Blockchain Principles and Applications

Amir Mahdi Sadeghzadeh, PhD

Data and Network Security Lab (DNSL)
Trustworthy and Secure AI Lab (TSAIL)

Most slides have been adapted from ECE/COS 470, Pramod Viswanath, Princeton 2024 and CS251, Dan Boneh, Stanford 2023.

Recap

Decentralized Blockchain

Block: Header + Data + Signature

Header: Pointer to previous block = hash of the previous block header and Merkle root of data of previous block

Data: information specific to the block

Signature: one of the users signs the block (header+data)

List of signatures known ahead of time: **permissioned** blockchains

Questions:

1. How is this list known ahead of time?
2. Which user in this list gets to add which block?
3. Who polices this?

Leader Election: Oracle

Time is organized into **slots**

Oracle selects one of the nodes (public identities)

random

everyone can verify the unique *winner*

The selected node is the proposer in that slot

constitutes a block with transactions

validates transactions

includes hash pointer to previous block

signs the block

Proof of Work

Practical method to simulate the Oracle

Mining

cryptographic hash function creates computational puzzle

Hash(nonce, block-hash) < Threshold

nonce is the proof of work

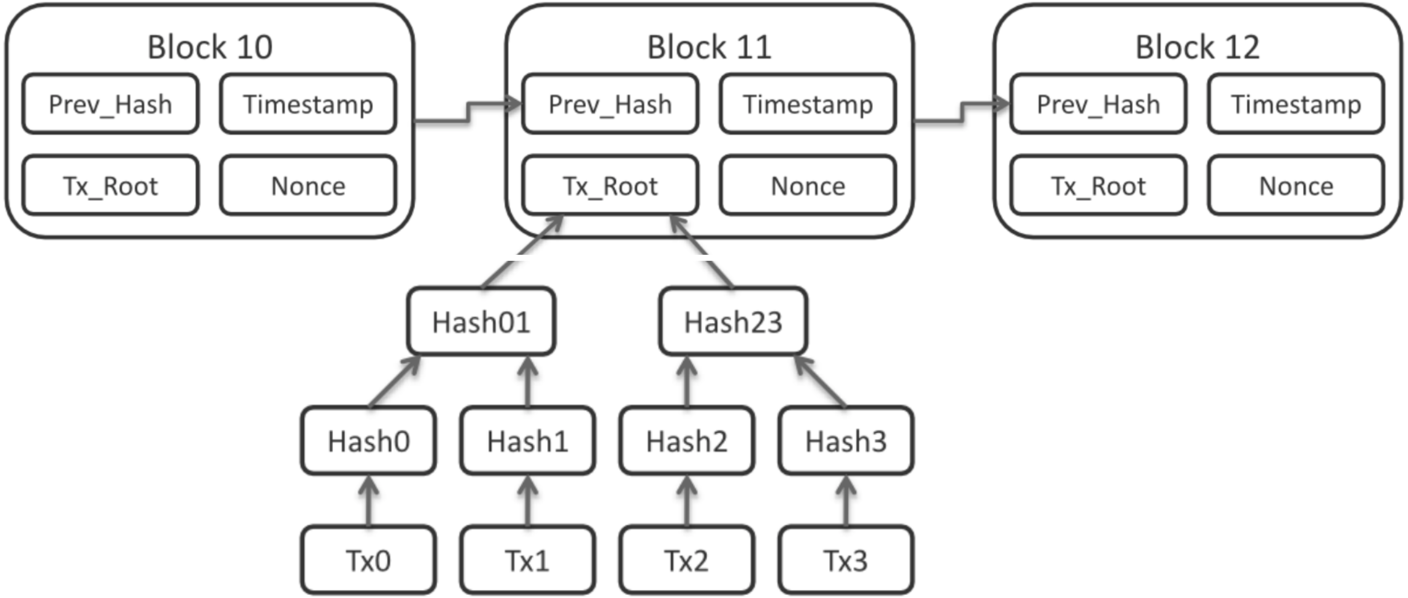
include nonce inside the block

Threshold

chosen such that a block is mined successfully on average once in 10 minutes

a successfully mined block will be broadcast to all nodes in the network

Block Constituents



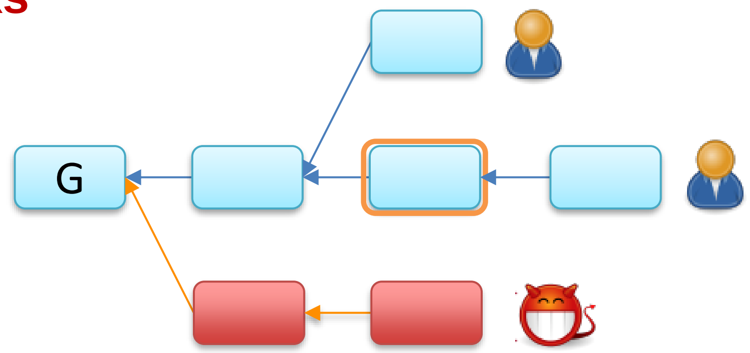
Longest Chain Protocol

Where should the mined block hash-point to?

