Bitcoin & Ethereum: Two Decentralized P2P Applications

Roy Shadmon

November 19, 2018

Abstract

Bitcoin and Ethereum offer a public network for peer-to-peer financial transactions without the need of a trusted third party. Through the blockchain and cryptographic techniques, users are able to make confidential and authenticated transactions by a fully distributed and decentralized proof-of-work consensus mechanism. We describe how the Bitcoin and Ethereum networks manage to create a public marketplace where–at the time of writing–$4.1 billion and $1.8 billion dollars respectively were transferred from peer-to-peer in the last 24 hour time period [1].

1 Introduction and Motivation*

Founded in 2008, Bitcoin became the first peer-to-peer (P2P) electronic payment system that took away control from financial institutions and gave control to a community at large. Therefore, rather than needing to trust a third party to manage funds, Bitcoin uses cryptographic proof to manage funds on the network. Additionally, Bitcoin proposes a trustless solution to the double spending problem discussed in Section 4.4.

Ethereum was first released in 2015. The main motivation of Ethereum is the same as Bitcoin while also offering a Turing-complete programming language that can be used to create Smart Contracts. Smart contracts allow users to create decentralized applications (DAapps), which we will discuss in Section 2.2. Additionally, Ethereum was built to be simple to use, universal to anyone, available anywhere, agile to architecture/protocol modifications, and non-discriminatory to all.

The rest of this section provides examples that motivated the creation of Bitcoin and ultimately Ethereum. Section 2 provides the foundation for Bitcoin and Ethereum. Section 3 discusses how transactions on the networks. Section 4 discusses challenges, security threats, and how those problems are mitigated on each network. Section 5 discusses a new consensus protocol and some interesting DApps currently being developed. Lastly, Section 6 contains our final remarks.

1.1 Centralized Banking

There are two guidelines a person should have regarding their money at any given moment: to have the capabilities to make financial transactions and to ensure only they have access to their funds.

Banks operate as the governance body that manages peoples' funds, while also enforcing rules that determine what interactions are allowed to be made with the funds. For example, banks restrict certain transactions during specific hours of the day or if they do not have the means to process a transaction. A person can only wire funds from one account to another during their bank's business hours, as well as, they cannot request to withdraw one million dollars from their account at a moments notice; most banks don't have that much cash on hand. In addition, bank accounts are vulnerable to fraud because handwritten signatures are easily fabricated and difficult to authenticate. Although there's insurance to recover stolen funds, the process is long and tiresome. Bitcoin and Ethereum, on the other hand, are networks that guarantee a person autonomous control of their funds given they can provide a valid digital signature. They also ensure that only the holder of the digital signature can make transactions from the corresponding account.

*In this paper, assume we are talking about both Bitcoin and Ethereum if we do not mention either specifically. Additionally, we will use the term Bitcoin and Ethereum when discussing the networks and BTC and ETH respectively when discussing the currency.

2 Bitcoin & Ethereum

The Bitcoin and Ethereum blockchains are both a distributed and decentralized peer-to-peer payment system based on cryptographic proof rather than trust. Both blockchains allow people to send money from one person to another without the need of a trusted third party. Instead of needing a financial institution to facilitate the transferring of money, a group of untrusted peers, called miners, use proof-of-work (explained in section 3.2) to confirm or reject the validity and authenticity of money transfers (called transactions).

2.1 Transactions

To make a transaction i.e. to transfer funds from one party to another, a party must hold a valid digital signature for the transaction to be validated, as well as pay a transaction fee (in Bitcoin) or gas (in Ethereum). Anyone with a copy of the digital signature (the private key) used to sign a transaction from the respective address has full control of the money stored by that address. Moreover, transactions are not encrypted, which allows anyone to view the entire ledger of the blockchain.[†] We discuss in Section 3.4 what happens when miners disagree regarding the transactions in the ledger.

When a user initiates a transaction in Bitcoin, they include a digital signature to verify they own the address which BTC is being sent from, the destination address, the transaction fee they're willing to pay, and some other details. Thus, the cost to transfer funds in the Bitcoin network is the amount to transfer plus the transaction fee.

