MONASH University

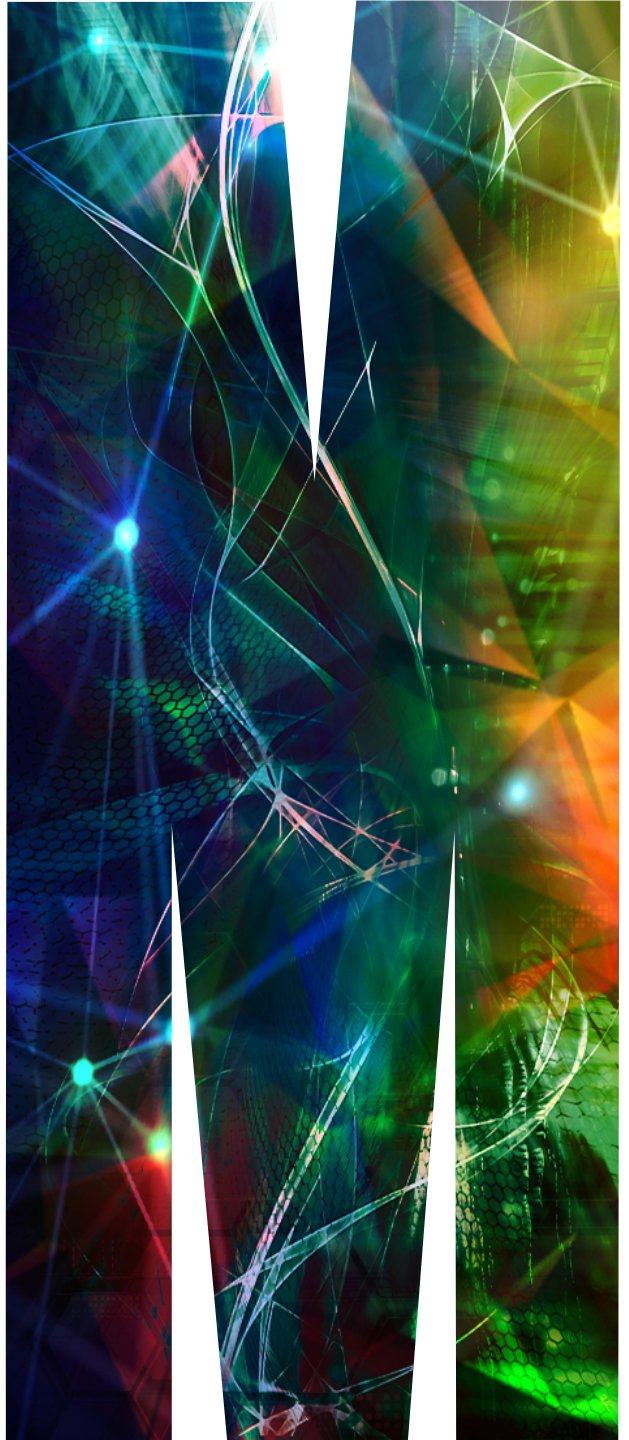
FIT5214: Blockchain

Lecture 11: Blockchain Network

Lecturer: Rafael Dowsley

rafael.dowsley@monash.edu

https://dowsley.net



Unit Structure

- Lecture 1: Introduction to Blockchain
- Lecture 2: Bitcoin
- Lecture 3: Ethereum and Smart Contracts
- Lecture 4: Proof-of-Work (PoW)
- Lecture 5: Attacks on Blockchains
- Lecture 6: Class Test/Alternatives to PoW
- Lecture 7: Proof-of-Stake (PoS)
- Lecture 8: Privacy
- Lecture 9: Byzantine Agreement
- Lecture 10: Algorand
- Lecture 11: Blockchain Network
- Lecture 12: Payment Channels

Unit Structure

- Lecture 1: Introduction to Blockchain
- Lecture 2: Bitcoin
- Lecture 3: Ethereum and Smart Contracts
- Lecture 4: Proof-of-Work (PoW)
- Lecture 5: Attacks on Blockchains
- Lecture 6: Class Test/Alternatives to PoW
- Lecture 7: Proof-of-Stake (PoS)
- Lecture 8: Privacy
- Lecture 9: Byzantine Agreement
- Lecture 10: Algorand
- Lecture 11: Blockchain Network
- Lecture 12: Payment Channels

Your Feedback

Your feedback is extremely important to us! You have a chance to provide:

(1) formal feedback about your learning experience

SETU: Go directly to https://monash.bluera.com/monash or follow the link from Moodle sidebar

(2) your recognition to a teaching staff/unit

Teaching Award Nomination: https://www.intranet.monash/it/education/ed-quality/awards

MONASH University

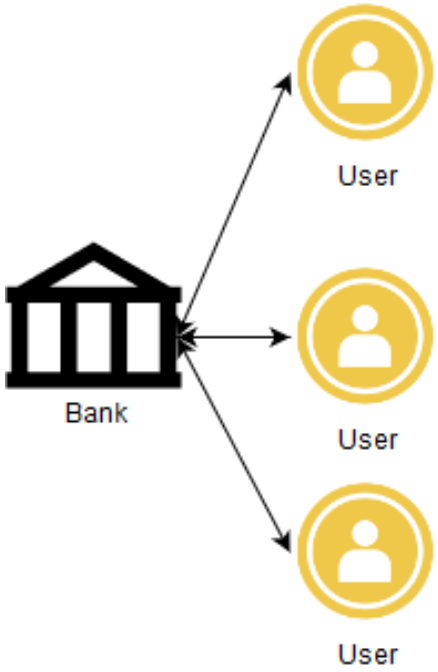
Blockchain Network

Agenda

1. Centralisation v.s. Decentralisation. (Why P2P network?)
2. Blockchain Network Overview. (What is a blockchain network?)
3. Network Protocol. (How does it work?)

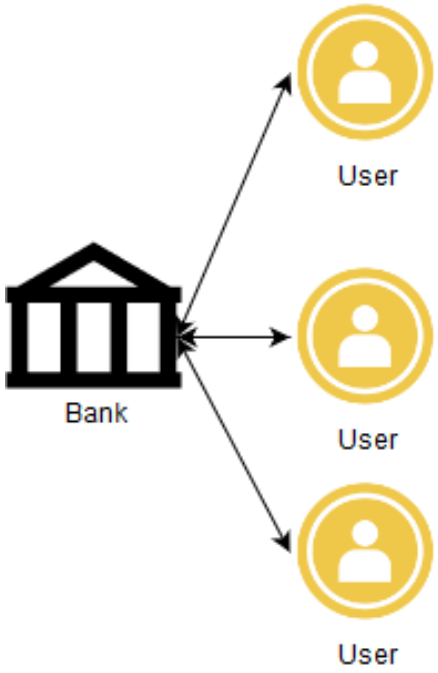
1. Centralisation vs Decentralisation. (Why P2P network?)

Centralised System



1. Trusted third party helps users to verify transactions.
2. Trusted third party protects users' privacy.
3. Trusted third party guarantees honest users' safety and security to avoid the attacks of malicious nodes.

