Principles of Blockchains
Princeton University,
Professor: Pramod Viswanath

Lecture 4

Peer to Peer Network and Bitcoin system

July 23, 2023

Contents

Chapter 1

Peer to Peer Network and Bitcoin system

1.1 Blockchains and Networking

A blockchain network is a system that provides ledger and smart contract services to applications without relying on a centralized server or authority. A centralized server can be a single point of failure or a target for censorship, which can compromise the security and availability of the network. Therefore, a blockchain network has the following networking requirements:

- The key primitive of the network is to **broadcast blocks and transactions to all nodes**.
  Blocks are data structures that contain transactions, which are records of value transfers or contract executions. Transactions are validated and appended to the ledger by consensus algorithms. Broadcasting ensures that all nodes have the same view of the ledger and can verify its integrity.
- The network also needs to be **robust and resilient to node failures or attacks**.
  Some nodes may go offline due to various reasons, such as power outages, network disruptions, or malicious attacks. The network should be able to tolerate a certain percentage of node failures without affecting its functionality. Moreover, the network should allow new nodes to join and synchronize with the existing nodes, as well as handle node churns and partitions.

1.1.1 Types of Network Architecture

There are two types of network architecture:

1. **Client Server** :
   This is a type of network architecture where one or more servers store most of the data and resources, and the clients access them through requests. The server is the central authority that controls the network, and the clients are dependent on the server for their functionality.
2. **Peer to Peer** :
   This is a type of network architecture where each node acts as both a client and a server, and there is no central authority or hierarchy. The nodes share their data and resources directly with each other, without relying on a server.

The need for robustness implies that we do not want a client-server relationship; we settle for a peer-to-peer (P2P) network where each node has identical behavior.

1.1.2 Overlay Networks

An overlay network depicts the connections between nodes, and is represented as a graph. It abstracts out the physical network switches and routers and defines virtual links between nodes. Two nodes that are connected by a link can exchange messages directly. Those that are not connected by a link must find a path connecting them on this overlay network in order to communicate messages.

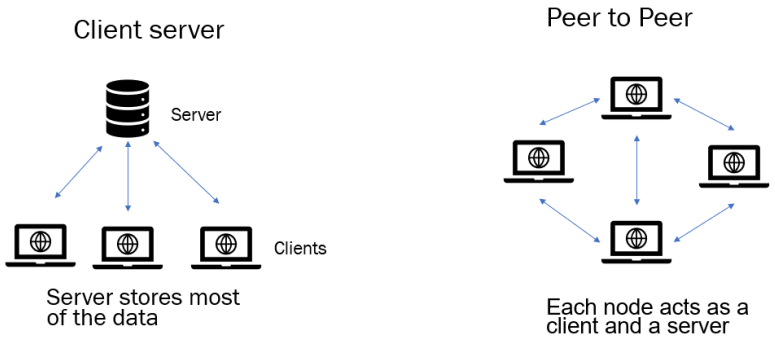

Figure 1.1: Types of Network Architecture

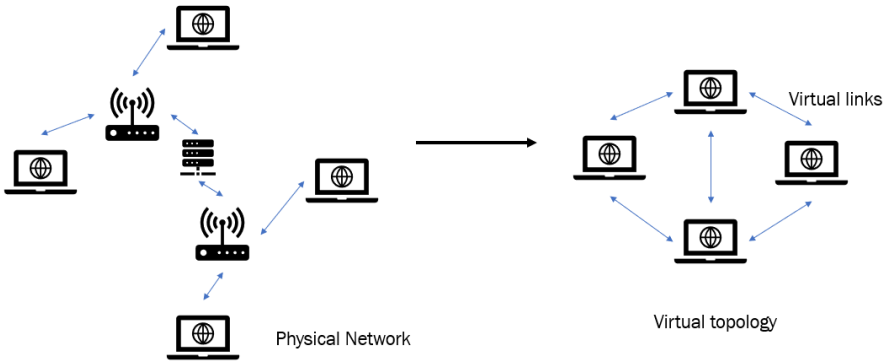

Figure 1.2: Overlay Network

There are different ways to classify overlay networks, but one common way is to distinguish between two types of overlay networks

- **structured** : These are overlay networks that have a specific topology or organization, such as a ring, a tree, or a grid. Structured overlay networks, like CHORD, assign an identifier to each node and uses that to construct welldefined routing rules. These networks are excellent for routing sending point to point messages, a use case not required for Bitcoin. Structured networks are suitable for broadcast, too; any message transmission takes O(logN) hops on CHORD with O(logN) connections per node.
- **unstructured** : These are overlay networks that have no specific topology or organization, and the nodes are connected randomly or based on some criteria, such as proximity, interest, or availability. Unstructured networks like d-regular graphs have no node identifiers; a node connects to d other nodes randomly.