A user initiating an Ethereum transaction inlucdes their digital signature to verify they own the address, the destination address, the gas willing to pay in Wei[‡], the max gas price which limits the number of computational steps a transaction execution is allowed to make, and some other security details (more than that of Bitcoin). The max gas price protects users from accidentally initiating an expensive transaction. Furthermore, if the user does not include enough gas, then the transaction is immediately rejected and the miner takes the gas already spent.

Ethereum contracts include more functionality and additional security measures. For example, to prevent the double spending problem (discussed in Section 4.4), Ethereum transactions include a nonce. Every transaction in the Ethereum network from an address must be verified in the exact order it was made. For example, if a miner tries to verify an Ethereum transaction from an address with a nonce value of 4 and has no record of a transaction with a nonce of value of 3 from the same address, the miner rejects the transaction. Moreover, Since Ethereum includes a Turing-complete programming language, Ethereum transactions aren't restricted to just sending money from one party to another; the transfer of money can be programmed to abide pragmatic logic through Smart Contracts.

2.2 Smart Contracts[§]

Smart Contracts are self-executing scripts that allow for dynamic, real-time execution of code triggered by transactions to the network. They also allow for two mutually distrustful counterparties to fairly interact, because each party can inspect the code prior to engaging in the contract. The network provides the guarantee that the contract will be explicitly followed as it is written.

More specifically, Smart Contracts published on the network have their own address. A function in the contract can either be triggered by addressing a transaction to the function or through data-driven events. Each time a contract is triggered, the transaction triggering the event is recorded on the blockchain. Contracts can contain an access control list (enforced by crpytographic signatures) that restricts who or when a person can make a transaction to trigger a function. For example, a contract can express the following type of logic: "If address X (the Smart Contract's address) has 10 or more ETH, then transfer 5 ETH to address Y (a secondary address)." The Smart Contract code for this contract can be seen in Figure 1.

[†] A ledger is the record of all transactions on the blockchain. When a set of transactions are added to the blockchain, each miner adds the set of transactions to their ledger. In short, the ledger is a log that records when valid transactions are made.

[‡] Wei is the smallest unit of measure of an Ether. 1 Ether = 10^18 Wei

[§] Both Bitcoin and Ethereum support Smart Contracts, however, they are significantly more extensible on Ethereum than on Bitcoin. We will, thus, only discuss Smart Contracts in relation with the Ethereum network.

```
pragma solidity ^0.4.3;

import "browser/mortal.sol";

contract TransferFunds is mortal {

    address myAddress = 0xdd870fa1b7c4700f2bd7f44238821c26f7392148; // My Secondary Address
    uint256 maxWeiToHold = 10000000000000000000; // 10 Ether
    uint256 tenEtherInWei = 5000000000000000000; // 5 Ether

    function () public payable {}

    function transfer10EtherIf () public payable {
        if (checkBalance())
            myAddress.transfer(tenEtherInWei);
        else {
            revert();
        }
    }

    function checkBalance () private view returns (bool) {
        if (address(this).balance >= maxWeiToHold) {
            return true;
        }
    }
}
```

Figure 1: This contract will have one triggerable function (transfer5Ether(), which when triggered will check the current balance of the contract and transfer 5 ETH to another address if the balance >= 10 ETH. Otherwise it will revert the transaction (i.e. cancel the transaction so it will not be recorded on the network). Additionally, we are including the contract mortal.sol to ensure only the owner of the contract can trigger the transfer5Ether function. If another party attempted to trigger that function, the transaction would immediately be rejected.

Since all transactions on the ledger are immutable[¶], a properly written Smart Contract should describe outcomes of all possible situations–including when the Smart Contract terminates[‖]. In this example, the empty function that acts as a constructor allows for the contract owner to kill the contract. Additionally, it is important to note that just like a normal transaction costs a certain fee, a transaction to create or trigger an Ethereum Smart Contract costs gas (paid by the address making the transaction).

2.3 Wallets

In contrast to a user holding physical currencies (such as US dollars) inside a physical wallet, Bitcoin and Ethereum addresses are stored in cryptographic wallets, where the ownership of an address referencing an amount of currency is established through providing the private key of the address. The address is created through a hash computation of the public key, which allows you to be able to create additional addresses that can be authenticated by the private key. Additionally, multiple addresses are contained within a wallet that point to a specific location on the Bitcoin or Ethereum network; where the official balance of each address is stored.

