2018 IEEE 38th International Conference on Distributed Computing Systems

# Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph

Federico Matteo Benčić and Ivana Podnar Žarko
University of Zagreb, Faculty of Electrical Engineering and Computing
Email: federico-matteo.bencic@fer.hr, ivana.podnar@fer.hr

***Abstract*—In addition to blockchain, a new paradigm is gaining momentum in the filed of distributed ledger technology—directed acyclic graphs. This paper compares the two paradigms focusing on features relevant to distributed systems using the following representative implementations: Bitcoin, Ethereum and Nano. We examine the applied data structures for maintaining the ledger, consensus mechanisms, transaction confirmation confidence, ledger size, and scalability.**

## I. Introduction

Distributed ledger technology (DLT) enables the maintenance of a global and append only data structure by a set of mutually untrusted participants in a distributed environment [1]. The most notable features of distributed ledgers (DLs) are immutability, resistance to censorship, decentralized maintenance, and elimination of the need for a trusted third party. This paper compares blockchains and distributed acyclic graphs (DAG) by focusing on features relevant to their distributed design, and explains how they deal with the known issues DLs are facing. We examine the applied data structures for ledger maintenance, consensus mechanisms, transaction confirmation confidence, as well as ledger size and scalability issues. A comparative qualitative analysis is presented using three reference implementations: Bitcoin [2] and Ethereum [3] serve as reference implementations for blockchain, while Nano [4] is used to represent DAG. They are chosen as representative solutions because of a relatively mature implementation with a notable developer community.

## II. Ledger Data Structures

Both DAG and blockchain store *transactions* in an open ledger maintaining its *state*. Transactions serve as inputs that cause the change to the state. However, the two approaches use distinct data structures for maintaining the ledger.

Blockchain consists of ordered units called *blocks*. Blocks contain *headers* and *transactions*. Each block header, amongst other metadata, contains a reference to its predecessor in the form of the predecessor's hash. An initial state is hard-coded in the first block called the *genesis block*. Unlike other blocks, the genesis block has no predecessor. Transactions in Bitcoin and Ethereum are hashed in Merkle Trees.

A DAG structure stores transactions in *nodes*, where each node holds a single transaction. A DAG holds a *genesis transaction* defining an initial state. In Nano, every account is linked to its own account-chain in a structure called the *block-lattice* storing the account's transaction/balance history. Each account is granted an *account chain*, which can be seen as a dedicated blockchain associated to a single account. Nodes are appended to an account-chain, each node representing a single transaction on the account chain. Nano uses two transactions to fully execute a transfer of value. A sender generates a *send* transaction, while a receiver generates a matching *receive* transaction. When a send transaction is issued, funds are deducted from the balance of the sender's account, and are pending in the network awaiting for the recipient to generate a corresponding receive transaction. While in this state, transactions are deemed *unsettled*. When the receive transaction is generated, the transaction is *settled*. The downside of this approach is that a node has to be online in order to receive a transaction.

## III. Consensus

For an entry to be appended to the ledger, *consensus* about the entry needs to be reached in the network. The assumption is that a supermajority of nodes are honest and reliable. Algorithms for achieving consensus with arbitrary faults generally require voting among a known set of participants. A popular method, often referred to as the *Nakamoto consensus*, elects a leader by some form of a lottery [5]. The leader then proposes an entry to be added to the ledger. The entries are checked for validity by all other nodes. Both Bitcoin and Ethereum use a lottery with a cryptographic puzzle in their *Proof of Work* (PoW) algorithms. An elected leader broadcasts the new entry to all other nodes. The function is difficult to solve intentionally since to manipulate the ledger, an attacker would need to have the supermajority of the computing power in the network, which makes an attack expensive to perform.

Since PoW is stochastic, it impossible to know which node will be elected as a leader. Furthermore, even though an entry has been added to the ledger, there is no guarantee it remains a valid entry. This phenomena is called a *soft fork*. A soft fork occurs when two different blocks claim the same predecessor. As nodes continue to build the chain on top of their received blocks, *two chains* are created. The problem is resolved when a block is mined that makes one chain longer than the other. The longer chain is adopted, while the shorter one is discarded or *orphaned*. As the chain increases in size, the probability that a block is discarded because of a soft fork decreases. Each system suggests the number of blocks that need to be appended above the block before it is safe to say that it will remain in the chain with great certainty.

2575-8411/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDCS.2018.00171

IEEE computer society

In Nano, there is no need for a leader election since users are obligated to order their own transactions. PoW is used as a spam protection measure to prevent over-generation of transactions by a malicious user. However, a different method for conflict resolution has been introduced, a system of *representatives*. When an account is created, it must choose a representative that can be changed over time. Representatives vote in order to resolve conflicts. Their votes are *weighted*: a representative's weight is calculated as the sum of all balances for accounts that chose this representative. In the case of a conflict, the wining transaction is the one that gained the most votes with regards to the voters weight [4].