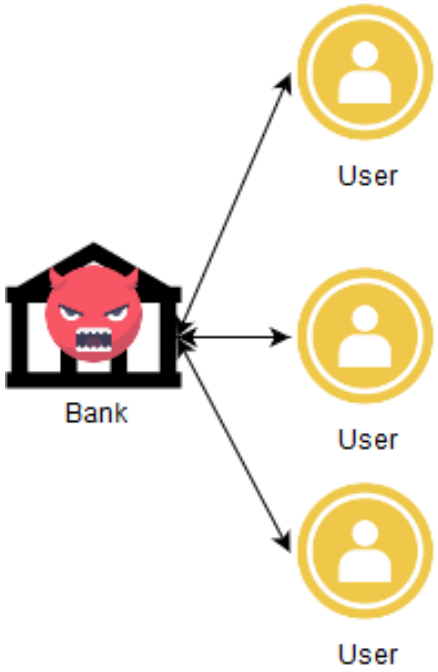
Centralised System



1. Trusted third party helps users to verify transactions.
2. Trusted third party protects users' privacy.
3. Trusted third party guarantees honest users' safety and security to avoid the attacks of malicious nodes.

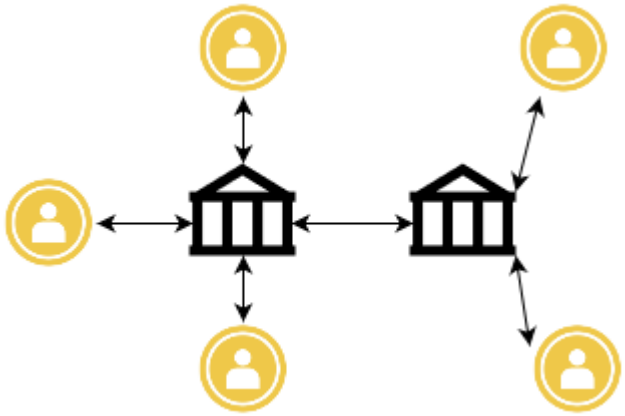
~~Trusted~~ third party

Centralised System



1. The centralised authority could reject any transaction without reason.
2. The centralised authority could modify any user's account.
3. The system will be collapsed if the centralised authority fails (single point of failure).

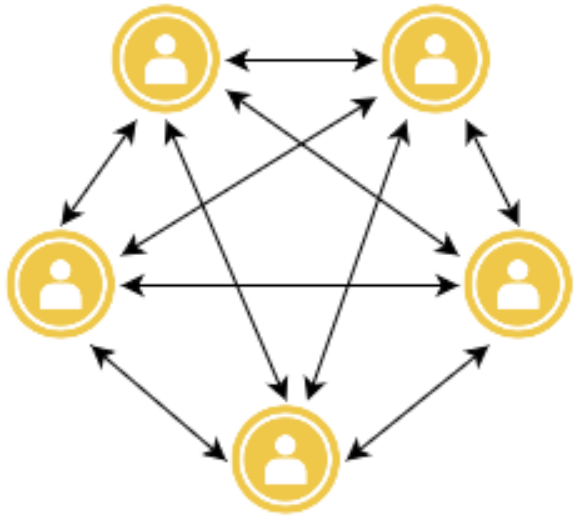
Decentralised/Distributed System



Decentralised System

1. No single point of failure.
2. System correctness is guaranteed by multiple nodes.

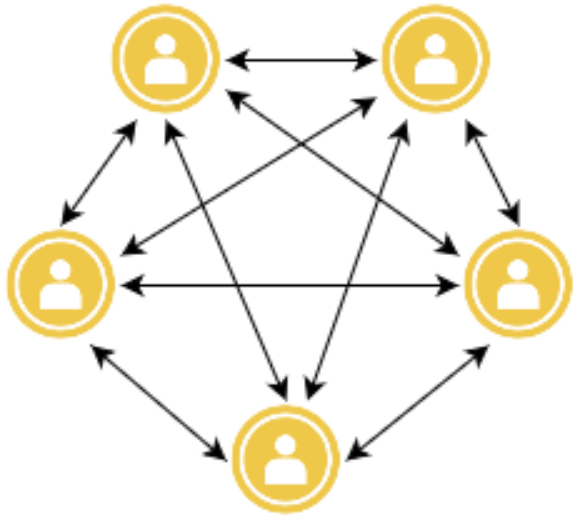
Decentralised/Distributed System



Distributed System

1. No single point of failure.
2. System correctness is guaranteed by the majority of nodes.

Decentralised/Distributed System



Distributed Network

1. No single point of failure.
2. System correctness is guaranteed by the majority of nodes.

A fully-connected distributed network is robust against failures, however, what if there are more than 10M nodes? Not every node can maintain more than 10M connections.

Peer to Peer (P2P) Network

1. Each node maintains a number of connections depending on its connectivity.
2. Each node follows a membership discovery protocol to refresh its connections.
3. Each node randomly selects a known peer as its new neighbour.

Source: http://www.processmodelcanvas.com/network/

Peer to Peer (P2P) Network

Source: http://www.processmodelcanvas.com/network/

1. Each node maintains a number of connections depending on its connectivity.
2. Each node follows a membership discovery protocol to refresh its connections.
3. Each node randomly selects a known peer as its new neighbour.

Blockchain peer to peer network is a dynamic distributed/decentralised network, that can disseminate messages in order to tolerant faults.

Peer to Peer (P2P) Network

Bitcoin is a peer-to-peer digital cash system by design, and the network architecture is both a reflection and a foundation of that core characteristic.

Source: http://www.processmodelcanvas.com/network/

2. Blockchain Network Overview

Blockchain network consists of different nodes to generate, validate, and disseminate messages.

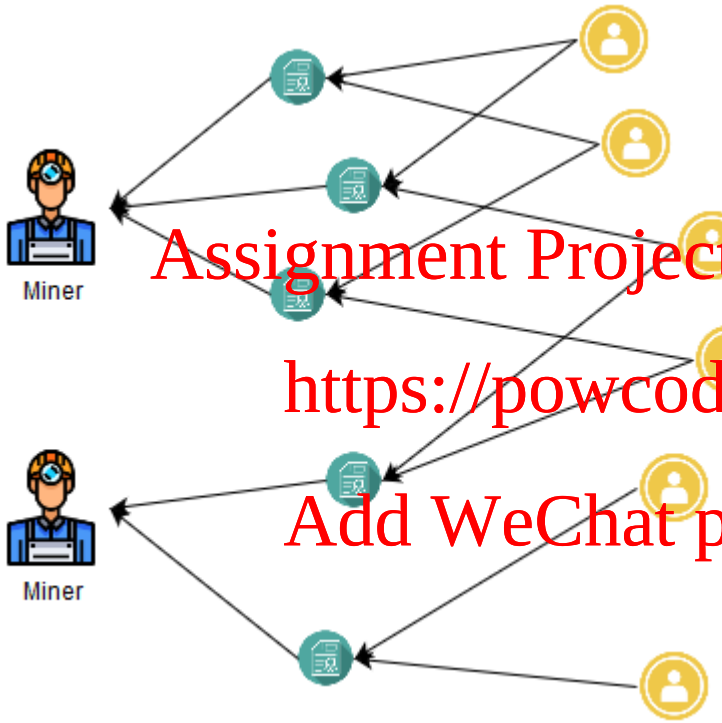
Participants

Users can exchange coins with each other.