In both Bitcoin and Ethereum, addresses should only be used once for every transaction to ensure the confidentiality of the receiver. The reason for this is made obvious in a simple event where Alice sends money to Bob's lone address that already contains say 10,000 BTC. For this event to occur, Bob must have provided Alice with his

[¶]Blockchains are append-only

[‖]Smart Contracts are deterministic. If Smart Contracts were non-deterministic, then when a triggered function executes, each node on the network could possibly return a different random result. Such an event would prevent the network from reaching a consensus on the transaction.

lone address. Since the ledger is not encrypted, Alice could query Bob's provided address to the blockchain network and notice that the address contains 10,000 BTC. If Alice were to leak Bob's single address with his name (since she knows it), then anyone could confirm Alice's insights regarding Bob's ownership of over 10,000 BTC. That action could endanger Bob's safety as he could become the target of a robbery or kidnap for ransom victim. This situation would be avoided if Bob had Alice send the payment to a different address containing 0 BTC. If so, Bob would be able to keep his address with 10,000 BTC confidential. Even if Bob sent money from his new address to his 10,000 BTC address, Bob's identity behind the 10,000 BTC address would remain confidential. If a person tried to trace transactions of the 10,000 BTC address, they would not be able to explicitly prove that the address belongs to Bob since Bob could have sent money from the address containing Alice's payment to someone else and not himself.

2.4 Confidentiality

Although anyone can query the network to retrieve the balance of a certain address, as long as a user does not publicize their address, their identity remains confidential.

2.5 Public / Private Keys

Digital keys, also known as digital signatures, come in pairs (public key, private key) and are used to ensure the authenticity of a user making a transaction from an address. When a user makes a transaction, they sign the transaction with their private key. The network client is then able to verify the authenticity of the sender making a transaction from an address by using the sender's public key to verify the signature made with the private key. If the sender is successfully authenticated, the transaction is valid and is sent to the miners. Moreover, the digital signature also ensures the integrity of the transaction since any modification to a transaction would had to have been resigned. This is important because–given no attacks on the network occur (discussed in Section 4.3 and 4.4)–once a transaction is included in the longest-chain it is immutable. Since the digital signature can only be made by an authentic user, the signature also ensures nonrepudiation[**]. As a result, the private key must always remain private because anyone with the private key of an address can make transactions from that respective address.

The foundation of Bitcoin and Ethereum's cryptography is a one-way cryptographic function called elliptic curve multiplication. The basic reason to use elliptic curve multiplication is that it's easy to calculate but impossible to reverse to find the inputs if given the outputs.

3 Miners

Miners are located all over the world and have independent decision power to select and verify transactions to include in a block on their nodes (hence, the distribution and decentralization). Miners retrieve pending transactions from either the respective Bitcoin or Ethereum client. They then group a set of transactions to create a block. Miners discard transactions that fail specific validity tests and can choose to ignore transactions with no fees or if the provided fee is too low. Additionally, each miner competes against the other miners to solve a cryptographic hash (a puzzle), where the first miner to build a block and solve the puzzle (by calculating the nonce[††] i.e. the proof-of-work) wins the race. The winning miner then broadcasts the new block to all other miners, and receives the sum of all transaction fees and the block's mining reward (we discuss block mining rewards in Section 3.2). After the losing miners validate that the nonce is correct, that all of the transactions in the block are valid (not already spent), and that the block extends the longest chain, they add the new block to the ledger. The process then repeats with new transactions grouped in the next block.

3.1 Miner Network

When a new miner wants to join the network to start competing for the next block, the new miner must establish connections with other peers. These connections are used to receive a copy of the ledger, to receive new

[**]The message sender cannot claim they did not send the message

[††]A different nonce than the one in an Ethereum transaction.

transactions, and to propagate new candidate blocks[‡‡]. This is important because the new miner must be able to build and propagate a new block with the found nonce to other miners to receive the reward. They must also maintain connections with other miners to learn if another miner finds a correct nonce to prevent wasting work (compute resources). Typically, each miner has between 8 and 125 connections at any given moment. Miners are able to connect with each other through a handshake, which establishes a timestamp (for time synchronization), IP addresses, and protocol version. A miner drops a connection with another miner if a message isn't returned after 3 heartbeats, where each beat occurs every 30 minutes. It's crucial to ensure active connections between other miners because miners may choose to stop or start mining as they please. Having no connections with other miners prevents the miner from propagating or receiving blocks.

