BLOCKCHAIN

Unit3: Class 1

1. Consensus Algorithms

1.1 What is distributed Consensus?

- Consensus is a procedure to reach a common agreement in a distributed and decentralized environment.
- Important for message passing system.
- A distributed system contains multiple nodes that are physically separate but linked together using the network. A distributed system is a group of computers working together to achieve a unified goal.
- A distributed system involves a set of distinct processes (e.g., computers) passing messages to one another and coordinating to accomplish a common objective (i.e., solving a computational problem).
- Computers in a distributed system communicate and coordinate by "message passing" between one or more other computers. Messages can be passed using any messaging protocol, whether that's HTTP, RPC, or a custom protocol built for the specific implementation. There are two types of message-passing environments:

  1)Synchronous

  - In a synchronous system, it is assumed that messages will be delivered within some fixed, known amount of time.
  - Synchronous message passing is conceptually less complex because users have a guarantee: when they send a message, the receiving component will get it within a certain time frame. This allows users to model their protocol with a fixed upper bound of how long the message will take to reach its destination.
  - However, this type of environment is not very practical in a real-world distributed system where computers can crash or go offline and messages can be dropped, duplicated, delayed, or received out of order.
  - **2) Asynchronous**
  - In an asynchronous message-passing system, it is assumed that a network may delay messages infinitely, duplicate them, or deliver

them out of order. In other words, there is no fixed upper bound on how long a message will take to be received.

- In synchronous environments, messages are delivered within a fixed time frame
- In asynchronous environments, there's no guarantee of a message being delivered.

As we know that blockchain is distributed, decentralized network that provides immutability, privacy, security and transparency. It is peer to peer network that has no central authority. Think of a normal central organization, all the decisions are taken by the leader or group of members. This is not possible in blockchain, because blockchain has no central authority.

- No central authority present to validate and verify the transactions, yet every transaction in the Blockchain is considered to be completely secured and verified.
- This is possible only because of the presence of the consensus protocol which is a core part of any Blockchain network.

Fig. 1

If there is single decision maker as in Fig. 1 then you do not require any consensus.

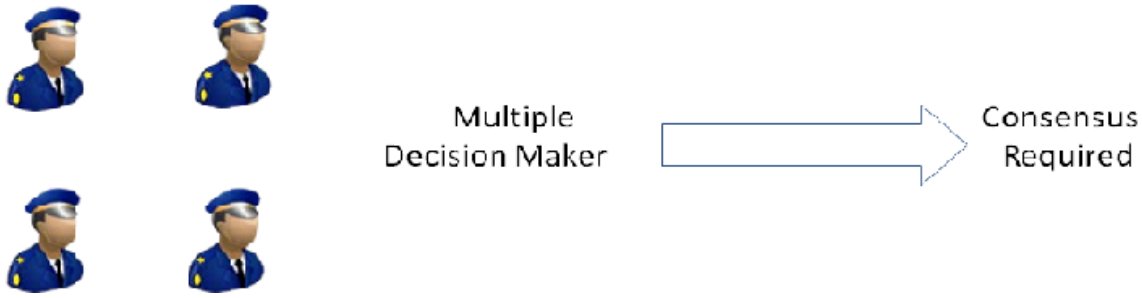

Fig. 2

If there are multiple decision maker as in Fig.2, then in a collective way, you need to come to a consensus using consensus mechanism.

- It sets the rules that all participants must follow to process transaction.
- A procedure through which all the peers of the Blockchain network reach a **common agreement** about the present state of the distributed ledger.
- A group of independent systems to agree on a single version of the truth.
- In this way, consensus algorithms achieve reliability in the Blockchain network and establish trust between unknown peers in a distributed computing environment.
- Essentially, the consensus protocol makes sure that every new block that is added to the Blockchain is the one and only version of the truth that is agreed upon by all the nodes in the Blockchain.
- A **consensus mechanism** is a set of steps that are taken by most or all nodes in a blockchain to agree on a proposed state or value.

1.2 Why Distributed Consensus protocols?

- Traditional Motivation: Reliability and fault tolerance in distributed systems
- Ensure correct operations in presence of faulty nodes.