Routing point to point messages is not feasible because it takes O(logN) hops and requires many peer queries to find the path from point A to point B. However, broadcast is very effective using gossip and also takes O(logN) hops. Therefore, the number of hops needed is equal to the structured network with the extra advantage of O(1) peer connections. That is why the Bitcoin network uses an unstructured d-regular overlay network.

1.1.3 Gossip and Flooding

Gossip and Flooding are two techniques for disseminating information in a network of nodes. Gossip is a technique that randomly sends the information to some of its neighbors, while Flooding sends the information to all of its neighbors. Both techniques can spread the information exponentially and reach all nodes in O(logN) time.
Gossip and Flooding can be used to achieve fast and reliable information dissemination in a network, but they differ in terms of efficiency, overhead, and robustness.

1.1.4 Expander Graph

Expander graph is a type of graph that has the property of being well connected but sparse.

Definition 1.1.1: Sparse Graph

A sparse graph $G(V, E)$ is a graph that has a small number of edges compared to the number of vertices, such that $|E| = O(|V|)$, where $|E|$ is the number of edges and $|V|$ is the number of vertices.

An Expander graph is a sparse graph that is also well connected, such that for any subset A of vertices, the number of vertices outside A that have at least one neighbor in A, denoted by $|A|$, is large.

A gossip message starts with $A(0)$ as the broadcasting node with $|A(0)| = 1$. In the next hop, it will reach $\partial A(0)$ with $|A(1)|$ being at least $(1+\epsilon)$ times $|A(0)|$. This process repeats and we get $|A(k)|$ being at least $(1+\epsilon)^k$ times $|A(0)|$ for any k. Therefore, the number of steps to get to half the number of nodes is proportional to the logarithm of the number of nodes. It can be proven that the other half of the nodes can also be reached in O(logN) time.

1.2 Bitcoin Network

Bitcoin is a peer-to-peer network that uses TCP protocol to exchange data. In Bitcoin network:

- The codebase limits the number of outgoing connections to eight and the number of incoming connections to 117.
- The network has a high churn rate (rate at which users join or leave the system); therefore, the node must be prepared to connect to new peers.
- The node maintains a large list of nodes running Bitcoin in the form of their (IP, port) pair and connects to one of them randomly when a slot becomes available.
- The node ensures that the peers it connects to are selected randomly.

1.2.1 Peer discovery

A node starts its list of peers by connecting to a group of DNS seed nodes. These are special nodes that provide a list of IP addresses and ports of active nodes in the network. A node can query a DNS seed node to get this list and then try to connect to some of the nodes in the list.
The seed nodes are not very distributed; so it is not wise to depend entirely on the peer list given by them. After connecting to the first set of peers, a node requests their peer list using **getAddr** and **Addr** messages. The node updates its peer list frequently by swapping peer lists with its peers.

1.2.2 Block transmission

The process of transmitting blocks and transactions involves the following steps:

1. A node sends an **inv** message to its peers, which is an inventory message that tells them what blocks and transactions are available.
2. When a peer receives an **inv** message, it checks if it already has the block or the transaction in its local storage. If not, it sends a **getData** message to the node to get those blocks and transactions.
3. Optionally, the peer can request only the headers of the blocks first, using a **getHeaders** message, before asking for the full blocks .
4. This header-first block transmission can reduce the bandwidth use and prevent invalid blocks from being accepted .