However, blockchain may have **forks**

because of network delays

because of adversarial action



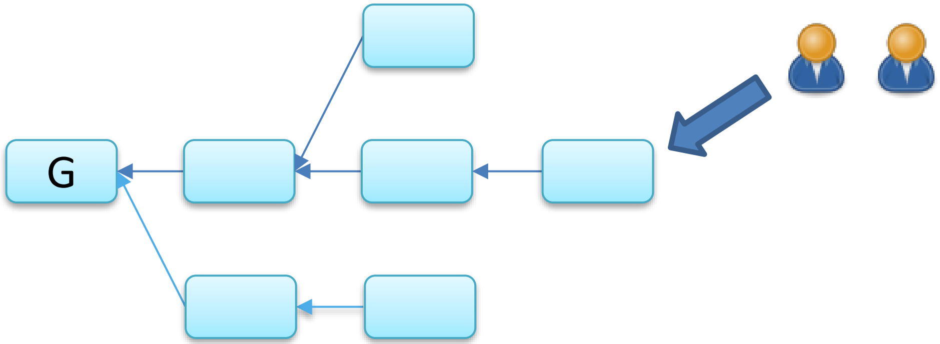
Longest Chain Protocol

Where should the mined block hash-point to?

Blockchain may have **forks**

because of network delays

because of adversarial action



Longest chain protocol

attach the block to the leaf of the longest chain in the block tree

Why Variable Difficulty

Block Difficulty

Example: in September 2022 the mining target or threshold (in hexadecimal) is:

$$0x\ \underbrace{0000000000000000000}_{\text{19 leading zeros}}8c8940000000000000000000000000000000000000000$$

The hash of any valid block must be below this value ~ 8/16 . 16^{-19} = 2^{-77}

Difficulty of a block:

Block_difficulty = 1/mining_target

Bitcoin Rule

(a) The mining difficulty changes every 2016 blocks

next_difficulty = (previous_difficulty * 2016 * 10 minutes) / (time to mine last 2016 blocks)

(b) Adopt the heaviest chain instead of the longest chain

chain_difficulty = sum of block_difficulty

(c) Allow the difficulty to be adjusted only mildly every epoch

¼ < next_difficulty/previous_difficulty < 4

Peer-2-Peer Networking

Networking Requirements

No centralized server (single point of failure, censorship)

