The Impact of Ethereum Throughput and Fees on the Transaction Latency during ICOs

Michael Spain
School of Computer Science, University of Sydney, Australia
mspa1382@uni.sydney.edu.au

Sean Foley
School of Business, University of Sydney, Australia
sean.foley@sydney.edu.au

Vincent Gramoli
School of Computer Science, University of Sydney, Australia
Data61-CSIRO, Australia
vincent.gramoli@sydney.edu.au

Abstract

The Ethereum blockchain has gained popularity for its ability to implement Initial Coin Offerings (ICOs), whereby a buyer enters a market order agreement with a seller in order to purchase cryptographic tokens at an agreed price. The popularity of ICOs in 2017 has created an increasingly adversarial environment among potential buyers, who compete for what is often a fixed supply of tokens offered for a limited period of time.

We study the impact of a series of ICOs in order to understand the relationship between transaction fees, throughput and latency in Ethereum. Our analysis considers the effects on both Ethereum's service providers, known as miners, and users who issue transactions in the network. Our results show that while buyers incentivise miners generously to include their transactions during ICOs, the latency of these transactions is predominantly determined by the levels of supply and demand in the network.

2012 ACM Subject Classification Human and societal aspects of security and privacy → Economics of security and privacy

Keywords and phrases ICO, Gas, Ethereum, Transaction Fee, Latency, Fairness

Digital Object Identifier 10.4230/OASIcs.Tokenomics.2019.23

1 Introduction

An *Initial Coin Offering (ICO)*, which typically consists of offering a fixed quantity of securities at a discounted price for a limited time, has popularized the use of the Ethereum blockchain [21]. Today, Ethereum is the second largest blockchain in terms of market capitalization after Bitcoin [14]. In Ethereum, the notion of gas was introduced in part for the need to incentivise miners to include, in the blocks they create, transactions of varying computational complexity [2]. Transactions may invoke smart contracts that allow a buyer and seller to transfer tokens at an agreed upon price expressed in Ether (Eth), the native token of Ethereum. This mechanism was used to raise more than $20B throughout 2017 and 2018.[1]

Research has revealed that in Bitcoin, the higher the fee users are willing to pay to the miners for including their transactions, the faster these transactions are included in the blockchain [13]. Interestingly, applying the same strategy in Ethereum could potentially

[1] `https://cointelegraph.com/news/ico-market-2018-vs-2017-trends-capitalization-\ localization-industries-success-rate`.

lead to front running, the act of "entering into an equity trade, options or futures contracts with advance knowledge of a block transaction that will influence the price of the underlying security to capitalize on the trade"[2]. A recent study which considered over 3 months worth of Ethereum transactions has shown that most take more than 3 minutes to be included [20], which is long enough to expose pending transactions to the risk of front running by way of issuing a similar transaction with a larger fee. However, it remains unclear whether transaction fees significantly impact the likelihood of successfully purchasing tokens during an ICO.

In this paper, we study this question empirically, first showing that during popular ICOs some participants are paying significantly higher fees. To begin, we retrospectively analyse the revenue and costs of the mining process during the ICO for the Basic Attention Token (BAT), where the entire supply of available tokens was sold in less than 30 seconds for $35M. During this ICO, participants paid on average more than 300× the average fee, providing rewards for miners that were orders of magnitude higher than the additional mining costs incurred during the ICO. These findings support our hypothesis that participants are willing to have their transactions included faster than others in the blockchain to ensure they could purchase tokens. More specifically, we measured the time it takes for transactions to be included in Ethereum's blockchain depending on their associated fees. Although we confirm our hypothesis that the fee of a transaction is inversely correlated to the latency, the correlation is surprisingly low, indicating that during the observed period, transaction fees were not a successful mechanism for front running in Ethereum. This observation, however, does not explain the large discrepancy in transaction latency we observed during this ICO.

To explain this discrepancy, we conducted a thorough analysis of transaction latency on the Ethereum blockchain for a period of 24 days in 2017, which included 19 ICOs, notably TenX and Tezos, that raised altogether more than $700M. By combining our analysis of the fees with the block gas limits in Ethereum, we identified other factors that contribute to the latency of a transaction. In particular, we observed that service demand was greater than the supply, which dramatically raises the latency of some transactions. More precisely, facing the rising popularity in ICOs, the volume of transactions issued exceeded the capability of the service. Then we also observed that, in mid-2017, when the Ethereum gas limit was increased, the capability of the service also increased. To conclude, this capacity analysis revealed that the effect of transaction fees were insignificant due to the high service demand.

