



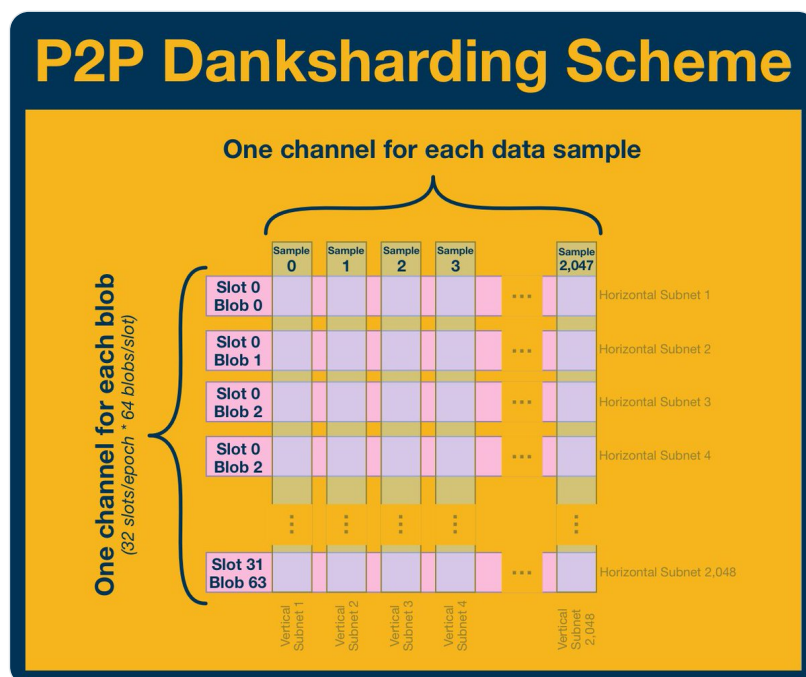
Haym @SalomonCrypto

Oct 25 · 25 tweets · [SalomonCrypto/status/1585042835459387392](https://twitter.com/SalomonCrypto/status/1585042835459387392)

(1/24) [@ethereum](#) Roadmap: Peer-to-Peer Networking

In order to achieve the vision of the World Computer, Ethereum needs more data bandwidth. But more bandwidth = higher node requirements = centralization. So what's the path forward?

Let's look into the future of the P2P network.



(2/24) [@ethereum](#) is the World Computer, a single, globally shared computing platform that exists in the space between a network of 1,000s of computers (nodes).

These nodes are real computers in the real world, communicating directly from peer to peer.

 **Haym**
@SalomonCrypto · [Follow](#)

(1/21) [@ethereum](#): The Big Picture

From 1492 to 2022, the context, technology and vision of the World Computer. The complete, top-to-bottom case for [\\$ETH](#).

An (unprecedented) mega-thread.



3:00 PM · Sep 3, 2022

 [Read the full conversation on Twitter](#)

 927  Reply  Copy link

[Read 42 replies](#)

(3/24) Today, the World Computer is... just ok. Execution is expensive and there isn't that much storage space. Yes, you can do (almost) anything, it's just going to be expensive and slow.

The solution: rollups. Independent blockchains that use [@ethereum](#) as a settlement layer.



Haym
@SalomonCrypto · Follow

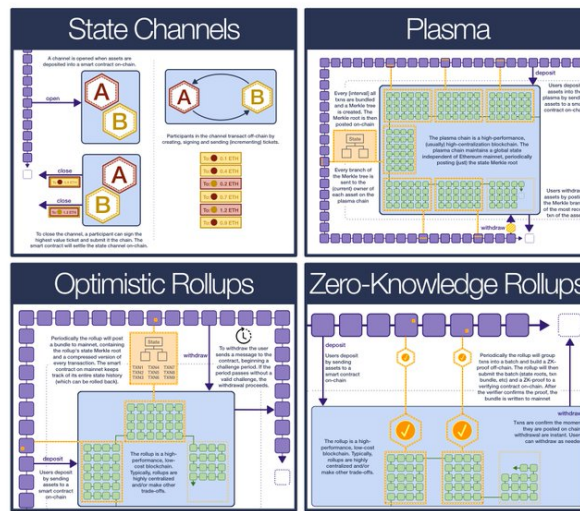


(1/15) [@ethereum](#) Scaling Technology

State Channels → Plasma → Optimistic Rollups → ZK-Rollups

Your guide to the technologies that will scale Ethereum from 12 to 100,000 txns/sec... at a lower cost than you pay today!

Ethereum Scaling



11:04 PM · Sep 12, 2022



[Read the full conversation on Twitter](#)



791



[See the latest COVID-19 information on Twitter](#)

[Read 17 replies](#)

(4/24) Rollups are fantastic; we're barely in the opening inning and we've already seen just how fast and cheap we can get execution.

Unfortunately, rollups only solve the execution problem. If anything they actually make handling data even more of a challenge.



(5/24) And so, this is the problem space we are entering: how can we scale the data capabilities of [@ethereum](#) without increasing the individual node requirements?

There really only is one option: we have to distribute the data.

(6/24) < NOTE >

Node - real computer running [@ethereum](#) software, can run many validators

Validator - a role granted by staking \$ETH, requires operating a node

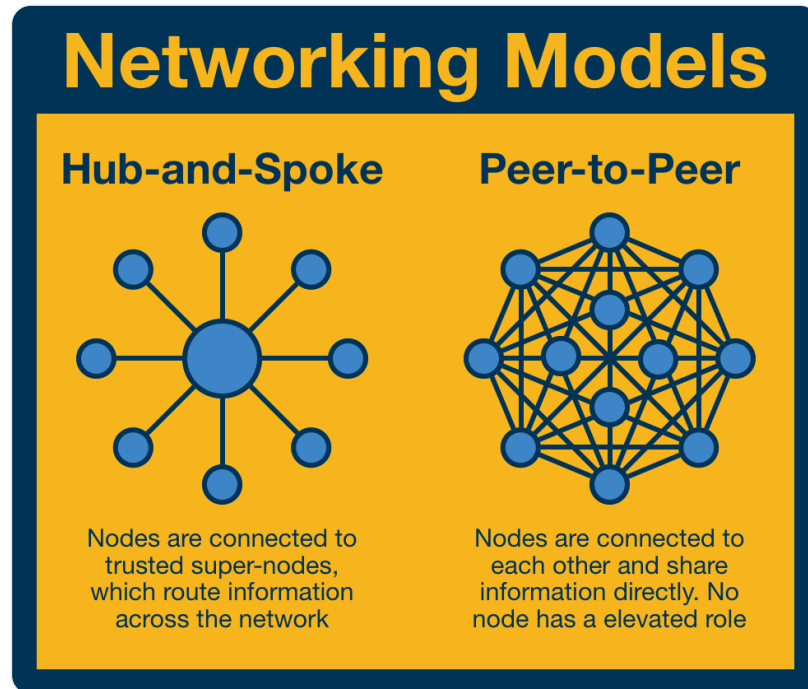
Ethereum itself doesn't distinguish; from the protocol perspective, a validator = node = IRL computer

< /NOTE >

(7/24) Let's consider two models of network: hub-and-spoke and peer-to-peer.

Hub-and-spoke = centralized (trusted) super-nodes route traffic through the network.

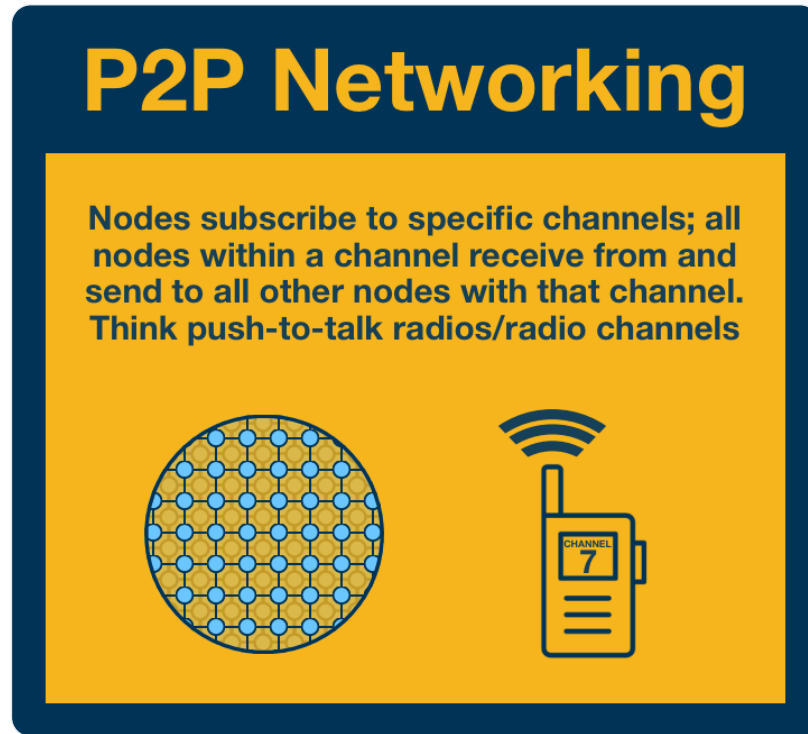
Peer-to-peer (P2P) = each node connects directly to its peer nodes. No node is more trusted than another.



(8/24) [@ethereum](#) is an example of a P2P network: each validator has the same privileges and responsibilities as the other validators and send messages directly to their peer nodes.

The problem: there are ~450k validators, all chattering back and forth. That is SO MUCH noise.

(9/24) Fortunately, we have a solution: we can implement communication channels. Remember those push-to-talk radios? Same idea, you can only send/receive messages from the channel you are currently tuned into.



(10/24) Our goal is to remain certain that 100% of data is available without forcing any one node to download 100% of the data.

If a user requests a huge amount of data, we need to be able to quickly and efficiently retrieve it, even though the entire dataset isn't in one place.

(11/24) We are going to apply two separate strategies. First, randomly sampled committees.

The validator set will be broken into committees, each responsible for ensuring data availability for only a single blob.

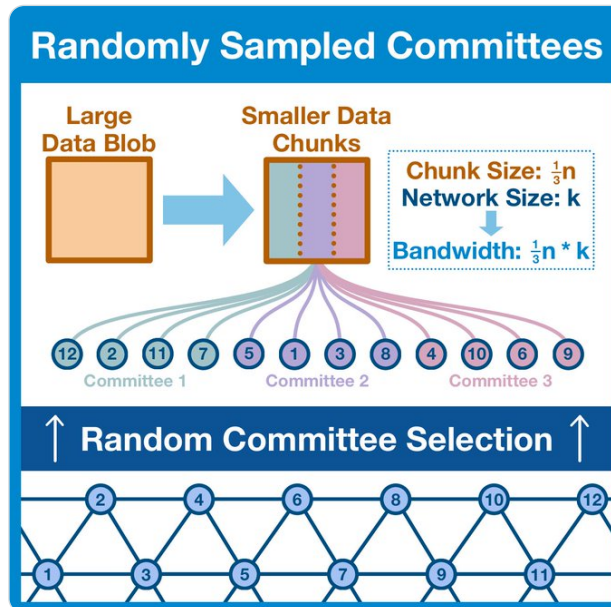


Haym
@SalomonCrypto · Follow



(1/17) The World Computer of tomorrow is a rollup-centric @ethereum. But rollups produce data, lots and lots of data. How can we scale a growing network without growing bandwidth?

We are going to need a couple solutions, let's start with Randomly Sampled Committees.



12:04 AM · Oct 24, 2022



Read the full conversation on Twitter

222 Reply Copy link

Read 8 replies

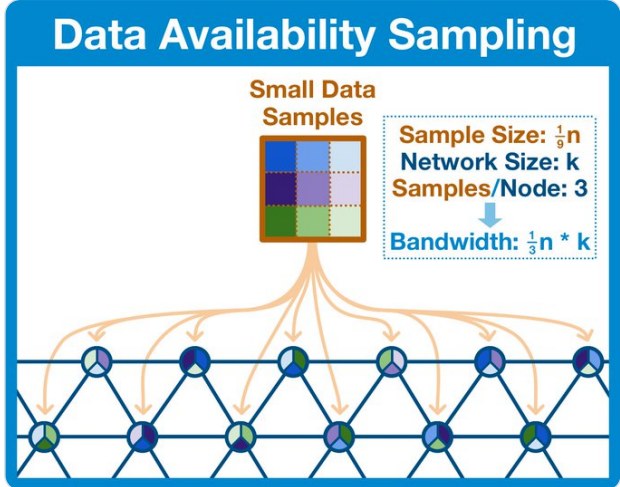
(12/24) Second, we will apply data availability sampling.

Every validator engages in data availability sampling, selecting a tiny portion of the data across every block/blob and attempting to download it.

Haym
@SalomonCrypto · Follow

(1/21) The World Computer of tomorrow is a rollup-centric @ethereum. But rollups produce data, lots and lots of data. How can we scale a growing network without growing bandwidth?

We've talked about randomly sampled committees, now it's time to talk Data Availability Sampling.



2:56 PM · Oct 24, 2022

[Read the full conversation on Twitter](#)

144 Reply Copy link

[Read 4 replies](#)

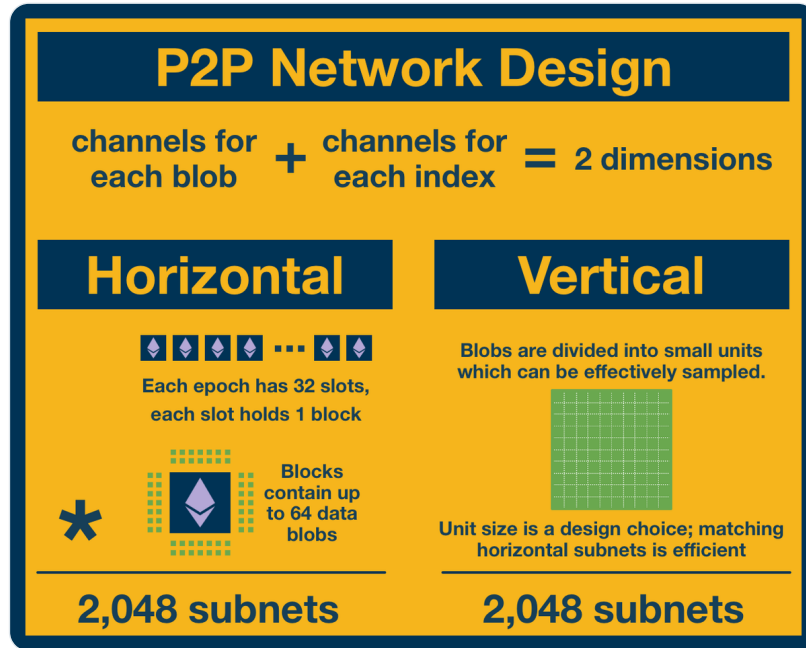
(13/24) Together, data availability sampling and randomly sampled committees give us powerful coverage:

- randomly sampled committees ensure every blob is downloaded in its entirety
- data availability sampling ensures every validator touches (a small part of) every blob

(14/24) Now we can begin to see our P2P Network design. We start with our unit of time: epochs.

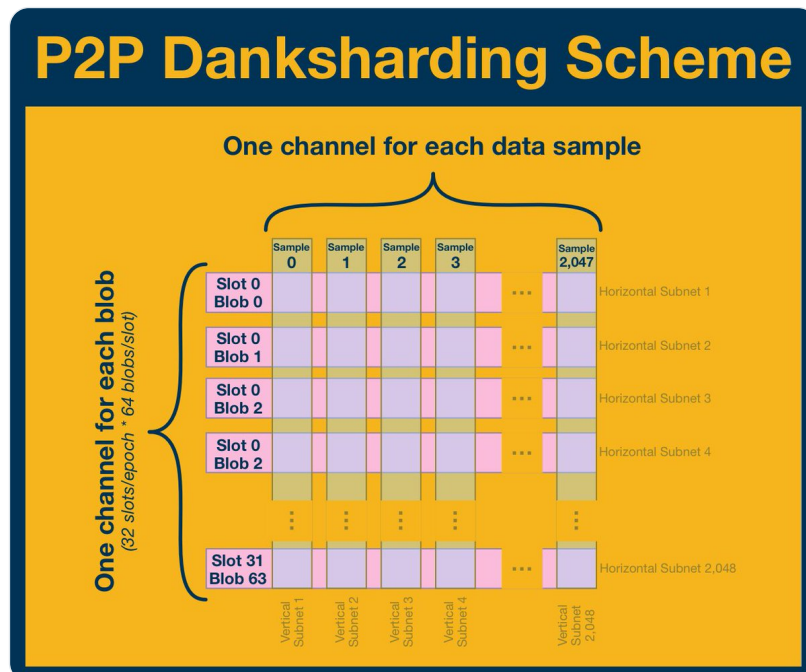
We need channels for each blob and each index, implying a two-dimensional design.

And so, we'll build out a P2P grid!



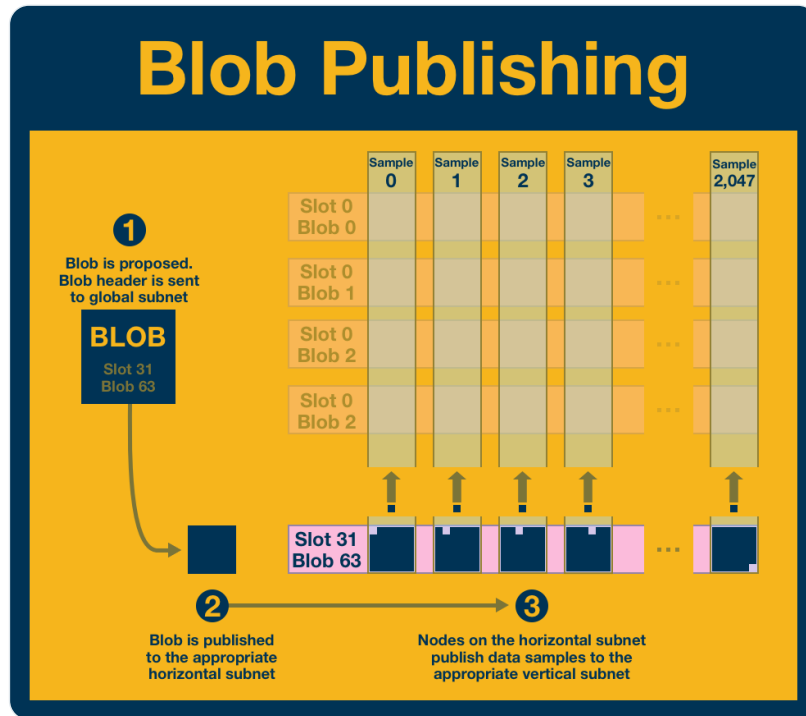
(15/24) Rows represent horizontal channels for randomly sampled committees (assigned at the beginning of each epoch).

Columns represent vertical channels for data availability sampling. Each channel is responsible for sampling ALL blobs at that specific index.



(16/24) Publishing blobs to the network is a 3 step process.

- 1) the blob proposer broadcasts the blob header to a global subnet. All nodes receive this header and use it for verification.
- 2) the proposer broadcasts the blob to all peers on the appropriate horizontal subnet.



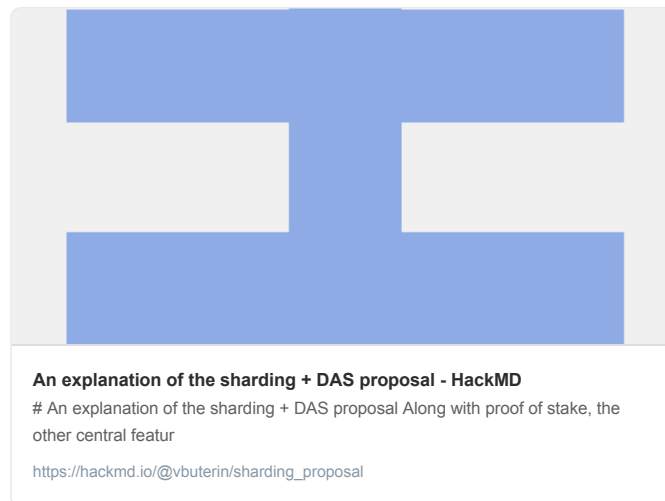
(17/24) Third and finally, as each node on the horizontal subnet receives a copy of the (full) blob, they identify the data samples corresponding to the vertical subnets they are subscribed to. They then isolate those samples and broadcast them to the appropriate vertical subnet.

(18/24) Let's add in some (realistic) numbers so you can get a sense of the idea. In his blog, Vitalik talks about:

- ~20 data samples per node/validator
- 512 byte data sample size
- 1 MB blob size

And some basics

1 epoch = 32 slots = 2,048 blobs



(19/24) Let's imagine an entire epoch has passed under this system and consider the data implications.

Each node is subscribed to a single horizontal subnet and so must download one blob. Each node is also subscribed to 20 vertical subnets and so must download 20 data samples.

(20/24) 1 blob (1 MB) + 20 samples (10 kB) = 1.1 MB of required data download/processing per epoch.

In contrast, if we made every blob download 100% of the data, each epoch would required 2,048 blobs (2 GB) and 0 samples.

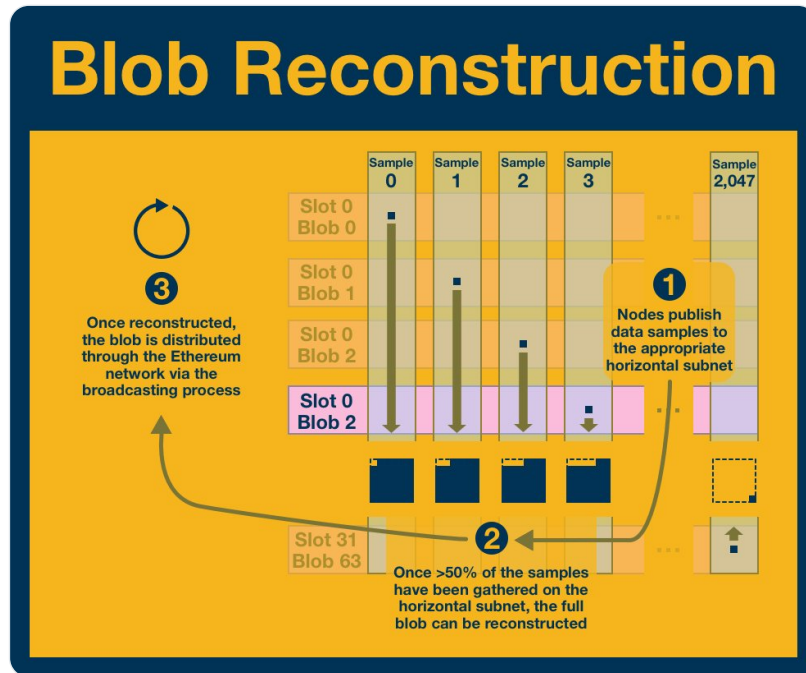
We are talking about a 99.95% reduction.

(21/24) BUT the result is that the data doesn't actually exist in its full form on the network. And so, before we can call our P2P design complete, we need to devise a way to recover the data.

We must reconstruct the blob using the pieces distributed around [@ethereum](#).

(22/24) Fortunately, our design makes this process incredibly easy; we just run the publication process in reverse!

We begin with the vertical subnets. Each participant publishes the relevant sample to the appropriate horizontal subnet where it is reconstructed.



(23/24) This is the basic structure behind data scaling in [@ethereum](#): we ensure all of the data is readily available within the system WITHOUT requiring any single node to hold it all.

The secret: clever P2P design that allows the efficient storage of movement and data.

(24/24) But designing the network and creating a data-scheme is only half the battle. The second half is actually designing the system.

For that, we are going to need something intricate... something secure enough to protect the World Computer.

Something super DANK!

Like what you read? Help me spread the word by retweeting the thread (linked below).

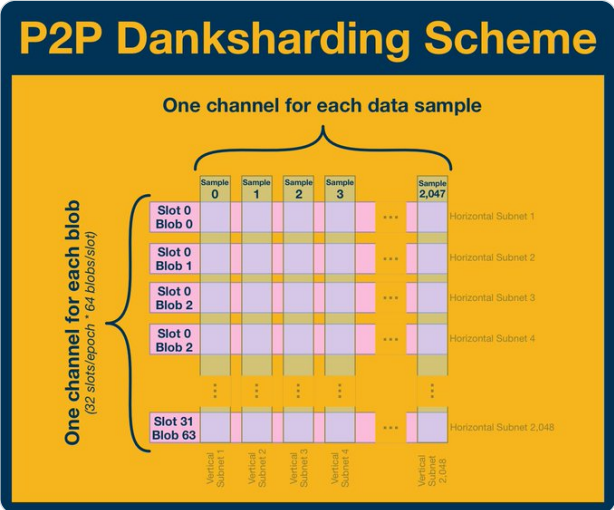
Follow me for more explainers and as much alpha as I can possibly serve.

Haym
@SalomonCrypto · Follow

(1/24) @Ethereum Roadmap: Peer-to-Peer Networking

In order to achieve the vision of the World Computer, Ethereum needs more data bandwidth. But more bandwidth = higher node requirements = centralization. So what's the path forward?

Let's look into the future of the P2P network.



The diagram, titled "P2P Danksharding Scheme", illustrates a data distribution model. It features a grid where the vertical axis represents "One channel for each blob (32 slots/epoch * 64 blobs/slot)" and the horizontal axis represents "One channel for each data sample". The grid is organized into "Horizontal Subnets" (rows) and "Vertical Subnets" (columns). The columns are labeled "Sample 0", "Sample 1", "Sample 2", "Sample 3", and "Sample 2,047". The rows are labeled "Slot 0 Blob 0", "Slot 0 Blob 1", "Slot 0 Blob 2", "Slot 0 Blob 2", and "Slot 31 Blob 63". The diagram shows that each data sample is distributed across multiple horizontal subnets, and each blob is distributed across multiple vertical subnets, demonstrating a peer-to-peer sharding scheme.

10:57 PM · Oct 25, 2022

[Read the full conversation on Twitter](#)

7 [Reply](#) [Copy link](#)

[Read 1 reply](#)

...