When a transaction is made, a miner must verify the authenticity and integrity of the transaction. Once successful, the miner broadcasts to their connections an inventory (inv) message that states: "I know about a new transaction" excluding the actual transaction data. If the neighbor doesn't know about the broadcasted transaction, then they will make a "getdata" message request. The miner will then send the complete transaction record to the requesting miner(s). Those miners will also verify the transaction and propagate the message to their connections. By having connections broadcast transactions to their connections, the networks can ensure that not only all transactions are eventually included in a block, but also miners receive every transaction. If a transaction does not make it into the next block, the miner must rebroadcast the transaction to its connections.

The protocol of asking before sending the full transaction data to other peers who have never seen the transaction helps minimize the total load on the network. In addition, miners in the Ethereum network must also check that the previous block referenced in the new block is valid, that the timestamp of the new block is greater than that of the referenced previous block, that the block number and the gas limit is valid, and that the total gas consumed by the transaction does not exceed the set gas limit. Moreover, since the most previous state of the blockchain is recorded in the Ethereum network, miners do not need to store the entire blockchain history to mine. Contrary to Ethereum, Bitcoin miners must store the entire blockchain ledger to mine and mitigate double spending attacks discussed in Section 4.4.

3.2 Proof-of-Work

Proof-of-Work is a repetitive calculation of large hashed numbers. To mine a block, miners must find a specific number that satisfies the target nonce. In addition, blocks are mined every 10 minutes in the Bitcoin network and every 15 seconds in the Ethereum network [2].

Bitcoin's target nonce value is adjusted every 2016 blocks (about every two weeks). This is to compensate for improvements in hardware performance. The calculation of Bitcoin's new target nonce is done through the following equation:

$$T_{\text{new}} = T_{\text{prev}} * \frac{t_{\text{actual}}}{(2016 * 10(\text{min}))}$$

where t_{actual} is the time it took to generate the last 2016 blocks, T_{prev} is the previous target, and T_{new} is the new target. For example, finding the target nonce works like finding a random hashed number that begins with 42 zeros given a specific range of numbers. Given this example, an average of 2^{42} attempts would be needed before the puzzle is solved. Moreover, if 2016 blocks were generated in less than 2 weeks, then the overall computing power of the network has increased, which results in the PoW difficulty to be increased (meaning a larger target nonce). Conversely, if it took more than 2 weeks to mine 2016 blocks, then the PoW difficulty would decrease.

Block rewards were initialized at the start of the Bitcoin network to subsidize the cost of each transaction. Each mined block initially had a 50 BTC block reward, which a winning miner would receive for publishing the next block on the longest chain. After every 210,000 blocks are mined, the mining reward halves. Eventually, however, the mining reward will decrease to less than the smallest denomination of BTC called a Satoshi (1 Satoshi = 10^{-8} BTC). Once this happens, the only incentive to mine the next block on the Bitcoin network would be for the transaction fees reward, which will make using Bitcoin more expensive. Ways to minimize transaction fees in the Bitcoin network will be discussed in Section 4.1 and 5.1.

[‡‡]A candidate block is a block a miner proposes to the network to be added to the longest-chain

Ethereum PoW is slightly different. While the PoW difficulty in Bitcoin is reset every 2016 blocks, the PoW difficulty in Ethereum dynamically changes per block. This helps ensure that a block is consistently mined every 15 seconds, which mitigates the threat of a soft fork overcoming the longest chain. In addition, block rewards in Ethereum are static at 5 Eth. Therefore, the overall reward a successful miner would get is the block reward plus the sum of gas included in the block's transactions.

3.3 Mining Pools

Mining pools are a group of miners who group their computer hash power to mine blocks on the blockchain. The percent compute resource one provides to the group, is the percent reward a member receives if their mining pool wins a race to mine the next block. Additionally, a mining pool's chance of success to mine the next block directly correlates to the proportion of the total amount of hash power they have in the network.

