoin's Answer

Towards Decentralization

Problem with Scroogecoin

Dependency on central agent Scrooge

Questions

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?

Bitcoin's Answer

- Bitcoin is organized as a network of equal nodes (i.e., a peer-to-peer network).

Towards Decentralization

Problem with Scroogecoin

Dependency on central agent Scrooge

Questions

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?

Bitcoin's Answer

- Bitcoin is organized as a network of equal nodes (i.e., a peer-to-peer network).
- Short answer to Questions 1- 4: Every node

Towards Decentralization

Problem with Scroogecoin

Dependency on central agent Scrooge

Questions

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?

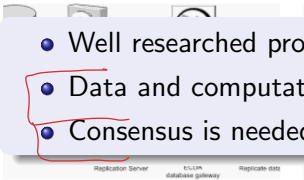
Bitcoin's Answer

- Bitcoin is organized as a network of equal nodes (i.e., a peer-to-peer network).
- Short answer to Questions 1- 4: Every node
- Chaos ensues without consensus...

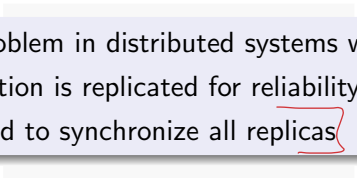
Contents

Distributed Consensus

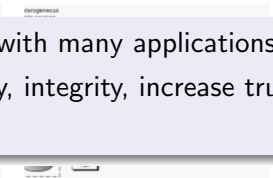
- Well researched problem in distributed systems with many applications
- Data and computation is replicated for reliability, integrity, increase trust etc
- Consensus is needed to synchronize all replicas



Heterogeneous Replication System
infocenter.sybase.com

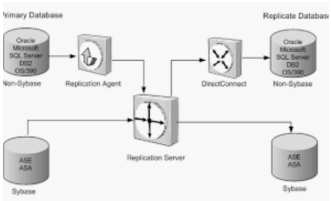


Working with SAP® Replication Server®
help.sap.com

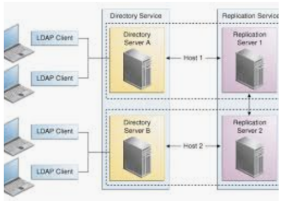


Replication Server ...
infocenter-archive.sybase.com

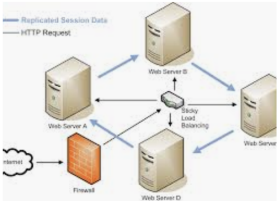
Data Replication Architectures for ...
blogs.sap.com



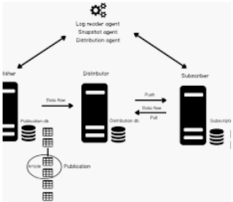
Sybase Replication deployment
infocenter-archive.sybase.com



Oracle Fusion Middleware ...
docs.oracle.com



Configuring Session Replication (Sun ...
docs.oracle.com



SQL Server replication: Overview o...
sqlshack.com

- Well researched problem in distributed systems with many applications
- Data and computation is replicated for reliability, integrity, increase trust etc
- Consensus is needed to synchronize all replicas

Distributed Consensus Protocol

Suppose there are n nodes that each have an input value.
Goal of a distributed consensus protocol: all nodes agree on an output value.

- The nodes can communicate.
- Some of the nodes are faulty / malicious.
- After finite time, all honest nodes agree on an output value.
- The output value must be generated by an honest node.

Traditional Consensus Protocols

Impossibility Results

- Byzantine generals
  impossible to achieve consensus if more than 1/3 of the generals are traitors
- Fischer-Lynch-Paterson
  no algorithm can always reach consensus in bounded time

Traditional Consensus Protocols

Impossibility Results

- Byzantine generals
  impossible to achieve consensus if more than 1/3 of the generals are traitors
- Fischer-Lynch-Paterson
  no algorithm can always reach consensus in bounded time

Main Issues

- Asynchronicity
- no global notion of time
- Determinacy (context: databases)

Breaking Bad

Bitcoin changes traditional assumptions

Breach 1: Incentives

- There are incentives for being honest
- . . . because there is a currency involved

Breaking Bad

Bitcoin changes traditional assumptions

Breach 1: Incentives

- There are incentives for being honest
- . . . because there is a currency involved

Breach 2: Embracing Randomness

- determinacy is not required
- violation of consensus is accepted but with exponentially decreasing probability

Breaking Bad

Bitcoin changes traditional assumptions

Breach 1: Incentives

- There are incentives for being honest
- . . . because there is a currency involved

