

TORRENT DRIVEN (TD) COIN: A CRYPTO COIN WITH BUILT IN DISTRIBUTED DATA STORAGE SYSTEM

ANIRUDHA PAUL

ABSTRACT

In recent years decentralized currencies developed through Blockchains are increasingly becoming popular because of their transparent nature and absence of a central controlling authority. Though a lot of computation power, disk space, and energy are being used to run this system, most of these resources are dedicated to just keeping the bad actors away by using Proof of Work, Proof of Stake, Proof of Space, etc., consensus. In this paper, we discuss a way to combine those consensus mechanism and modify the defense system to create actual values for the end-users by providing a solution for securely storing their data in a decentralized manner without compromising the integrity of the blockchain.

1 INTRODUCTION

When Bitcoin [1] was first introduced, its Proof of Work consensus showed us a different way to combat adversaries. Rather than blocking the adversaries directly, it made an attack on the network extremely expensive. Though this system is preventing any major attack on the system, the tendency of centralized mining is increasing exponentially. Nowadays, ordinary people can not participate in bitcoin mining. As a result, the system is not as democratized as initially thought.

Another widespread consensus is Proof of Stake, which is adapted in Ethereum 2.0 [2], Cardano [3], etc., coins, where you stake your own money to participate. Any bad behavior can cost the miners losing their staked money. Though it is power efficient, staking actual currency without doing any work can cause nothing at stake, whale problems, etc.

To address this, we propose a consensus that uses this distributed network of the miner to store user data. By securely storing this data and continuously proving the proof of storage, the miners will earn seed points. Instead of staking the currency they maintain, they will use these earned seed points to claim a spot in the mining round. In a nutshell, miners should provide some utility to earn points that enable them to mine further to get actual cryptocurrency. Spoofing the process of earning these seed points is hard enough so that honest miners have no incentive to deviate from the intended flow. And as the underlying mining technology is basically already tested proof of stake, the integrity of the network is as good as other similar currencies like Ethereum 2.0, Cardano, etc.

2 PREVIOUS WHITEPAPER SHORTCOMINGS

In the Bitcoin whitepaper [1], the author introduced the idea of Proof Of Work, where miners need to continuously try different nonce to generate a hash with the required number of zero bit. But as time goes by, this challenge of generating the intended hash has become so complex that nowadays it is near impossible to mine effectively without designated ASICs, which is a huge waste of money and resources for doing calculations that have no other purpose other than showing commitment.

To address this, Ethereum is introducing Proof of Stake in its system. But it also introduces new problems - as now the "work" part is removed to show commitment, miners have the opportunity to sign multiple blocks from parallel chains, making it hard for forks to converge. Though Ethereum punishes this behavior by slashing the coin of bad actors, it is still not completely secured from manipulation by the whales who have the majority coin in control.

Another consensus mechanism is proof of space [4], where the miner fills up its disk space with garbage information generated by mathematical hash functions, and the verifier sends them challenges from time to time to validate whether the miner is holding the data or not. The issue is the disk space the data is taking is not meaningful. It is there to show commitment, just like Bitcoin's proof of work.

We shall see later in this paper that it is possible to address some of the shortcomings mentioned above.

3 ARCHITECTURE

The whole mechanism of this blockchain can be described with three sections.

1. Block structure
2. Consensus
3. Method of issuing the token

The block structure follows the standard Bitcoin block structure. The consensus follows the proof of stake mechanism, but it is different in a sense it doesn't use the main coin as a stake but instead uses a secondary Seed Bonus Token. The main deviation comes in the method of issuing the tokens. Though the main Torrent Driven Coin (TD Coin) is issued with a standard block reward, the secondary coin (Seed Bonus Token) needed for staking and mining has a different structure for minting.