Participants

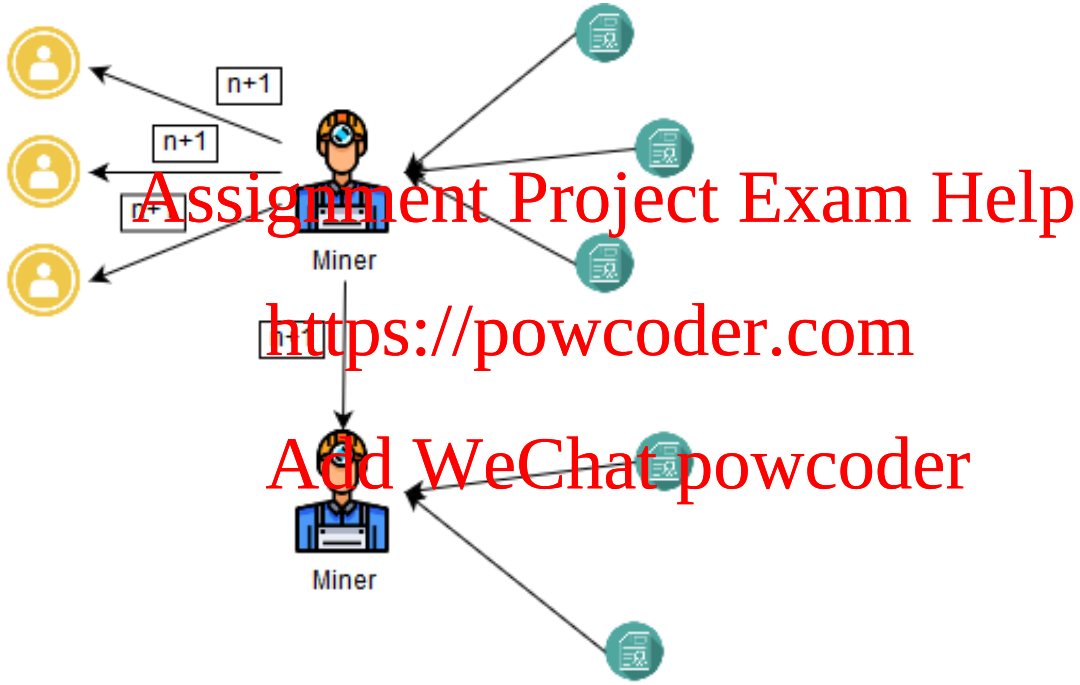
Users then broadcast the new transaction to their neighbours.

Participants



Miners select different transactions to do a computing competition. The transactions will be recorded in blocks. Only the winner can add his block into the chain of blocks.

Participants

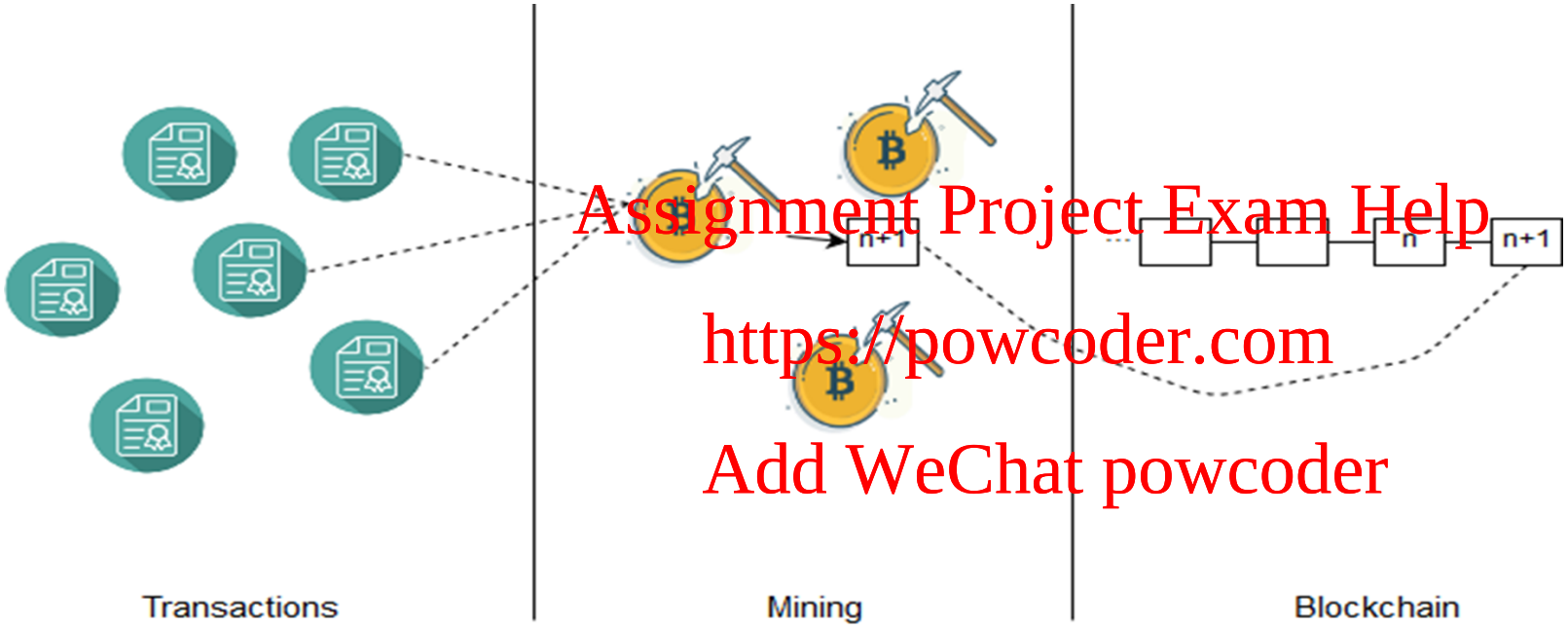


As long as a miner discovers new block (n+1), it immediately "broadcasts" this block to all its neighbours, which could be normal users or other miners.

Participants

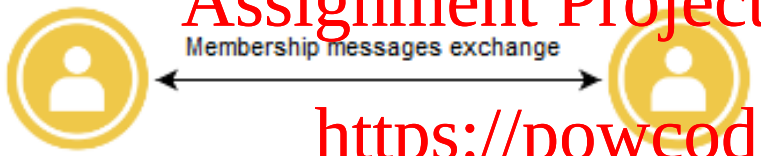
- Blockchain network consists of normal users and miners.
- Mining pool owns multiple front-end nodes to connect with users or other miners/pools.
- Each participant is allowed to have different connections.