Key Primitive

Broadcast blocks and transactions to all nodes

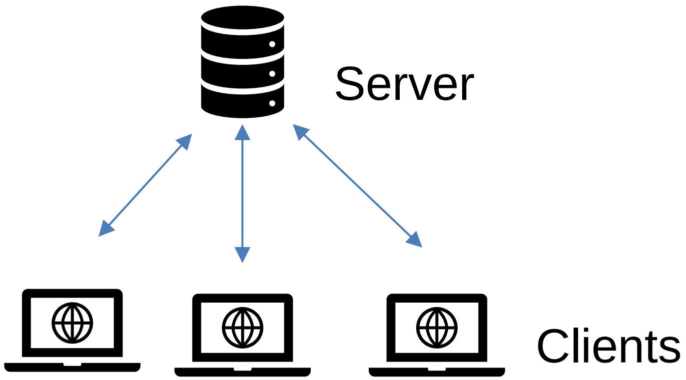
Robustness

some nodes go offline

new nodes join

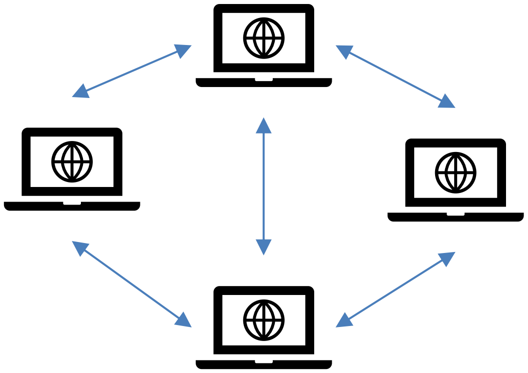
Types of Network Architecture

Client server



Server stores most of the data

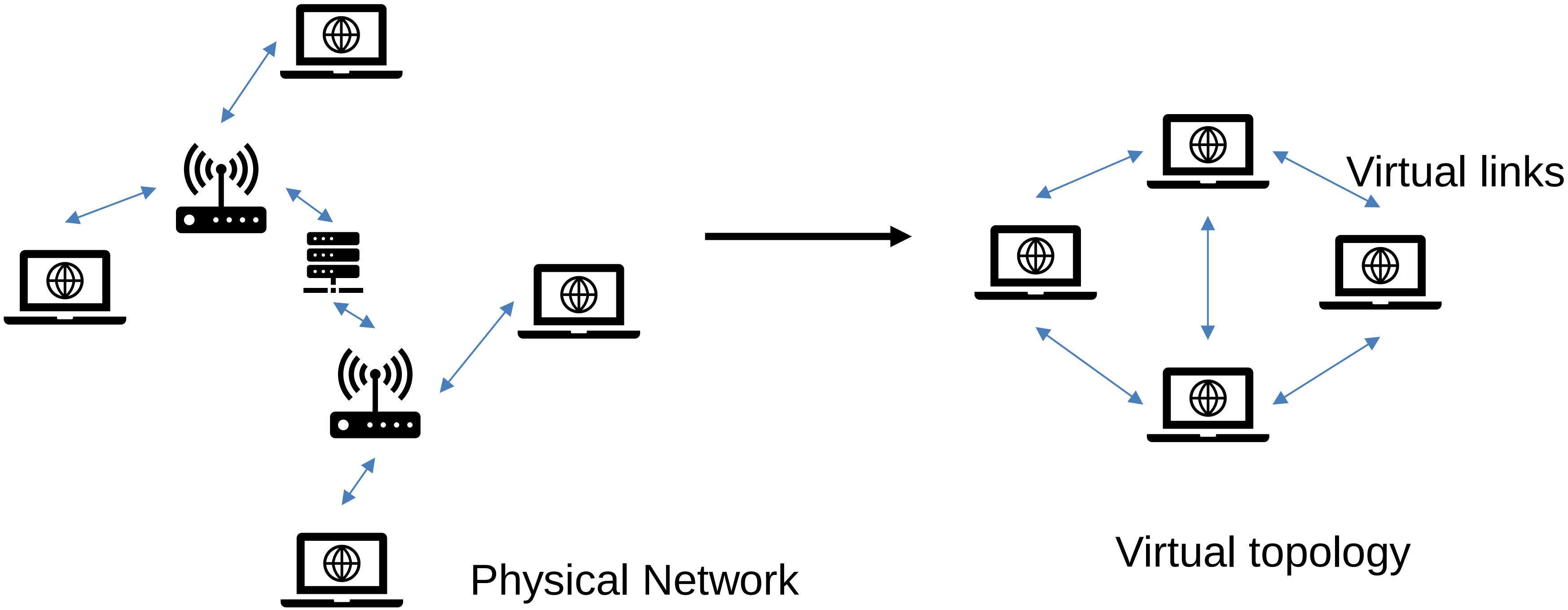
Peer to Peer



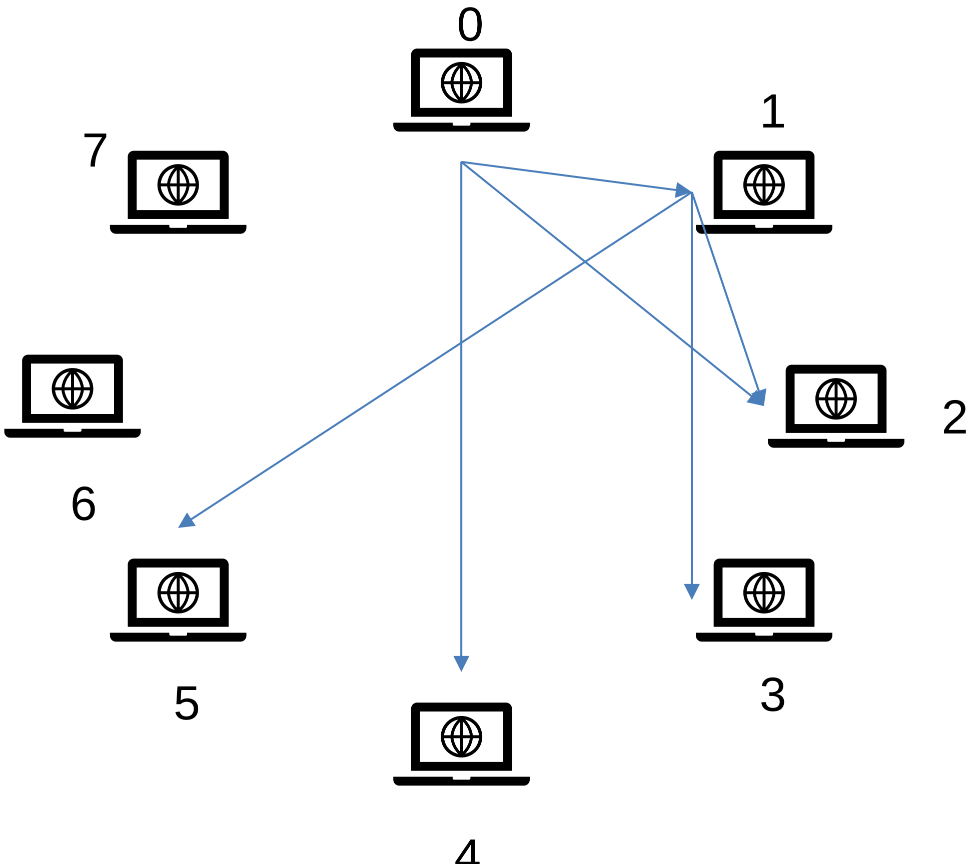
