Introduction to Avail Light Clients

Introduction

Avail light client (LC) lets you interact with Avail DA without requiring a full node and without trust assumptions towards remote peers. This is accomplished by leveragingData Availability Sampling (DAS) performed by the light client on every newly created block.

The light client listens on the Avail network for finalized blocks and performs DAS on a predetermined number of cells on each new block. After successful block verification,block confidence is calculated for a number of cells in the matrix, with the number depending on the percentage of certainty the users wishes to achieve.

This screenshot is from a purely in-browser interface built by the Avail team that uses a light client to determine the confidence of blocks in real time. You can check it out atlight.avail.tools(opens in a new tab) . Light client functionality is separated into two logical parts - thelight client and theapp client . While the LC is primarily focused on DAS, the app client is used to perform data reconstruction.

Light client

Thelight client mode of operation is active regardless of whether the app client is also active or not. Light client connects to an Avail node via a WebSocket connection and waits for a newly finalized block, with the header containing its KZG commitments.

Avail blocks are chunked and divided into equal sized cells as a part of that blocks matrix. Each row in the matrix is then erasure coded usingReed-Solomon (RS) erasure codes and committed withKate-Zaverucha-Goldberg (KZG) commitments. On each received header the client does random sampling of the matrix cells, which are retrieved using one of two mechanisms:

1. DHT
2.
    - the client first tries to retrieve cells via Kademlia DHT, on the LC-only high availability peer-to-peer network.
3. RPC
4.
    - if some or all of the required cells can't be found on the DHT, LC uses RPC calls to the Avail node(s) to retrieve the data. Cells not already found in the DHT will be uploaded thus increasing blocks availability in the LC P2P network

Once the data is received, light client verifies individual cells and calculates the confidence that is then stored locally.

Light client useslibp2p withKademlia as a DHT implementation. Peer-to-peer network is able to perform NAT traversal, both symmetric and asymmetric, enabling easy connectivity with various network configurations (e.g. symmetric and asymmetric NAT). On fresh startup, the LC performs a block sync with the node, using both DHT and RPC mechanisms. The block depth to which the sync is going to be done is set with thesync_block_depth config parameter, which needs to be set to the max number of blocks the connected node is caching (if downloading via RPC).

When and how to embed the light client

The Avail light client plays a vital role in ensuring the availability and correctness of data within the Avail network. By employing random sampling, it achieves security levels comparable to full nodes. Furthermore, by leveraging the peer-to-peer network, it enhances overall data availability while reducing the load on full nodes.

The light client is capable of downloading and verifying application-specific data submitted to Avail, which can be conveniently queried using the light client API.

You can check out all the different methods supported by the Avail light client in ourLight Client API Reference . The light client exposes an HTTP API that enables users to query the status, confidence, and application data for each processed block. When a block is finalized in Avail, the light client performs random sampling and verification, calculates confidence in the given block data, and if the confidence is high, retrieves the application data from the block. This data is then verified and stored locally for easy access.

Run Avail Light Client Run a Light Client