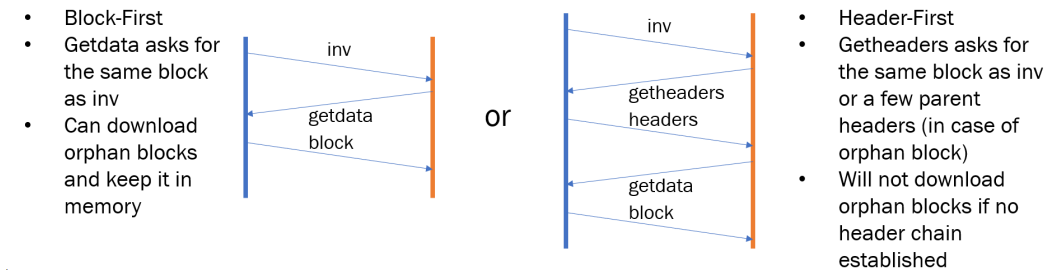

Figure 1.3: Block transmission in blockchain network

1.2.3 Data Broadcast

Now we explain how data, such as transactions, are broadcasted in a peer-to-peer network

- Each node maintains a non-persistent memory to store unconfirmed transactions, which are transactions that have not been included in a block yet. This memory is called the mempool.
- When a node receives a new transaction, it sends an inv message to its peers, which contains the transaction id (txid). The inv message is a way of checking if the peer already has the transaction in its mempool.
- If not, the peer sends a getdata message to request the full transaction data from the node. The node then sends the transaction data (tx) to the peer.
- Some unconfirmed transactions might be removed from the mempool due to various reasons, such as low fees, expiration time, or double spending.

1.2.4 Compact Blocks

Here we discuss two methods of relaying blocks in a peer-to-peer network:

1. **legacy relaying** :
   Legacy relaying is the traditional way of sending the full block data to each node.
2. **compact block relaying** :
   Compact block relaying is a protocol that reduces the amount of bandwidth and latency required to transfer a block that confirms many transactions that the nodes already have in their mempools.

There some differences between the two methods in terms of the messages exchanged between the nodes .Here are the advantages of compact block relaying over legacy relaying:

- Compact block relaying uses shortened transaction identifiers (txids) instead of full transaction data, which reduces the size of the block data.
- Compact block relaying allows nodes to request only the transactions they are missing from their mempools, which reduces the redundancy of sending transactions twice.
- Compact block relaying has two modes: low bandwidth mode and high bandwidth mode.
  Low bandwidth mode minimizes the bandwidth usage by asking for permission before sending a compact block.
  High bandwidth mode reduces the latency by sending a compact block as soon as possible without asking for permission.
- Compact block relaying can verify the proof of work (PoW) from the block header before requesting the full block data, which prevents invalid blocks from being accepted.

There is a trade-off between capacity and propagation delay in a peer-to-peer network. **Capacity** (C) is the maximum amount of data that can be transmitted in a given time, measured in bits per second (bps). **Propagation delay** (D) is the time it takes for a signal to travel from one node to another, measured in seconds. The end to end delay, which is the total time it takes for a block to reach all nodes in the network, increases with the increase

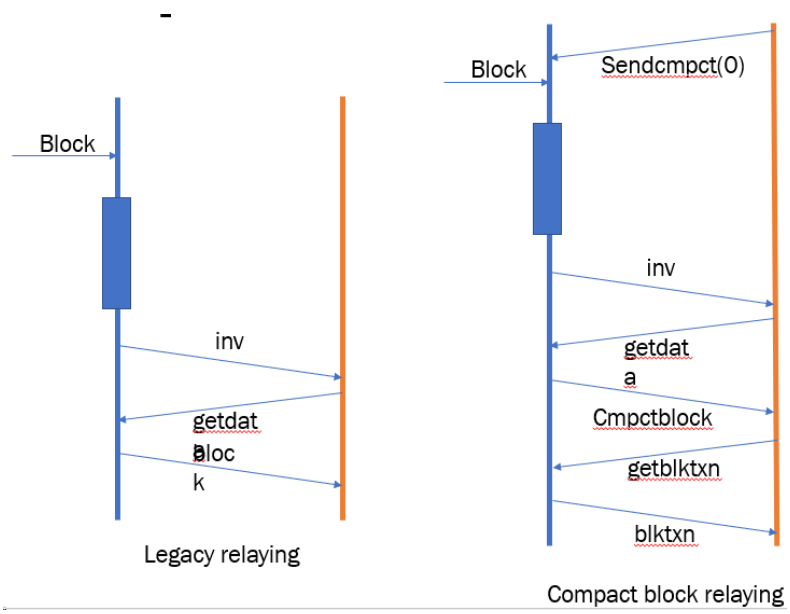

Figure 1.4: Compact Block

in block size. This is because larger blocks take longer to transmit and verify, which can cause congestion and orphaning. Therefore, there is a trade-off between increasing the capacity and reducing the propagation delay of the network. This increase in delay affect the system in following ways:

- Wasted Hash power
- Forking

1.2.5 Disadvantages of current p2p network

- **Efficiency:**
  It requires a lot of communication between nodes. The total communication is proportional to the number of nodes (N) and the average degree of connectivity (d). This means that as the network grows, the communication overhead also increases, which can affect the performance and scalability of the system.
- **Privacy:**
  It can link the transaction source to the IP address of the node that broadcasts it. This means that anyone who monitors the network traffic can potentially identify the sender and receiver of a transaction, which can compromise their anonymity and privacy.
- **Security:**
  It allows for plausible deniability for forking. Forking can cause inconsistency and double spending. Plausible deniability means that a node can claim that it did not receive a valid block from another node, even if it did, and create a fork on purpose. This can be used as an attack strategy to disrupt the consensus and integrity of the system.

1.3 Geometric Random Network

The current network topology is based on random IP addresses, which are not related to geographic distances. This means that nodes may be connected to peers that are far away from them, which can increase the latency and bandwidth usage of the network.

A better network topology would be based on a **geometric random network**, which is a network where nodes are placed according to some geometric criteria, such as proximity or similarity. This way, nodes can connect to peers that are closer or more relevant to them, which can improve the efficiency and performance of the

network. The challenges are creating a self-adapting network topology based on measurements, such as latency, bandwidth, or trust. This means that nodes can dynamically adjust their connections based on the changing conditions and preferences of the network.

1.3.1 Perigee

Perigee is a self-adaptive network topology algorithm for peer-to-peer networks. Perigee is a decentralized algorithm that selects neighbors based on past interactions. It aims to retain neighbors that relay blocks fast and disconnect from neighbors that do not relay blocks fast. It also explores unseen neighbors to discover new potential connections.

'Perigee is motivated by the multi-armed bandit problem, which is a problem of finding the optimal strategy for choosing among several options with uncertain rewards. Perigee tries to balance between exploration and exploitation, which means finding new neighbors and using existing neighbors, respectively. Perigee can improve the efficiency and performance of the peer-to-peer network by reducing the latency and bandwidth usage.

Algorithm

The Perigee Algorithm works as follows:

- It assigns scores for each subset of neighbors based on how fast they relay blocks.
- It retains the subset of neighbors with the best score and disconnects the node that is not in the subset.
- It forms a connection to a random neighbor to explore new potential connections.

1.4 Basic requirements for a banking system

In any currency/banking system, there are some basic requirements that the system must provide for. We list them here:

1. There should be a unit of currency/money.
2. There should be a standard way of keeping accounts, i.e., keeping track of how much money each person owns, and transferring money between accounts.
3. No user should be able to create new money from thin air. Put differently, there should be a fixed amount of money in the system at any given time, and new money should be introduced in a systematic manner.
4. A user should not be able to spend more money than he/she owns. There should be a way to verify whether or not this happens.
5. One user should not be able to spend someone else's money (at least, not without their permission).

Let us see how the Bitcoin system provides these features.

1.5 Bitcoin and Satoshi

1. The basic unit of currency in the Bitcoin system is Bitcoin, and the smallest denomination is called a Satoshi, which is equal to 10^{-8} Bitcoins.
2. All transactions in Bitcoin must be an integer multiple of a Satoshi.
3. Just like other currencies, there is an exchange rate between Bitcoin and the dollar. As of July 22, 2023, one Bitcoin is worth 30,000 US dollars. However, the exchange rate between Bitcoin and dollars is very volatile due to various factors, including greed and speculation.
4. The classification of Bitcoin as a currency (like the US Dollar) or a store of value (like gold) has been widely debated. The mainstream view is that Bitcoin is a combination of both.

5. Economically valuing Bitcoin, both in the short and long terms, is an active area of research.
6. Understanding the economic aspects of Bitcoin, both as a currency and a store of value, involves studying various aspects of the Bitcoin system.

1.6 Transactions

In ordinary parlance, the term **transaction** refers to an exchange of something of value. In the context of Bitcoin and cryptocurrencies, a transaction is simply a message that specifes the transfer of money from one entity to another. In fact, transactions are the data-values that get recorded on the blockchain. The blockchain as a ledger is therefore an ordered list of transactions. From this publicly verifable ledger, any user can detect whether transactions are made according to certain rules or not, thereby lending credibility to the ledger and the currency.