The rest of the paper is organised as follows. Section 2 describes the important concepts about ICOs and the Ethereum blockchain. Section 3 illustrates the tremendous increase in Ethereum transaction fees during a successful ICO. Section 4 correlates the increased transaction fees with the latency of transaction. Section 5 indicates how the supply and demand of the service impacts latency. Finally, Section 6 lists the related work and Section 7 concludes.

2 Background

A blockchain is a chain of blocks distributed among multiple participating nodes, where *miners* create blocks to include transactions issued by any participating client node [14]. In proof-of-work blockchains, miners solve a computationally intensive cryptographic problem to prove that their block is legitimate.

[2] `https://www.nasdaq.com/investing/glossary/f/front-running`.

2.1 Initial coin offering

An Initial Coin Offering (ICO) is a method of raising funds through a blockchain system for mostly blockchain related projects. Ethereum, being the largest blockchain with the ability to conduct ICOs, has experienced hundreds of ICOs in 2017 alone [17]. ICOs are an attractive alternative to other early stage funding processes such as Venture Capital, because they circumvent many of the legal and regulatory requirements and facilitate individuals' participation. Projects are often able to raise significantly more capital through an ICO than is possible with traditional approaches. We focus our study on Ethereum.

2.2 Mining in Ethereum

Miners participating in the Ethereum network run the Ethereum Virtual Machine (EVM) which executes smart contracts. Unlike transactions in Bitcoin, Ethereum transactions can invoke arbitrarily complex functions through smart contracts. This increased functionality requires the protocol to measure the amount of computation each transaction performs for two reasons [2]. First, a miner needs to be able to determine ahead of time whether the transaction they are about to execute will ever finish. Second, there needs to be a mechanism for users to incentivise miners to include computationally intensive transactions. This is why Ethereum uses the concept of *gas*, whose unit represents one computational step in the EVM—all the opcodes in the EVM have a cost measured in gas. Every transaction in Ethereum must include both the gas limit, which is the maximum amount of gas that can be used executing the transaction and the gas price, which is the price, measured in *Wei* ($1\,Eth = 10^{18}\,Wei$), that the sender will pay per unit of gas. If the transaction execution is not finished after the gas limit is reached, the EVM will abort the transaction and revert any state changes. Hence the fee in Ether associated with transaction t in Ethereum is:

$$fee_t = \frac{gas\text{-}price \times gas\text{-}used(t)}{10^{18}}.$$

The Ethereum mining algorithm is Ethash, which is a memory hard algorithm designed to reduce the level of centralisation risks compared to Bitcoin's Hashcash algorithm that is now dominated by centralised pools of Application Specific Integrated Circuits (ASICs).

2.3 Incentives

When a block is created in Ethereum, the miner of the block can vote to increase, decrease or maintain the total gas limit of the next block. This allows the maximum throughput of Ethereum to adjust over time with the capabilities of the miners. The miner of a block b in Ethereum receives 5 Ether plus the sum of the fees for all transactions included in b:

$$reward_b = 5 + \sum_{\forall t \in b} fee_t.$$

Ideally, the miner will include as many transactions as they can (up to the gas limit of the block). However, the block reward usually exceeds the marginal increase in revenue that is gained from including more transactions, since the miner must restart the process each time it includes new transactions (as the block content changes). The primary incentive for the miner is the block reward, rather than the fees gained from filling the block. This becomes a problem when the number of transactions issued starts to approach the maximum theoretical throughput [13].

3 The Basic Attention Token ICO

In this section we study the impact of an ICO that raised $35M in less than 30 seconds on 31st of May 2017 on the Ethereum economy. We show that the Basic Attention Token (BAT) ICO impacted the relationship between mining revenues and costs.

This experiment extends the research of Möser and Böhme [13] to the context of Ethereum, considering the impact of impatient users on mining revenue and costs. We find that high demand for the network could create an inequitable environment for Ethereum users. These findings serve as motivation for our study of transaction latency, presented in Section 4.

3.1 Experimental settings

This experiment studies the transactions confirmed by the Ethereum network during the BAT ICO on the 31st of May 2017, that started with block 3798640 and ended with block 3798642. The data was obtained from the block explorer Etherchain which provides a public API for Ethereum block and transaction data [8]. Data was gathered for a total of 10003 blocks, which includes 5000 before the BAT sale and 5000 after. This number represents roughly one day before and one day after the sale in order to approximate average network conditions, so that the effect of the BAT ICO can be effectively quantified.

3.2 Mining revenue

The analysis compares the average of a variety of metrics in the period directly before and after the BAT ICO with those observed between blocks 3798640 and 3798642.