Each node acts as a client and a server

BitTorrent, Napster

Overlay Networks

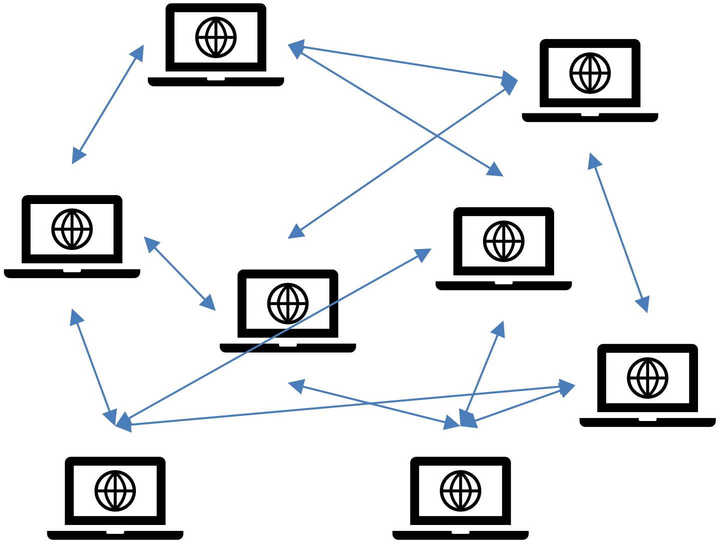


Structured Overlay Networks



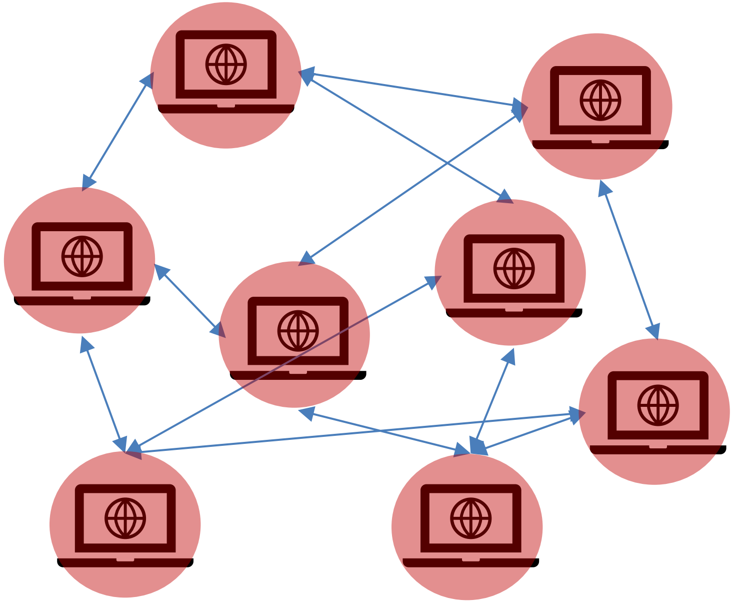
- Example: CHORD
- Assign a graph node identifier to each node
- Well defined Peer routing rules
- O(log N) routing
- O(log N) connections per node