1.6.1 Addressing

1. **Bitcoin Address and Its Generation:**
   - In Bitcoin, the notion of traditional accounts is replaced with addresses.
   - An address is simply the hash of a user's public key. It is a unique alphanumeric string that serves as a destination for receiving bitcoins.
   - Addresses are also used to determine where bitcoins will be sent in a transaction.
   - New pairs of public and private keys, and thus new addresses, can be generated at will by a single user.
2. **Receiving vs. Spending Coins:**
   - To receive coins, a user only needs to share their Bitcoin address with others. The address serves as a public identifier for receiving funds.
   - However, to spend coins, a user must also reveal the corresponding public key associated with the address.
   - This is because spending requires providing proof of ownership through a digital signature, which is created using the user's private key.
   - The idiosyncrasy lies in the fact that while an address (hash of public key) is publicly visible and used for receiving, the associated public key is only revealed during the spending process for cryptographic verification.

1.6.2 Transaction inputs and outputs

1. **Transaction Inputs:**
   - Each transaction input represents the amount of Bitcoin being spent from a specific address.
   - When a user initiates a transaction, they reference one or more previous unspent transaction outputs (UTXOs) from the blockchain that they have the right to spend.
   - These UTXOs serve as the inputs to the new transaction and determine the source of the funds being spent.
   - Each input includes a reference to the UTXO's transaction ID and its output index, along with a cryptographic signature to prove ownership.
2. **Transaction Outputs:**
   - Each transaction output represents the amount of Bitcoin being received by a specific address.
   - When a transaction is created, it typically includes multiple outputs, each specifying the amount of Bitcoin and the recipient's address.
   - These outputs determine the destinations of the funds being transferred in the transaction.
   - Each output locks the specified amount of Bitcoin to the recipient's address using a locking script that can only be unlocked with the corresponding private key.

3. **Balance Consistency:**
    - In every valid transaction, the total amount of Bitcoin being spent (sum of inputs) must equal the total amount being received (sum of outputs).
    - This ensures that the transaction preserves the overall balance of the Bitcoin system, i.e., no new bitcoins are created, and no bitcoins are lost during the transaction process.

1.6.3 Signatures on transactions

1. **Signing Transactions for Safety:**
    - Each transaction in Bitcoin must be signed by the users who are spending money. This process is a fundamental safety feature that prevents unauthorized access to someone's funds.
    - For every transaction input, the user creating the transaction must sign it using the corresponding private key associated with the address from which the funds are being spent.
    - By signing the transaction, the user proves ownership of the private key and authorizes the transfer of funds.
2. **One-to-One Correspondence: Public and Private Keys:**
    - As mentioned earlier, each address in Bitcoin has a one-to-one correspondence with a public key, and each public key has a corresponding private key.
    - When a user wishes to spend Bitcoins associated with a particular address, they create the transaction and then sign it using the private key linked to that address.
3. **Verification of Signatures:**
    - After a user signs a transaction, they broadcast it to the network along with the corresponding public key (not the private key).
    - Anyone who sees the signed transaction can verify its authenticity by checking whether it was signed with the private key associated with the public key provided.
    - By doing so, other users can validate that the transaction was indeed signed by the rightful owner of the address (and the coins in that address).
4. **Multiple Addresses, Multiple Signatures:**
    - In transactions that spend Bitcoins from multiple addresses, there must be signatures corresponding to each of these addresses.
    - Each input requires a valid signature to prove ownership and authorization for spending the funds associated with that particular address.

1.6.4 UTXO

The UTXO (Unspent Transaction Output) model is a key feature of the Bitcoin system that ensures the security and validity of transactions. Let's summarize the important points about the UTXO model and how it facilitates transactions with multiple inputs and outputs:

1. **UTXO Model for Validating Transactions:**
    - In the UTXO model, every transaction input must be a transaction output of a previous transaction, linking the spending of funds to their source.
    - When a new transaction output is created, it is considered unspent until it gets consumed as an input in a future transaction when the funds are spent.
    - A valid transaction can only include unspent transaction outputs (UTXOs) as its inputs, providing proof that the address indeed has sufficient funds to spend.
2. **Preventing Double-Spending:**

- By keeping track of UTXOs at all times, honest users can validate every new transaction against this set to prevent double-spending attempts by dishonest users.
- To spend their money, users must provide a valid chain of ownership through the UTXO history, ensuring that each input is a previously unspent output.

