Consensus Protocols

Name: Jaden Ade
AUD: 15697

Introduction

We all have heard of cryptocurrencies such as Bitcoin, Ethereum and so on, all over social media and on the news. [1] Although Blockchain is nothing new, Bitcoin was one of the first to successfully implement it and attract the public's attention. After which, many crypto companies and currencies emerged thereby making Blockchain a highly discussed and popular topic. Surprisingly, the technology adopted by blockchain is nothing new or just created. Blockchain combines technology like cryptography, P2P networking, distributed system and so on. In addition to this, Blockchain is also very secure because the transactions can't be tweaked or tampered with and all the entities/parties which make the transactions are kept anonymous. Nevertheless, New challenges and security issues keep on emerging every day due to which the design of a consensus remains a major issue. The consensus of Blockchain is that every entity/node maintains the same distributed ledger. In the blockchain, the entity/party is both the server and the host. It requires exchanging information with other similar entities/parties to arrive at a consensus. The Consensus Protocol adopted must also be suitable for a certain blockchain type. [1][2] Some of the Consensus Protocols are:

Proof of Work (PoW)

Proof of Work is used by Bitcoin and more popular Cryptocurrencies. This protocol provides security in the form of block mining. But this protocol achieves its consensus by placing blocks in a distributed system even though the notes may show faults such as them being unreliable, unavailable, malicious, and more. The security of these kinds of the network is supported by specialized hardware and electricity, which makes them costly and inefficient from a resource and environmental standpoint. [3] One of the main challenges of this protocol is Selfish mining. It was an attack discovered by Eyal and Sirer. Selfish mining is when a miner keeps his/her discovered blocks private and continues to mine over them. This attack enables Selfish miners to avail unfair rewards. This scenario destroys the decentralized structure of the system and raises the success rate of various similar attacks. Another kind of attack is Double Spending. Double Spending is when the payment is reversed after the delivery of goods and services. Before, double spending was thought to be difficult due to the mining power. But in 2016, a paper published by Sompolinsky and Zohar exposed that an attacker could easily launch this attack on low mining power with the combination of Selfish Mining. Another kind of attack is Feather Forking. Miller recommended this strategy. The attacker forks the blockchain in this attack, rendering all blocks invalid and verifying the target. The attacker keeps mining until the main chain has advanced k blocks. Fortunately, the attack is not profitable, and the success rate with the mining power is extremely low.

Proof of Stake (PoS)

[1] The stake is used instead of processing capacity to determine which node creates a new block in Proof of Stake. However, the SHA256 puzzle must still be solved by nodes in this Protocol. The difference between this and Proof of Work is that the nodes do not need to change their nonce numerous times; instead, they can solve this with the stake amount. Because it does not require a lot of computer resources to reach a consensus, proof of stake is an energy-saving consensus. [4] For the block to be valid, it needs to have an address and a timestamp. Along with that, the user also needs to provide proof of ownership of the address. The Proof of stake algorithm uses the following conditions:

$Hash(\ Hash(\ Block_{prev}),\ A,\ time) \leq balance(A)\ M\ /\ D$

In the above algorithm, $\text{Block}_{\text{prev}}$ is the block the user is making, and the time contains the current UTC timestamp.

[5] Also, due to one block being created per round of Proof of Stake, the generation and transaction speeds of blocks are much quicker than Proof of work. This has led to the recent popularity of this protocol. The security of Proof of Stake relies on many factors. A few of the Proof of Stake protocols are considered secure as long as the messages sent by the network reach their destinations within a certain time limit.