Since the process of mining a block is done through brute force, the mining pool's server assigns each member of the mining group a range of nonces to search, with respect to the percent compute resource contribution the individual provides to the group. Once a member of the pool finds the correct nonce, the member communicates the nonce to the pool operator who then communicates with the pool server. For the mining pool server to communicate with the respective network, the pool server is connected to the blockchain network through a gateway that uses either the Bitcoin or Ethereum Remote Procedure Calls (RPC) protocol, respectively. This requires the mining pool's communication to have low latency, because any delay could result in another miner or mining pool mine a candidate block quicker. Furthermore, every member of the mining pool must also learn about new blocks and transactions quickly to avoid wasted work from being done on an old block. The more wasted work done on an old block results in less work being done to mine the next block, which negates the pool's profits.

To prevent pool members from cheating their proof-of-work assignments, the mining operator may randomly ask any member of the group to show their proof-of-work results. Although this is computationally expensive, this prevents pool members from cheating the group as the punishment for cheating results in the pool member being removed from the mining pool.

If a mining pool or single miner ever controls more than 51% of the total hash power in the network, then the blockchain becomes at risk to mutability. We discuss this attack on the blockchain and how it's mitigated in Section 4.3.

3.4 Consensus when disagreements occur among miners

Miners rejecting another miner's block can happen. For example, a rejection of a block can occur if two miners think they were first to create a new block by finding a valid nonce. If this does happen, a fork of the blockchain occurs (a disagreement) and the disagreement is resolved by favoring the longest chain of blocks. With this approach, miners supporting shorter chains will eventually acquiesce and accept the fork containing blocks of the longest chain because only the first miner to confirm a block to the longest chain is given the financial reward. This is because shorter chains are discarded, as well as the rewards for adding a block to a shorter chain are returned to each unprocessed transaction. Miners doing work on a shorter chain eventually adopt the longest chain because by building blocks on a shorter chain the miner is, in a sense, doing "free work"[§§]. Thus, they are incentivized to drop the shorter chain and support the longest chain as soon as they recognize they're working on a shorter chain. Moreover, as long as at least 51% of the total compute power of the network agrees on the blocks of the longest chain (hence, 51% of the compute power is being used to find the next block of the longest chain), it is computationally impossible for a shorter chain, over time, to overcome the longest chain length.

4 Challenges

Ethereum and Bitcoin are both limited with the number of transactions that can be handled by the network[¶¶]. Each transaction also costs some fee to make (as solving the puzzle and creating new blocks use compute re-

[§§] By continually losing the race, the miners never win the financial reward.

[¶¶] Ethereum ≈25tx/second, Bitcoin ≈10tx/second

sources that someone needs to pay). Since it is predicted for fees to become increasingly more expensive (because tougher proof-of-work problems will be needed to solve to confirm a block and the decrease in the block reward prize in Bitcoin's case), and because of throughput limitations, not all transactions need to immediately be on the blockchain.

A centralized solution to offline transactions is Venmo (a centralized payment intermediary app) that provides the means to make transactions without the centralized bank keeping track of who spent money, who received money, and how much money was transferred. Venmo works as such, person A sends money to person B's Venmo account. Whenever person B desires, they can send money back to person A, to another person C, leave it in their Venmo account, or transfer it to their bank account for a fee. In some cases, person B might consider waiting to transfer the money to the bank if they're expecting another payment to only pay bank fees once. This process illustrates how Venmo operates as an offline payment method to avoid some transaction fees. Both Bitcoin and Ethereum are capable of Venmo's features through payment channels.

4.1 Bitcoin's Lightening Network

Bitcoin resolves the low throughput transaction problem through the lightening network. The lightening network creates an off-chain, bi-directional payment channel between two parties with minimal transaction fees [***]. Every off-chain transaction requires the digital signature of both the sender and recipient in case a disagreement occurs. If no disagreements occur, either party can close the channel where the final signed transaction is reported to the network. For example, party A sends 5 BTC to party B, who signs the transaction. Party B then sends 2 BTC back to party A, who also signs the transaction. The payment channel now has a record that A's balance is reduced by 3 BTC and B's balance is increased by 3 BTC. Let's assume the two parties are done making transactions to each other. Thus, they send the final transaction of A's balance is 3 less, and B's balance is 3 more, which gets validated because both parties signed the last transaction.

If a disagreement occurs off-chain, then the blockchain will deterministically enforce the last mutually signed transaction. There are problems with this solution as one person can cheat once.