Breach 2: Embracing Randomness

- determinacy is not required
- violation of consensus is accepted but with exponentially decreasing probability

Breach 3: No Identities

- creates new complications: Sybil attacks

Breaking Bad

Bitcoin changes traditional assumptions

Breach 1: Incentives

- There are incentives for being honest
- . . . because there is a currency involved

Breach 2: Embracing Randomness

- determinacy is not required
- violation of consensus is accepted but with exponentially decreasing probability

Breach 3: No Identities

- creates new complications: Sybil attacks

Fundamental assumption

We can somehow pick a random node in the network

Implicit Consensus

Simplifying assumption

It is possible to randomly select a node. Randomness is not disturbed by Sybil attacks

Bitcoin consensus algorithm (simplified)

1. Every new transaction is broadcast to all nodes

Implicit Consensus

Simplifying assumption

It is possible to randomly select a node. Randomness is not disturbed by Sybil attacks

Bitcoin consensus algorithm (simplified)

1. Every new transaction is broadcast to all nodes
2. Every node collects new transactions into a block

Implicit Consensus

Simplifying assumption

It is possible to randomly select a node. Randomness is not disturbed by Sybil attacks

Bitcoin consensus algorithm (simplified)

1. Every new transaction is broadcast to all nodes
2. Every node collects new transactions into a block
3. Each round, a **random node** is selected to broadcast its block

Implicit Consensus

Simplifying assumption

It is possible to randomly select a node. Randomness is not disturbed by Sybil attacks

Bitcoin consensus algorithm (simplified)

1. Every new transaction is broadcast to all nodes
2. Every node collects new transactions into a block
3. Each round, a **random node** is selected to broadcast its block 
4. Other nodes accept the broadcast block if all its transactions are valid (unspent, signed)

Implicit Consensus

Simplifying assumption

It is possible to randomly select a node. Randomness is not disturbed by Sybil attacks

Bitcoin consensus algorithm (simplified)

1. Every new transaction is broadcast to all nodes
2. Every node collects new transactions into a block
3. Each round, a **random node** is selected to broadcast its block
4. Other nodes accept the broadcast block if all its transactions are valid (unspent, signed)
5. Acceptance of the block is expressed by including it in the hash of the next block

Possible Threats

Stealing Bitcoins

Q: If Alice gets to propose the next block, can she steal coins?

Possible Threats

Stealing Bitcoins

Q: If Alice gets to propose the next block, can she steal coins? A: No. To do so

- she'd have to create a valid transaction
- she'd have to forge the sender's signature
- but we assumed a secure cryptographic signature scheme

Possible Threats, II

Denial of Service Attack

Q: If Alice doesn't like Bob, can she defer his transactions forever?

Possible Threats, II

Denial of Service Attack

Q: If Alice doesn't like Bob, can she defer his transactions forever? A: No.

- Alice may ignore Bob's transactions entirely, but
- Bob waits for the next honest node chosen at random

Possible Threats, III

Double Spending

Scenario

- Bob sells digital downloads
- Alice wants to buy from Bob's webshop
- Alice pays with Bitcoin
- Alice downloads the ware

Questions

1. Can Alice spend her coin twice?
2. When is it safe for Bob to let Alice download the ware?

Double Spending, I

Can Alice spend her coin twice?

1. Alice broadcasts a transaction to pay Bob
2. This transaction is included in the next broadcast block
3. Alice gets to propose the subsequent block
4. She includes a transaction to spend the same coin elsewhere
5. Alice ignores the previously broadcast block
6. Only one of the blocks will be accepted by the network, eventually

Double Spending, II

When is it safe for Bob to let Alice download the ware?

Double Spending, II

When is it safe for Bob to let Alice download the ware?

Bob sees Alice's transaction

- Alice could immediate propose further transaction(s) for the same coins
- Gamble which transaction gets included in the next block

Double Spending, II

When is it safe for Bob to let Alice download the ware?

Bob sees Alice's transaction

- Alice could immediate propose further transaction(s) for the same coins
- Gamble which transaction gets included in the next block

Alice's transaction accepted in head block

- Alice could have proposed further transactions
- another head block might be elected by the majority of nodes

Double Spending, II

When is it safe for Bob to let Alice download the ware?

Bob sees Alice's transaction

- Alice could immediate propose further transaction(s) for the same coins
- Gamble which transaction gets included in the next block

Alice's transaction accepted in head block

- Alice could have proposed further transactions
- another head block might be elected by the majority of nodes

Best Practice

- Each new block on top is further confirmation for a transaction
- If a transaction has received k confirmations, then the probability that this transaction will **not** stay in the blockchain goes down exponentially as a function in k.
- Recommendation on the Bitcoin network: $k = 6$

Contents

Incentives

Q: Can we give nodes an incentive to behave honestly?