3.1 Block structure

Transactions are arranged in standard Bitcoin format, where every transaction is spent from the coin, not from the account. The coin owner transfers the coin by digitally signing the hash of the previous transaction and adding the next owner's public key at the end. The new owner can verify the signatures by following the chain of ownership. And to avoid double-spending, only the oldest transaction of any coin is considered valid. [1](#)

In the chain, many of these transactions are arranged in a block, and their hash value links them. The size of the blocks can be dynamically adjusted like Ethereum based on the network congestion. [2](#)

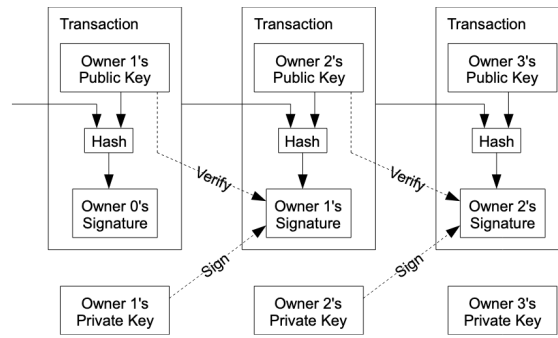


Figure 1: Transaction structure (Standard Bitcoin) [1]

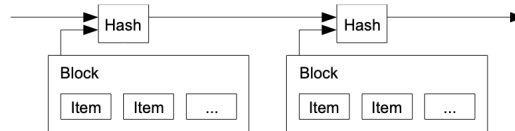


Figure 2: Block structure (Standard Bitcoin) [1]

3.2 Consensus

In this modified proof of stake consensus mechanism [3](#), miners agree to lock up the whole amount of their secondary coin, "Seed bonus token," for getting the chance to validate new blocks of data to be added to a blockchain.

The blockchain algorithm selects validators from the pool of queued miners based on how much seed bonus token their accounts have. The more seed bonus token a miner has, the better chance of being chosen to mine and earn newly minted primary crypto - "Torrent Driven Coin" as a reward if the block gets added to the main chain. A portion of their seed bonus token is burnt to encourage future data seeding, and the rest is returned back to their wallet again.

If a validator is caught cheating, they could be punished by burning all their seed bonus tokens and sending them to an unusable wallet address to which nobody has access, making them useless forever.

3.3 Method of issuing the token

There are three types of token present in this system.

1. TORRENT DRIVEN COIN This is the standard-issue coin that can be exchanged between any parties present in the blockchain. It can only be minted by mining a block in this blockchain. Other than exchanging it as money, users can also burn an

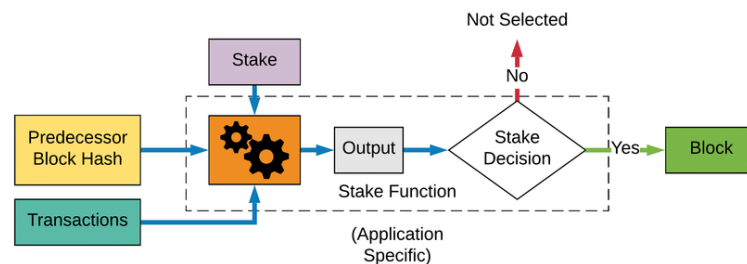


Figure 3: A high level architecture of Proof of Stake consensus [5]

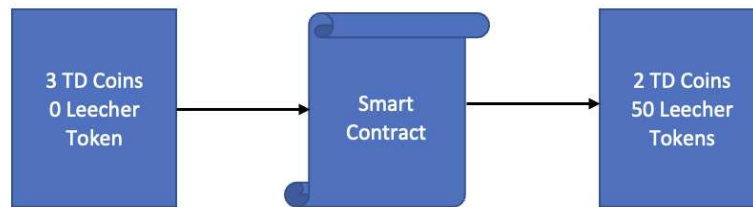


Figure 4: Minting Leech Token

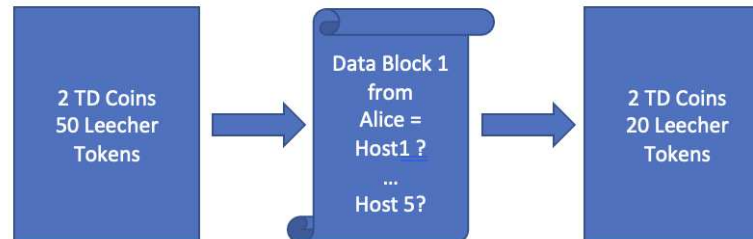


Figure 5: Hosting Request

amount of this coin through a smart contract present in layer 1 of this blockchain to get the Leecher Token described below. The exchange rate between Torrent Driven Coin and Leecher Token will be adjusted dynamically based on supply and demand.