4.2 Ethereum Payment Channels

Ethereum's payment channel operates similarly to Bitcoin's. The main difference is that Ethereum payment channels are operated through Smart Contracts, which could be used to solve the problem Bitcoin faces when one party prematurely closes the channel. To do this, both parties can initiate another Smart Contract that acts as the "trusted" third party arbitrator. In the same example above, rather than party A needing to trust party B to sign the transaction acknowledging they're receiving 5 BTC, party A could send 5 BTC to a secondary Smart Contract. For party B to receive the 5 BTC, they must sign the transaction, which would trigger the Smart Contract to release the 5 BTC to party B. If party B doesn't sign the transaction after a set amount of time, then the 5 BTC would be returned to party A. Another interesting solution is Sprite Payment Channels [3].

4.3 51% Attacks

A 51% attack on the blockchain occurs when a miner or a group of miners control more than 51% of the total compute power on the network. Since the main chain is the longest chain, a miner(s) who controls more than 51% of the compute power can fork the chain, pick whichever transactions to verify, and eventually validate enough blocks for their chain to become the longest chain; i.e. their chain would be the main chain with all the new transactions. The problem with this is that an attacker can make a large payment for some item, receive the item after 6 successive blocks, and then fork the directly previous block containing their transaction. Since they control more than 51% of the hash power, they are able to mine the next block faster than the minority mining on the longest chain. Eventually the fork will become longer than the longest chain, which causes the other miners to accept the fork as the longest chain so they can continue competing for the block rewards.

[***]It costs a transaction fee to initiate the payment channel and to close the payment channel, but no payments in between.

There is currently no way to prevent such attacks on any PoW consensus network, as the point of the consensus mechanism is 1 CPU = 1 vote. However, even if an attacker did manage to control more than 51% compute resource, they are–to a debatable extent–disincentivized to construct such an attack as it would cause severe distrust in the network and rapid selling of the currency. Ultimately, this would cause the overall price of the currency to drastically decrease. On the other hand, if the miner(s) used that power to ethically mine every next block, they would win the mining reward of next block more often, which will reduce the amount of "wasted" work done when losing the race and increase their payout. A new consensus mechanism called Proof-of-Stake (PoS) is currently being researched that prevents a 51% attack problem, which will be discussed in Section 5.1.

4.4 Double Spending Problem

The double spending problem is when an attacker attempts to trick the network into spending the same money twice. This attack can occur as a result of a 51% attack (discussed in Section 4.3) or a transaction getting grouped in a losing block. If a transaction isn't recorded in the current longest chain, the transaction is considered not complete. For example, consider the following series of events: party A makes a transaction T sending 2 units to party B for a cup of coffee. Party A then makes another transaction T' sending the same 2 units to party C for a pastry. If each transaction is simultaneously mined by different miners in different candidate blocks (causing a soft fork described in Section 3.4), both party B and C believe that the transaction transferring them money is verified as they can see their transaction was recorded in a block (not necessarily a block on the longest chain). Eventually, one blockchain will overcome the length of the other, which will cause the miner of the shorter blockchain to accept all transactions on the longest chain. As a result, the losing miner would then need to rebroadcast all transactions not included in the winning block to be re-authenticated and verified for the next block. Ultimately, either party B or party C will receive the 2 units, because only one transaction will be recorded in the longest chain. When the the transaction that was not included in the longest block is re-verified, the miner will deem the transaction invalid as the funds were already spent and discard the transaction. As a result, either party B or C will not receive their funds after already giving the coffee or pastry.

Bitcoin is able to highly guarantee a solution to this problem only if a receiver waits for 6 successively mined blocks on the longest chain after the block including their transaction is mined before accepting that the transaction is securely final. Requiring a receiver to wait 6 blocks, however, is problematic as a seller would need to wait an hour before knowing they are highly likely to receive their payment. Because this requirement isn't reasonable when purchasing an item such as coffee or a pastry, sellers must accept that some transactions will be voided as a result of a buyer double spending (knowlingly or not).

Conversely, Ethereum prevents this problem by including a nonce (different from the PoW nonce) that acts like a counter on each transaction. A new transaction from the same address increments the nonce by one. When a miner verifies a transaction, it looks in its ledger for the nonce of the most previous transaction from that respective address. If it is one less, then it knows that this transaction is next. Otherwise, it rejects the transaction because there must be a more previous transaction. Thus, miners mine transactions in exact order they were made. Because of the risk of a soft fork overcoming the longest chain (hence, the transaction not actually getting mined), receivers are advised to wait for 12 successive blocks (3.5 minutes) to be mined on the longest chain after the block containing their transaction is mined on the longest chain. 3.5 minutes is a more reasonable waiting period to ensure the transaction is completely accepted by the network.