Table 1 Average statistics vs. BAT blocks

	Average	Block 3798640	Block 3798641	Block 3798642
Total Block Fee (Eth)	0.08	28.05	35.29	12.14
Total Block Size (Bytes)	6952	10359	9479	5403
Tx (Per Block)	39	85	42	42
Gas Used (Per Block)	2247441	4313308	4262947	4326224

Table 1 compares metrics observed in the three BAT blocks to the cumulative averages for these metrics recorded before and after. We can see that the average transaction fee per block recorded over the period (excluding the BAT ICO) was 0.08 Ether, 314.5× lower than during the ICO.

In Figure 1(left) we plot the total block fees for each block in the dataset. The impact of the BAT ICO on mining revenue is immediately evident. The log scale used on the y-axis is needed to represent an increase of hundreds of times the average fees per block, with a noticeable residual effect in the blocks that immediately followed the ICO. We now show that this level of mining revenue is not proportional to the increased cost incurred by the miners of these blocks or reflected in the performance of the network.

The first and second blocks of the BAT sale were 49% and 36% larger than the average block size before and after the ICO, respectively. However, the final BAT block was smaller than the average block size over the period. So, while on aggregate the raw size of the blocks appended by miners during the BAT ICO increased, it was insignificant compared to the increase in revenue. The next metric considered is the number of transactions per block, which reveals the achieved throughput of the network. While the first BAT block recorded

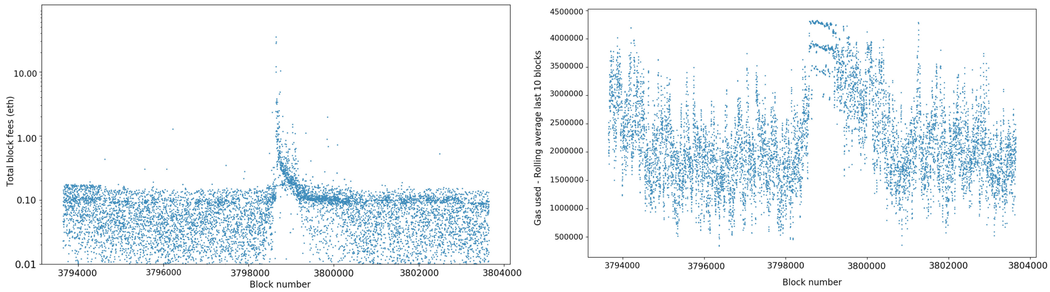


Figure 1 Total fees per block and rolling average of last 10 blocks around the BAT ICO

twice as many transactions than the average, the other two BAT blocks were close to the average, showing that the network did not provide any significant improvement in throughput during the BAT ICO.

The most appropriate proxy for the computational effort expended by mining is gas used per block. This is because the EVM performs transactions of arbitrary complexity and gas is used to measure the total computational demand of the transaction. The BAT blocks consumed almost twice as much gas as an average block over the period.

Figure 1 (right) depicts a rolling average of gas used in the last 10 blocks. It reveals that around the BAT ICO, the majority of blocks mined were close to the gas limit. This indicates that, although neither block size or throughput increased significantly, the computational work performed by the miners was significantly higher than any other period in the sample.

3.3 Users pay substantial transaction fees during an ICO

Analysing the mining environment around the BAT ICO shows that during periods of high demand in Ethereum, some users are willing to pay massive transaction fees in order to have their transactions included quickly. Consequently, the miners of the BAT blocks received a total block reward that was much larger than usual over the period. In contrast, while the computational costs associated with the mining environment increased, they were insignificant compared to the change in revenue.

This experiment has revealed the actions of impatient users when there is an excessive demand to transact. We hypothesize that due to the fees shown in Figure 1, many users attempting to enter the ICO or transacting during this time were negatively impacted due to miners prioritising high fee transactions first. While in principal it seems fair that high fee transactions should be prioritised, it raises a question of fairness in blockchains. Wealthy users could possibly front run the transactions of others by setting a fee that is large enough.

The cost of immediacy in Ethereum is clearly subject to significant variation based on network activity. This presents a substantial disadvantage to users wishing to transact small during periods of high activity. Whilst it is possible to quantify the effect of large transaction fees on mining revenue and environment, we are not able to determine the effect that these transactions have on other users wishing to transact at a similar time. This observation serves as motivation to study the latency of transactions in Ethereum, to determine how the transaction fee impacts the latency of that transaction.

4 The Impact of Transaction Fees on Latency

As discussed previously, high transaction fees paid by participants during ICOs could possibly be motivated by a front running attempt. In this section, we study how successful high fees are at reducing transaction latency. We start by introducing the concept of transaction latency before describing our experimental setup and conclude that, as expected, transaction fee is inversely correlated to the transaction latency, but surprisingly, that this correlation is negligible.

