Lecture Notes

Blockchain Basics

Session 1 – Basic Idea Of Blockchain

In this session you have learnt the basic components of blockchain technology as follows:

- Database systems
- Role of intermediaries
- What is a blockchain?

Further in the session, you learned important concepts related to blockchain technology as follows:

- Compare blockchain with traditional database systems

Comparison of Database Systems

We classify databases into three types — **centralised**, **decentralised** and **distributed**.

- Centralised Databases: A single centralised database stores all the data
- Decentralised Databases: All the information is not stored in one place but multiple places or databases.
- Distributed Databases: Systems where data processing is **shared** across all the nodes, but the system decision **might** still be centralised, based on the complete system knowledge.

Listed below are the critical points of differentiation between the three types of systems we encountered so far.

Feature	Centralised	Decentralised	Distributed
Security	Low; Most vulnerable to data security issues	Moderate; Data can be rebuilt from parallel servers if backed up	Highest; Very difficult to lose data completely
Response Speed (**Applicable in case the networks having large amounts of data)*	Bottlenecks can cause response speed to reduce significantly	Quick response speed depending on the distribution of data	Fastest response rates
Overheads and Costs	Low; Redundancy is minimized	Substantial processing overheads to ensure proper coordination among servers	Massive overheads to ensure appropriate coordination among multiple nodes
Points of Failure / Maintenance	Single point of failure; Easy to maintain	A limited number of points of failure; Maintenance more	Multiple points of failure; Difficult to maintain

Feature	Centralised	Decentralised	Distributed
		complex than centralised systems	
Stability	Highly unstable; if the central server fails, entire network collapses	Stability better than centralised systems; the network can continue to operate at a reduced level if any one server fails	The highest level of stability; single node failure doesn't affect the network
Scalability	Low scalability	Moderately scalable	Infinitely scalable
Ease of Setup	Easy to set up	Difficult to set up	Difficult to set up

Intermediaries and their disadvantages

Intermediary

An intermediary generally plays the role of a **trusted middle party** in any transaction, which connects two parties that are **usually unknown** to each other.

Disadvantages of Intermediaries

Traditionally, the significance of intermediaries lay in **establishing trust** among unknown parties. However, involving intermediaries also carries with it a fair share of drawbacks such as:

- **Monetary charges**
- **Process inefficiencies**
- **Security concerns**.

Basic Idea of Blockchain

To overcome the challenges of centralised systems and disadvantages caused by having multiple intermediaries, Satoshi Nakamoto devised a revolutionary technology that has the potential to disrupt existing business frameworks and come up with a new way of doing business. This revolutionary technology is known as "**Blockchain**".

Blockchain, is a combination of **decentralised and distributed database** containing a registry of transactions that are distributed among **peers** or fellow participants in the network. The registry includes a long list of transactions and is continually updated with new transactions as they take place. Starting from the very first transaction, a bunch of transactions is grouped into a **block** as per a predefined block size (1MB in case of Bitcoin). Once the block size is achieved by one block, the next set of transactions forms another block which is then **linked** to the block **previously** formed. Over time, a **series of blocks** is formed where each block is **connected** to another block that was created just before it. Thus, we call this chain of blocks as the **blockchain**.

To create the blockchain network, Satoshi used various existing technologies and techniques. You learnt those in detail.

Digital Signatures

Just like the normal signature, a digital signature also acts as verifier of the authenticity of the sender. There are two types of digital signatures:

- **Symmetric digital signatures:** In symmetric digital signatures, one single key is used to encrypt the messages. The sender encrypts the message with that key and sends it to the receiver. Once the receiver receives it, they need the same key to decrypt or unlock that message. So, the sender also shares the key he/she encrypted the message with so that the receiver can use it to decrypt the message.
- **Asymmetric Digital Signatures:** In Asymmetric Digital Signatures, a pair of public and private key is used. A message encrypted with a public key can only be decrypted with the corresponding private key of the public-private key pair and vice versa. The public key of each participant is shared across the network and private key is held secret only with the individual.

Hashing

Hashing is an encryption technique which is used to encrypt the data to ensure data security. Hashing is done using hash functions. The hash function is a mathematical function that can take in **any length of input** and convert it to an **output of fixed length**.

The hash of a data serves like a fingerprint for that data. Hash uniquely represents that data. Even a small change in the data generates an entirely random and different hash. Also, you learnt that each hashing algorithm, irrespective of the size of the data generates the hash of a fixed length which is a characteristic of that hashing algorithm.

The hash function has many properties. Some of them are listed below.