Example:

- Commit a transaction to a database
- State machine replication
- Clock synchronization
- Eg: Imagine you are in charge of backend database for a company like google or Facebook. These companies typically maintain thousands or even millions of servers, which form massive distributed databases, that records all of the actions that happens on the systems like user's comments, posts and likes and so on. If any new comments come in, that will be recorded in 10 to 15 different server's backend databases. These server needs to make sure that these comments get recorded in all of the databases or none of the databases. For sometimes, because some nodes

might be faulty, These comments are recorded in none of the databases. It is ok. You can go back to the user and say that there is a problem in posting your comments, could you please try again. On the other hand, these actions are recorded in some of the databases and others not. You will be in trouble, because you have inconsistent databases. This is the key problem that motivated the traditional research on distributed consensus protocols.

Byzantine General Problems

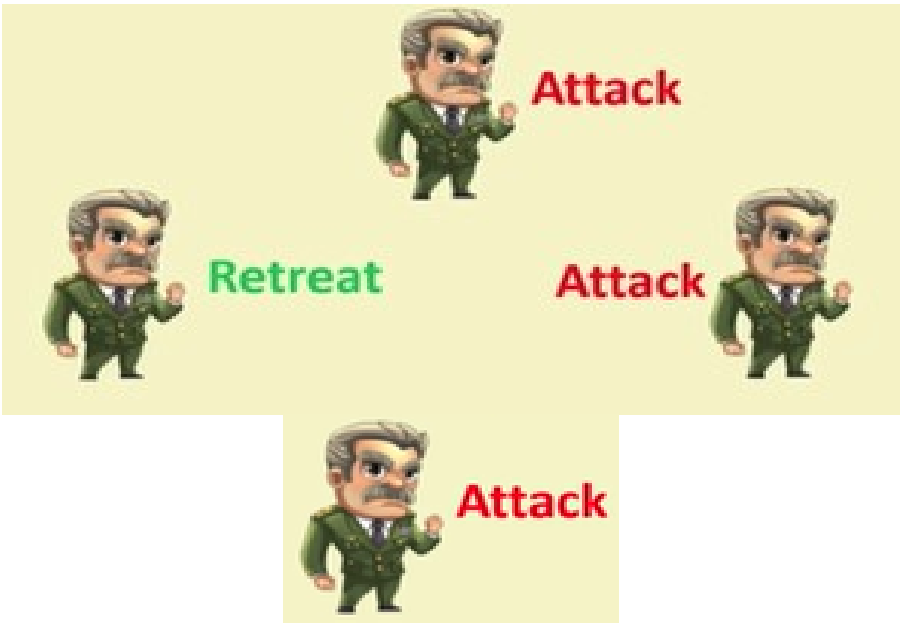

Fig. 3

Imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching an agreement. The generals must decide on when to attack the city, but they need a strong majority of their army to attack at the same time.

