ZENON: The Network of Momentum

Lightpaper

Revision: 05/29/2018

Abstract

Zenon aims to develop an open, worldwide decentralized network using a highly scalable consensus protocol based on a mathematically verifiable cryptographic mechanism called proof-of-momentum.

This thesis outlines a novel consensus protocol and a new foundational network architecture that is linearly scalable, tolerates Byzantine adversaries[1] and is resistant to Sybil attacks[2], while offering equivalent guarantees on security and immutability like blockchain implementations.

1. Introduction

The Internet, a worldwide system of computer networks that use the Internet protocol suite to exchange information is the foundation for any distributed technology nowadays. It was started as a project by the Advanced Research Projects Agency of the U.S. government and funded by DARPA in the late 60' and was first known as the ARPANet[3].

The goal was to create a mesh network of research computers from different universities that would be operational even if a military attack or other disaster occurred, by using a design in which messages could be routed or rerouted in more than one direction.

A distributed ledger technology (DLT), also called a shared ledger is a consensus of replicated, shared, and synchronized digital data that is globally distributed and spread across multiple regions.

Four decades after the inception of the Internet, an unknown entity that goes by the pseudonym of Satoshi Nakamoto released to the public a novel DLT implementation, the blockchain, in the form of a cryptocurrency called Bitcoin built on top of the existing Internet infrastructure that allowed assets to be transferred from one party to another without any trusted third parties. In other words, a blockchain is a distributed network of computing nodes that periodically agree on a set of new transactions that are grouped into blocks, forming a cryptographic hash chain.

A definition for a currency used for Internet value exchange lies in the Bitcoin whitepaper[4], where Satoshi Nakamoto describes the need of a global network for transfer of value: "What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party".

It further explains the foundation of the peer-to-peer electronic cash system with its proof-of-work (PoW) consensus mechanism: "Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes". Also known as the Nakamoto consensus, proof-of-work is a mechanism to probabilistically elect a leader to generate a block, thus ensuring a fair participation within the network. Bitcoin is currently one of the largest decentralized systems globally, using massive amounts of computational power (reflected in the network's hash-rate[5]) from billions of CPUs.

With the success of Bitcoin, many other cryptocurrencies emerged with different architectures and consensus protocols, most notable being Ethereum that proposed smart contracts using a Turing-complete scripting language called Solidity[6] or IOTA that proposed a directed acyclic graph (better known by its abbreviation, DAG) architecture called Tangle[7] to achieve greater scalability over blockchains.

In this thesis we propose a new approach for a decentralized architecture called The Network of Momentum that can scale in a linear way as it grows and an efficient consensus protocol involving a peer-to-peer mesh of nodes that cryptographically verify Momentum Proofs.

The network will also feature an existing, Turing-complete scripting language that will enable developers to build and run zApps without any special setup; moreover, key elements of the environment for the scripting language such as versatility, portability and efficiency will be addressed in order to enable the deployment of decentralized Internet of Things applications that will allow low-resource devices to actively participate in the open network.

As the network grows exponentially, a new method of storing the global ledger between nodes will be implemented such as key properties like immutability and privacy are retained.

2. Network Consensus

One can define consensus as a general agreement that generates an irreversible state within a network. The goal of achieving consensus is to facilitate the synchronization across a public distributed network of computer nodes that can exhibit Byzantine behavior.

Typical requirements for a byzantine consensus problem, where each process has an initial value and all non-faulty processes must agree on a single value are: validity - if all the non-faulty processes have the same initial value, then the agreed upon value by all the non-faulty processes must be that same value, agreement - every non-faulty process must agree on the same single value and termination - every non-faulty process must eventually decide on a value. and integrity - every non-faulty process decides at most one value, and if it decides some value x, then x must have been proposed by some process. One of the first algorithms for solving the Byzantine agreement problem was proposed by Lamport[8].

We assume an "eventual-synchrony" network model, introduced by Dwork[9], that presents an asynchronous network where messages among honest nodes can be delayed arbitrarily, but ultimately it behaves synchronously and delivers all messages within an unknown fixed time interval. During the synchronously period of the network, nodes are guaranteed to terminate the protocol (liveness). Given the "FLP Impossibility Result", a fundamental discovery of Fischer[10] demonstrates that deterministic protocols cannot guarantee safety, liveness and fault tolerance in fully asynchronous networks.

In the following part, we will propose a protocol and specific algorithms that will be implemented to reach consensus in the Momentum Network.