Properties of cryptographic hash functions

Property	Description	Effect on the hash function
Fixed-length output	The outputs of hash functions have a fixed length	Ability to hide information
Pseudo-randomness	Hash outputs seem random, but are deterministic*	It is very difficult (almost impossible) to form patterns in outputs of hash functions or make predictions about 1. The hash output for a given input 2. The changes in the given hash output for a change in a given hash input
Public availability	Hash functions are publicly available	Encryption services using hash functions can be easily rolled out
Any-length input	The inputs to cryptographic hash functions can be of any length	Any length of input strings can be processed through the hash function for encryption

Merkel Tree

Merkel Tree is a way of organizing data points in the blockchain. The output of a Merkle tree is a Merkle root. The Merkle tree utilizes hash functions to arrive at the Merkle root by recursive hashing.To arrive at the Merkle root of "n" number of elements, you have to follow the below steps:

1. Compute the individual hashes of each element.
2. For this set of hash elements, form strings by concatenating each pair of consequent hashes. In case a consequent hash is not available for the last hash, it must be concatenated to itself.
3. Compute the individual hashes of the obtained strings.
4. If the number of hashes obtained is more than one, go back to step 2 and perform steps 2 and 3 till one single hash is obtained.

.Even the smallest of the changes in any of the data points will change its hash and hence the root hash will also be changed. Thus root hash is the fingerprint representing all the datapoints.

The Merkle trees help in verifying transaction data on a block by acting as a unique identifier for the transactions.

Hash Cash

Hash cash

Hash cash is a proof of work the sender of an email must do before sending an email. The sender must calculate the hash of the email data and make sure that the calculated hash satisfies a predefined condition. This limits the scope of spam emails. The technique of hash cash is used in the process of creation of blocks in the blockchain.

TCP/IP and Peer to Peer Network

TCP/IP

The TCP/IP is a communication protocols which sends the data in the form of packets over a server using the IP addresses of sender and receiver.

Peer to Peer Network

In a client-server architecture, a client can be a machine or a program that allows the user to make requests to the server. For example, WWW is a client-server program. Users can request data and the web will provide it. On the other hand, a server is essentially a program and **NOT A DEVICE**. In contrast, in a peer-to-peer network, each device can act as a server and as a client at different points in time.

In a blockchain peer-to-peer network, the peers use TCP/IP protocols to connect to other peers and transfer the data.

Basic Architecture of Blockchain

Satoshi used all the above-mentioned techniques to create the ideal business network, i.e., blockchain. The architecture of blockchain network is as follows:

- There is a peer-to-peer network created between all the participants of the network.
- Each network participant has a digital signature for identifying the participants

- Any transaction happening between two network participants gets flooded in the network with the help of gossip protocol to all the participants for validation.
- Each participant has a transaction pool which is a memory space allocated to store the verified transactions.
- Each transaction is secured using the hashing algorithm.
- The transaction pool is at a node level.
- There are designated nodes which create the blocks out of the transactions happening in the network at a frequency known as mining rate.
- Once the block is created it gets flooded in the network and each network participant verifies the block of data and once there is a single source of truth for the block, the block gets added to the blockchain.

Blockchain Basics-Session 2

Introduction

You already learnt about the various technologies that Satoshi envisioned for the ideal network that he proposed. In this session, you learnt how those technologies come together and create the first blockchain network, the Bitcoin blockchain network. The main topics you learnt as part of this session are:

- How blockchain has evolved
- Layers of blockchain network
- Fundamentals of Bitcoin blockchain network
- Consensus mechanism
-

Evolution of Blockchain Networks

Since Satoshi's network was an open source network, the developers across the world started working on it and created many variants of the same technology, also known as forks.
Two main networks have evolved in the market today:

1. **Cryptocurrency Networks:** The main intention of these networks is to introduce cryptocurrencies that work the same as common currencies such as rupees, dollars etc. However, these currencies only have a digital life and can be used only on a digital network. Examples of this type of network are Ethereum, Bitcoin etc. These are used to process transactions without the use of any central organisation like a bank
2. **Enterprise Networks:** Enterprises use these networks for record keeping and many other purposes. Examples of these networks include Hyperledger, R3 Corda etc.

Layers of A Blockchain Network

A blockchain network has the following layers:

1. **Hardware Layer:** The hardware layer can be a cloud or a server that hosts the entire network.
2. **Ledger or Fabric Layer:** It forms the base of the blockchain network and constitutes of blocks also known as a ledger that hold the transaction data.
3. **Smart Contract or Logic Layer:** It forms the business logic of the network and ensures that the network follows all the rules and regulations that govern the blockchain network.
4. **Interface layer:** It is the set of API's that are used to communicate with the blockchain and get the required result such as retrieval of data, the addition of data etc.
5. **User Interface or Application Layer:** This layer is the front end of the application and runs the entire network. It interacts with the rest of the layers.