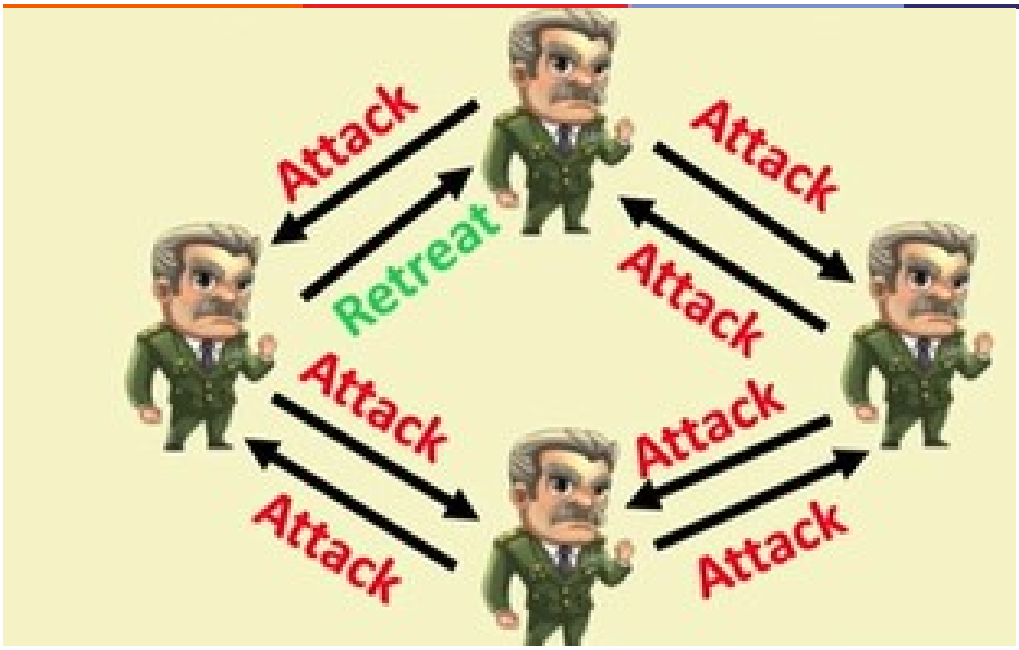

Fig.4

- Consider a message passing systems, a general behaves maliciously. In distributed systems achieving consensus is difficult when some of the nodes are malicious. If node behaves maliciously, we call it as Byzantine node.

The generals must have an algorithm to guarantee that (a) all loyal generals decide upon the same plan of action, and (b) a small number of traitors cannot cause the loyal generals to adopt a bad plan. The loyal generals will all do what the algorithm says they should, but the traitors may do anything they wish. The algorithm must guarantee condition (a) regardless of what the traitors do. The loyal generals should not only reach agreement, but should agree upon a reasonable plan.

1.3 Objectives of Consensus Algorithm

- **Agreement**: All honest nodes decide on the same value
- **Termination**: All honest nodes terminate execution of the consensus process and eventually reach a decision
- **Validity**: The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node

- **Fault tolerant**: The consensus algorithm should be able to run in the presence of faulty or malicious nodes (Byzantine nodes)
- **Integrity**: This is a requirement that no node can make the decision more than once in a single consensus cycle.

1.4 How does Consensus algorithm Works?

- Alice wants to pay Bob. Alice broadcasts the transaction to all nodes. This transaction details contain Alice's signature, because it came from Alice and Bob's Public key as in Fig.3. Different users are performing transactions at same time. Put all transaction into blocks.

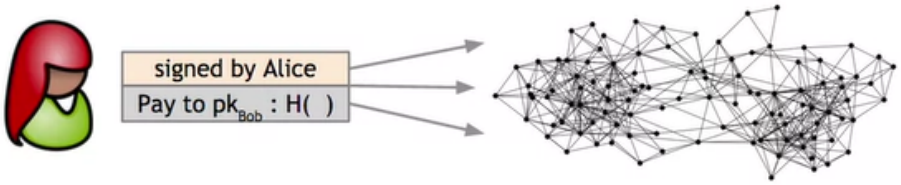

Fig. 5

Miners:

Consensus process are carried out by special peer nodes. Each node has set of outstanding transactions it's heard about. Each node might have a slightly different versions of the outstanding transactions. Peer to peer network is not perfect, some node may heard about a transaction, but not other nodes

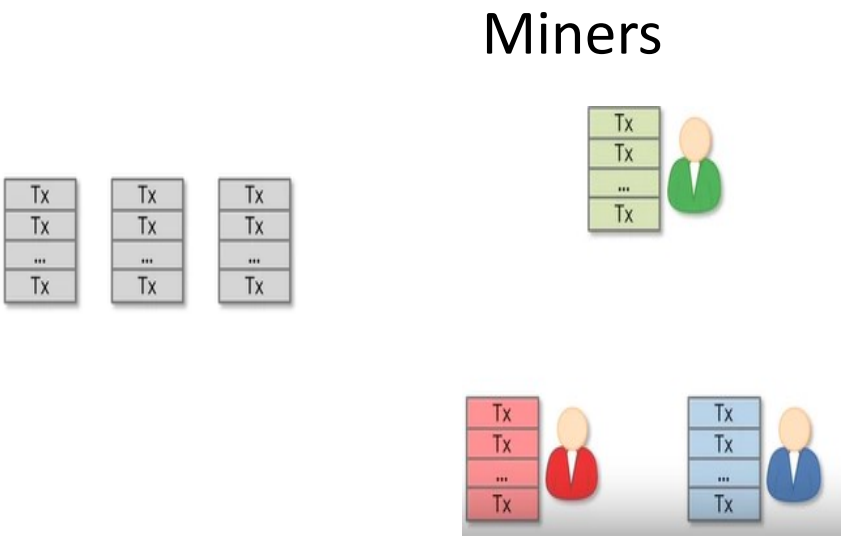

Fig.6

We have sequence of blocks that everybody has agreed upon. There are three miners as shown in Fig. 4, each of them proposes a block, but valid block is added to the existing chain.