In order to achieve a good positioning in the distributed ledger trilemma there is a need to address three key challenges. In the first place, the network must select statistically representative groups of validators regularly through a permission-less Sybil attack resistant algorithm using a custom hybridized proof-of-work. Secondly, it must guarantee an insignificant probability that any shard is corrupted during the system lifetime by periodically reshuffling, spawning or reorganizing shards that are sufficiently bias-resistant. Lastly, the network should handle accurately using an atomic mechanism inter shard transactions.

Also, a cryptocurrency built on top of a distributed ledger technology should take in consideration the fairness attribute that can be further classified as follows:

- fairness of access i.e. no one can stop or delay transactions to enter into the system;
- fairness of ordering, i.e. no one can manipulate the order of transactions;
- fairness of timestamping, i.e. the timestamp is not given by a single entity.

In the Network of Momentum, transactions are issued by users connected to nodes that are participating in the open network. A random sampling is made to select nodes that assign Momentum Proofs containing cryptographic hashes computed from the details of a transaction as it traverses the sharding spaces.

A node that participate in the pre-approval process of the transactions presented to the network is a sentry and it must meet several requirements, such as having a collateral that can be confiscated in case of malicious behavior (i.e. signing conflicting transactions / double spending).

Sentinels are special type of sentries that will enable inter-sharding communication channels within the Network of Momentum, participating in the global consensus by validating the integrity of the shards.

During an epoch, pillars and sentries can privately check whether they are selected to participate in the consensus protocol using an algorithm similar to Algorand[11]. The identity of the selected committee members is unknown for the rest of the nodes in the network until they achieve consensus. A cryptographic proof allows other members to formally verify and accept the identity of new committee members, also limiting the number of nodes of a malicious actor proportional with their computational power.

A round consists of two phases: a broadcasting phase to propagate the transactions into the network using well established gossip algorithms to the committee members and the consensus phase where the committee decides upon the final state of the transactions, updating at the end the involved accounts. In case the consensus is not achieved within a specific time interval, the network penalizes the committee nodes from that particular shard and the round is considered void; a new committee is formed in order to reach consensus. An epoch ensures the global consensus; at the end of the epoch, all altered accounts are also updated with the latest hash of the consensus vote, enforcing the immutability for the transactions applied to that account. The state of the shard is passed to all other shards and a consensus committee agrees on the global state of the ledger computing a cryptographic digest and broadcasting it to all network participants. A pseudo-random function is used as a randomness generator in order to shuffle the shards and to choose a new committee to achieve again global consensus.

Pillar nodes represent the foundation layer of the Momentum's network. By randomly interconnecting with each other, they generate sharding spaces throughout each epoch. The number of shards of the network linearly increases with the computational power within the network.

3. Network Architecture

The current cryptoeconomic space is dominated by monolithic blockchain implementations, mainly using the Nakamoto consensus to process transactions in a permission-less and trust-less environment, limiting a widespread adoption and being outperformed by centralized processors like Visa[12]. Existing approaches are bounded to a limited transaction throughput because all consensus member nodes must redundantly process all outgoing transactions and storing all states (including balances of accounts or different payloads such as smart contracts code), slowing the entire system down as more activity gradually increases coordination costs.

The Network of Momentum is the foundation technology underlying the Zenon cryptocurrency, a decentralized cyberspace sharing a global distributed cryptographic ledger that is replicated across participating computer nodes. The network is designed to scale-out by expanding current processing constraints linearly with the number of nodes without compromising decentralization while preserving long term security properties under a permission-less operation.

The security and immutability guarantees provided by the Momentum architecture are enforced by using a bias-resistant mechanism based on Verifiable Random Functions[13] (VRF) for selecting nodes to construct momentum proofs and validate transactions and an efficient atomic inter-shard communication protocol for transactions traversing the multidimensional sharding space.

In order to create a robust transaction ledger, one should consider focusing on persistence and liveness properties where the finality of honestly generated transactions is achieved in the shortest t time intervals.

Thinking outside the "block", we make the following several remarks regarding the modus operandi of transactions within a payment system.

Firstly, we assume an independent transaction model where the execution of transactions can be done in any order if they do not have a dependency between them and if they exist in different shards, providing that they will generate the same final state, in the same way addition has an associative property in mathematics. However, we do impose several restrictions (e.g. transactions must be processed in a particular interval of time, have a specific amount, etc.) or discarded otherwise. Also, we supply a priority function for any transaction to be used as a way for a total ordering resulting in a set of transactions that can be uniquely ordered by their fee (the transaction hash can be used to distinguish between transactions with the same fee).

