

# Argument for Incentives

Based on DH7 from: <https://www.lesswrong.com/posts/FhH8m5n8qGSShsAgG/better-disagreement>

This is not

an actual argument for incentives but rather me trying to understand the argument for incentives. I attempt to make the strongest argument for incentives in a stateless network and in doing so found some useful insight which I want to share. This is mostly a loose stream of thought and has not been wordsmithed.

The Ethereum network has a lot of data.

- 40gb of account data
- 100gb of chain data

By their nature, stateless clients are leachers. Beyond simple block verification which scales fine via gossip, they cannot expose the standard client functionality without making on-demand requests for data from the network. Some nodes may be semi-stateful or retain a cache of recently seen or requested state, allowing them to serve some requests, however for the purposes of this document we can treat the majority of such semi-stateful nodes roughly the same as fully stateless ones since they would still exhibit similar needs, just on a potentially smaller scale.

For the purposes of this document we will be exploring the idea of a network dominated by stateless clients which run on lightweight hardware and retain very little state. These clients would expose the maximum possible JSON-RPC API, specifically supporting `eth_call`

which requires on demand access to arbitrary accounts and storage from the state. These clients would also participate in relaying transactions, requiring them to have on demand access to a broad range of account balances and nonces.

## Asymetry in the Network

For now, we make no assumptions about the network topology. This means we won't start with assuming continued use of the DevP2P ETH

network topology, nor do we assert any new network topology.

However, what we can observe is that there will be some number of "full" nodes which have data, and a much larger number of "stateless" nodes which need the data. The full nodes have some combined capacity  $C$

and the stateless nodes have some combined need  $N$

. In the event that  $N > C$

the network can no longer meet the demands of the stateless clients.

The situation is more complex than this, because even in the case where  $N \leq C$

the full nodes are still paying a cost to serve this data. This means that there is still a cost to full nodes even when operating below this capacity. This cost manifests in as non-trivial disk i/o and CPU overhead. For a miner, this can have a direct negative impact on their earnings.

## Natural tendency towards selfishness

In such an asymmetric network, stateless clients are consuming a limited "public" resource and the full nodes providing that resource will naturally exhibit self preserving behavior. Thus, the "tipping" point where demand exceeds capacity is likely much lower than the actual theoretical tipping point.

Other factors that play into this are basic client defaults. Currently it isn't easy

for a client to be selfish in a manner not already codified by the client developers, and the network is dominated by a single client, go-ethereum, which naturally has incentives to be sure it shares well enough with itself.

If we think ahead to a network that supports statelessness, where clients are easier to write and maintain, and there are a larger number of different clients operating on the network, it becomes more likely for one of more clients to exhibit behaviors that take advantage of the network as well as clients that facilitate more selfish behaviors by sharing less data than they could.

The natural end result is a network comprised of stateless nodes which take without giving back and full nodes which act selfishly with their data. In order to change the outcome, we have to change the behavior.

## Changing Selfish Behavior

In decentralized networks we will tend to have little direct control over how people behave. Our only areas of concrete control are in the bounds placed by the protocol rules.

Behavior changes thus come either from internal or external motivation. Internal motivations can be considered but seem unreliable since we expect the network participants to be a diverse group with different and often disparate internal motivations. This leads us to external motivators, of which there is potentially more easy common ground.

1. People are universally lazy, meaning that we can reliably expect most to take the path of least resistance.
2. People are motivated by money, meaning that people will offer a service if they are adequately compensated for it and people will pay for a service if it provides enough value.

## If full nodes are hard to run...

If a full client is hard to run and maintain then the lazy option is the one that preserves the clients ability to function. Hard to run full clients naturally result in fewer of them.

In this mode, very few will serve data if it is off by default

, and eventually, most will turn the feature off if it negatively effects their client's performance which would likely occur in the network described above.

## If full nodes are easy to run...

If a full client is easy to run and maintain... then we've solved a problem we don't know how to solve yet, and we don't need stateless clients.

## If contribution can happen beyond full nodes...

If we lower the barriers to contributing to the data network then ... why do we need incentives??? It seems that as long as you constrain contribution to the data network to full nodes, there remains a reasonable argument that incentives are necessary. However, once you open up the option for stateless nodes to make meaningful contributions back to the network, the asymetry in the network goes away, and the argument that financial incentives are necessary no longer holds.

## Failure case for the homogenous network

The "Homogenous" network fails in the same way as the asymmetric network. There is still some "capacity" and "demand" and when demand outpaces capacity the network can no longer serve requests. To examine this, we need to examine the expected behaviors of network participants.

- Full Nodes: stay online, maybe serve some data or even a lot of it.
- Stateless and Semi-Stateless: some stay online for long periods. Some may hop on and off quickly.

I assert that the ones that stay online for a long time will predominantly have more capacity to serve data than they consume. There will be actors that consistently consume more than they provide but I would expect those to be in the minority as such a node would be less performant than a full node and the use cases which would put such a node under constant demand are those that would typically have inherent incentives to pay the costs associated with running a full node (backend infrastructure).

Thus, the leachers are those that jump on the network briefly, and then exit. It seems that the very nature of this type of node is such that it's consumption would naturally be quite low. Examples are, quickly performing one-off operations like checking a balances, issuing a transaction, or reading some state. All of these are pure leaching behaviors but they all exhibit a very low usage footprint.

So for this network to fail, it would need to be dominated by leacher nodes which consume more than they provide. It seems this would involve a massive number of transient stateless nodes, or a large number of stateless nodes which have consistently high usage. Neither of these situations seems likely...