Hi, in the wiki post about erasure codes it is claimed that when there is an “honest minority of light clients” that randomly probe a block via its Merkle tree and reject blocks that fail this test

it provides a limited availability guarantee:https://github.com/ethereum/research/wiki/A-note-on-data-availability-and-erasure-coding#erasure-codes-as-a-solution

My question is: What is the purpose of the random probing… is it to make sure that unavailable blocks are rejected, or is it to make sure that the data is stored by other entities?

If the purpose of the sampling is to make sure unavailable blocks are rejected, then how does the information about a failed data probing by a single client lead to all honest clients rejecting the unavailable block? I think it’s not possible to do this by publishing the evidence of unavailability on the root chain (since we know an attacker can then simply provide the data belatedly in arbitration, which opens up DDOS attacks of the arbitration system) Alternately, the client with the evidence of unavailability could just pass its evidence to other clients directly over the P2P network, but intuitively it seems this would just open the P2P network to similar DDOS attacks, since the attacker can then again make the data available to the other light clients after they have been kept “busy” for an amount of time checking this claim of unavailability. (but maybe my intuition around this is incorrect.)

If the purpose of the sampling is to make sure data is available at another location, what would be the mechanism a node could use to find the other light client nodes that have the data that it needs?

Thanks to anyone who can help me clarify my thinking around data availability!