All these layers together form the blockchain network and users can have a different set of capabilities or applications that they use over the network.

Bitcoin Blockchain Network

Bitcoin was the first implementation of the blockchain, and it has many use cases. In the bitcoin network the transaction is validated as follows:

- A transaction is performed between two nodes
- The transaction gets flooded across the entire network.
- All the nodes in the network validate the transaction and add it to their transaction pool in case the transaction is valid otherwise reject it.
- Each node has its own transaction pool that it maintains.
- All the transactions that are validated as valid are placed inside the transaction pool.

All the above-mentioned steps are a part of the record keeping mechanism of the blockchain network.

Recordkeeping in the Bitcoin Blockchain Network

There are several methods in which records are maintained in networks. One of the basic methods is the Account/Balance model wherein balance related to every account is maintained. However, bitcoin uses a different method for recordkeeping. Bitcoin uses the concept of unspent transaction output (UTXO) for record keeping.

Every transaction in a bitcoin is based upon an unspent transaction output (UTXO). Whenever a transaction is performed in bitcoin, it primarily consumes existing UTXOs called as inputs and creates new UTXOs called outputs. The outputs generated from a transaction can be utilized further in the network by the node.

There are two primary ways to get the UTXO in a bitcoin network:

- Either from another node in the network whenever the transaction is performed.
- Network reward which a miner gets for mining a new block.

There is also a transaction fees which can be a part of a transaction which is paid to the miner for adding the transaction into a block.

Types of Nodes

There are three types of nodes in the bitcoin blockchain network:

1. **Miner nodes:** They create the blocks in the network
2. **Full nodes:** They store the details of all the blocks in the network
3. **SPV nodes:** They are nodes that store only the partial details of the blocks in the network

The miner node takes the unconfirmed transactions in the mining pool and creates a block. Once a block is created, it gets flooded across the network using the gossip protocol. All the nodes receive the block and first validate the block if the block passes the validation, they add it to their respective blockchain otherwise not.

Block Anatomy

A blocks header is comprised of the following components:

- **Merkle root** - aggregation of all the hash values of the transactions into a single hash value.
- **Timestamp** - Timestamp of the block creation time.
- **Nonce** - Random Value that is altered/updated to try different permutations to achieve the required difficulty level. You will learn more about this in the upcoming section.
- **Transaction counter -** Count of the **number of transactions** in a block.

Mining Process

The process of mining means that a new block is created and added to the blockchain. Every miner creates a block from the transaction existing in its transaction pool.

- For a block to be valid and added to the network, the miner needs to solve a puzzle which is defined as below in the case of bitcoin blockchain.

  Hash of the block header < Value
  Hash of the header is the hash of all the data present in the header appended together.

- In case the above condition is not met, the miners need to recompute the hash.
- To recompute the hash nonce is used whose value is changed in every iteration to arrive at a new hash. This is because all the other data in the block such as time stamp, Merkel root, number of transactions etc. are all fixed and can not be changed.
- To change the hash the only thing that can be changed is the nonce value. The difficulty to attain the acceptable hash in the network is termed as the difficulty of the network. Once the condition is satisfied, the block becomes a valid block.

- The miner who calculated a valid hash first can claim the creation of the block

Mining Reward

A miner **competes** with other miners in the network to form a block. The miner who is able to first generate a block in the network is given a block reward. The block reward started from **50 BTC** and currently, the miner earns a block reward of **12.5 BTC** for every block.

Block Validation

Once a block gets created it is propagated to the entire network and all the peers don't add the block as such to their respective blockchains. Each peer node performs validations on the incoming block.

The block header also contains a previous hash pointer of the block which precedes this block. Blockchain being a chain of blocks has link between the blocks. The previous hash acts as this link between the blocks. Each node in the network checks whether the last block in its blockchain has the same hash as the previous hash pointer of the incoming block, it adds the block otherwise not.

Consensus Mechanism: Proof of Work

In the blockchain, mathematical algorithms are used to verify transactions and ensure trust between transacting parties. Transacting parties have to trust the output achieved using these mathematical algorithms. These algorithms together are known as the consensus mechanism in the blockchain. The consensus can for a transaction or for an entire block.

- The consensus for the transaction verification of transaction includes the checks for the following:
  - Sender Balance
  - Valid Authority
  - Valid signatures
- The consensus for a block is for the following condition:

  Hash(HDR) < Difficulty Level

The consensus mechanism is difficult to achieve and easy to verify. The above-mentioned consensus mechanism is known as proof of work. In the proof of work consensus, miner nodes have to solve a computationally difficult problem to compute the hash for the current block. The network sets a difficulty target for the miners. This difficulty level plays a vital role in the consensus as well as the mining process.