3. **Multiple Transaction Outputs (Outputs and Change):**
   - A transaction may have multiple outputs, allowing users to send funds to multiple recipients in a single transaction.
   - For example, if a user owns an address with 2 Bitcoins in an unspent output and wants to spend 1 Bitcoin, the transaction will have two outputs: one for the recipient and one for themselves (the change).
   - The change output sends the remaining 1 Bitcoin back to the user's address or a new address.
4. **Multiple Transaction Inputs:**
   - Users can include multiple transaction inputs in a single transaction to combine funds from different unspent outputs for larger transactions.
   - For instance, if a user owns an address with 1 Bitcoin from two separate unspent outputs and wants to pay 2 Bitcoins to another address, they can include both unspent outputs as inputs in the transaction.

The UTXO model's design ensures that each transaction is valid, and its structure allows for flexibility in handling various scenarios, such as spending from multiple sources and sending funds to multiple recipients in one transaction. The adoption of the UTXO model is one of the key factors that contribute to the security and integrity of the Bitcoin blockchain.

1.6.5 Cryptocurrency wallets

Cryptocurrency wallets play a crucial role in simplifying the process of managing and transacting with cryptocurrencies like Bitcoin. Here are the key points about cryptocurrency wallets:

1. **Complexity of UTXO Management:**
   - Keeping track of UTXOs and measuring them for each transaction can be complex and challenging for regular users.
   - Additionally, to maintain anonymity, users often generate new addresses regularly, which requires careful handling of private keys and secure storage.
2. **The Role of Cryptocurrency Wallets:**
   - Cryptocurrency wallets are software applications that handle many of the complex tasks in the background, making it easier for users to transact in cryptocurrencies.
   - Wallets manage UTXOs, track balances, create new addresses, and handle cryptographic signatures for transaction verification.
3. **Anonymity and Security:**
   - Wallets ensure that new addresses (and the corresponding keys) are generated securely without revealing private keys.
   - For maintaining anonymity, wallets often offer features like generating new addresses for each transaction, making it harder to link transactions to a specific user.
4. **Secure Storage and Usage:**
   - Wallets provide a secure storage solution for private keys, protecting them from unauthorized access or theft.
   - Users must be cautious and responsible for securing their wallet's private keys since compromising them could result in the loss of funds.

5. **Trust in Wallet Software:**
    - Using a cryptocurrency wallet requires placing trust in the software's functionality and security.
    - Using a cryptocurrency wallet requires placing trust in the software's functionality and security.

1.6.6 Transaction fees

Transaction fees play a crucial role in the Bitcoin network and serve as an incentive for miners to include transactions in the blocks they mine. Here are the key points about transaction fees:

1. **Total Value in Inputs and Outputs:**
    - In a Bitcoin transaction, the total value in the inputs (UTXOs being spent) must be equal to or greater than the total value in the outputs (newly created UTXOs).
    - The difference between the total value in inputs and outputs is known as the transaction fees.
2. **Transaction Fees as Miner Incentives:**
    - The transaction fees are claimed by the miner who successfully includes the transaction in a block and adds that block to the blockchain.
    - Miners compete to add transactions to their blocks because the fees they collect are an additional reward on top of the block reward (newly minted Bitcoins).
3. **Transaction Prioritization and Speed:**
    - Transactions with higher fees are more attractive to miners because they earn more rewards for including them in their blocks.
    - Miners prioritize transactions with higher fees, leading to faster inclusion of these transactions in the blockchain.
    - Transactions with lower fees may take longer to be confirmed since they are lower in the priority list.
4. **Automatic Fee Calculation:**
    - Wallets automatically calculate transaction fees based on a particular fee rate, measured in Satoshi per kilobyte (Satoshi/kB).
    - The fee rate determines the amount of Satoshi to be paid per kilobyte of transaction data.
    - Higher fee rates result in faster confirmation times, while lower rates may lead to delayed confirmations.
5. **Dynamic Fee Variation:**
    - Transaction fees can vary over time due to fluctuations in network demand and block space availability.
    - During periods of high network congestion, transaction fees may increase as users compete for limited block space.

