

Stateless Ethereum involves greatly increasing the block size. This is dangerous because larger blocks propagate slower, and [slower block propagation means less security](#). There are [a number of proposals](#) to decrease the witness size, but here I want to approach this problem from another direction, what if blocks propagated faster?

This post is a request for help

: I'm pretty sure this will help blocks propagate faster but I'm not sure how big the effect will be, and I hope someone else knows how I could find out!

Currently, nodes first fetch an entire block. They then check the proof of work and pass it on. This introduces a propagation delay which scales linearly with the number of hops the block makes. The alternative is to pipeline block propagation. The entire header needs to be fetched in order for the node to check the proof of work, but after that each packet of the block body and also of the witness can be verified and immediately forwarded along. A picture explains it best:

[

PipelinedWitnessRelayNoPacketLoss

1102x524 7.84 KB

](<https://ethresear.ch/uploads/default/original/2X/d/d41cfdbe0f8418f174187c1347675c9b58abd6c9.png>)

Since blocks commit to their uncles, transactions, and receipts (the block body) [using merkle tries](#), packets containing parts of those tries are easy to verify as long as they arrive in roughly the correct order. This also applies to witnesses, which are just subtrees of the previous header's state root.

This is important! The maximum block size ethereum can support [might be around 1MB](#). However, if better block propagation lets us bump that limit up to 2MB, then maybe there's no need to migrate to using a binary trie. If that were true, it would let us ship Stateless Ethereum much sooner!

My question is, how much does this increase the maximum supportable block size?

I have some ideas for tackling this question:

- The current approach gets slower as there are more hops between the miners. So, if all the miners form a fully-connected subgraph then switching to pipelined block propagation won't help at all. (although there a different approach could significantly speed up block propagation) Some sort of [network topology inference](#), maybe based on [rumor source detection](#), could tell us whether that's true.
- On the other hand, assume the network forms a [random regular graph](#), given we have ~2000 nodes and each of them are connected to up to 25 other nodes, the diameter of the network is around 6. (The diameter is still about 6, even if we have 8000 nodes). I expect that there's some kind of coordination between miners, so 6 seems like a reasonable upper bound on the number of hops between miners. If we had an estimate of the latencies and bandwidth of links in the network this might be enough to estimate how much pipelining would help?
- We could do something like [the starkware experiment](#), but instead of creating large blocks and looking at orphan rates, we could instead create large blocks and then look at how often mining pools orphan blocks from each other. This would give us a sloppy estimate of the latency and bandwidth between each mining pool? If the latency between two pools seems high, that would indicate that there are multiple hops between them?