Messages



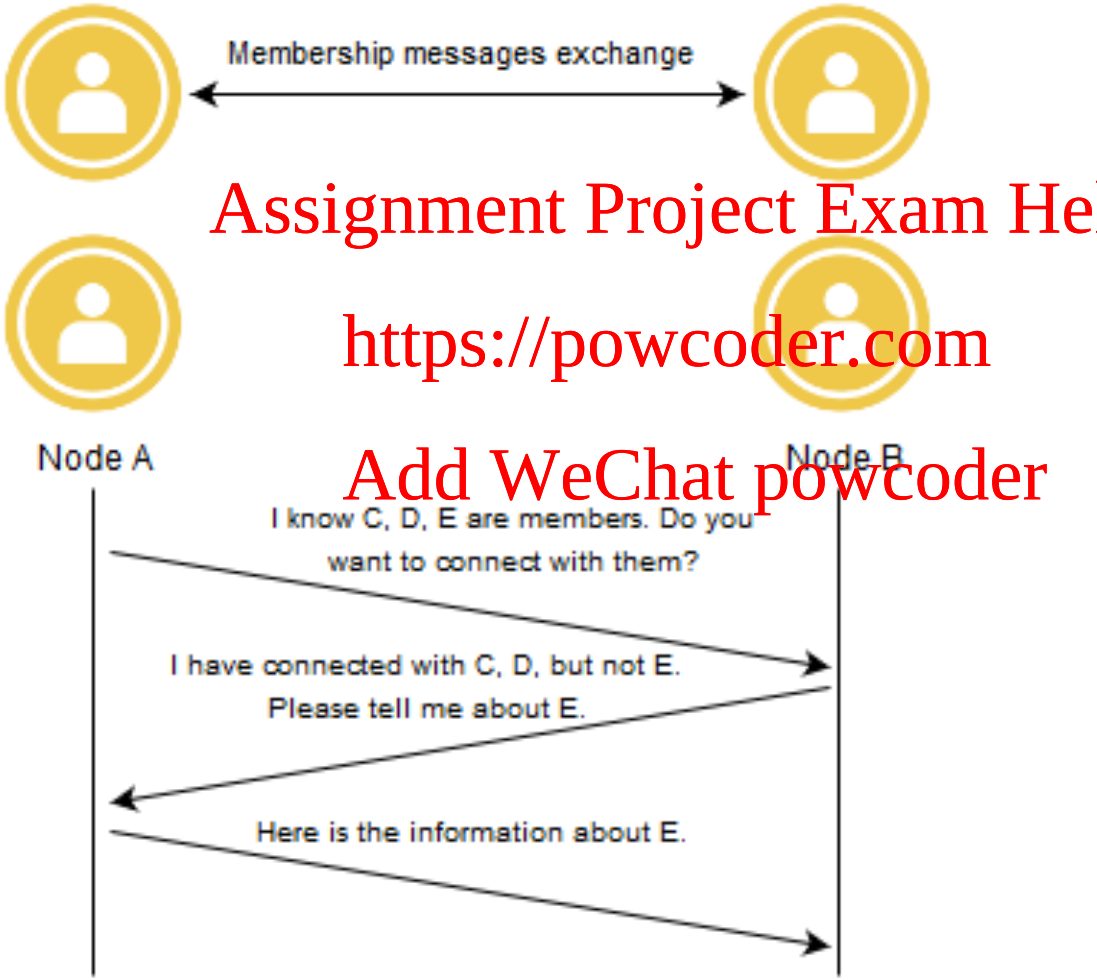
- Transactions are generated from users, and validated by miners.
- Blocks are generated from miners/mining pools, and validated by the majority of participants.

Messages

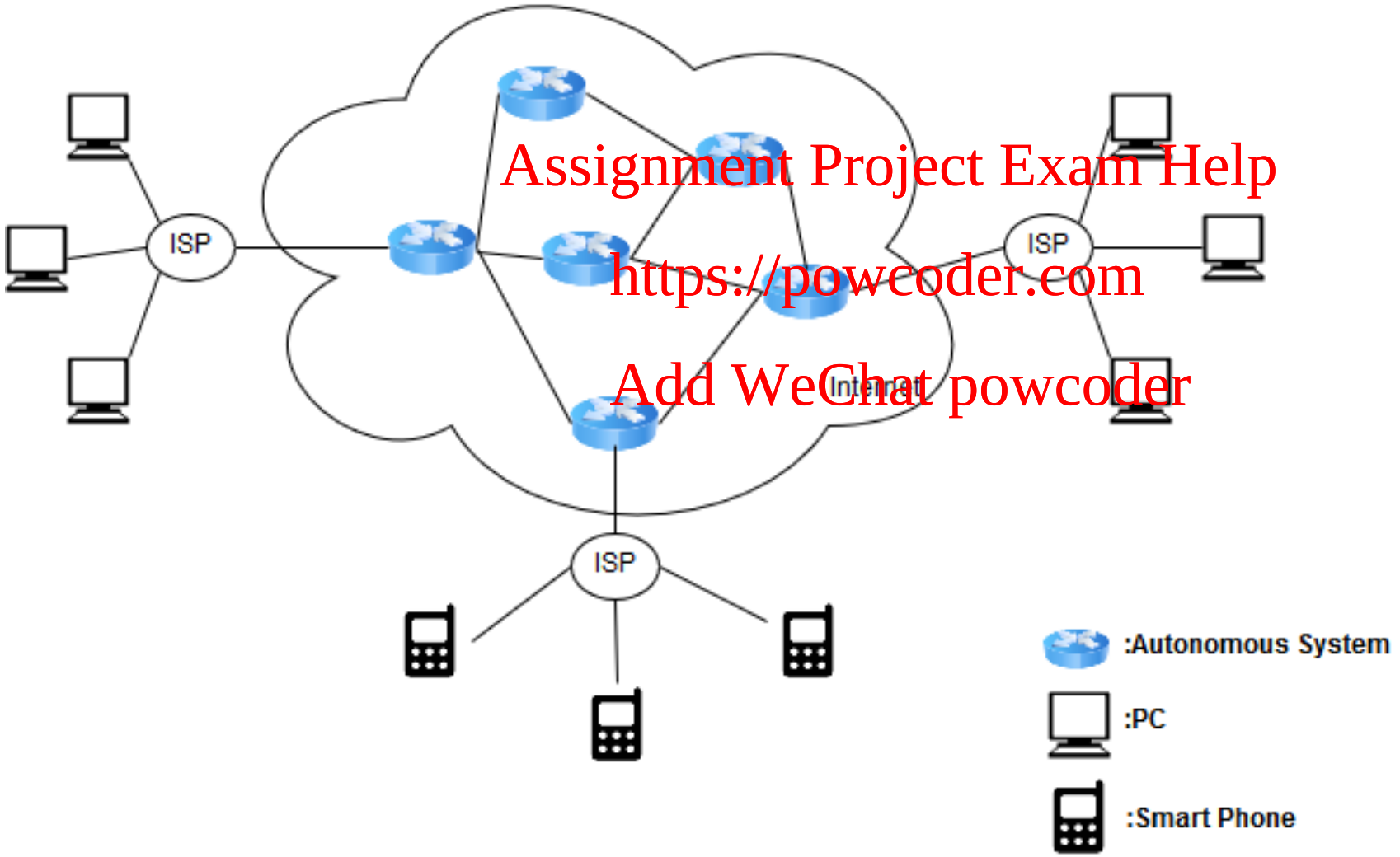


Each participant owns its unique membership message. Participants can find new peers by exchanging membership messages.