Incentives

Q: Can we give nodes an incentive to behave honestly?

Can we penalize nodes that accept double spend transactions?

Incentives

Q: Can we give nodes an incentive to behave honestly?

Can we penalize nodes that accept double spend transactions?

Can we reward nodes whose blocks remain on the blockchain?

Incentives

Q: Can we give nodes an incentive to behave honestly?

Can we penalize nodes that accept double spend transactions?

Can we reward nodes whose blocks remain on the blockchain?

- Bitcoin has two mechanisms for that: block rewards and transaction fees

Incentive 1: Block Reward

- block creator can include a coin-creating transaction that generates the **block reward**
- block reward
  - started at 50 bitcoins/block
  - halves every 210000 blocks
  - corresponds to about four years
- status as of 11th of May, 2020: block reward is 6.25 bitcoins
- timing considerations
  - goal: one block every 10 minutes (hence four years)
  - block rewards runs out in 2140
  - no more coin creation thereafter: cap of 21 million bitcoins
- why does it work?

Further Financial Implications

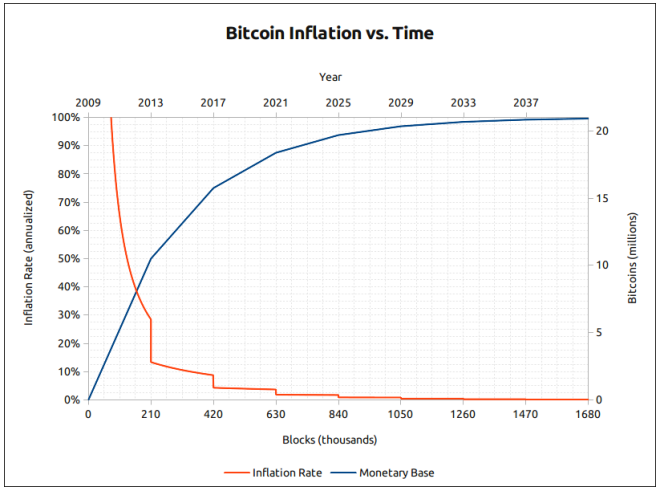


Image source https://www.bitcoinblockhalf.com/images/bitcoin-inflation-chart.png

Deflationary currency

- total amount of bitcoins is predetermined
- hence, it will keep or even increase its value
- contrast with behavior of fiat currency

Incentive 2: Transaction Fees

Transaction Fee

- total output of a transaction may be less than the total input
- the block creator can cash the difference

Incentive 2: Transaction Fees

Transaction Fee

- total output of a transaction may be less than the total input
- the block creator can cash the difference

Status

- transaction fees are voluntary
- inclusion may increase quality of service
- may become mandatory as block rewards shrink

Mining and Proof Work

Remaining Issues

- How can we pick the random node?
- How do we throttle a rush to try and create bitcoin blocks?
- How can we avoid Sybil attacks?

Mining and Proof Work

Remaining Issues

- How can we pick the random node?
- How do we throttle a rush to try and create bitcoin blocks?
- How can we avoid Sybil attacks?

One answer: Proof of Work

- Proof of Work: select nodes in proportion to their computing power
- Set up a competition among all nodes
- (We'll discuss alternatives later)

Proof of Work in Bitcoin

Hash Puzzle

Task: create a block

Input: content of the block (previous hash and all transactions) and a **target**

Procedure: find a **nonce** such that

$$H(\mathbf{nonce} \| \mathbf{prev\_hash} \| \mathbf{tx}_1 \| \dots \mathbf{tx}_n) < \mathbf{target}$$

where H is a given hash function

Proof of Work in Bitcoin

Hash Puzzle

Task: create a block

Input: content of the block (previous hash and all transactions) and a **target**

Procedure: find a **nonce** such that

$$H(\mathbf{nonce} \| \mathbf{prev\_hash} \| \mathbf{tx}_1 \| \dots \mathbf{tx}_n) < \mathbf{target}$$

where H is a given hash function

Remark

- **nonce** to be included in the block
- puzzle friendliness of H implies that there is no better algorithm than brute force trial and error to find **nonce**, but checking is efficient
- **target** can be used to gauge the difficulty of the puzzle

Mining

Solving hash puzzles competitively to collect the block reward and the transaction fees.

Probability that Alice succeeds mining depends on the fraction of global hash power that she controls.