4.1 Defining transaction latency

We refer to the *latency* of a transaction t as the time taken for it to be included in a block. Note that this does not necessarily correspond to the inclusion time of t, as Ethereum cannot deterministically define the inclusion of a transaction as a number of appended blocks or confirmations [15].

For the latency of transaction t to be well-defined, the block that includes t must be part of the canonical blockchain as determined in Ethereum so that blocks that are part of forks are not considered. Provided that clocks are synchronized, we can say that for some transaction t, $t_{broadcast}$ is the timestamp when the transaction t was initially broadcast, $t_{included}$ is the timestamp of the block creation that included t and $latency_t = t_{included} - t_{broadcast}$, where $t_{included} > t_{broadcast} > 0$.

Unfortunately, we will never know the real value of $t_{broadcast}$ unless the transaction was issued by a node that we controlled. The reason is that the transaction must propagate from the originating node through the peer-to-peer network and the clocks are not perfectly synchronized. We can however approximate latency using the earliest known time for $t_{broadcast}$. Approximating our definition from above, we can say for some transaction t, $t_{received}$ is the earliest timestamp when transaction t is received by some of our nodes, hence:

$$latency_t \approx t_{included} - t_{received}.$$

4.2 Experimental settings

For these experiments, we used two datasets: The first dataset, labelled Geth Data, includes the time at which each transaction was relayed, and the second dataset, labelled Blockchain Data, includes the time at which each transaction was included.

The Geth Data included the timestamps of a transaction when the transaction was relayed in Ethereum using the `geth` client between 2017-06-24 00:00:00 and 2017-07-18 00:00:00. This period includes 19 ICOs: Dao.casino (BET), Pillar (PLR), Mothership (MSP), Blocktix (TIX), TrueFlip (TFL), EOS (EOS), Binance Coin (BNB), InsureX (IXT), CoinDash (CDT), Press.One (PRS), Tezos (XTZ), Nimiq (NET), Polybius (PLBT), Rialto (XRL), Santiment (SAN), Starta (STA), OpenANX (OAX), OmiseGO (OMG), EncryptoTel (ETT) that raised a total of more than $700M [17]. Each time a node received a transaction, it recorded the transaction hash, the $t_{received}$ timestamp, and the IP address of the node that relayed the transaction. This dataset contained 66,472,214 transactions, which is significantly larger than the actual number of transactions included in the blockchain for that period. There are two reasons for this. Firstly, this dataset contained many duplicates of the same transaction as transactions are relayed multiple times in Ethereum. Secondly, this dataset included many transactions that were either never included in a block or part of a fork and not visibly included in the blockchain.

The Blockchain Data were obtained via the blockchain company Infura. They provide a publicly accessible interface to the internal API of a `geth` node. Using this service, we extracted transaction and block data from the Ethereum blockchain for a 24-day period between 24/06/17 00:00:00 - 18/07/17 00:00:00.

In order to determine transaction latency, the datasets were combined by removing duplicate transactions and selecting only the earliest of these timestamps from the Geth Data and matching, using its hash, each transaction in the Blockchain Data to the transaction in the Geth Data.

4.3 The significant variation of latency depending on request time

Figure 2 (left) plots the latency of every transaction committed in the Ethereum blockchain throughout the period observed. Note that the y-axis uses a log scale, indicating that the latency of a transaction varies exponentially depending on the time it was issued. This variation provides a disappointing initial impression of performance. Ideally, latency would not vary by orders of magnitude and should be independent of the time when the transaction was issued. It is noteworthy that Figure 2 (left) does not show the distribution of transaction latency which can give a misleading representation of performance due to outliers.

Figure 2 Ethereum transaction latency and empirical cumulative distribution function

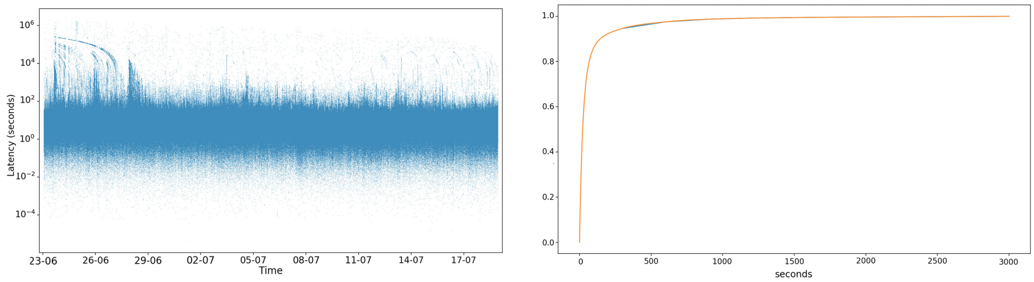


Table 2 shows that the median transaction latency observed was 22.13 seconds. Consider that the average block time for the period was 17.63 seconds. This means that over half of all transactions issued at depth i in the blockchain were included by the block at depth $i+2$. This observation significantly improves the initial impression given by Figure 2. However, while median performance is strong, the peaks that are seen in Figure 2 (left) are also quantified in the table. The latency data is extremely positively skewed above the 90th percentile.