Delegated Proof of Stake (DPoS)

[7] In Delegated Proof of Stake, the node which creates the block is also responsible for validating the transactions. In this protocol, a token holder can vote for their preferred block producer and their votes will be prioritized based on their stake. Also, Token Holders may give their stake to another voter and cast a vote on their behalf in the block producer election. Once they are elected, they may create blocks after verification of transactions that were created for the last block time in the order in which they were selected. If all blocks in the transactions are verified and signed, they receive rewards. Also, the rewards are shared equally with the users who voted for them. But the rewards will not be given if they don't do their task in the allowed time. After which the block is missed, and transactions remain unverified. However, in Delegated Proof of Stake, everyone earns no matter the concept of rich and poor which makes this protocol more decentralized with better reward distribution. Users could also vote out any offending party/delegate if they believe that there is malicious intent/activity involved thereby keeping real-time voting secured. [6] Another advantage is its efficiency for Generating Blocks. The number of blocks created by this protocol increases as time increases. The reason for this is because the time taken to select the node remains unchanged, which is why the protocol's time increase. This leads to a slight increase in the creation of blocks. Compared to traditional protocols, this one has improved security, efficiency and better decentralization of block generation.

Practical Byzantine Fault Tolerance (PBFT)

[8] Bitcoin employs Proof of Work (PoW), while Ethereum employs Proof of Stake (PoS), both of which are constantly improving. IBM employed PBFT (Practical Byzantine Fault Tolerance) in their blockchain. This so-called "Hyperledger" includes member registration, identity management, auditability, and other features that aid in achieving high performance in specific situations. But, its scalability is limited. This algorithm has been created from Byzantine Generals Problem. It remains unharmed by malicious attacks and has been implemented in many distributed computing environments. The only problem of this protocol is

its speed and scalability. Due to this protocol's fault tolerance, many experts add this to their blockchain systems. [9] PBFT works in a synchronized system and needs many voting rounds which makes it impractical. In PBFT, there is a group which requires a leader, who is then elected using a 'Election Algorithm'. The other nodes are called Replicas. If a primary node fails, A new leader maybe selected with the creation of a new group. However, the nodes do not change their state and the requests remain organized. This is what makes this algorithm safe. [10] PBFT has inspired many BFT protocols with better security and performance. For synchronous networks, PBFT may exploit the timeout algorithm and detect of any issues or anomaly of the primary node.

Conclusion

[10] Ever since the conception of Blockchain implemented by Bitcoin, the world of crypto has not only popularized but also has improved in terms of speed, scalability and so on. What we have discussed in this paper is just four of many consensus protocols. We have been able to analyze how they work, what is their fault tolerance. We have even been able describe their performance in terms of each of them and have identified the current problem / security issue. However, these all are at the time of writing this paper. New Protocols are introduced, and current Protocols upgrade regularly with better security, scalability, and speed. We hope that you were able to grasp the gist of few of the consensus protocols and their functions.

Reference:

[1] Zhang, S. and Lee, J.H., 2020. Analysis of the main consensus protocols of blockchain. *ICT Express*, *6*(2), pp.93-97.

[2] Evermann, J. and Kim, H., 2021. Workflow Management on Proof-of-Work Blockchains: Implications and Recommendations. *SN Computer Science*, *2*(1), pp.1-22.

[3] Zhang, R. and Preneel, B., 2019, May. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 175-192). IEEE.

[4] Vashchuk, O. and Shuwar, R., 2018. Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake. *Electronics and Information Technologies*, *9*(9), pp.106-112.

[5] Nguyen, C.T., Hoang, D.T., Nguyen, D.N., Niyato, D., Nguyen, H.T. and Dutkiewicz, E., 2019. Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. *IEEE Access*, *7*, pp.85727-85745.

[6] Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N.N. and Zhou, M., 2019. Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism. *IEEE Access*, *7*, pp.118541-118555.

[7] Kaur, S., Chaturvedi, S., Sharma, A. and Kar, J., 2021. A Research Survey on Applications of Consensus Protocols in Blockchain. *Security and Communication Networks*, *2021*.

[8] Feng, L., Zhang, H., Chen, Y. and Lou, L., 2018. Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain. *Applied Sciences*, *8*(10), p.1919.

[9] Zoican, S., Vochin, M., Zoican, R. and Galatchi, D., 2018, November. Blockchain and consensus algorithms in Internet of Things. In *2018 International Symposium on Electronics and Telecommunications (ISETC)* (pp. 1-4). IEEE.

[10] Xiao, Y., Zhang, N., Lou, W. and Hou, Y.T., 2020. A survey of distributed consensus protocols for blockchain networks. IEEE Communications Surveys & Tutorials, 22(2), pp.1432-1465.