1. **Consensus Mechanism**

*-“Consensus is the core of Blockchain*. *They decide how a blockchain works”-*

**1.1 What is Consensus Mechanism ? (Cơ chế đồng thuận)**

<https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp>

A consensus mechanism is a fault-tolerant mechanism that is used in computer and blockchain systems to achieve the necessary agreement on a single data value or a single state of the network among distributed processes or multi-agent systems, such as with cryptocurrencies. It is useful in record-keeping, among other things.

(Consensus Mechanism ( Cơ chế đồng thuận ) là một cơ chế chịu lỗi được sử dụng trong các hệ thống máy tính và blockchain để đạt được thỏa thuận cần thiết về một giá trị dữ liệu hoặc một trạng thái duy nhất của mạng giữa các quy trình phân tán hoặc hệ thống đa tác nhân. Nó rất hữu ích trong việc lưu trữ hồ sơ, trong số những thứ khác.)

Ở các cấu trúc hệ thống truyền thống, luôn có cơ quan đầu não tập trung chịu trách nhiệm giải quyết các vấn đề xử lý và thống nhất thông tin. Tuy nhiên, với kiến trúc phân tán peer to peer của Blockchain thì cũng cần đảm bảo sự đồng thuận thống nhất trong luồng xử lý thông tin nhằm đảm bảo tính an toàn và bảo mật cho hệ thống không bị xung đột và mâu thuẫn.

Nói một cách đơn giản, sự đồng thuận là một cách năng động để đạt được thỏa thuận trong một nhóm. Trong khi bỏ phiếu chỉ giải quyết cho một nguyên tắc đa số mà không có bất kỳ suy nghĩ nào về cảm xúc và phúc lợi của người thiểu số, trong khi một sự đồng thuận đảm bảo rằng một thỏa thuận đạt được có thể đem lại lợi ích cho toàn bộ nhóm.

**1.2 Why do we need Consensus ?**

Trong một hệ thống blockchain mà không cần sự cho phép, kiểm tra tính danh, ai cũng có thể trở thành 1 node trong mạng lưới và chỉnh sửa,các giao dịch, bao gồm chúng trong 1 block mới. Khi đó, chúng ta có thể gọi các chỉnh sửa này là một fork, fork này bao gồm các giao dịch đã bị chỉnh sửa, khi đó mạng lưới của chúng ta có nhiều phiên bản khác nhau về thông tin của các giao dịch. Mục tiêu của consensus là để tránh những fork này và tất cả mọi người đều chấp nhật 1 phiên bản thật sự. Kể cả trong một blockchain mà các node được kiểm tra và cho phép, chọn lọc, chúng ta vẫn cần consensus bởi vì chúng ta cũng không chắc chắn 100% rằng các node này tin tưởng được.

**1.3 More explaine about Consensus**

In any centralized system, like a database holding key information about driving licenses in a country, a central administrator has the authority to maintain and update the database. The task of making any updates – like adding/deleting/updating names of people who qualified for certain licenses – is performed by a central authority who remains the sole in-charge of maintaining genuine records.

Public blockchains that operate as decentralized, self-regulating systems work on a global scale without any single authority. They involve contributions from hundreds of thousands of participants who work on verification and authentication of transactions occurring on the blockchain, and on the block mining activities.

In such a dynamically changing status of the blockchain, these publicly shared ledgers need an efficient, fair, real-time, functional, reliable, and secure mechanism to ensure that all the transactions occurring on the network are genuine and all participants agree on a consensus on the status of the ledger. This all-important task is performed by the consensus mechanism, which is a set of rules that decides on the contributions by the various participants of the blockchain.

**2. What types of Consensus Algothrims ?**

Cơ chế đồng thuận thực hiện được dựa trên giải thuật đồng thuận. Có nhiều giải thuật đồng thuận khác nhau.

**2.1 Byzantine General Problem**

<https://blockgeeks.com/guides/blockchain-consensus/>

Ok so imagine that there is a group of Byzantine generals and they want to attack a city. They are facing two very distinct problems:

The generals and their armies are very far apart so centralized authority is impossible, which makes coordinated attack very tough.

The city has a huge army and the only way that they can win is if they all attack at once.

In order to make successful coordination the armies on the left of the castle send a messenger to the armies on the right of the castle with a message that says “ATTACK WEDNESDAY.” However, suppose the armies on the right are not prepared for the attack and say, “NO. ATTACK FRIDAY” and send back the messenger through the city back to the armies on the left.