Unstructured Overlay Networks



- Example: d-regular graph
- No node graph identifier
- Connect to any random d-nodes
- O(log N) routing (difficult to route)
- O(1) connections per node
- O(log N) broadcast

Gossip and Flooding



- Mimics the spread of an epidemic
- Once a node is “infected”, it “infects” its peers
- Information Spread exponentially and reaches nodes in O(log(N)) time

Expander graph

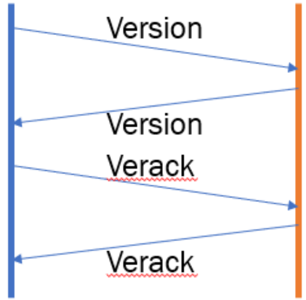
- Well connected but sparse graph
- Sparse graph G(V,E): |E| = O(|V|)
- Expander graph: $|\partial A| \geq \varepsilon|A|$
  - $|\partial A|$ = number of vertices outside A with at-least one neighbor in A.
- **A random d-regular (d $\geq$3) graph for large |V| is an expander graph** (with high probability)
- Intuition for O(log N) broadcast

Bitcoin network

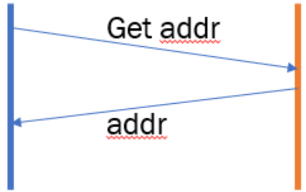
- TCP connection with peers
- At most 8 outbound TCP connections
- May accept up to 117 inbound TCP connections
- Maintains a large list of nodes (IP, port) on the bitcoin network
- Establishes connection to a subset of the stored nodes

Peer discovery

- DNS seed nodes (Hard coded in the codebase)
- Easy to be compromised, do not trust one seed node exclusively
- Hardcoded peers (fallback)
- Ask connected peers for additional peers



Connecting to a peer

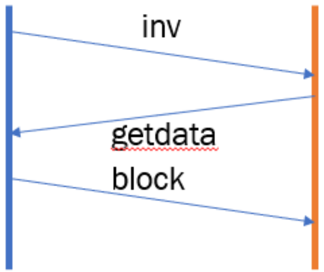


Gathering additional peers
Addr: contains list of up to 1000 nodes

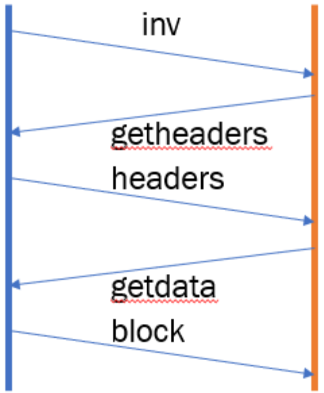
Block transmission

- Block is broadcasted to the network using gossip-flooding
- Standard block relay protocol to gossip blocks
- Relay after block validation
- Inv(blockhash): inventory message containing blockhash

- Block-First
- Getdata asks for the same block as inv
- Can download orphan blocks and keep it in memory



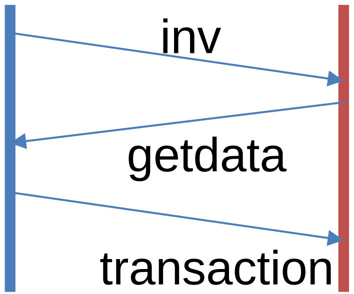
or



- Header-First
- Getheaders asks for the same block as inv or a few parent headers (in case of orphan block)
- Will not download orphan blocks if no header chain established

Data Broadcast

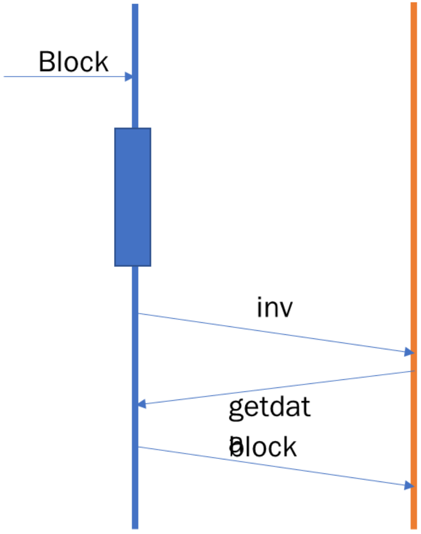
- Data (transactions) broadcasted using Gossip-flooding
- Each node maintains a non-persistent memory to store unconfirmed tx (mempool)
- inv(txid): Check if peer has a transaction with id: txid in mempool



