

# EIP 4844: What does it mean for L2 users?

[Ed Felten](#)

[Follow](#)

Offchain Labs

--

1

Listen

Share

As of this writing, EIP-4844 and the rest of Dencun upgrade are currently scheduled to go live on Ethereum mainnet on March 13.

Dencun, the next Ethereum hard fork, is on testnet and will be coming to mainnet soon. Dencun is big news for L2 users, because it brings EIP-4844, a new data availability system that promises a potentially big reduction in L2 costs. If you use an L2 like Arbitrum, how does this affect you? Let's investigate.

Before we start, a big shout-out to my research collaborators, Davide Crapis of the Ethereum Foundation and Akaki Mamagishvili of Offchain Labs, for the [research](#) underlying this post.

## Rollup data availability and EIP-4844

Rollups spend a lot to make sure that their transaction data is available to all of their nodes and validators. Most rollups do this by writing their data to Ethereum as calldata. That costs about \$1000 per megabyte at current prices. Good rollups cut that to \$300 per megabyte by using advanced data compression. Still, data posting cost makes up the largest portion of L2 transaction fees.

Why is Ethereum calldata so expensive? Because Ethereum is in high demand, and rollups have to outbid other parties who want to use Ethereum's resources.

EIP-4844 adds a new type of data availability to Ethereum, called blob data

blob data

. Blobs lack some features of calldata — they're not available to Ethereum contracts, and they're only stored for 18 days — but they meet the data availability needs of rollups, so they provide a new, and hopefully cheaper, way for rollups to record their data.

How much cheaper? That's the critical question.

## 4844 basics

A 4844 data blob holds around 125 kilobytes of data. Each Ethereum block can include up to 6 blobs, with a target of 3 blobs per block. The price of a blob will fluctuate over time — if an Ethereum block has more than 3 blobs, the blob price will go up; if fewer than 3 blobs, the blob price will go down.

A price adjuster like this works like a thermostat. If your home is too cold, the thermostat turns up the heater so the place gets warmer; if your home is too hot, the thermostat turns down the heater. The result is an equilibrium

where the temperature stays at the desired level.

Blob pricing is similar, but instead of seeking a constant temperature, it seeks a constant rate of blob usage. If rollups are using too many blobs, the price goes up so the rollups will back off. If rollups are using less than the target number of blobs, the price goes down to encourage more use. The result should be an equilibrium where rollups are using about 3 blobs per Ethereum block, and the price is set just right to maintain this.

Of course, conditions will change over time, so the equilibrium price may change with conditions, but the use of 3 blobs per block on average will hold — just like how the thermostat keeps your home at the right temperature when the outside weather changes, using the heater more when it is colder outside.

This suggests a recipe for finding the equilibrium price. We can ask ourselves: If the blob price was 10 cents, how many blobs would rollups use? What if the price was \$1? What if \$10? Etc. The equilibrium price will be the one where rollups use exactly three blobs per Ethereum block.

So let's think about ...

## How rollups will use 4844 blobs

A good rollup will use blobs in whatever way best serves the rollup's users. And users want two main things from the rollup's data poster: low cost and fast posting.

These two goals are in tension. The lowest cost way to use blobs is to wait until the rollup has enough data to fill a blob, and only then buy a blob and post the data. But posting is faster if the rollup doesn't wait so long, and instead posts partially full blobs more often. That's more expensive because you have to pay for the full blob even if you don't use it all, so the frequent-posting strategy will buy more blobs.

How to balance these factors depends on the price of blobs. If blobs are super-expensive, you want to use as few of them as possible, so it's worth waiting longer and posting full blobs. On the other hand, if blobs are super-cheap, then you can post very quickly, posting blobs that are mostly empty, at almost no cost. The lower the blob price, the more often you should post, and the emptier your blobs will be.

The [research paper](#) makes this tradeoff precise, by assigning a notional dollar cost to the time spent waiting for posting, then finding the blob usage that minimizes the sum of blob cost plus waiting cost. The result, as suggested above, is that rollups' blob use moves inversely to the blob price: when the price is high they use fewer blobs, and when the price is low they use more blobs and those blobs are less full.

## Rollup size matters

A surprising takeaway from the research is that rollups that are busier (having more transactions) do better than less busy rollups. For a busier rollup, the tradeoff between cost and finality time is less harsh, because a busy rollup will accumulate a full blob of data more quickly. If the busy rollup is making smart decisions, it can have both better cost and faster posting time than its less-busy rivals.

For example, if Rollup A has twice the traffic of rollup B, then if both rollups are making optimal decisions in the cost-delay tradeoff, Rollup A will have data cost per transaction about 41% lower than Rollup B and

will post its transactions about 41% faster.

In some cases, a small rollup will prefer to use Ethereum calldata rather than blobs. This is because when you use blobs, you have to pay for the whole blob regardless of whether you're using it, but with calldata you only pay for the number of bytes you use. So a low-traffic rollup might find that waiting to build up a large batch imposes too much delay on its users, and having decided to use a small batch, the cheapest way to post that small batch is to buy a small amount of Ethereum calldata rather than a whole blob.

All in all, smaller rollups face worse tradeoffs than large ones. What can smaller rollups do?

## Teaming up

The best tactic for smaller rollups is to team up and share blobs. As an example, suppose that right now, Rollup C wants to post a blob that is 70% full and Rollup D wants to post a blob that is 30% full. If they team up, they can post a single blob that contains both of their data. Instead of buying one blob each, they now get to split the cost of a single blob. It's win-win.

But why stop there? If the two rollups form an alliance and always post together, they can get lower cost and better delay than they would get separately. It's long-term win-win.

Even the busiest rollup can sometimes benefit from teaming up. Instead of waiting for enough transactions to fill a blob completely, the busy rollup might accumulate enough of its own transactions to fill 90% of a blob, then sell the remaining 10% of space to a less-busy rollup. Depending on the circumstances, that might or might not turn out to be win-win for the two.

## Splitting up the savings

When rollups share a blob, they save money. How should they split up the savings? Or equivalently, how should they split up the cost of posting the shared blob?

An obvious approach is to pay in proportion to how much data they use. In the 70/30 combination from above, rollup A would pay 70% because it provided 70% of the data, and rollup B would pay 30% because it provided 30% of the data. That's one way to do it.

Another way is to argue that if rollup A is getting an X% reduction in its cost due to teaming up, then rollup B should also get the same X% reduction. This suggests that they should each pay for half of the shared blob. If they hadn't teamed up, Rollup A would have bought a whole blob, so paying for half of the shared blob is a 50% savings. Similarly, if they hadn't

teamed up, Rollup B would have bought a whole blob, so paying half is also 50% savings. By this argument, teammates should always pay for equal shares of a shared blob, regardless of how much data each one contributed.

You can argue for either of those splits, or perhaps you'll say both have some merit so the best answer is somewhere in between. The research paper discusses a rule called the "Nash bargain" that has some nice technical properties and turns out to be in-between the proportional-split and equal-split rules.

## **Enough theory. What do we actually expect?**

The fact of the matter is that nobody can say for sure what will happen when EIP-4844 rolls out. We can be confident that rollup transactions will get cheaper. But we can't say with any confidence how much cheaper they'll be.

And we know one more thing as well. Though 4844 is an important step, Ethereum will continue to evolve and improve its data availability support, and rollups will continue to evolve their strategies for using it. We'll see increasing technical sophistication in what is available and how rollups use it — and the result will be a better deal for users.