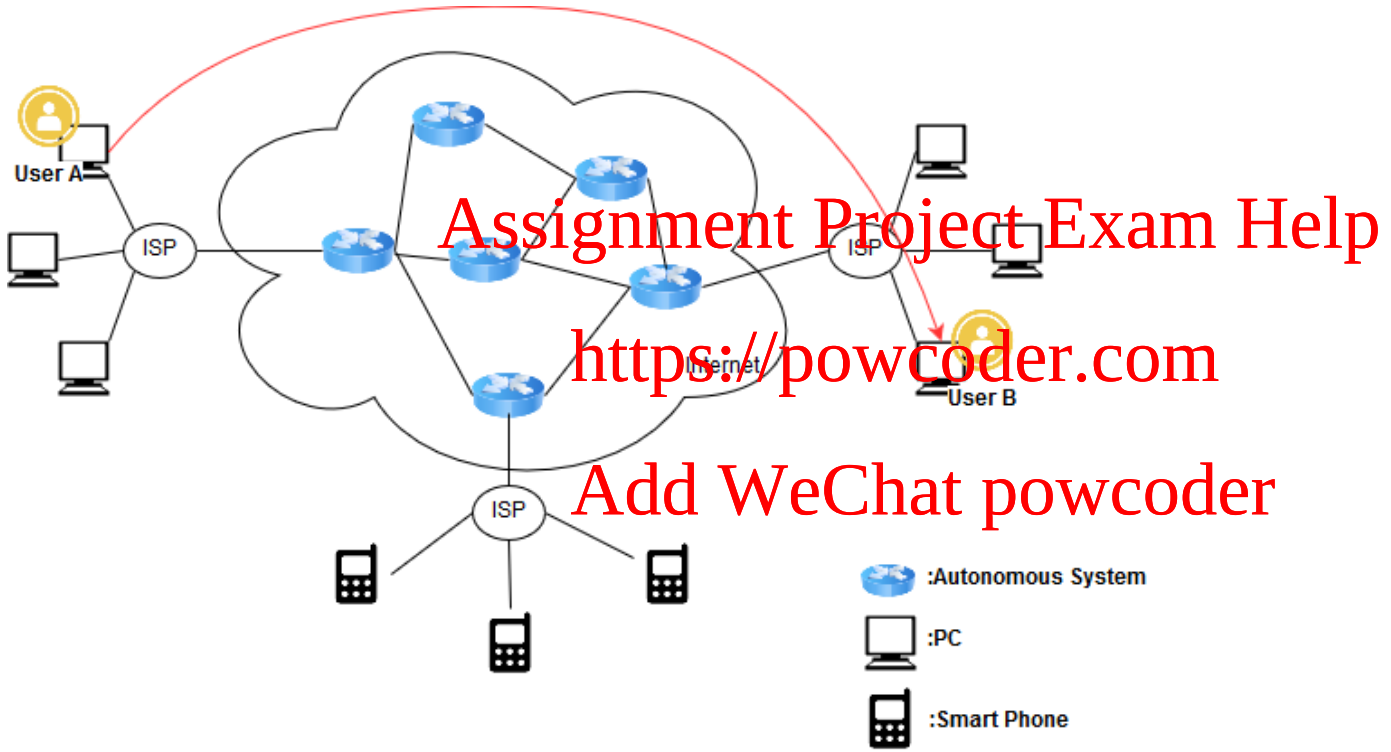
Messages



Network Infrastructure

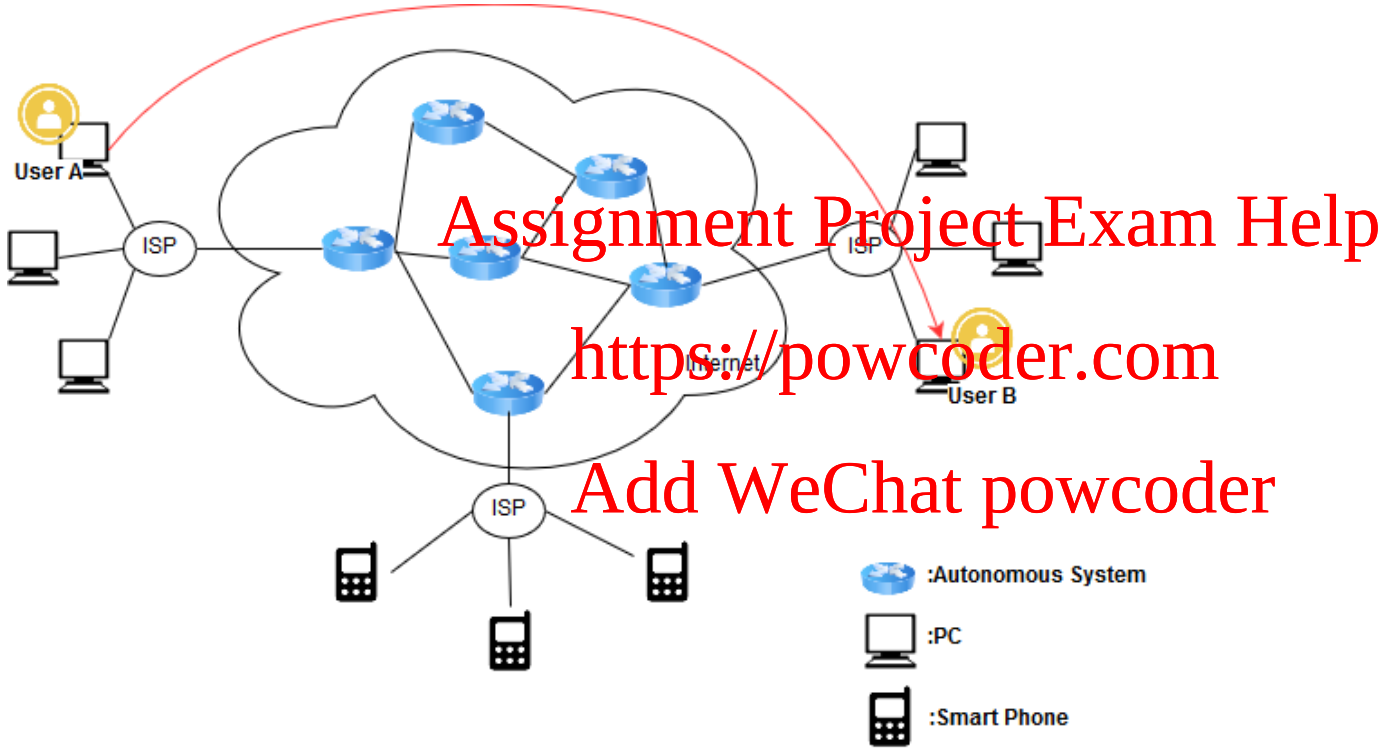


Network Infrastructure



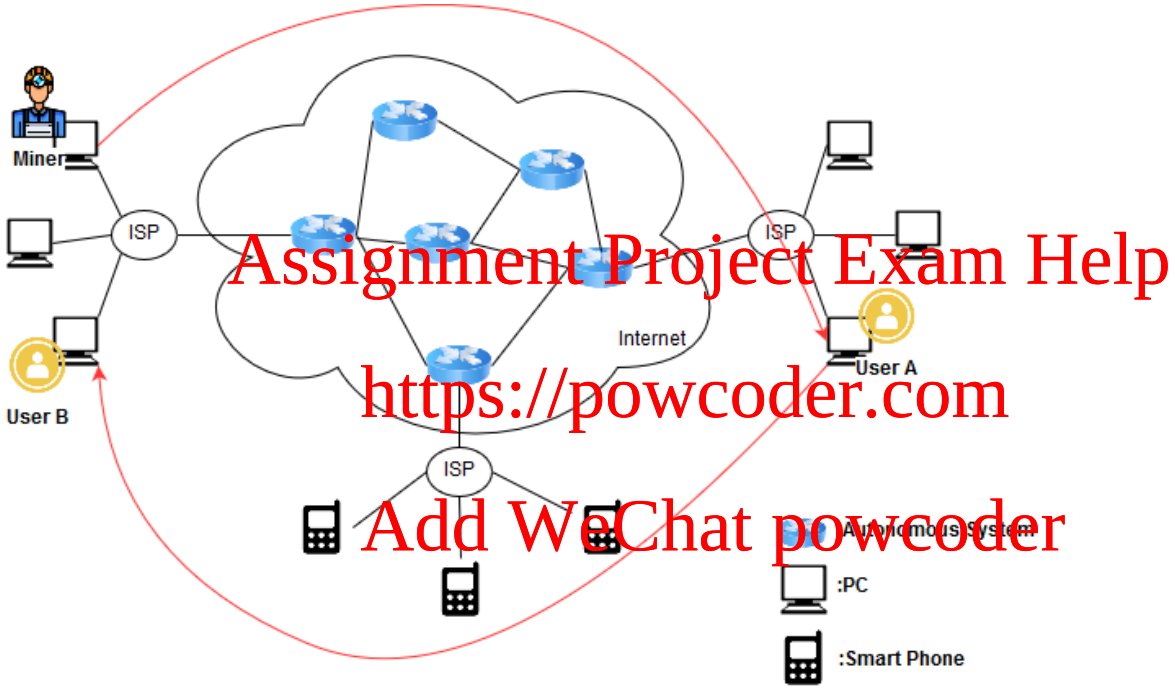
User B is the neighbour of user A in this P2P network. The messages from user A to user B have to go through several hops in Internet.

Network Infrastructure



Bitcoin and other blockchain platforms are structured as a peer-to-peer network architecture on top of the Internet (overlay network).

Network Infrastructure



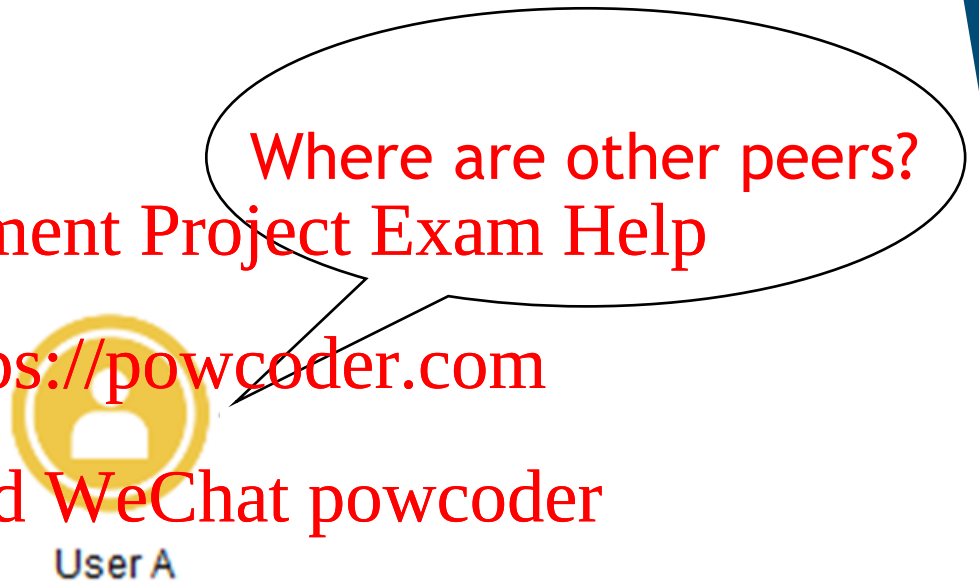
As long as a block is found by a miner, it will immediately be sent to the neighbours in P2P overlay network, and then relay to neighbours' neighbours.
The block messages have to go through several hops in Internet.

3. Peer to Peer Protocol

Connection Initialisation

If the node is new, then it initialises the connections through:

- Hardcoded seed nodes.
- Hardcoded DNS nodes.




```
vSeeds.emplace_back("seed.bitcoin.sipa.be"); // Pieter Wuille, only supports x1, x5, x9, and xd
vSeeds.emplace_back("dnsseed.bluematt.me"); // Matt Corallo, only supports x9
vSeeds.emplace_back("dnsseed.bitcoin.dashjr.org"); // Luke Dashjr
vSeeds.emplace_back("seed.bitcoinstats.com"); // Christian Decker, supports x1 - xf
vSeeds.emplace_back("seed.bitcoin.jonasschnelli.ch"); // Jonas Schnelli, only supports x1, x5, x9, and xd
vSeeds.emplace_back("seed.btc.petertodd.org"); // Peter Todd, only supports x1, x5, x9, and xd
vSeeds.emplace_back("seed.bitcoin.sprovoost.nl"); // Sjors Provoost
vSeeds.emplace_back("dnsseed.emzy.de"); // Stephan Oeste
```

maintained by the core developers

Source: github/bitcoin

Connection Initialisation

If the node joined the network before, then it initialises the connections through:

- Its peer list.

