# Hashgraph the Future of Decentralized Technology and the End of Blockchain

**Ledina Hoxha**

European University of Tirana

## Abstract

Hashgraph is data structure and consensus algorithm that is fast, with a very high throughput and low consensus latency, secure because of the asynchronous byzantine fault tolerant and fair due to the fairness of access, ordering, and timestamps. These properties enable new decentralized applications such as a stock market, improved collaborative applications, games, and auctions. Hashgraph is a new consensus protocol that has garnered attention lately by being projected as a technology that will make blockchains obsolete. Hashgraph currently scales only in the number of transactions processed but does not scale with the number of nodes in the network. Hashgraph will face the same issues that other public blockchains are facing today and may not be able to maintain its security and performance. In fact, scalability is still an open problem for public blockchains.

**Keywords:** Byzantine fault tolerance, latency, fair, fairness, hashgraph, blockchain, Swirlds

## 1. Introduction

The hashgraph is a distributed data structure that maintains a growing list of data among a large amount of network peers. It was first proposed in a white paper by Leemon Baird titled "The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance". Hashgraph is a new consensus alternative to the blockchain. It uses a gossip protocol that works in the following manner: Every node in Hashgraph can spread signed information (called events) on newly-created transactions and transactions received from others, to its randomly chosen neighbors. These neighbors will aggregate received events with information received from other nodes into a new event, and then send it on to other randomly chosen neighbors. This process continues until all the nodes are aware of the information created or received at the beginning. Due to the rapid convergence property of the gossip protocol, every piece of new information can reach each node in the network in a fast manner. The history of the gossip protocol can be illustrated by a directed graph, i.e., each node maintains a graph representing sequences of forwarders/witnesses for each transaction. In the ideal case, all the nodes have the same view of all transactions and their witnesses. Further, by performing virtual voting, each node can determine if a transaction is valid based on whether it has over two-thirds of nodes in the network as witnesses. Note that Hashgraph runs in the Byzantine setting, where the assumption is that less than a third of nodes are Byzantine (nodes that can behave badly by forging, delaying, replaying and dropping incoming/outgoing messages). Hashgraph is a new approach that greatly differs to other interpretations of the distributed consensus and looks to provide an upgrade to the current systems of distributed ledger technology (DLT). Hashgraph can resolve today's scaling and security issues, while also pushing the use of distributed consensus applications into new areas.Essentially, Hashgraph is a data structure and consensus algorithm that is faster, fairer, and more secure than blockchain. It uses two special techniques in order to outperform the blockchain. These include: Gossip about Gossip and Virtual Voting.

Gossip about Gossip involves attaching a small additional amount of information to a pair of hashes (Gossip) that contain the last two people talked to. By doing this, a Hashgraph can be built and updated whenever additional information is gossiped, on each node. When the Hashgraph is ready, we are also aware of the information that each node has and exactly when they knew it. As a result, it becomes straightforward to know how a node would vote and this data can be used as an input to the voting algorithm and to find whichever transactions have reached consensus quickly

## 2. Hashgraph Consensus

Hashgraph consensus is an algorithm that operates on a list of events contained within a hashgraph data structure and maintains consensus over the correct chronological order of the events among an arbitrary number of participating nodes.

Use cases can be found in any distributed system that benefits from maintaining a consensus among the participants over the global state of the system and the order of state changes applied to it. Good examples are log replication, distributed ledgers, shared state machines, etc. The algorithm is based on the following core concepts that together provide the properties listed further below:

- Famous Witnesses
- Strongly Seeing
- Gossip about gossip
- Virtual voting

## 3. Has Blockchain been surpassed by Hashgraph?

Blockchain technology operates as reliable digital ledger that can be used to record financial transactions, ownership, and almost everything of value. Any information held on a blockchain is shared across its existing network and is consistently updated and also incorruptible. This system ensures that data is not stored in any individual location, and that the blockchain cannot be controlled by any single entity. Hashgraph is a consensus algorithm which has many benefits over Blockchain. It is a new distributed ledger technology which is around fifty thousand times faster than Blockchain, more efficient, more cost-effective, safer and mathematically fairer. Importantly, it does not require significant computation and energy consumption, so it is efficient and sustainable. Blockchain was born out of the 2008 financial collapse and ever since it has been at the forefront of distributed, peer-to-peer transactions. But currently it is limited to 7 transactions per second and a miner can choose the order for which transactions occur in a block, which can delay orders by placing them in future blocks, or even prevent them from entering the system. However, the Hashgraph consensus time stamping prevents an individual from affecting the consensus order of transactions. Hashgraph consensus uses a gossip protocol. This means that a member such as Alice will choose another member at random, such as Bob, and then Alice will tell Bob all of the information she knows so far. Alice then repeats with a different random member. Bob repeatedly does the same, and all other members do the same. In this way, if a single member becomes aware of new information, it will spread exponentially fast through the community until every member is aware of it.