5 Future Work

The communities of both Bitcoin and Ethereum is facing environmental and security problems. The total amount of energy consumed by the Bitcoin network, for example, is about 73 terawatt-hours (TWh) per year, which is more than the total energy consumed in Austria [4]. Thus, a new Ethereum consensus mechanism called Casper that uses Proof-of-Stake is currently being developed in an attempt to reduce the amount of energy consumed by the networks.

5.1 Proof-of-Stake

Proof-of-Stake is a consensus algorithm for permissionless blockchains that depends on a validator's economic stake in the network. Rather than using expensive compute resources to validate transactions, a set of validators take turns proposing and voting on the next block, where the weight of each of the validator's vote directly relates on the size of its stake (how much ETH or BTC that validator owns). There are two types of PoS algorithms: chain-based and Byzantine Fault Tolerant (BFT) [5].

Chain-based PoS randomly selects a validator during every block vote and grants that validator the right to create one block, which must point to the previous block of the longest chain.

BFT PoS randomly assigns validators the right to propose a block, where the agreement of the block is done through a multi-round process. Validators are then responsible for voting for some block at each round. The important difference of this method is that the consensus on each block comes within one block. This means that consensus does not depend on the length of the chain.

Moreover, since validators must deposit currency to vote, a malicious validator will be punished by losing their deposit if they vote against the majority. Therefore, by giving the majority power to validators who have the largest stake, validators will be disincentivized to act malicously, as any malicious act will reduce trust in the network. Diminished trust would then result in the overall value of the currency to decrease which would affect the validators with the largest stake the most.

5.2 Dfinity

Dfinity is building a decentralized, public cloud computing resource, where users pay to have access to compute resources. This project is an extension of the Ethereum network and it also introduces a different consensus protocol. Rather then "The code is law" mentality in Ethereum, Dfinity introduces governance by decentralized intelligence, which they call the Blockchain Nervous System.[†††]

5.3 StorJ

StorJ is a decentralized cloud file storage system. Similarly to Box, where users can store files on the cloud, StorJ allows users to loan out extra storage on their machines or pay to store files on other people's machines. Through client-side encryption and decentralized file-storage techniques, data owners can trust that only they can access their files whenever they want. [‡‡‡]

5.4 AnyLog

AnyLog is a decentralized IoT platform where data owners can capture the full value of their data. Through massive parallelism, a globally distributed P2P network, and incentive platform, users will be able to effectively store and query IoT data in a distributed, cost-effective way.[§§§]

6 Conclusion

Bitcoin and Ethereum provide not only a solution to secure decentralized banking, but also a platform to create highly secure and decentralized applications. Similarly to the internet boom in the early 2000's, the emergence of cryptocurrency and blockchain DApps is near. In time, we will see which DApps succeed.

References

[1] coinmarketcap.com, "Cryptocurrency market capital." https://coinmarketcap.com/.

[†††] More information regarding this project can be found at https://dfinity.org/.

[‡‡‡] More information regarding this project can be found at https://storj.io/.

[§§§] More information regarding this project can be found at https://anylog.co/.

[2] etherscan.io, "Ethereum average blocktime chart." https://etherscan.io/chart/blocktime.

[3] A. Miller, I. Bentov, R. Kumaresan, and P. McCorry, "Sprites: Payment channels that go faster than lightning," *CoRR abs/1702.05812*, 2017.

[4] Digiconomist, "Bitcoin Energy Consumption Index." https://digiconomist.net/bitcoin-energy-consumption.

[5] Ethereum, "What is proof of stake." https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs#what-is-proof-of-stake.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[7] A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring, and B. Bhattacharjee, "Discovering bitcoin's public topology and influential nodes," 2015.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[10] A. M. Antonopoulos, *Mastering Bitcoin: Programming the open blockchain.* O'Reilly Media, Inc., 2017.

[11] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pp. 1–10, IEEE, 2013.

[12] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," *https://github.com/ethereum/wiki/wiki/White-Paper*, 2015.