

Harvesting

Approximately every 4.69 seconds, the beacon client sends a new signage point to the farmer, who sends it to each harvester.

The exact protocol message sent for each signage point is the following:

```
class NewSignagePointHarvester:
    challenge_hash: bytes32
    difficulty: uint64
    sub_slot_iters: uint64
    signage_point_index: uint8
    sp_hash: bytes32
```

1. The harvester receives a signage point, and computes the plot filter:

```
plot_filter_bits = sha256(plot_id + challenge_hash + sp_hash)
```

If the resulting bits start with 9 zeroes, then the plot passes the filter. This does not require disk access, since the plot_ids are stored in memory.

2. For each of the plots passing the filter, a new thread is started, which performs the quality lookups. Recall that this requires around 7-9 random reads into the plot, one for each table. This is where the majority of the disk activity will be. On average, 1 of every 512 plots will perform this step.

3. `required_iterations` is computed. If `required_iterations` is less than the `interval_iterations`, this proof of space is good (it has won a block). Most proofs will not pass this step.

5. For winning proofs, the whole proof is fetched on disk (approximately 64 random reads in the plot).

6. The proof is sent back to the farmer.

Revision #4

Created 5 June 2023 14:42:22 by Admin

Updated 30 October 2024 18:02:20 by Admin