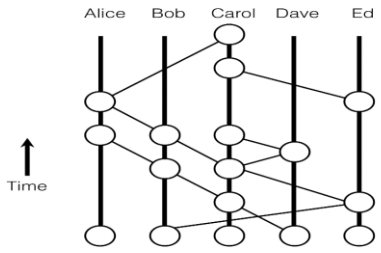

Figure.

Security is also a key issue. With Blockchain there is only an increased possibility of consensus, and when a consensus is not reached forks occurs such as the recent Bitcoin Gold and upcoming Segwit2x forks. With Hashgraph's Byzantine security, a consensus will be reached every time. Also, in hashgraph, every container is used and none are discarded, so this eliminates the problem of stale blocks forming. The end result, Hashgraph is like Blockchain on steroids. It will allow faster, cheaper, safe and more efficient trading of cryptocurrency. If Blockchain had the traditional heavyweights of the financial world quaking in their boots, then Hashgraph will make them a gibbering mess on the floor!

## 4. Comparison with Blockchain

Similar to systems based on blockchain technology, hashgraph provides desired properties that makes common issues like double spending impossible.

Table 1.

| | Bitcoin Blockchain | Hashgraph |
|---|---|---|
| **Fair** | No | Yes |
| **Low Computation** | No | Yes |
| **Byzantine fault tolerant** | No | Yes |
| **Distributed Trust** | Yes | Yes |
| **Solves double spending** | Yes | Yes |
| **Mining required** | Yes | No |

Let's analyze the three claims put forward by Hashgraph.

1) Fast. Hashgraph is fast, as it uses the gossip protocol to spread messages to the network and also performs some optimization of the gossiped messages to reduce the communication overhead. The gossip-about-gossip also yields a consensus protocol.However, there is another reason why Hashgraph is fast: it currently works in a permissioned setting. "At this time, Hashgraph is only deployed in private, permissioned-based networks"—— Hashgraph Team on Telegram. Below, we discuss the difference between permissioned and non-permissioned networks. The difference is crucial as it has a direct impact on the throughput of the consensus solution. In a non-permissioned setting (aka public blockchain) like in Bitcoin/Ethereum, the nodes participating in the consensus protocol are not known beforehand and untrusted since any node is allowed to join or leave the network at will. Moreover, the consensus mechanisms for such a setup have to account for maliciousness, particularly Sybil attacks where a single user generates multiple entities to influence the consensus process and for instance, mounts double spend attacks. Solving these issues in a non-permissioned setting affects the overall throughput.

On the other hand, in private (permissioned) distributed ledgers, identities of all nodes are known beforehand and the network is not open to an arbitrary participant. The prior knowledge of the identities of the participating nodes provides a natural protection against Sybil attacks and makes it easier to reach consensus. This means that no Sybil resistance mechanism needs to be put in place and hence the throughput can be increased dramatically (when compared to public blockchains). As Hashgraph is currently a private distributed ledger, its throughput should be compared with the likes of other private blockchains, e.g., IBM HyperLedger Fabric (700 transactions per second) or Red Belly (400,000 transactions per second). Its throughput should not be compared with public blockchains like Bitcoin or Ethereum (10 transactions per second) as it is equivalent to comparing apples with oranges. Currently, Hashgraph has yet to release concrete technical details for its deployment as a public ledger.

2) Fair. Hashgraph also provides fairness via consensus time stamping. This means that if one transaction reaches two-thirds of the network ahead of other transactions, it is considered to be the first. It is a relatively fair system, as two-thirds of the network are witnesses and it is difficult for a majority of them to make unfair decisions. However, Hashgraph is based on the gossip protocol and this means that when a node chooses its successors uniformly at random, there is some probability (e.g., one-third, if the node's neighbors are chosen globally and uniformly at random) that all the chosen nodes may be Byzantine or malicious. These malicious successors can stop passing the transaction to the next group of nodes, thereby preventing the transaction from reaching two-thirds of the network which would result in an unfair outcome for the honest creator. In addition, trying to ensure that every honest node is connected to some honest nodes, and that every message can be transmitted to other honest nodes without being stopped by intermediate Byzantine nodes, is an open problem in itself. While this is currently not an issue given the private nature of Hashgraph, this will be a hurdle to overcome when releasing a public distributed ledger.