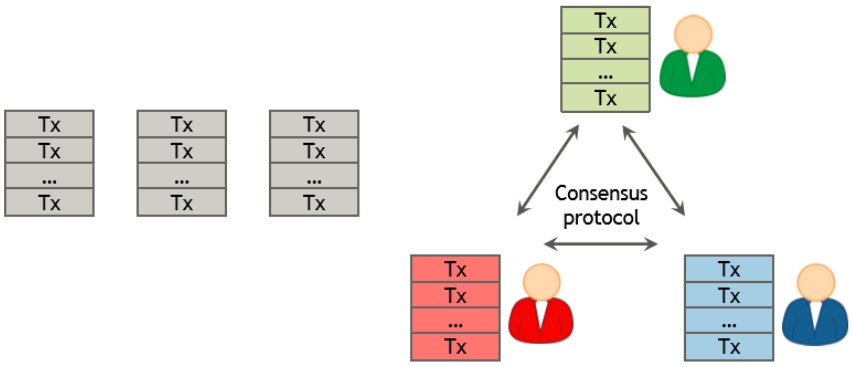

Fig. 7

Fig. 8

A secure chain is a single main chain with a consistent state.

1.5 Why consensus is hard?

- Nodes may crash
- Nodes may be malicious
- Network is imperfect
    - Not all pairs of nodes connected
    - Faults in network

- Latency

1.6 Types of Consensus algorithms

- Proof of Work (PoW)
- Proof of Stake (PoS)
- Delegated Proof of Stake (Dpos)
- Proof of Authority
- Proof of Elapsed Time
- Proof of Scope
- Proof of Space
- Proof of Burn
- RAFT
- PAXOS
- Byzantine Fault Tolerance System
- PBFT.

What are Permissionless Blockchains?

- Popular blockchains such as Bitcoin, Ethereum, Litecoin, Dash, and Monero fall under this category. Also known as public blockchains, these allow anyone to transact and join as a validator. The data on these blockchains is publicly available, and complete copies of the ledgers are stored across the globe. This is what makes it hard to censor or hack these systems. This blockchain does not have anyone who controls it, and one can remain relatively anonymous as there is no need for identifying themselves to get an address and perform transactions.

What are Permissioned Blockchains?

- These blockchains, also known as private blockchains, can be thought of as closed ecosystems that can only be accessed by those who are allowed access. Anyone who is interested in validating transactions or viewing

data on the network needs to get approval from a central authority. This is useful for companies, banks, and institutions that are comfortable to comply with the regulations and are very concerned about having complete control of their data. Ripple is a perfect example of a permissioned blockchain.

1.7 How can you choose the right consensus protocol for your blockchain?

If you are trying to decide on the right consensus protocol for your blockchain, there are some things that you should take into account. They include

- The speed in which your blocks will need to be written into the blockchain? Consensus formation can take time. If consensus is faster, the trust guarantees will be much lower (and vice versa).
- What type of network will you be using? Is it synchronous, partially synchronous, eventually synchronous or asynchronous?. For example, the Internet does not guarantee message delivery and is generally considered to be eventually synchronous.
- How many miners, writers, or validators do you think you will need? These special blockchain nodes are the ones which will select blocks to write to the chain.
- How "final" does a block need to be? Banks and other financial institutions most often expect any transactions to be immediately final (that is, they cannot be rolled back). Some consensus protocols treat block decisions to be conditional on future actions. In others, blocks may eventually be final but are not immediately so.
- To what degree do you put your trust in the nodes/operators? Are you trying to protect your blockchain from nodes that might crash, nodes that might actively attempt to hack the blockchain, or both?
- In a permissioned blockchain, choosing the right **consensus protocol for permissioned blockchain** depends on factors like the extent of decentralization required (For example, how much the participants in a network trust each other, the number of permissions that must be granted to all the participants to carry out important tasks on the network etc).
- Most of the time, these types of blockchains use PBFT algorithms (Practical byzantine fault tolerance) and its variants including voting and

lottery-based consensus as opposed to consensus models like the Proof-of-Work consensus that are prevalent in the major permissionless public networks.

- The adoption of blockchain technology can provide many benefits like enhanced transparency, security and traceability to businesses in all industries. However, blockchain networks cannot function properly without consensus algorithms to verify each and every transaction that is being committed. This is why it is crucial to choose the right consensus model for your specific blockchain.