This is where we face a problem.

A number of things can happen to the poor messenger. He could get captured, compromised, killed and replaced with another messenger by the city. This would lead to the armies getting tampered information which may result in an uncoordinated attack and defeat.

This has clear references to blockchain as well. The chain is a huge network; how can you possibly trust them? If you were sending someone 4 Ether from your wallet, how would you know for sure that someone in the network isn’t going to tamper with it and change 4 to 40 Ether?

What these generals need, is a consensus mechanism which can make sure that their army can actually attack as a unit despite all these setbacks.

We are now going to go through a list of consensus mechanisms which can solve the Byzantine Generals problem.

* 1. **Proof Of Work (PoW) (Bằng chứng công việc)**

Đại diện: **Bitcoin, Litecoin**

Satoshi Nakamoto, Bitcoin’s creator, was able to bypass the problem by inventing the proof of work protocol.

First let’s see how it work in context of the Byzantine Generals problem.

Suppose the army on the left want to send a message called “ATTACK MONDAY” to the army on the right, they are going to follow certain steps.

Firstly, they will append a “nonce” to the original text. The nonce can be any random hexadecimal value.

After that, they hash the text appended with a nonce and see the result. Suppose, hypothetically speaking, the armies have decided to only share messages which, on hashing, gives a result which starts with 5 zeroes.

If the hash conditions are satisfied, they will send the messenger with the hash of the message. If not, then they will keep on changing the value of the nonce randomly until they get the desired result. This action is extremely tedious and time consuming and takes a lot of computation power.

If the messenger does get caught by the city and the message is tampered with, according to hash function properties, the hash itself will get drastically changed. If the generals on the right side, see that the hashed message is not starting with the required amount of 0s then they can simply call off the attack.

However, there is a possible loophole.

No hash function is 100% collision free. So what if the city gets the message, tampers with it and then accordingly change the nonce until they get the desired result which has the required number of 0s? This will be extremely time consuming but it is still possible. To counter this, the generals are going to use strength in numbers.

Suppose, instead of just one general on the left sending messages to one general on the right, there are 3 generals on the left who have to send a message to the ones on the right. In order to do that, they can make their own message and then hash the cumulative message and then append a nonce to the resulting hash and hash it again. This time, they want a message which starts with six 0s.

Obviously, this is going to be extremely time consuming, but this time, if the messenger does get caught by the city, the amount of time that they will take to tamper the cumulative message and then find the corresponding nonce for the hash will be infinitely more. It may even take years. So, eg. if instead of one messenger, the generals send multiple messengers, by the time the city is even halfway through the computation process they will get attacked and destroyed.

The generals on the right have it pretty easy. All they have to do is to append the message with the correct nonce that will be given to them, hash them, and see whether the hash matches or not. Hashing a string is very easy to do. That in essence is the process behind proof-of-work.

The process behind finding the nonce for the appropriate hash target should be extremely difficult and time consuming.

However, the process of checking the result to see if no malpractice has been committed should be very simple.

So, if we are to summarize how Proof Of Work Protocol works with the blockchain.

The miners solve cryptographic puzzles to “mine” a block in order to add to the blockchain.

This process requires immense amount of energy and computational usage. The puzzles have been designed in a way which makes it hard and taxing on the system.

When a miner solves the puzzle, they present their block to the network for verification.

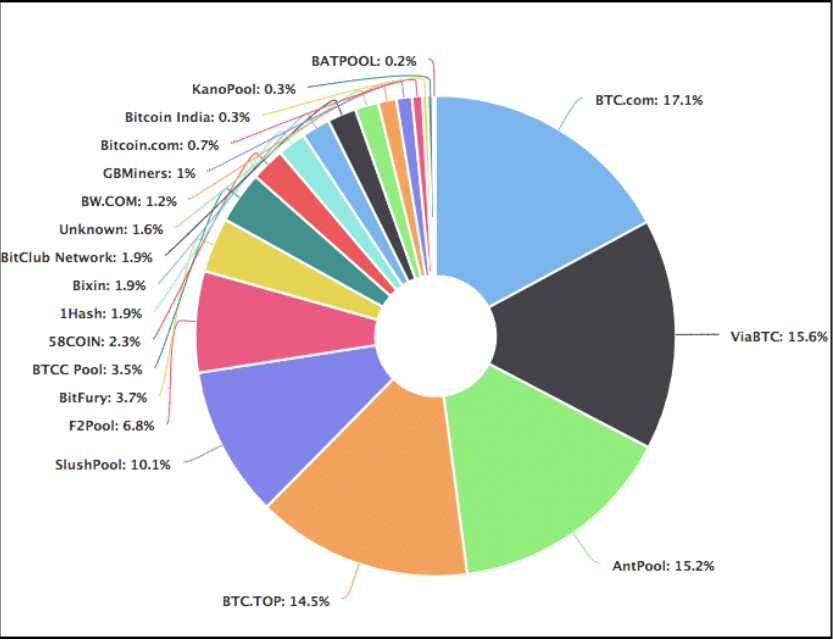
Verifying whether the block belongs to the chain or not is an extremely simple process.

