The history of consensus

Of late, in the field of computer science, the term ‘consensus’ has been mostly associated in the context of blockchain, and it is believed that the consensus algorithm was introduced because of blockchain. However, contrary to conventional consensus, consensus in computer science has existed for near three decades. The concept of consensus in distributed systems has been in discussion since the 1980s – much prior to the existence of blockchain. 1

Today, consensus algorithms continue to be an active area of research, and new algorithms are frequently designed and developed. There are different types of consensus algorithms, and each is suited for different implementations of blockchains. Here are a few well-known algorithms:

- Proof-Of-Work (PoW) – Bitcoin, solving mathematical puzzles, high-energy consumption
- Proof-Of-Stake (PoS) – networks participators put in a stake to be part consensus
- Byzantine Fault Tolerant (BFT) – Protection against bad actors (e.g., Istanbul)
- CDP Fault Tolerant (CDT) – Protection against bad actors (e.g., Cetus)
- Byzantine-Resilient Fault Tolerant (BRFT) – A variant of BFT above (e.g., HotStuff)

The current Ethereum uses BFT-based RAFT consensus algorithm, JPM Coin uses BFT-based Istanbul, and Metax Labs used a BFT-based algorithm called HotStuff.

Defining consensus algorithms

What does it really mean to have a consensus in a decentralized world? Why do we have so many consensus algorithms? Why can we not reach a consensus on consensus algorithms?

An intuitive definition would suggest that two or more individuals agree on a specific item, e.g., if they agree on the shape of an object to be a rectangle, they have reached consensus. As the object changes its shape, so does their square or circle; humans will use their senses to identify and agree on the new shape. However, in the world of computers, machines rely on the consensus algorithms to determine the changes in the state and a set of rules that finalize by applying those changes.

Those rules in the blockchain world are known as smart contracts – a set of rules or logic that, when executed, will agree on the state of objects. The concept of consensus in blockchain networks from the current state to a new state. Since the rules (smart contracts) are clearly defined and well known, one would think it should be easy for computers to reach a consensus following those rules. However, in reality, it is extremely difficult to achieve consensus because computers do not have a common sense. In a blockchain network, computers do not communicate with each other, and they have no way of knowing what other computers are doing. In order to achieve consensus, consensus algorithms are designed, and each is suitable for different implementations of blockchains.

Applicability for Link

How does this apply to our current work in Liink? For that, let’s consider inquiries in Liink. As each transaction is confirmed by the Issuer, the transaction is committed to the chain, and Liink maintains two states – ‘Private’ and ‘Public’. This is in contrast to a general blockchain where there is only one state, and all nodes are publicly notified. This also implies that on Link, we have two consensus—one for private and another for public. A detailed implementation of private/public consensus is described in the FAQ.

Each state of the inquiry (creation, acknowledgment, response, etc.) is agreed upon by participants of that inquiry. Inquiry based on a set of rules (Inquiry Smart Contract). Further, this update is also recorded on the Liink network as a unique transaction reference number (Txn: 0x4d4, 62b, 0x277, 4937 and so forth) that can be retrieved in case of non-repudiation.

Inquiry in Link

Consensus algorithms govern the state of the system within a blockchain network changes. This someone should provide a general idea about the role of consensus algorithms in blockchains and its applicability in Link applications.

If you are interested in working with blockchain technologies, you can find current opportunities at Onyx.