2. LEECHER TOKEN Leecher token grants the ability to upload your data to other users or the ability to host other's data in your machine and get a seed bonus. The only way to get this token is to send an amount of Torrent Driven coin to the predefined smart contract, and the smart contract will burn the coin and give the sender an amount of Leecher Token in exchange. The exchange rate is controlled algorithmically to address the supply-demand issue.

Each leech token grants access to upload or host one MB of data. If a user wants to host 30MB of data in one additional copy in the network, the process will work like this -

For example, Alice wants to make a copy of her data on the peer-to-peer network, and Bob wants to host data for the seed bonus token.

First, Alice will send an amount of Torrent Driven coin to a specific smart contract to get 30 or more leecher tokens as seen in figure 4.

Then she will join the pool of hosting requests to find possible hosts with the 30 leecher tokens in place like in figure 5.

On the other side, Bob also needs to exchange Torrent Driven coins to show commitment as a seeder. So, for example, if he has 50 leecher tokens, he can request the smart contract to match him with $50 * 5 = 250$ data blocks. Five is a constant here, representing each data block will be saved by at least five seeders.

Then what the smart contract will do is match each block Alice wants to host with five different Host addresses. Alice will see all the public keys associated with each of her data blocks on the blockchain. The application on her side will make five copies of data, add the public key to those blocks, make a torrent tracker and publish the tracker on the blockchain. When the tracker gets published, the group of hosts, including Bob, can see the tracker and use that to only download the data attached with their individual public keys in header like in figure 6.

Not only that, each payload is also appended with the seeder's public key and random value before encryption which prevents seeders from swapping the payload between them. Each seeder has his own unique version of the same payload

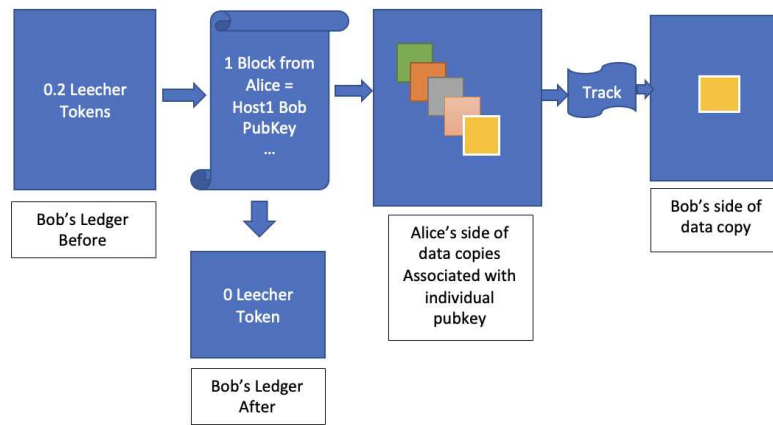


Figure 6: Data Distribution Process

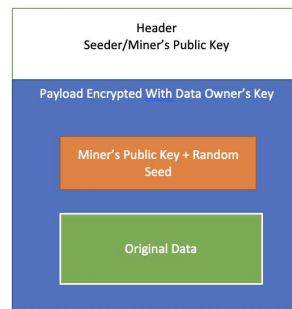


Figure 7: Payload Structure

that the data owner himself can only decrypt. Sample payload structure can be seen in figure 7

The entire process will be covered with Byzantine Fault tolerance and other cross-chain swap techniques if any involved party deviates from the intended path like - going offline, not signing, false signing, etc. The smart contract will ensure that other parties won't have to consume the loss.

So this leecher token is not exchangeable between addresses. It can only be used for hosting purposes.

3. SEED BONUS TOKEN The sole purpose of this token is to ensure data integrity and facilitate staking for proof of stake.

The first issue is - what is the guarantee the host is storing the data? The standard checksum used in torrent systems is not enough to solve this issue. Because to do that, they need to send the data back to the validator, which is slow and unnecessary. Instead, the original data owner or some other validator can save some part of that data in their system, and after random intervals, they can request zero-knowledge proof from the host. The time to response window is short to avoid data recreation. If they can't respond with the proof within that window, that proves the miner is either offline or is not actually storing the data - both of these behaviors are unacceptable.