The proof-of-work mechanism definitely answered a lot of questions when it came to solving the Byzantine General’s Problem, but unfortunately there are some issues with proof-of-work.

First and foremost, proof of work is an extremely inefficient process because of the sheer amount of power and energy that it eats up.

People and organizations that can afford faster and more powerful ASICs usually have better chance of mining than the others.

As a result of this, bitcoin isn’t as decentralized as it wants to be. Let’s check the hashrate distribution graph:



As you can see, ~65% of the hashrate is divided among 5 mining pools alone!

Theoretically speaking, these big mining pools can simply team up with each other and launch a 51% on the bitcoin network.

* 1. **Game Theory**

<http://web.cecs.pdx.edu/~nbulusu/papers/gamesec18.pdf>

Although blockchain technology has gained considerable attention from the computer science and economics communities, the use of game theory methods in this technology is limited [16]. In this section, we present the most relevant and recent works that utilize game theory in blockchain technology.

Xu et al. [20] proposed a game theoretical approach to suppress the attack motivation on a blockchain that consists of mobile devices and edge servers. The game is formulated between a mobile device and an edge server, where the mobile device can send a request to the server to acquire a real-time service or launch an attack. On the other hand, the server chooses to either provide the service or to attack the mobile device. The authors introduced a punishment mechanism according to the action record to mitigate the attacks on the blockchain. They have concluded that both players tend to behave finely when the punishment weight is significant. The proposed approach was designed to deal with attacks like zero-day attack, DDOS attacks, and password-based attacks.

Johnson et al. [10] employed a game theoretical model to analyze the incentives for a mining pool to launch a DDoS attack against another mining pool. The players in the game are two competing mining pools, where each one may utilize additional computing resources to increase the chance of winning the mining race, or to trigger a DDoS attack to lower the expected success of the other competing mining pool.

Luu et al. [14] studied the block withholding attack on mining pools using a game theoretical approach by formulating the Bitcoin mining as a game. They analyzed the block withholding attack and concluded that the attack is profitable and well-incentivized in the long-term. The authors derived the game equilibrium state, which is a mixed strategy where all clients are incentivized to attack rather than participate honestly to maximize their payoffs. Finally, the authors concluded that the PoW protocol is vulnerable to such an attack.

In a paper entitled ’The Miner’s Dilemma,’ Eyal [5] studies the scenario when pools attack each other. Open pools (i.e., pools of miners that allow any miner to join the mining work) are vulnerable to block withholding attacks performed by infiltrated miners from competing pools. This paper defines a game where pools recruit some of their participants to infiltrate other pools to diminish their mining capabilities. This game is called the miners dilemma where players are two pools, and their strategies are whether or not to attack each other. The author observes that attacking is the dominant strategy for each player.

All the above works have introduced game theoretical approaches to the PoW mining protocol. As previously discussed in Section 1, PoW is not an attractive approach for blockchains that are efficient-sensitive due to its massive computation demands. In a more relevant work presented by Kiayias et al. [11], Ouroboros consensus protocol was proposed. Similar to our protocol, Ouroboros eliminates the need for an energy-hungry PoW protocol. Ouroboros is based on the Proof of Stake (PoS) protocol. It works by dividing the time into rounds called slots in which each slot is assigned to a leader. The leaders are picked based on the stake they have. A chosen leader is responsible for producing a block for its time slot. The authors utilized game theory to introduce a reward mechanism to incentivize the participants in the blockchain. By means of the game theoretical design, attacks such as selfish-mining and block withholding are mitigated. The rewarding mechanism works by awarding a positive payoff for participants who do not diverge from the protocol.