1.6.7 Coinbase transactions

Coinbase transactions are a crucial mechanism for introducing new Bitcoins into the Bitcoin system and rewarding miners for their efforts in processing transactions and adding blocks to the blockchain. Here are the key points about coinbase transactions:

1. **Introduction of New Bitcoins:**
    - New Bitcoins are introduced into the system as a reward for miners who successfully mine a new block.
    - Every block includes a special transaction known as the "coinbase transaction," which allows the miner to claim a fixed number of newly created Bitcoins.
2. **Block Reward Halving:**
    - Initially, when Bitcoin was launched, the block reward for miners was 50 BTC per block.

- Approximately every 210,000 blocks (about four years), the block reward is halved through an event known as "halving."
- This means that after each halving, miners receive half the number of Bitcoins as the previous reward.

3. **Current Block Reward:**
   - As of now, there have been three halvings, and the current block reward for miners is 6.25 BTC per block.
   - The block rewards will continue until the year 2140 when the total supply of Bitcoin will reach its cap.
4. **Fixed Supply of Bitcoin:**
   - The total supply of Bitcoin is capped at 21 million coins.
   - This fixed supply ensures that there will never be more than 21 million Bitcoins in circulation.
   - Approximately 19.4 million coins are already in circulation, and the remaining Bitcoins will be gradually introduced through coinbase transactions until the cap is reached.
5. **Incentive Mechanism:**
   - Coinbase transactions, along with transaction fees, serve as incentives for miners to actively participate in the network and secure the blockchain through the proof-of-work process.
   - In the early years of Bitcoin, coinbase transactions formed a significant portion of the rewards for miners, but over time, the contribution of transaction fees has increased.

The combination of coinbase transactions and transaction fees ensures the proper functioning and security of the Bitcoin network. Miners are motivated to continue mining, securing the network, and processing transactions as they receive both block rewards and transaction fees as incentives for their efforts. The gradual reduction in block rewards through halvings also creates a deflationary monetary policy, which further contributes to Bitcoin's scarcity and potential value appreciation over time.

1.6.8 Transaction mempool

The transaction mempool plays a crucial role in the process of how Bitcoin transactions are propagated, verified, and eventually included in blocks. Here are the key points about the transaction mempool and its significance:

1. **Transaction Propagation:**
   - When a user wants to make a Bitcoin transaction, they create a transaction message, sign it using their private key, and broadcast it across the Bitcoin network.
   - The transaction message is a few kilobytes in size and contains information about the transaction inputs, outputs, and the corresponding digital signatures.
   - Miners in the network receive these transactions and verify the signatures before adding them to their memory pool, known as the "mempool."
2. **The Mempool:**
   - The mempool is a temporary storage area where miners keep pending transactions that they have received and verified.
   - Transactions in the mempool are waiting to be included in a block and added to the blockchain.
   - Miners prioritize which transactions to include in their working block based on factors like the transaction fee rate (higher fee with smaller size).
3. **Block Size Limit:**
   - Each block in the Bitcoin blockchain has a maximum size limit, currently set at 1 megabyte (MB) in the Bitcoin network.
   - Miners aim to maximize the total transaction rewards in their block while adhering to this size limit.

4. **Transaction Fee Rates:**
    - Miners prioritize transactions with higher transaction fees because they want to maximize their potential reward for including transactions in a block.
    - Transactions with higher fees are more likely to be included in blocks sooner, incentivizing users to offer higher fees if they want their transactions processed quickly.
5. **Confirmation Latency:**
    - There is a certain latency between the time a transaction is issued and when it is confirmed on the blockchain. This latency is influenced by two factors: (a) the time it takes for a transaction to be included in a block (can be reduced by offering a higher transaction fee), and (b) the time it takes for that block to be buried deep enough in the longest chain for the transaction to be considered confirmed.
    - Users may choose to trade off latency for security, deciding whether to use higher transaction fees for faster confirmation or lower fees for potentially longer confirmation times.
6. **Blockchain Design Trade-offs:**
    - The trade-off between latency and security is specific to the Bitcoin blockchain design.
    - Other blockchain designs may have different approaches and trade-offs, which will be explored in future lectures.

The mempool and the prioritization of transactions based on fees are essential components that allow the Bitcoin network to efficiently process and confirm transactions while incentivizing miners to maintain the security of the blockchain through the proof-of-work process.