But if they can respond to the challenge properly along with able to send a small amount of data to show seed liveliness and passing the checksum test, the host will be rewarded with a small amount of new token - the "Seed Bonus Token." The download and checksum check is similar to a normal torrenting system [6]. And the zero-knowledge proof is similar to this proof of space and time described in this paper [7]. The whole process can be done in side chain and only publish the result in main chain to claim reward.

Now there is a new token in the account for potential miners - which is really hard to spoof and also not transferable. That's why the reason to be a bad actor to earn this token is very slim. So the possibility of the host honestly seeding the data is much higher.

But now the question is, what is the incentive for them to earn this reward? The answer is - they can use this reward to stake to get a chance to validate a block and get the block reward in TD coin, which is actually transferable and can be used as actual currency.

From this point on, the consensus falls back on the proof of stake. The difference here is - instead of staking the main currency, the miners are staking their off-chain hosting work reward earned through proof of space.

4 SCALABILITY, SECURITY, AND RESOURCE-EFFICIENCY

The system is scalable in a sense underlying everything; it is still using the proof of stake as its consensus, which is pretty scalable and secured under heavy transactions. One might think the whole process of downloading and uploading data can slow down the main blockchain - which is not true. Cause the upload, download, and zero-knowledge prove that part of the system can be considered off-chain work and side chain transactions. Only the result of each checkpoint is published on the main chain. So the main chain is not slowed down by all the bottlenecks associated with proof of space.

The main network will continue to use proof of stake, but the staked coins are influenced by tokens earned in the independent torrenting mechanism and proof of space consensus.

The whole system is resource-efficient because though miners have to do a lot more work, especially allocating more disk space than the traditional proof of stake, the silver lining is that those works are not wasted work. By using the storage, they are providing actual value to the users. There is no unnecessary computation power used like Proof of Work.

Overall the security is also up to par with other traditional coins. The original main network is using proof of stake, which after many iterations and research, is now pretty secure to be relied upon. The staked point is going through the zero-knowledge proof of space and byzantine consensus, making it extremely hard to profit from being a bad actor. There is no way to replicate the data as the packets are encrypted with individual miner's public key, and the only way to list your public key as a valid host is by burning actual Torrent Driven coins. As the exchange rate is controlled algorithmically, it is not easy to inflate or deflate the staking pool or make a hostile takeover.

So by combining all those techniques, the proposed blockchain is scalable, secured, and resource-efficient and provides actual value to users instead of the perceived value of traditional crypto coins.

5 CONCLUSION

The primary goal of this white paper was to create intrinsic value for the whole distributed mining network of blockchain. By incorporating torrenting mechanisms in the proof of space framework, we have introduced a solution for securely and reliably storing multiple copies of data on the internet. Moreover, as there are monetary exchanges and interests involved in the process, the chance of abandoning the torrent by the seeder is really low. And it is also a great way to incentivize hosting private encrypted data. In the process, it is now replacing the redundant generated data of Proof of Space with actual meaningful information.

By adding signatures, zero-knowledge proof, and byzantine consensus to track the seed status and facilitate a reward mechanism, we have minimized the risk of bad actors creating fake seed data and, in the process, attacking the staking pool.

On the other hand, removing staking coins with staking work has introduced a toned-down version of Proof of Work in the Proof of Stake mechanism, making the architecture secure like PoW and scalable like vanilla PoS.

Finally, introducing a utility inside the network and regulating the exchange rate algorithmically has the possibility of reducing the current deflationary nature of the crypto coins where no one wants to use it in the real-world other than speculative investment. Because in this architecture, seeding more data is beneficial for the miners, that's why the competition among miners can reduce the cost for the storing data for the normal users.

REFERENCES

- [1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009.
- [2] Vitalik Buterin. Ethereum white paper: A next generation smart contract & decentralized application platform. 2013.
- [3] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual international cryptology conference*, pages 357–388. Springer, 2017.
- [4] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. Cryptology ePrint Archive, Report 2013/796, 2013. <https://ia.cr/2013/796>.
- [5] Christopher Natoli, Vincent Gramoli, and Paulo Veríssimo. Deconstructing blockchains: A comprehensive survey on consensus, membership and structure, 08 2019.
- [6] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72. Berkeley, CA, USA, 2003.
- [7] Tal Moran and Ilan Orlov. Simple proofs of space-time and rational proofs of storage. In *Annual International Cryptology Conference*, pages 381–409. Springer, 2019.