Difficulty Level

The difficulty value, also known as the target value, sets the difficulty level of the network. The difficulty level is used to regulate the mining of blocks in the network. The bitcoin network has a block creation time set to 10 mins and this value has to be maintained. The difficulty value adjusts with every block creation such that the value of block creation remains constant.

The formula for calculation of difficulty level in bitcoin blockchain network is:

Difficulty Level = (Previous Difficulty Level * 20160) / (Time taken to mine last 2016 block)

In case the time taken for the last block is greater than 10, the new value of difficulty level is lower than the previous one such that the blocks are created faster. In case the time taken for the last block is less than 10, the new value of difficulty level is greater than the previous one such that the blocks are created slower.

Data Immutability in Blockchain Network

Blockchain offers one more important feature that differentiates it from other networks: Data Immutability. Immutability means something which is permanent and cannot be changed. In practical terms, it refers to the extreme difficulty that one will face in trying to alter or make changes to the existing data.

So how does blockchain ensure data immutability? Blockchain offers data immutability as follows:

- Blockchain forms a chain of blocks which are connected to each other via a link also known as the previous hash pointer or simply previous hash.
- The hash of a block is calculated for all the contents in the block header using a hashing algorithm.
- Merkel root is a part of the block header and any change in any constituent transaction results in the change in the Merkel root.
- Hashing algorithms are deterministic and result in a different output in case the input is changed.
- Change in the Merkel root will result in changing the hash of the entire block.
- In case the hash of one block is changed, the block next to it will not have a link to this block as the previous hash will not match the new block.
- Hence, the link between the blocks will be broken and these blocks will become invalidated.

Immutability is an important feature of the blockchain which helps in data from being manipulated and also helps in identifying malicious nodes in the network.

SPV Nodes

The primary purpose of having SPV nodes is to validate the transactions without having computationally intensive machines. The SPV nodes just store the block headers and can validate whether a transaction is present in a block or not. They don’t have the entire blockchain data and use the Merkel Root to arrive at a conclusion regarding a transaction.

The use of SPV nodes is done due to the fact that proof of work is computationally heavy and requires a lot of resources by the nodes. This is one of the challenges that has prompted developers to think of alternative solutions to proof of work.

Challenges in Consensus Mechanism

In Bitcoin, a lot of miners can come together and form a pool known as mining pool where they compute the block hash together. The mining pool divides the nonce value into different ranges, and all the miners calculate the hash within the assigned range. This makes the mining process much faster, and less computational power is required for the process of mining to be carried out.

The major challenge with pool mining is that it creates a monopoly in the network and only specified nodes get the reward. In case the mining pool exceeds 51% of the network strength, the miners can derail the network from its original intent and tamper with the data as their data could get accepted by the majority rule.

Proof of work faces three major challenges:

a. It is energy inefficient: In proof of work millions of miners try to mine one block, and only one miner is successful. Rest of the energy spent by the miners goes to waste.

b. It is computationally heavy: Proof of work requires a lot of computation by the miners to mine one block successfully.

c. 51% attack: In case more than 51% of the nodes in the network are malicious the network could become unstable.

Bitcoin follows the longest chain rule which states that in case the blockchain branches out into multiple branches, the longest branch is accepted by the network. The branches are known as forks.

In case there 51% of the network comes together to form a mining pool they can create the longest chain with faulty transactions as they are computationally superior making the network unstable.

Byzantine Fault Tolerance

Apart from the 51% attack which can cause the blockchain network not to behave properly, there is one more problem similar to 51% attack known as Byzantine Generals problem. The Byzantine Generals problem occurs when a malicious node propagates wrong message or tampered transactions in the network that could compromise the security of data in the blockchain network. The blockchain network needs to be tolerant to such activities and needs to ensure that all the transaction data in the network is tamper free. Such networks are called as Byzantine Fault Tolerant Networks and they use the Byzantine fault tolerant consensus.

BFT Network

For a network to be byzantine fault tolerant the number of malicious nodes in the network should be less than **1/3rd** of the total nodes in the network. Whenever a node receives two conflicting messages, it goes for a majority vote and accepts the message which comes from majority number of nodes. To ensure that the correct message is accepted by the nodes in the network the total number of malicious nodes needs to less than 33.33% or 1/3rd of the network.

Proof-of-work is a BFT consensus mechanism provided the 51% attack and pool mining does not happen.

Evolution of Proof of Work

The proof of work is a very energy intensive mechanism and requires a lot of computational power. One one node comes out as a winner and rest all the energy used by other nodes is wasted in the proof of work mechanism. The evolution of consensus mechanisms is going and one of the methods to overcome the challenges of proof of work is to choose a leader node based on the stake in the network. It is very unlikely for a node to be malicious if it has the maximum stake in the network.