3) Secure. Hashgraph is an asynchronous BFT, but it is not deterministic. In Fischer et al. (1985) it was shown that in asynchronous systems, deterministic consensus protocols are impossible even in the simple case of only one faulty node. A consensus protocol can be either non-deterministic asynchronous or deterministic non-completely asynchronous in the Byzantine setting. For deterministic protocols, all honest nodes reach consensus by round r for some a priori known constant r. For non-deterministic or probabilistic protocols, the probability that an honest node is undecided after r rounds approaches zero as r approaches infinity. For synchronous protocols, roughly speaking, messages are guaranteed to be delivered after a certain bound $\Delta$, but the asynchronous protocols don't have such a bound.

Hashgraph is a non-deterministic asynchronous protocol by adding randomness. The compromise is that the consensus protocol will terminate eventually but there is uncertainty as to when termination will occur. In its current design, Hashgraph employs coin toss (i.e., middle bit of a signature) for nodes to make decisions when there is no progress in the consensus protocol. Therefore, there is non-zero probability of all honest nodes having the same value after numerous rounds of coin toss. Eventually all the honest nodes will become unanimous. However, if all the Byzantine nodes try to disrupt the protocol by manipulating the gossip protocol as detailed in point 2 above, the effectiveness and efficiency of this coin toss approach becomes questionable, as it may take numerous rounds to reach consensus.

## 5. Conclusions

Hashgraph is an interesting consensus protocol that has been shown to yield high throughput in a private and static setting. Hashgraph is fast, fair, and secure within the permissioned setting it currently operates in. However, if and when used in a public setting, Hashgraph will face the same issues that other public blockchains are facing today and may not be able to maintain its security and performance. In fact, scalability is still an open problem for public blockchains. It is interesting to see new solutions being proposed by the community. For instance, Ethereum uses PoS for their Casper protocol, NEO employs dBFT, EOS has a dPoS-based solution, and Zilliqa implements sharding. All these solutions have their own benefits and weaknesses as there is no silver bullet to solve the scalability problem and there has never been one for many scientific problems. Another important question is what does a scalable solution actually mean? Does this mean that the solution is scalable in the number of users or in the number of transactions, or in the size of the network? If a P2P network is capable of processing thousands of transactions, can we call the solution scalable? If so, what happens when the network doubles its size—— can the throughput be maintained? In fact, a solution that is scalable in a single dimension may not be well-suited for a use case that requires scaling in a different dimension. Hashgraph currently scales only in the number of transactions processed but does not scale with the number of nodes in the network. Zilliqa for instance scales with the number of nodes in the network. So, let us all appreciate the underlying technology in Hashgraph but not with closed or blinded eyes. And while we appreciate Hashgraph, we should also appreciate blockchains which have paved way to the likes of Hashgraph in just the same way, they have paved the way for Ethereum, EoS, NEO, Zilliqa and many others. Hashgraph is proven to be fully asynchronous Byzantine. This means it makes no assumptions about how fast messages are passed over the internet and this makes it resilient against DDoS attacks, botnets, and firewalls.

The Hashgraph algorithm works without needing to use the Proof of Work or Leader systems, and can also deliver low-costs and high performance levels without a single point of failure. Hashgraph does away with the need for extensive computation and energy consumption and improves on the performance statistics of the Bitcoin network. Bitcoin operates at a maximum of 7 transactions per second. While Hashgraph is only limited in relation to bandwidth and allows for over 250,000 transactions per second.

In addition to this, Hashgraph also allows more a fairer system of operations as currently, miners can choose the order for which transactions occur in a block, and can even delay orders by moving them into future blocks, or even stop them from entering the system if necessary. Hashgraph utilizes consensus time stamping and prevents any individual from changing the consensus order of transactions by denying the ability to manipulate the order of transactions.

Despite its obvious benefits, Hashgraph has some way to go before it can boast of the network effects enjoyed by both Bitcoin and Ethereum. However, in the ever evolving world of blockchain technology it seems we are on the cusp of the next stage of evolution.

## References

[1] https://steemit.com/bitcoin/@tommytwohats/has-bitcoin-been-surpassed-by-hashgraph
[2] https://coincodex.com/article/1151/hashgraph-vs-blockchain-is-the-end-of-bitcoin-and-ethereum-near/
[3] https://hashgraph.com/
[4] https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Hashgraph.html
[5] https://hackernoon.com/demystifying-hashgraph-benefits-and-challenges-d605e5c0cee5