Note that even though a transaction may be deemed settled, it is only confirmed when it receives a majority vote for the send and receive transactions. Beside voting on conflicts, representatives vote automatically on blocks they have not seen before. A representative that sees a new transaction forwards it with a vote-signature attached if the transaction is valid. This means that the network automatically broadcasts consensus information, while the transaction is making its way through the network.

## IV. Ledger size

Bitcoin clients offer a pruning mode, allowing users to delete raw block data after the entire ledger has been downloaded and validated, keeping only a small subset of the data. The advantage of the method is that disk space is saved. The downside is that other nodes are no longer able to download the entire history of a pruned node.

Ethereum offers a state delta pruning mechanism. Ethereum keeps track of the deltas in the global state maintained by a Merkle state tree. A delta in a global state is the difference between two states of the ledger. However, if one is not interested in past states, the deltas can be discarded without harming the chain integrity. A *fast sync* algorithm has been implemented to tackle this issue. The result of the mechanism is a database pruned of the state deltas, reducing ledger size.

Nano distinguishes between three types of nodes: *historical* which keep record of all transactions, *current* which keep only the head of account-chains, and *light* that do not hold any ledger data and only observe or create new transactions (in the current implementation, all nodes are historical nodes). To reduce the ledger's size, Nano plans to implement *pruning*. Since the accounts keep record of account balances instead of unspent transaction inputs, all other historical data can be discarded to decrease the ledger size.

## V. Scalability

In blockchain, the transaction rate is limited by the periodicity at which blocks are created and the block size. In Bitcoin, a block is mined roughly every 10 minutes with a maximum block size of 1 MB (Bitcoin transaction rate is between 3 and 7 transactions per second, depending on the size of individual transactions). In Ethereum, a block is mined roughly every 15 seconds with a dynamic block size limited by the maximum amount of *gas*, a unit used to measure the fees required for a particular computation [3]. This enables Ethereum's transaction rate to be roughly between 7 to 15 transactions per second. The transaction rate is rather low compared to established payment solutions: e.g. *Visa* can process up to $56,000$ transactions per second.

Potential approaches to increase scalability are the following: to increase the block size, which increases the maximum amount of transactions that fit into a block; to create off chain *channels* through which users can bypass the network transaction rate cap; to create nested blockchain structures in the form of sidechains (e.g. Plasma) or to split the network in partitions (i.e. sharding). A more detailed analysis is available in [6].

Opposed to blockchain technology where dedicated validators must exist in order to generate and order blocks, a user in Nano must sort his/her own transactions. This approach vastly differs from the way transactions are executed on blockchain systems: transaction ordering is done asynchronously by the account owner being in charge of the ordering. The consequence is that there is no inherent cap in the transaction throughput in the protocol itself. However, peak throughput on a test reached on the main network was $306$ Transactions Per Second (TPS) with an average of $105.75$ TPS [7]. The limit is currently determined by the quality of consumer grade hardware and network conditions.

## VI. Conclusion

To conclude, DAG based ledgers store transactions as edges in a directed acyclic graph while blockchains bundle transactions in blocks. Blockchain technology determines the global truth by choosing the longest branch. Transaction ordering in blockchains is generally done by some sort of a stochastic leader election algorithm (e.g. PoW). Nano's DAG abandons leader election and delegates transaction ordering to users and their representatives to resolve conflicts. A transaction has been confirmed in blockchain when a number of blocks is appended above a referent block. In Nano's DAG, a transaction is confirmed when there is a majority of votes cast in favor of a transaction by the representatives. Decreasing ledger size is achieved by federating past transactions to historical nodes. A number of approaches to scale DLTs are being investigated, however they are yet to be proven in a production environment.

## References

[1] B. Y. A. Narayanan, J. Clark, and I. F. Y. O. U. Have, "Bitcoin ' s Academic Pedigree," *Communications of the Acm*, 2017.

[2] P. Franco, "The Blockchain," *Understanding Bitcoin*, pp. 95–122, 2014. [Online]. Available: http://dx.doi.org/10.1002/9781119019138.ch7

[3] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, pp. 1–32, 2014.

[4] C. Lemahieu, "RaiBlocks : A Feeless Distributed Cryptocurrency Network," pp. 1–8, 2008.

[5] "Sawtooth v0.8.13 documentation." [Online]. Available: https://sawtooth.hyperledger.org/docs

[6] F. M. Benčić and I. P. Žarko, "Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph," 2018. [Online]. Available: http://arxiv.org/abs/1804.10013

[7] "Stress Testing The RaiBlocks Network: Part II – Brian Pugh – Medium." [Online]. Available: https://medium.com/@bnp117/stress-testing-the-raiblocks-network-part-ii-def83653b21f