If there are no active nodes in its peer list, then through:

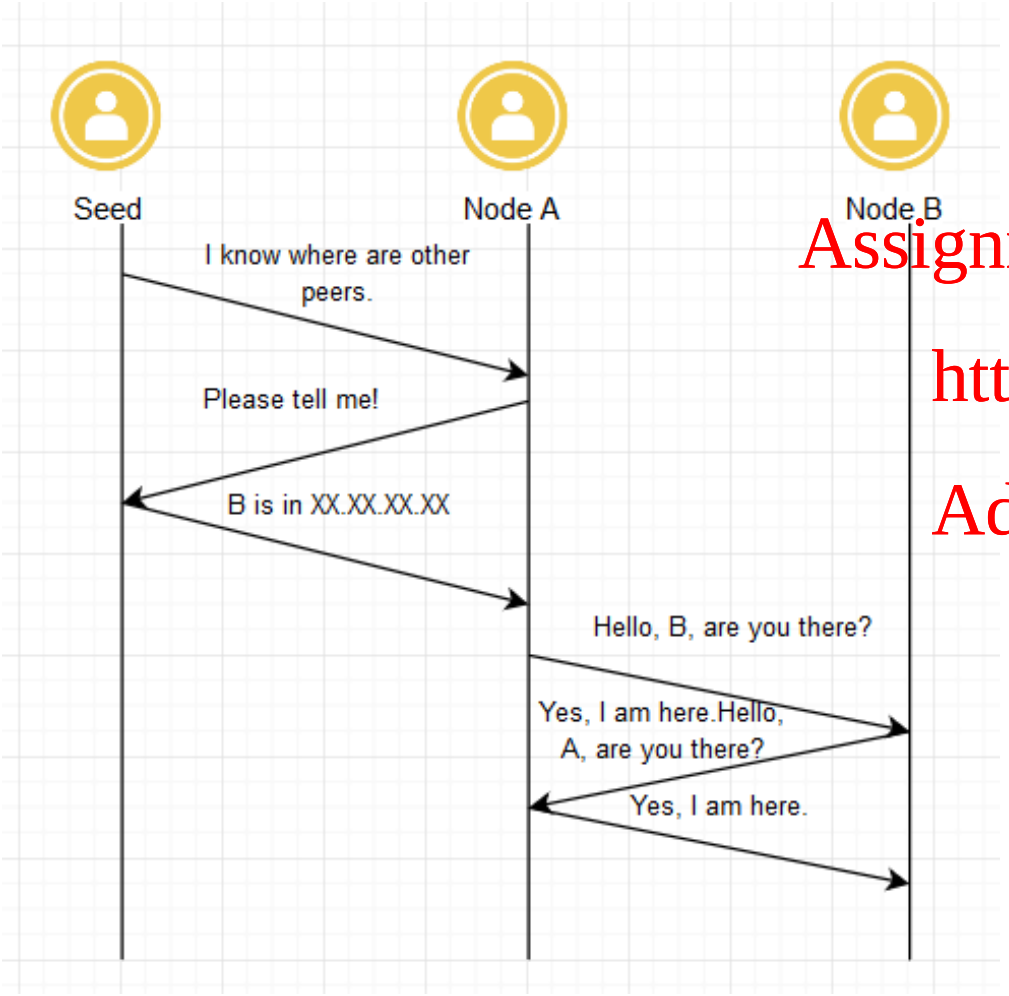
- Hardcoded seed nodes.
- Hardcoded DNS nodes.

User A

```
vSeeds.emplace_back("seed.bitcoin.sipa.be"); // Pieter Wuille, only supports x1, x5, x9, and xd
vSeeds.emplace_back("dnsseed.bluematt.me"); // Matt Corallo, only supports x9
vSeeds.emplace_back("dnsseed.bitcoin.dashjr.org"); // Luke Dashjr
vSeeds.emplace_back("seed.bitcoinstats.com"); // Christian Decker, supports x1 - xf
vSeeds.emplace_back("seed.bitcoin.jonasschnelli.ch"); // Jonas Schnelli, only supports x1, x5, x9, and xd
vSeeds.emplace_back("seed.btc.petertodd.org"); // Peter Todd, only supports x1, x5, x9, and xd
vSeeds.emplace_back("seed.bitcoin.sprovoost.nl"); // Sjors Provoost
vSeeds.emplace_back("dnsseed.emzy.de"); // Stephan Oeste
```

Source: github/bitcoin

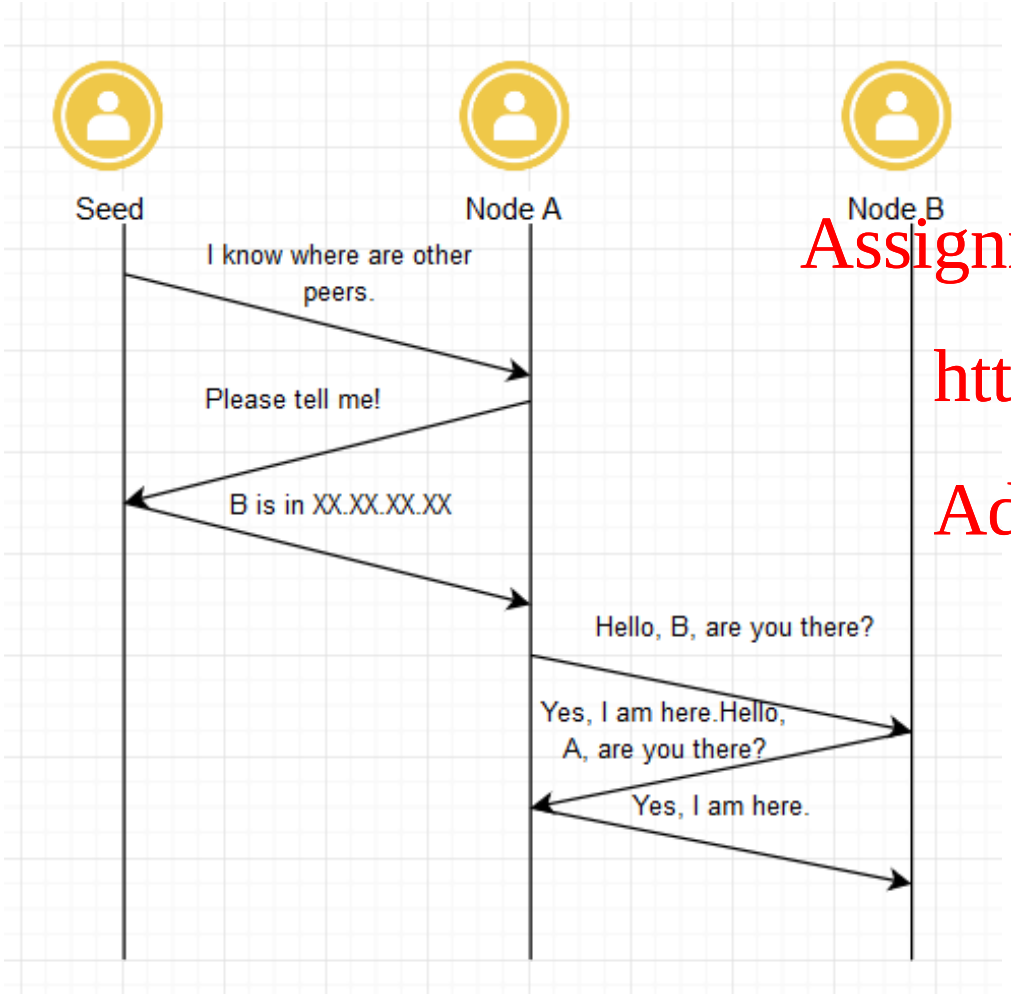
Membership Discovery



- Each node learns other members' information from its neighbours.
- Each node maintains a database to manage members' information.
- Each node disseminates its members' information to its neighbours.