* 1. **Proof of Stake (PoS) (Bằng chứng cổ phần)**

Đại diện: **Ethereum**

(Một miner nắm 2% tổng lượng tiền thì chỉ có thể validate tối đa 2% tổng lượng giao dịch có giá trị tương ứng)

<https://blockgeeks.com/guides/blockchain-consensus/>

Ethereum is soon planning to move on from Proof of Work (POW) to Proof of Stake (POS).

Proof of stake will make the entire mining process virtual and replace miners with validators.

This is how the process will work:

The validators will have to lock up some of their coins as stake.

After that, they will start validating the blocks. Meaning, when they discover a block which they think can be added to the chain, they will validate it by placing a bet on it.

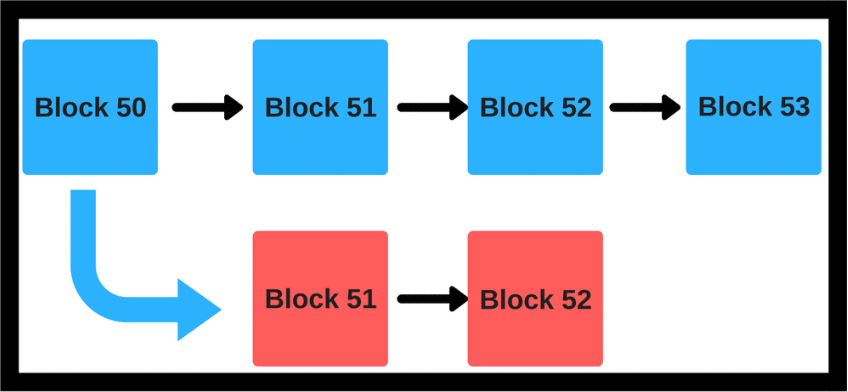
If the block gets appended, then the validators will get a reward proportionate to their bets.

As you can see, the POS protocol is a lot more resource-friendly than POW. In POW you NEED to waste a lot of resources to go along with the protocol, it is basically resource wastage for the sake of resource wastage.

\* The Biggest Roadblock to Proof of Stake

Ethereum developers always planned to eventually move on to proof of stake, that was always their plan. However, before they could do so, they had to address one of the biggest flaws of proof of stake(POS).

Consider this scenario for a moment:



Suppose we have a situation like the one above. There is a main blue chain and a red chain which sort of branches from the main itself. What is there to stop a malicious miner from mining on the red blocks and force a hardfork?

In a proof-of-work(POW) system, this risk can be mitigated.

Suppose malicious miner Alice wants to mine on the red chain. Even if she dedicates all of her hash power to it, she won’t get any other miner to join her on the new chain. Everyone else will still continue to mine on the blue chain, because it is more profitable and risk-free to mine on the longer chain.

Now remember, POW is extremely expensive resource-wise.

It makes no sense for a miner to waste so much resource on a block that will be rejected by the network anyway. Hence chain splits are avoided in a proof of work system because of the amount of money that the attacker will have to waste.

However, things look a little different when you bring in POS.

If you are a validator, then you can simply put your money in both the red chain and blue chain without any fear of repercussion at all. No matter what happens, you will always win and have nothing to lose, despite how malicious your actions maybe.

\* This is called the “Nothing at Stake” problem.

In order to incorporate the POS consensus system, cryptocurrencies need to address this issue and Ethereum is going to do this in a very interesting way by adapting their Casper protocol.

Casper is the POS protocol that Ethereum has chosen to go with. While there has been an entire team busy creating it, Vlad Zamfir is often credited as being the “Face of Casper”.

\* So how is Casper different from other Proof of Stake protocols?

Casper has implemented a process by which they can punish all malicious elements. This is how POS under Casper would work:

The validators stake a portion of their Ethers as stake.

After that, they will start validating the blocks. Meaning, when they discover a block which they think can be added to the chain, they will validate it by placing a bet on it.

If the block gets appended, then the validators will get a reward proportionate to their bets.

However, if a validator acts in a malicious manner and tries to do a “nothing at stake”, they will immediately be reprimanded and all of their stake is going to get slashed.