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

Gauging the Difficulty

- the current_target at end May 2020:
  0000000000000000001297f6000000000000000000000000000000000000000000

Gauging the Difficulty

- the current_target at end May 2020:
  0000000000000000001297f6000000000000000000000000000000000000000000
- target gets recalculated every 2016 blocks to obtain an average time of 10 minutes between blocks

Gauging the Difficulty

- the current_target at end May 2020:
  00000000000000000001297f6000000000000000000000000000000000000000
- target gets recalculated every 2016 blocks to obtain an average time of 10 minutes between blocks
- it gets adjusted by the factor time_actually_taken / time_alloted_for_2016_blocks, but limited to the interval $[0.25, 4]$

Gauging the Difficulty

- the current_target at end May 2020:
  00000000000000000001297f6000000000000000000000000000000000000000
- target gets recalculated every 2016 blocks to obtain an average time of 10 minutes between blocks
- it gets adjusted by the factor time_actually_taken / time_alloted_for_2016_blocks, but limited to the interval $[0.25, 4]$

$\Rightarrow$ if mining power increases, then mining gets harder

Gauging the Difficulty

- the current_target at end May 2020:
  00000000000000000001297f6000000000000000000000000000000000000000
- target gets recalculated every 2016 blocks to obtain an average time of 10 minutes between blocks
- it gets adjusted by the factor time_actually_taken / time_alloted_for_2016_blocks, but limited to the interval $[0.25, 4]$

$\Rightarrow$ if mining power increases, then mining gets harder

- first block's target value difficulty_1_target:
  00000000ffff0000000000000000000000000000000000000000000000000000

Gauging the Difficulty

- the current_target at end May 2020:
  00000000000000000001297f6000000000000000000000000000000000000000
- target gets recalculated every 2016 blocks to obtain an average time of 10 minutes between blocks
- it gets adjusted by the factor time_actually_taken / time_alloted_for_2016_blocks, but limited to the interval $[0.25, 4]$

$\Rightarrow$ if mining power increases, then mining gets harder

- first block's target value difficulty_1_target:
  00000000ffff0000000000000000000000000000000000000000000000000000
- the **difficulty** is computed as follows

```
1 difficulty = difficulty_1_target / current_target
```

Contents

Cost of Mining

Question

Is mining worth the effort?

Cost of Mining

Question

Is mining worth the effort?

Mining reward

$$\text{mining reward} = \text{block reward} + \text{transaction fees}$$

Cost of Mining

Question

Is mining worth the effort?

Mining reward

$$\text{mining reward} = \text{block reward} + \text{transaction fees}$$

Mining cost

$$\text{mining cost} = \text{hardware cost} + \text{operating cost}$$

Cost of Mining

Question

Is mining worth the effort?

Mining reward

$$\text{mining reward} = \text{block reward} + \text{transaction fees}$$

Mining cost

$$\text{mining cost} = \text{hardware cost} + \text{operating cost}$$

Further considerations

- exchange rate of bitcoin, block rate, ...
- cost of electricity, taxes, property rental, ...

Orphan Blocks

- A transaction is “included” in the blockchain if it is sufficiently often confirmed
- Certainty is never achieved
- An **orphan block** is a block that does not make it in the confirmed chain
  - might contain an invalid transaction
  - might contain a double spend attempt (so it could have been part of a mined block)
  - might be caused by network latency

Implications of Distributed Consensus

All aspects of Bitcoin are subject to DC

- Exchange rate
- State of the ledger (i.e., all account balances)
- Alice's balance = sum of balances of all account identities controlled by Alice
- Rules of the system (halving, transaction fees, target recalculation)
  but no fixed procedure, rather external agreement (or disagreement)
  $\Rightarrow$ soft forks / hard forks

How do bitcoins obtain exchange value?

- Question of bootstrapping the currency
- Initially: no value
- Factors: security of the blockchain, health of the mining ecosystem, value of the currency

$\Rightarrow$ social process of accumulating (relative) trust in the currency

- Beginning of Bitcoin
  - No currrency value
  - No miners except Nakamoto

  $\Rightarrow$ Insecure because anyone could have hijacked the mining process

The 51 Percent Attack

Consider an attacker Don that controls >50% of the mining capacity ...

- Don can perform denial of service attacks
- Don cannot fake transactions (digital signatures are secure)
- Don cannot include faulty transactions (e.g., double spending; nobody else would accept them $\Rightarrow$ soft fork)
- Don cannot change the infrastructural constants like the block reward (it would lead to a hard fork)
- Don's acting might disrupt trust in Bitcoin due to the fork

Thanks!