Secondly, we consider a global set of accounts, each account being associated with a single public key and having its own tamper-proof transaction history. Like other cryptocurrencies, Zenon also relies on digital signatures to provide strong authentication guarantees and account ownership: the accounts can be altered only by signed transactions approved by the private-key holder at any given moment in time and each modification on the state of the ledger must be confirmed or rejected by the network with finality in an optimal time interval. In this context, only accounts modified by transactions need to reach consensus.

Lastly, recall that we make the Diffie-Hellman[14] assumption (i.e. computational Diffie-Hellman problem is hard) and we also rely that any malicious party is computationally bounded and that cryptographic primitives that are used in the system are considered safe by design.

Leveraging on these assumptions, Zenon aims at the following fundamental objectives with respect to scalability, security and decentralization:

- full decentralization - Zenon does not rely on any trusted third parties and has no single points of failure;
- shard health - each shard regularly and correctly handles transactions associated to it;
- secure transactions - each transaction is committed atomically or eventually canceled, intra and/or inter shard;
- scale-out - the throughput is linearly dependent to the number of validators participating in the network;
- low storage overhead - nodes do not need to store the full global ledger but only a periodically calculated reference point that encapsulates a shard's state;
- low latency - Zenon facilitates fast confirmations times.

Because running nodes will consume resources such as electricity, bandwidth, storage and computing power in order to validate transactions, the network will feature an internal cryptoeconomic incentive layer that will describe how the system is architected so that different stakeholders and users are motivated based on economic incentives to ensure the functionality and sustainability of the network.

Zenon will use Ed25519 [15], an implementation of Schnorr signature algorithm for the process of address generation. Ed25519 is faster than ECDSA[16] and has a smaller size, thus making it an ideal signature algorithm for a cryptocurrency.

A quad-core Intel Xeon E5620 2.4G CPU can validate ~100k of Ed25519 key-pairs per second.

The Steps for generating an address:

a. An Ed25519 key pair is generated;

b. A SHA3-256 hash is generated from step a;

c. The RIPEMD-160 of step b is calculated;

d. The Base58 string is computed from step c and outputs the final unique address.

Furthermore, the Momentum Network implements performance optimizations to intra-shard transactions by parallelizing processing, local ledger pruning and an almost "instant-confirmation", low-latency.

4. Conclusion

We present Zenon, a cryptocurrency based on a novel and secure sharding protocol for a permission-less distributed public digital ledger. At its core, the protocol is different from blockchain implementations in a way that it doesn't gather transactions in batches and it enables a non-probabilistic finality on transactions.

The network will be deployed in a two-step process detailed below. During the first phase, known as the "Pillar Formation" phase, an initial set of nodes called Pillars will be created to sustain an incipient version of the network. In between the two phases, a testnet will be deployed and Pillar nodes will have exclusive participation access. The second phase of the process, called "Network Genesis" is the starting point of the Momentum Network that involves the deployment of the Pillars to the new network.

In this lightpaper we do not emphasize on an academic style writing and mathematical demonstrations, but rather a more general approach and conceptual understanding of the core attributes of the proposed cryptocurrency and its underlying consensus protocol. For a more formal approach one can review the whitepaper with all the technical detail implementations and mathematical demonstrations that will be available to the public after academic review distributed through a set of official academic channels.

References

[1] Byzantine Agreement with a Rational Adversary, https://homepages.inf.ed.ac.uk/vzikas/pubs/GKTZ12.pdf
[2] SRJE: Decentralized Authentication Scheme against Sybil Attacks, https://ieeexplore.ieee.org/document/5350024/
[3] Advanced Research Projects Agency Network, https://en.wikipedia.org/wiki/ARPANET
[4] Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf
[5] Bitcoin hash-rate, https://blockchain.info/charts/hash-rate
[6] Solidity, https://github.com/ethereum/solidity
[7] The Tangle Revision 1.3, http://untangled.world/iota-whitepaper-tangle/
[8] The Byzantine Generals Problem, https://people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf
[9] Consensus in the Presence of Partial Synchrony, https://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf
[10] Impossibility of Distributed Consensus with One Faulty Process, https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf
[11] ALGORAND, https://arxiv.org/pdf/1607.01341.pdf
[12] Visa TPS, https://usa.visa.com/run-your-business/small-business-tools/retail.html
[13] Verifiable random function, https://en.wikipedia.org/wiki/Verifiable_random_function
[14] Diffie-Hellman problem, https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_problem
[15] Ed25519: high-speed high-security signatures, https://ed25519.cr.yp.to/
[16] Elliptic Curve Digital Signature Algorithm, https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm