An exploratory data analysis on Filecoin gas fees

Maria Silva, January 2022

Table of contents

---

1 - Context and dataset

This report is the result of an exploratory data analysis of some gas fee-related metrics extracted from the Filecoin network. The goal was to extract insights into the gas dynamics and how users responded to the mechanisms in place.

For this analysis, we used a dataset of gas fee metrics over the month of June 2021. Each row corresponds to an average over all the blocks produced within a specific minute. Since each epoch is 30 seconds and each epoch has on average 5 blocks, each row aggregates the information of 2 epochs and, approximately, 10 blocks of transactions.

The original dataset had 34273 data points and 10 features (the table below describes these features). There were no null values, so there was no need to drop any observations or input values.

Field	Unit	Description
`timestamp`	ns	timestamp of minute of aggregation
`mean_gas_fee_cap`	attoFIL / GasUnit	The maximum FIL amount that a sender is willing to pay per GasUnit for including a message in a block
`mean_gas_premium`	attoFIL / GasUnit	A priority fee that is paid to the block-producing miner.
`mean_gas_limit`	GasUnit	The limit on the amount of gas that a message's execution can consume
`mean_gas_used`	GasUnit	The amount of gas that a message's execution actually consumes

`mean_parent_base_fee`	attoFIL / GasUnit	The base fee in the parent epoch. The base fee is the amount of FIL that gets burned. It is set dynamically by the network
`mean_base_fee_burn`	attoFIL	The amount of FIL burned due to the base fee
`mean_over_estimate_burn`	attoFIL	The amount of FIL burned to compensate for gas estimation errors.
`mean_gas_refund`	GasUnit	The gas units left after discounting the gas burned and the gas used by the transaction. This is returned to the sender.
`mean_gas_burned`	GasUnit	The total amount of gas units burned

Besides these features, we create some new features, namely:

- `days` - the day of June of the row
- `mean_gas_estimation_error` - the difference between `mean_gas_limit` and `mean_gas_used`. It is the average error the sender makes when estimating its gas limit.
- `mean_over_estimation_penalty` - the average penalty in gas units due to overestimation errors. It is the difference between `mean_gas_estimation_error` and `mean_gas_refund`.

For some of the plots, we also converted the units of the features to make the plot easier to read. The conversions were the following:

- `1 attoFIL = 1E-9 nanoFIL`
- `1 nanoFIL = 1E-9 FIL`
- `1 million GasUnit = 1E-6 GasUnit`

Finally, for the rest of the report, we will refer to each feature by its name excluding the "mean". For instance, when we talk about the *base fee*, we mean the *average base fee* for the minute. However, for readability, we will drop the references to the averages.

2 - Base fee

The base fee is an essential component of Filecoin's gas fee mechanism. Filecoin uses an adapted version of Ethereum's [EIP1559](https://eips.ethereum.org/EIPS/eip-1559) and, as such, introduces a fee that is dynamically set by the network meant to control supply and demand, instead of a pure first-price auction.

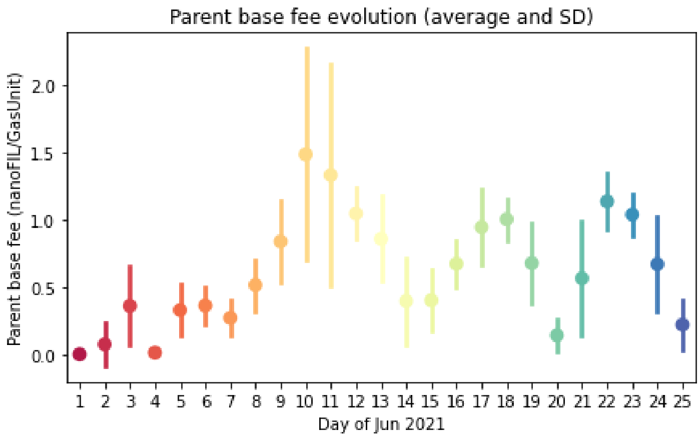
In the total fee paid by message senders in the Filecoin network, a big part is governed by the base fee. It was designed to control supply and demand by dynamically adjusting the minimum price per unit of gas based on how congested the network is. If there are too many transactions being sent to the network, the base fee increases, thus balancing the number of people that actually want to send transactions during that time.

At a minimum, all transactions need to pay the base fee times the amount of gas used, an amount which is burned when the block is mined. Additionally, senders may specify a higher fee than the base fee and, in that case, the difference goes to the miner. Here, the total FIL that is given to the miner is the gas limit (i.e. the gas amount estimated by the sender) times the premium gas fee (which is the difference between the gas fee cap specified by the sender and the base fee). In Section 3, we explore this premium gas fee and the gas fee caps in more detail.

Finally, an additional amount of FIL is burned every time there is an overestimation of the amount needed for each message. This amount is computed as the estimation error (i.e. the gas limit minus the actual gas used) times the base fee. So, the base fee is also used to price gas overestimation errors. We discuss this overestimation penalty in more detail.

Now that we have described how fees are set in Filecoin, let's take a deeper look at the base fee. In this dataset, we have the parent base fee, which is the base fee applied to the parent epoch. So, even though we don't have the base fee of the current epoch, this is the best approximation we have. In addition, because we are using an average of 2 epochs, we should not see huge variations.

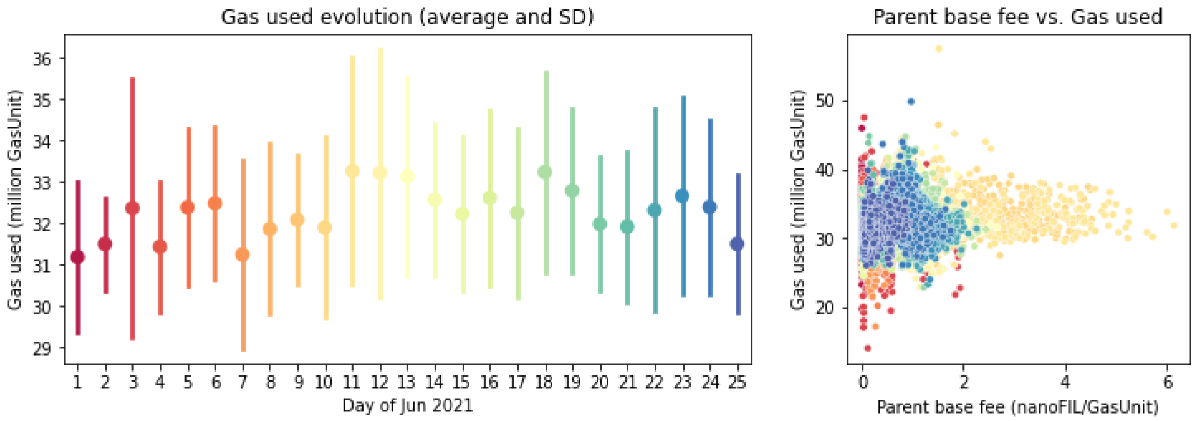
The next plot shows a daily summary of the parent base fees for each day in June 2021. Here, each point corresponds to the average of the day, while the error bars encode the standard deviation for the day.



It is interesting to observe that there is some fluctuation in the base fee during the month of June 2021. Interestingly, the fees start very small and, after day 8, they quickly increase to the highest values observed in June. After this, they experience some ups and downs but never return to those values.

From the data, we cannot tell what happened to the network after the 8th of June, but it seems to be some sort of upward shift in the base fees, accompanied by an increase in volatility.

Now, let's look into gas consumption. The figure below contains two plots related to the gas used. On the left, we see the same daily summaries and on the right, we have a scatter plot that compares the parent base fee with the gas used. Ideally, we would be comparing the gas used in the previous epoch with the base fee used in the current block, however, this is the closest comparison we can do. Additionally, even if it is not perfect, we would expect some level of correlation between the base fees of adjacent epochs.



Surprisingly, we do not see a strong correlation between the gas used the parent base fee. In the daily averages, we see slight increases in gas usage on days 3, 11, 12, 18 and 23. However, it is not even close to the variations experienced in the base fees. Note that we also made a similar analysis for the gas limit and the conclusions are the same.

Thus, we reach a surprising conclusion that needs further investigation. The first step would be to get data for the base fee (instead of the parent base fee) and compare it against the total gas used in all the blocks in a certain epoch. It may be that we are missing some activity due to the minute averages. For instance, if we do the average at the level of the message, this would hide the total gas used in a block because we do not take into account *how many transactions are being added.*

Another data point we could analyse to confirm that this increase is due to network congestion would be to look at the mempool size. If we see the mempool increasing, then we know that the network is not being able to process all the messages.

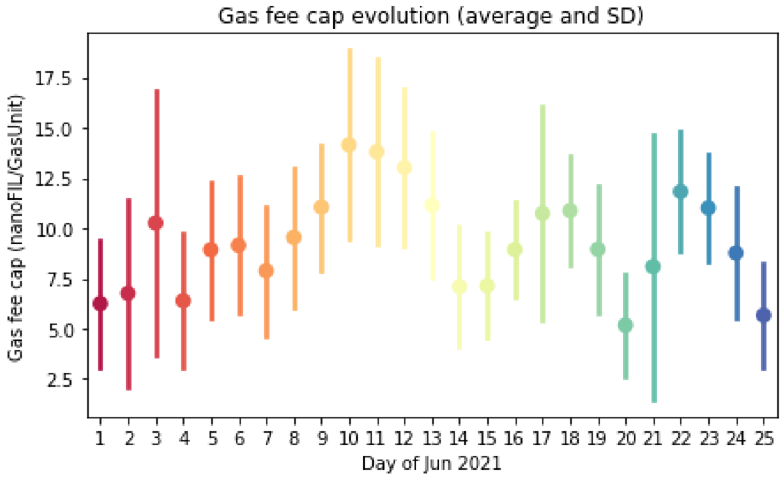
If we confirm that the increase is due to congestion, we would need to understand why the network got congested. We could look at the number of deals being made during that time and at the number of new actors joining the network as a first step.

On the other hand, if we cannot find congestion as the cause for the increase in the base fee, one possible explanation could be a change in the base fee computation mechanism. We would need to check the repository and dedicated channels and search for any changes made during that time.

3 - Gas fee cap and Premium fee

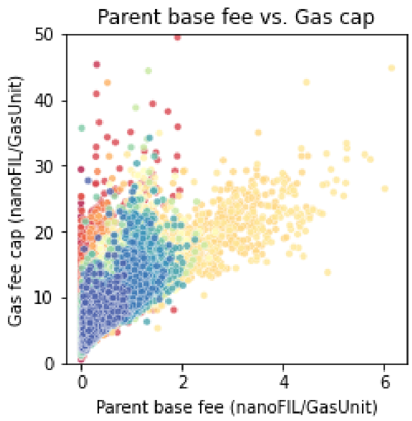
In this section, we analyse how users responded to the observed fluctuations in the base fee. Have they adapted to the parent base fee and changed their gas caps accordingly? To answer this question, let's start by looking into the gas fee cap, which is the maximum nanoFIL that a sender is willing to pay per GasUnit.

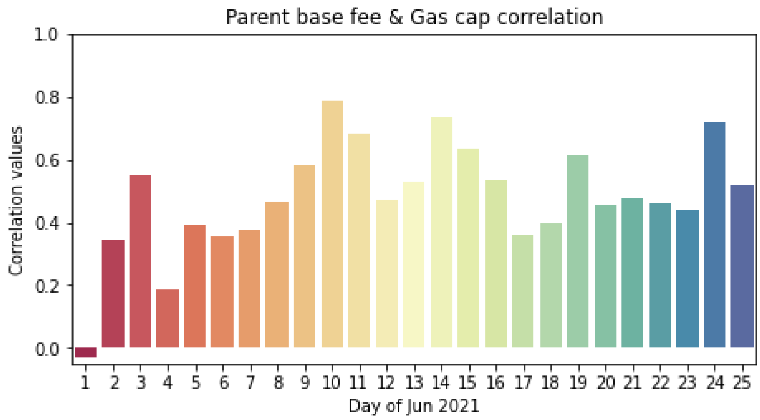
The next figure contains a daily summary of the gas fee caps for each day in June 2021. Each point corresponds to the average of the day, while the error bars encode the standard deviation for the day.



Looking at the gas fee caps per day, we see a similar trend to the one observed in the parent base fees. In other words, we see the same up and down fluctuations after the 10th. However, at the beginning of the month, the relationship is not as clear. Let's take a closer look at this in the next figure.

Here we present two different plots. On the left, we have a scatter plot of the gas fee caps against the parent base fees, where the colour encodes the day of the month. On the right, we have a bar chart where each bar represents the correlation coefficient of the parent base fee and gas cap correlation for each day of the month.

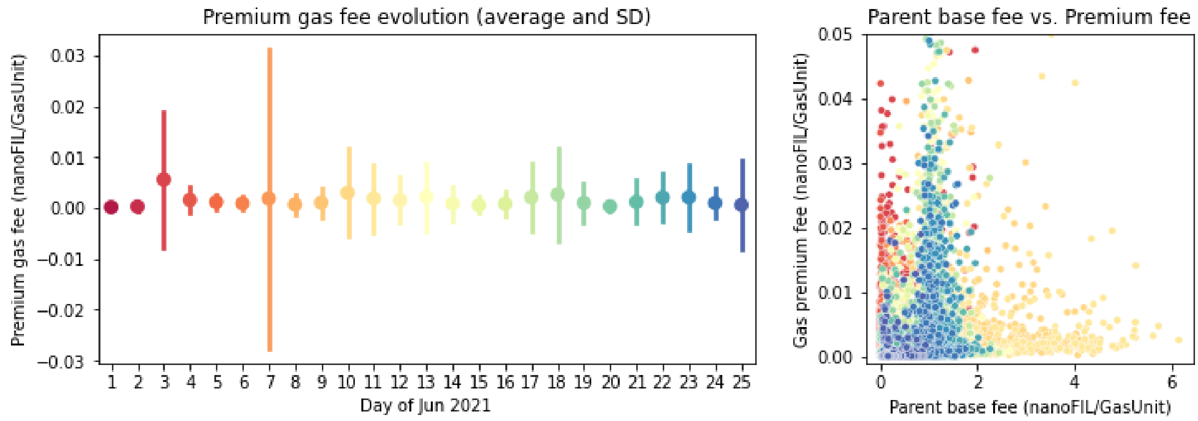




The correlation between the parent base fees and the gas fee caps is 0.64, which indicates that network users do try to follow the parent base fee when setting their gas caps. However, we also observe a certain level of resistance. Concretely, in the days where the base fee is smaller, the correlation also tends to decrease.

Now, let's take a look at the gas premium fees. The premium fee is defined as the difference between the fee cap and the base fee and corresponds to the amount of FIL per GasUnit that is awarded by the user to the miner. Thus, for a given base fee, the higher the fee cap, the higher the final premium fee will be and, consequently, the most likely it is for a transaction to be picked from the mempool by the miners.

In the next figure, we show two plots. On the right, we have the daily summaries of the premium gas fee. As before, the points are the averages for the day while the error bars are the standard deviations. On the right, we plot the parent base fees against the gas premium fees.



On the 3rd and 7th of June, we see some extreme values. However, excluding those, the premium fee has remained more or less stable during June 2021. In addition, the premium fee is not strongly correlated with the parent base fee, which indicates that changes in the base fee are not coupled with either positive or negative changes in the premium fee.

In conclusion, users seem to loosely adjust their fee caps with changes in the base fee. However, this change is not necessarily transferred to the premium fees. The premium fee is not correlated with the base fee, which means that in times of high congestion the average user does not increase the miner rewards to make sure that their transaction goes through. In these situations, we tend to see an increase in the fee caps, but these only cover the increase in the base fees.

This is a very interesting phenomenon that we could explore in more detail. One could see if this trend is maintained on all message types, which would require a split of gas fees per message type. A possible avenue would be to see if the trend is maintained in two distinct groups of messages, namely proof messages and deals messages. Since the first are mostly sent by miners while the second are agreements between miners and clients, we may see different dynamics in these two groups.

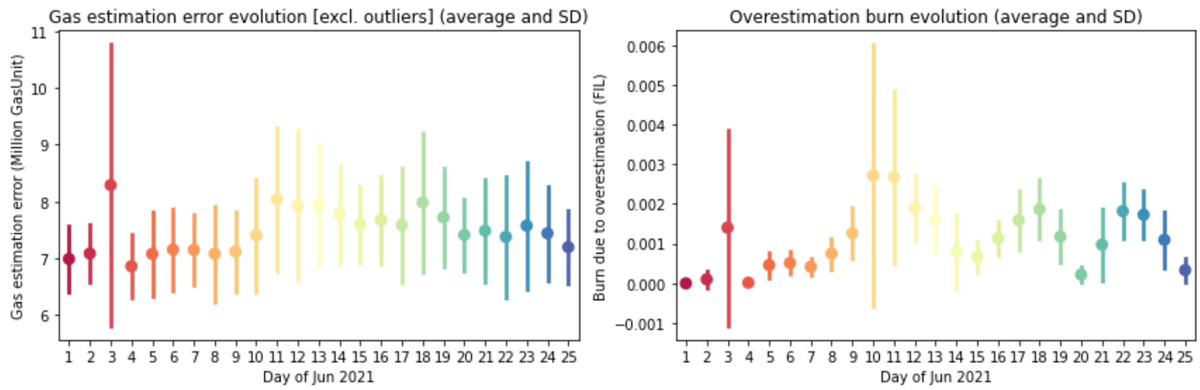
In addition, one may investigate whether this trend is consistent through time. It could be that this trend only happened at the beginning of the network and, recently, with more people joining the network, new dynamics may have emerged. For this, we would need to collect data for a longer time period.

4 - Gas estimation and overestimation penalty

Another interesting mechanism of the Filecoin gas fees is the overestimation penalty. This penalty depends on the difference between the estimated gas limit and the actual gas used by the message (the higher the difference, the more FIL is burned).