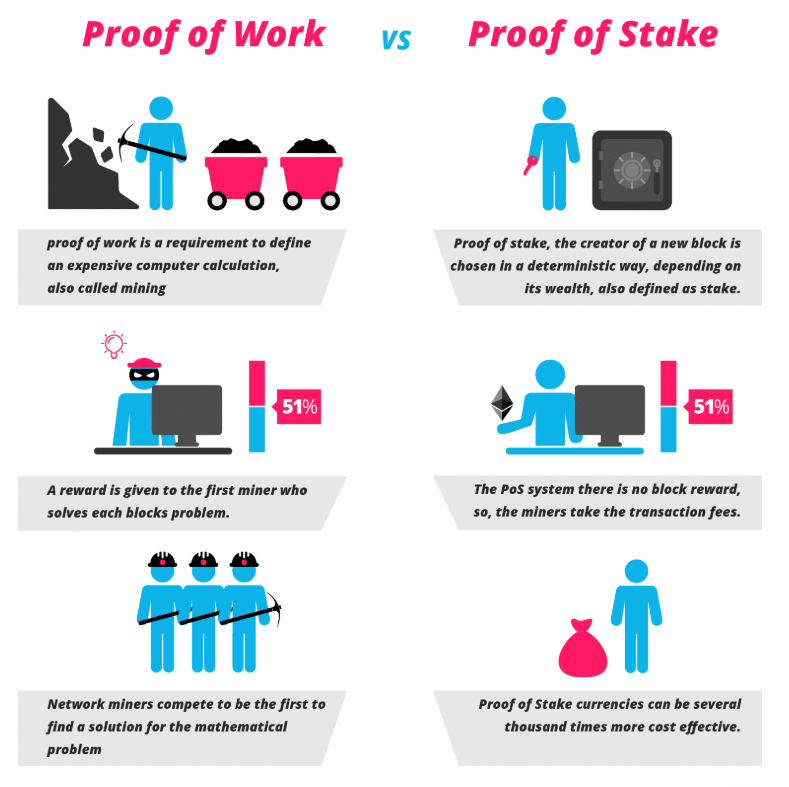
As you can see, Casper is deigned to work in a trustless system and be more Byzantine Fault Tolerant.

Anyone who acts in a malicious/Byzantine manner will get immediately punished by having their stake slashed off. This is where it differs from most other POS protocols. Malicious elements have something to lose so it is impossible for there to be nothing at stake.

Flawlessly implementing Casper and Proof Of Stake will be critical if Ethereum plans to scale up.

Notable Crypto That Uses This Protocol: Ethereum (in the near future)

\*Compare PoW and PoS



* 1. **Delegated proof-of-stake algorithm (DPoS)**

Đại diện: **EOS**

<https://blockgeeks.com/guides/blockchain-consensus/>

* 1. **Practical Byzantine Fault Tolerance(pBFT)**

<https://www.geeksforgeeks.org/practical-byzantine-fault-tolerancepbft/#:~:text=Practical%20Byzantine%20Fault%20Tolerance%20is,optimized%20for%20low%20overhead%20time.>

Đại diện:

**Zilliqa** – pBFT in combination with PoW consensus

**Hyperledger Fabric** – permissioned version of pBFT

**Tendermint** – pBFT + DPoS(Delegated Proof-of-Stake)

**2.7 Delegated Byzantine Fault Tolerance (dBFT)**

<https://blockgeeks.com/guides/blockchain-consensus/>

Đại diện: **NEO**

* 1. **Proof of Activity (PoA)**

<https://www.investopedia.com/terms/p/proof-activity-cryptocurrency.asp>

**2.9 Leased Proof-of-Stake (LPoS)**

<https://academy.binance.com/vi/blockchain/leased-proof-of-stake-consensus-explained>

Đại diện: **Waves**

**2.10 Proof-of-Capacity (PoC)**

<https://www.investopedia.com/terms/p/proof-capacity-cryptocurrency.asp>

Đại diện: **Burstcoin**

**2.11 Proof-of-Importance (PoI)**

<https://digiforest.io/en/blog/blockchain-consensus-algorithms>

Đại diện: **NEM**

**2.12 Proof-of-Authority (PoAuthority)**

<https://digiforest.io/en/blog/blockchain-consensus-algorithms>

**2.13 Proof-of-Burn (PoB)**

<https://digiforest.io/en/blog/blockchain-consensus-algorithms>

<https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp>

Đại diện: **Slimcoin**

**2.14 Proof of Elapsed Time (PoET)**

<https://www.investopedia.com/terms/p/proof-elapsed-time-cryptocurrency.asp>