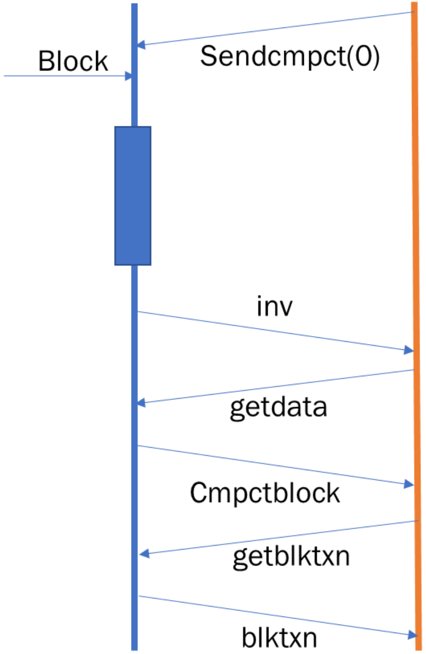
- Some unconfirmed tx might be removed from mempool

Compact blocks

Compact blocks



Legacy relaying



Compact block relaying

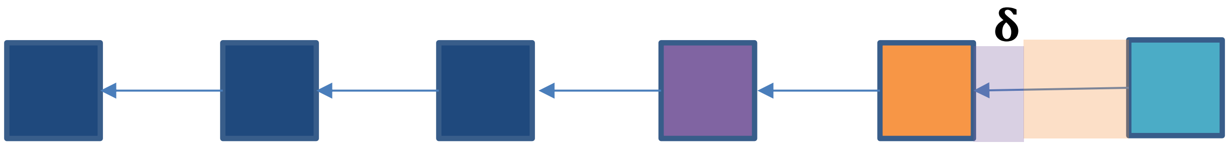
- Legacy relaying sends transactions twice
- Guess the mempool of the receiving node
- Compact block has block header, txids, some full transactions
- The receiving node sends getblktxn to receive missing transactions

Capacity and Propagation Delay

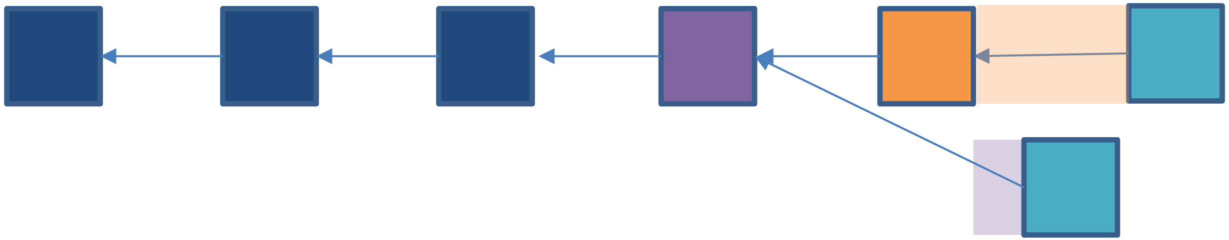
- C = communication/processing capacity of the network (tx/s)
- D = speed-of-light propagation delay
- End to end delay
  1. Propagation delay: D
  2. Processing delay: B/C (where B is block size in tx)
  3. Queuing delay
- Delay increases with increase in block size

Effects of delay

- Wasted hashpower



- Forking



Disadvantages of current p2p network

- Efficiency
  - Not efficient (total communication is O(Nd))
- Privacy
  - Can link transaction source to IP address
- Security
  - Plausible deniability for forking

Improving P2P Network Topology

- Random IP network
  - Not related to geographic distances
- Need a **geometric** random network
  - IP addresses do not necessarily reveal location
- Challenge
  - Self-adapting network topology based on measurements

Perigee

- A self-adaptive network topology algorithm
  - Goal: mimic random geometric network

- Decentralized algorithm that selects neighbors based on past interactions

  retain neighbors that relay blocks fast

  disconnect from neighbors that do not relay blocks fast

  explore unseen neighbors

- Motivated by the multi-armed bandit problem
  - Explore vs exploit tradeoff

Perigee Algorithm

- Assign scores for each subset of neighbors based on how fast they relay blocks
- Retain subset neighbors with best score
- Disconnect node not in the subset
- Form a connection to a random neighbor

Resources

- ECE/COS 470, Pramod Viswanath, Princeton 2024
- CS251, Dan Boneh, Stanford 2023