The goal of this penalty is to force the users of the network to estimate their gas consumption correctly. But why is this important? The reason is that blocks don't have infinite space. Miners need to make sure that the total gas of their block stays below a specified threshold. To do this, they add all the gas limits of the transactions they want to include in the block and, if the total surpasses the threshold, they know they have to exclude some transactions. Therefore, the estimated gas limit occupies space in each block and overestimations leads to a loss of gas that could have been used in other transactions.

So, in this section, we explore the gas estimations errors and the penalties paid by the users for these errors. Our first figure shows the same daily summaries we have seen before (where the points represent the averages for the day and the errors bars represent the standard deviations). On the left, we plot the gas estimation errors in million GasUnits. Note that in order for the plot to be more visible, we exclude a few outliers from the sample, which were located on the 3rd of June. On the right, we have the amount of FIL burned due to overestimation errors. This amount is the product of the base fee and the overestimation penalty.

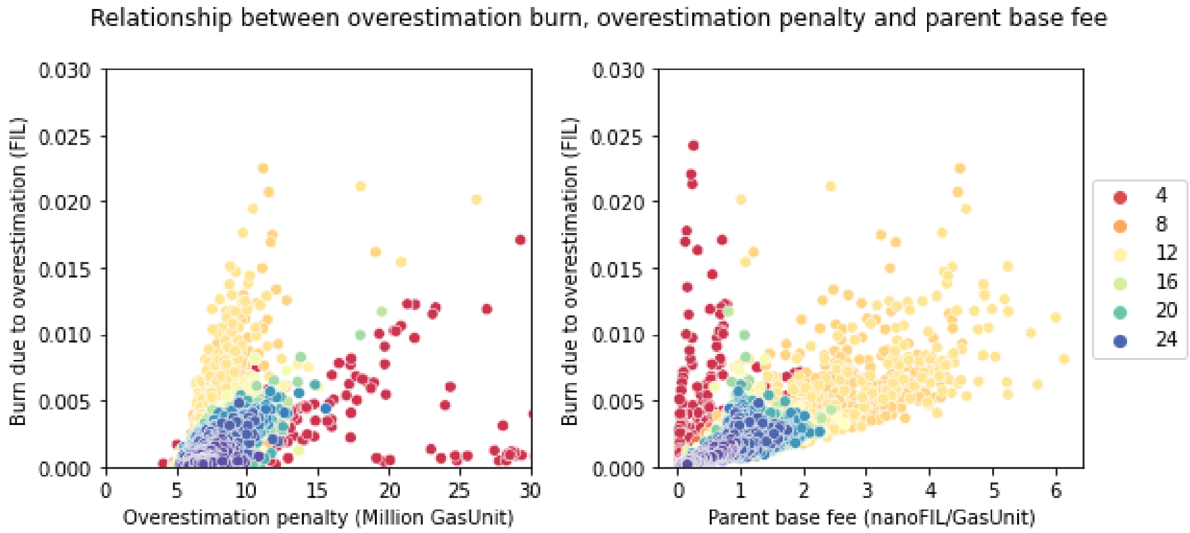


In these plots, we observe daily fluctuations similar to the ones experienced with the base fee. Concretely, on the 3rd, 11th, 18th, and 23rd of June, there is an increase in gas estimation errors and, consequently, in the amount of FIL burned due to overestimation.

This effect was especially felt on the 3rd, where a significant number of large estimation errors skewed the average of that day. This was the day when larger than normal premiums were paid to miners and when we started to observe a sustained increase in the base fee.

From the data we have, it is not clear what happened on that day. However, it is clear that we had an influx of messages that were clearly overestimating their gas limits. In order to fully understand this effect, one could analyse how the number of new actors and active actors per day evolved during this month. Looking at the total number of actors and its distribution by actor type (e.g. miner, token holder, client, etc.), could give us an idea of what was causing this phenomenon. This could also be an effect of a new feature that was introduced around that time or a bug in the tools used to create and send messages to the mempool. To check this, one could go to the repositories of the main projects around the Filecoin network and search for any activity around this time.

The other time where we see very high overestimation burn amounts is around the 11th of June. However, looking at both the base fees and the estimation errors during this time, we conclude that the high overestimation burn amount was mostly caused by the increase in the base fee and not by the actual overestimation errors. The next plot showcases this. On the dark red points (i.e., day 3), we see high overestimation penalties and low base fees, while, on the yellow points (i.e., day 10th), we see "normal" overestimation penalties and high base fees.



So, for the 11th of June, we go back to the issue of very high base fees discussed in Section 2.