Espresso HotShot: Consensus Designed for Rollups

Espresso Systems

Follow

--

Listen

Share

The [Espresso Sequencer

](https://www.espressosys.com/blog/decentralizing-rollups-announcing-the-espresso-sequencer) is a system that decentralizes transaction sequencing for layer-2 scaling solutions on Ethereum without compromising on their scale and speed.

At the core of the Espresso Sequencer is a consensus protocol that prioritizes high throughput and fast finality. We have designed this protocol to complement the trade-offs inherent to Ethereum's consensus, which prioritizes liveness under pessimistic conditions rather than responsiveness under optimistic conditions. Rollups on the Espresso Sequencer can offer users a more performant experience than rollups that directly rely on Ethereum for sequencing. We believe in empowering users to access a combination of speed, scale, and security they require. Building on the HotStuff protocol, we call the Espresso Sequencer consensus protocol HotShot.

The Espresso Sequencer is designed around a single decentralized proof-of-stake security model that underpins both a consensus protocol for ordering transactions and a data availability mechanism which allows for further performance benefits. It also encompasses a system of rollup contracts that (a) register committed blocks of sequenced transactions, verifying their consistency with the consensus protocol and availability certificates, (b) register updated state commitments for each zk-VM deployed to the Espresso Sequencer, and © receive and validate proofs for the state updates.

Our first public milestone in this effort, the [Americano

](https://www.espressosys.com/blog/releasing-espresso-testnet-1-americano) testnet, implements the first version of HotShot.

Optimistic Responsiveness

HotShot prioritizes high throughput and fast finality, complementing the dynamic availability of Ethereum's consensus (Gasper). Fast finality,

or

more formally optimistic responsiveness,

is the ability of the protocol to confirm transactions as fast as the network will allow. Confirmation can be nearly instantaneous under optimistic network conditions. This stands in contrast to protocols in which the confirmation delay is tuned to worst-case network conditions, or where transactions are only probabilistically final. Dynamic availability,

the hallmark achievement of Nakamoto's longest-chain consensus protocol, is the ability of a protocol to remain live under sporadic participation, even if most nodes at any given time are offline. Consensus protocols must choose between optimistic responsiveness and dynamic availability — these two properties are incompatible. Most practical BFT protocols to-date, including Tendermint and Casper, achieve neither property.

HotShot extends HotStuff to the decentralized "proof-of-stake" setting with large-scale dynamic participation, while retaining optimistic responsiveness.

Web2 Performance with Web3 Security

Scalability in consensus systems is measured by throughput and latency

. Throughput is best described by the bytes of data that can be finalized by the system per unit of time (e.g. per second). This is more precise than TPS as it accounts for the variability of size and complexity across transactions. Meanwhile, latency can be defined as the average time it takes for a transaction to be finalized after it's submitted. The primary scalability challenge

of consensus protocols is to achieve the highest possible throughput while maintaining decentralization and a reasonably low latency.

Consensus, or state-machine replication,

is not only a protocol for all participating nodes to agree on an ordering of transactions, but also to replicate the state (or at least a transaction log that can be replayed). While in theory these two functionalities can be separated, and the quantity and/or identity of nodes participating in each could be distinct, both are substantially large and diverse in a decentralized system. Thus, at the heart of any decentralized blockchain is a mechanism for propagating information in a resilient way among all nodes participating in the protocol.

Resilient communication protocols (e.g., peer-to-peer gossip) are one reason why decentralized blockchains achieve much lower throughput than traditional "Web2" transactional systems, particularly when there is extreme heterogeneity among nodes participating in the network. The typical "Web2" architecture utilizes a star network configuration, whereby all traffic is routed through one or more designated high-bandwidth servers. This optimizes the communication (particularly the broadcast rate) in a network where most participating nodes have much lower bandwidth than these central servers, but it is less resilient to byzantine corruption.

A primary advantage of optimistically responsive consensus protocols (e.g., HotStuff) is the ability to perform better when network conditions are favorable. Such protocols can even leverage a typical "Web2" architecture to optimistically achieve extremely high throughput, and in the worst case fall back to a high-resilience gossip-based path with lower-throughput.

In this sense, optimistically responsive protocols have the potential to achieve the best of both worlds: Web2 performance with Web3 security.

Even in our initialtestnet implementation, Hotshot already demonstrates the scalability benefits of optimistic responsiveness. As we extend HotShot, leveraging SNARKs, verifiable information dispersal (VID), and other techniques, it will be able to sustain high throughput even under pessimistic conditions when the only available communication channel is a lower-throughput gossip protocol. Read more about our first testnet, Americano, and our future plans here.