If a connection is dropped, the node can select a member from the database, and establish a new connection.

Membership Discovery



For each participant:

- The memberships are dynamic.
- Database is limited.
- There is a timing-based eviction mechanism to keep the freshness.
- Only its neighbours can receive its membership messages.

Communication Protocol

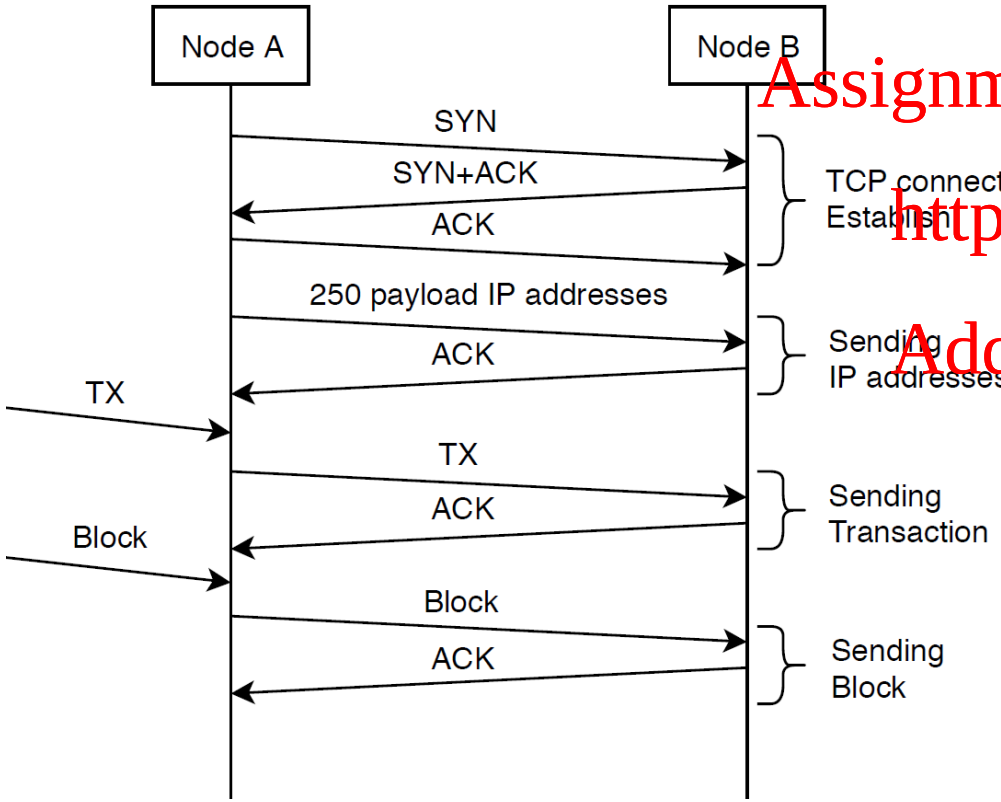
- TCP.
- UDP.
- Gossip.
- Stratum protocol: for supporting pooled mining.

Communication Protocol

TCP	UDP
Connected (handshake used)	Connectionless
Reliable	Lossy
Error Free	Error Packets Discarded
Ordered Data Delivery	No Sequence Guarantee
Slower	Faster

Communication Protocol

- TCP

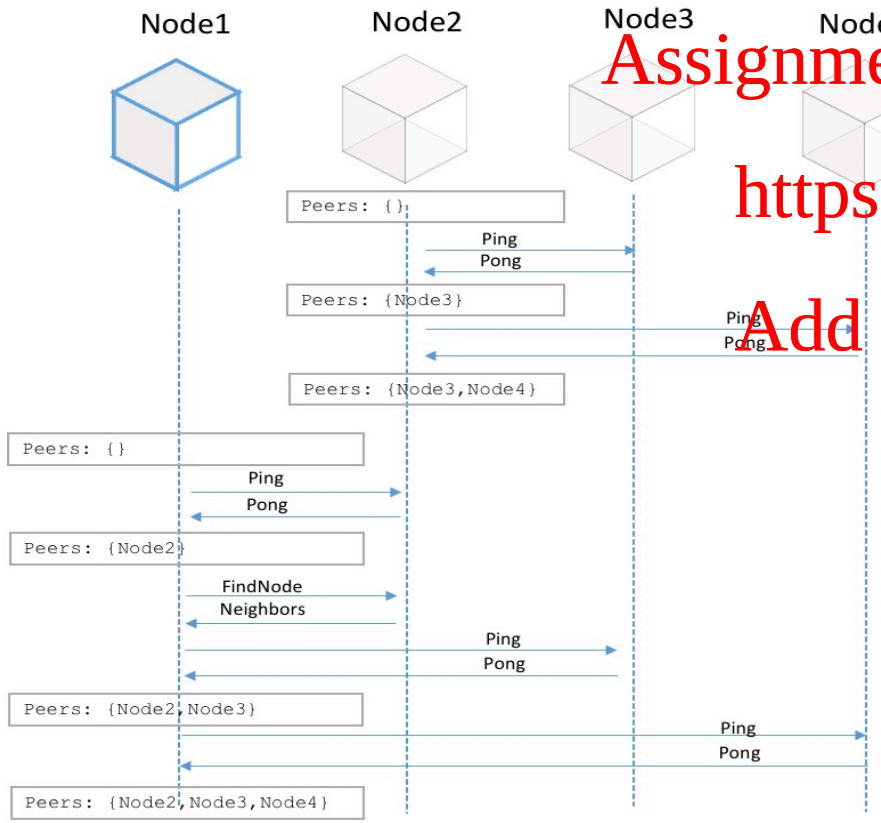


Most blockchain networks are using TCP.

Membership, blocks and transactions messages can be sent using TCP.

Communication Protocol

- UDP



Ethereum defines 4 types UDP-based messages: Ping, Pong, FindNode, and Neighbors for membership discovery.

Source: http://block.bendakhlia.org/main-blockchains/blockchains/ethereum/ethereum-discovery

Broadcast

- Bitcoin uses a multi-hop broadcast to propagate transactions and blocks through the network (i.e., each node propagates the information to its neighbours, and so on).
- At each hop in the broadcast the message incurs a propagation delay (sum of transmission time plus verification time)

Broadcast

- After receiving a valid block or transaction in the Bitcoin network, a node sends an “inv” message to its neighbours to check if they already have the transaction or block.
- If the neighbour doesn’t have, it sends a “getdata” response back to request the details of the transaction or block.
- “getblocks” message allows a peer which has been disconnected or started for the first time to request the blocks it hasn’t seen (at most 500 blocks are sent in the answer to “getblocks”, possibly multiple “getblocks” messages are needed).

Additional Reading

- https://developer.bitcoin.org/reference/p2p_networking.html
- Information Propagation in the Bitcoin network https://ieeexplore.ieee.org/document/6688704

Next Week

- Payment Channels