Preface

Hi Everyone! Im quite new in this forum and this is my first post. After readingwelcome notes I see there is a goal set for increasing economic incentive confluence in all aspects of the Ethereum protocol.

I want to bring this up because I am looking to understand whether there is anyone or any research groups that are working in this direction and if help is needed in that direction.

Abstract

Every rational participant who is looking to participate in Ethereum protocol network will be looking to maximise his earnings while cutting down costs such as occur from expensive hardware, high energy costs, amount of data stored and network traffic consumed.

From the network perspective, however opposite qualities of node operators required - the security guarantees, privacy, censorship resistance and overall speed and accessibility imply that nodes should willing to exchange data to the maximum extends, store data indefinitely.

I invite in this topic to discuss and find out solutions that can model such system as a demand and offer market with a goal to create financial model and possibly touch some technical implementation details how such could be implemented.

Problems

1) Financial incentivise to communicate with light nodes are missing

Most of blockchain users do not run their nodes in order to connect to Ethereum. Instead most projects and individuals prefer using RPC endpoint providers, who pose a centralisation risk and can be seen as web2.0 legacy solution that just solved issue for those who were unable to launch their node.

Common scenario for light nodes attempting to speak to full nodes is that counterpart simply refuses connection;

From rational standpoint it makes sense as even reading state requires some processing to be done by the node for each connected end-user and that must be somehow compensated in order to be sustainable.

This creates problem of free endpoints being slow or unresponsive and even paid RPC endpoints will throttle connection and apply rate limits;

2) Data retention

Today Ethereum archive node state will be growing over time. AFAIK (correct me if Im wrong) in current model, even with sharding - amount of data stored in the Ethereum can be expressed as $\lim_{t\to\infty} f(t)=\infty$

which obviously will not be sustainable in long run and will make syncing node times impractical in some future

3) Geospatial centralisation

Already raised in geographical decentralisation discussion so I won't stop much on this, but obviously may be solved by if financial incentivises for Node runners are in place that allow to benefit from running in remote locations

4) Data validity

Whenever data is exchanged between peers there is a risk of data being corrupt or manipulated. If application developer today connects to RPC endpoint and reads some on-chain state - most of the time he will not bother checking block headers to ensure that state he got actually makes sense from chain integrity standpoint.

Thoughts on possible solution

Decentralized marketplace model

Nodes can offer their quotas for read/write operations as well as some latency guarantees and light-nodes set prices they are ready to pay;

In a simple manner we can imagine chain-entry point marketplace implemented as various nodes or clusters of nodes offering their services on some conditions which can vary from cluster to cluster; Same manner today the RPC providers do their business;

Clients who want to operate with the cluster must enter the contract conditions by locking asset as payment guarantee and

store any extra on-chain data that is required for particular SLA to operate if any.

Light nodes

Incentivise to run instead of wallet

Ideally, we want every user actually run his own node.

Light nodes fit in perfectly well for validating that read data is indeed correct and can resolve problem of data validity.

If we have decentralised market model for p2p data, than light nodes themselves can act as data relays and caches by re-distributing data to the peers and further optimising network traffic and can be explored on it's own as financial incentivise to run light-nodes in mass or providing access in a bottle-neck geospatial cases.

Bootstrap incentivise

One issue when booting in to p2p network is that you need to have an access to bootstrapping nodes. And since new peer is using bootstrap nodes - obviously he might be a low (zero) value to the network (has 0 Eth value).

With a marketplace model, bootstrapping node can actually require newcomer to perform some work - cache data or distribute traffic in order to gain some credit and therefore even public/free

rpc endpoints operating under such protocol can be financially incentivised.

Define slashing rules for data

If such marketplace is treated as a network and is secured by underlying assets than slashing rules

Some thoughts possible technical implementation details

Once match is found the deal can be formalised as smart contract and hence such marketplace can be secured by technologies such as EigenLayer nodes to restake their ETH in to the data marketplace and hence also provide security and data validity guarantees with clearly defined slashing rules.

All communication can be moved off-chain and on the chain onlystate-channel or some kind of rollup is needed.

Once this is done, user can bring up his light node and set his peer connections to those cluster(s) (in the best scenario for redundancy user would want to have few clusters availible).

When Full-nodes receive such light-node connection request they verify on their service contract that light node address is payment capable, and hence are financially incentivised to accept connection;

Data exchange protocol

Once connected, RPC requests between light and full node might look the similar to what exists today with some additional data fields that will be required as receipts and as proof of command execution.

Payment receiver

Requires all request contain signed receipt field. Collects receipts for all requests and can batch them together to pull payments. Processes all requests and for transactions returns a proof that transaction was submitted to mempool.

Payee

After sending a request, expected scenario is that service provider (payment receiver) will shortly respond and then client can:

- validate that chain state actually makes sense (data is not corrupt and is consistent)
- Send acknowledgement to provider

SLA

There should be pre-defined service-level agreement in form of contract.

If data is inconsistent:

Client can submit this on chain and get provider slashed; This creates strong incentivise for full-nodes to operate honestly;

Provider on his end collects acknowledgements from client and can submit them in case of any dispute as proof of his work.

If due to some reason light-client fails to acknowledge the RPC method execution:

Cluster can refuse any future connections to this node. In strictest mode light node can be dropped after one failing acknowledgement.

To incentivise light-clients to act honestly a penalty can be provisioned in the SLA contract that penalize for exiting contract without any transactions, or more generally speaking:

Cost of entering + exiting contract should be higher than the potential price of unacknoeledged RPC calls made until provider closes the connection;

Further thoughts

Im essentially looking for a decent peer-to-peer human networking to help improving protocol.

Please get in touch and leave your feedback.