Table 2 Ethereum transaction latency distribution

Percentile	10	25	50	75	90	95	99
Latency (seconds)	2.81	8.40	22.13	54.51	158.83	379.22	2854.31

Figure 2 (right) depicts the empirical cumulative distribution function obtained from the empirical study in order to visualise the skew of the latency data. This chart shows there is a point of inflexion in the distribution of latency around the 90th percentile.

The distribution of transaction latency poses an obvious question, why does the shape of the curve changes drastically above the 90^{th} percentile. In other words, what is different about this group of transactions that makes them take significantly longer to be included in a block?

4.4 On the minor impact of transaction fees on latency

The first thing to consider when attempting to explain latency is the transaction fee. All transactions in Ethereum specify a gas price, representing the price the user is willing to pay the miner for each computational step. Since the gas price is at the discretion of the user, perhaps the shape of the data can be explained by the transaction fee. If the gas price is too low, the total transaction fee may not provide enough of an incentive for the miner to include it. In order to examine this hypothesis, we compare the fees paid by transactions that are in the fastest 90% (latency < 158.83 seconds) with the fees paid by transactions in the slowest 10% (latency > 158.83 seconds).

Table 3 Comparison of fee distribution between fastest and slowest transactions

	25	50	75	90	95	99
Fastest 90% of latency (majority)	5.00^{14}	1.27^{15}	3.90^{15}	8.00^{15}	1.14^{16}	3.15^{16}
Slowest 10% of latency (outliers)	6.60^{14}	1.80^{15}	3.16^{15}	5.10^{15}	8.00^{16}	3.30^{16}

Table 3 shows that the transaction fees paid by the slow outliers are generally higher than in the fast group, except for the 75th and 90th percentiles. This means that these transactions were generally paying a higher fee but experiencing substantially worse latency. Initially, these results seemed counterintuitive. We know that users in a blockchain expect their latency to be correlated with the transaction fee they pay, but these results challenge this assumption. We thus calculated the covariance between the fee and the latency and obtain: $covariance(fee, latency) = -1.453 \times 10^{17}$.

From the covariance we can derive that there is a negative correlation between fee and latency, indicating, as we expect, that the fee is inversely related to the latency:

$$correlation(fee, latency) = -0.0001606.$$

This correlation suggests however that there is very little causal relationship between the transaction fee and latency. Recall that transaction fees in blockchains are supposed to allow a user to incentivise the miner of a block to include a transaction. This statistic challenged the common assumption of users in a blockchain that they can significantly impact the latency of their transaction through the level of the transaction fee.

5 The Impact of Supply and Demand on Latency

The weak correlation between fee and latency raises the question of what is the dominant factor in determining transaction latency. In this section, we study the relation between the supply and demand and how it affects latency. In particular, we study the Ethereum blockchain over the same period as the transactions were issued, where supply increased significantly and deduce the relationship between supply, demand and latency.

5.1 Supply side - gas limit

We are trying to explain why some transactions take significantly longer to be included than others. Our first attempt considering only transaction fees did not only fail to do so, it suggested that the fee itself may be insignificant. In order to try understand these strange results, we now consider the theoretical bounds of the Ethereum network. Recall that in proof-of-work blockchains the only way transactions are included is when a new block is mined:

$$capacity\text{-}throughput = \frac{block\text{-}gas\text{-}limit}{block\text{-}interval}.$$

The maximum number of transactions able to be included in a block is determined by the gas limit, since the Ethereum protocol targets a constant block interval by modifying the difficulty of proof-of-work. In particular, Ethereum's yellow paper [21] states the gas limit H_l of the block must satisfy the following relation: $H_l < P(H)_{H_l} + \frac{P(H)_{H_l}}{1024}$, $H_l > P(H)_{H_l} + \frac{P(H)_{H_l}}{1024}$, where $P(H)_{H_l}$ is the gas limit of the parent block. This mechanism was designed to allow the gas limit to evolve slowly over time to adapt to changes in the mining environment [21]. By allowing each miner to vote independently of one another, Ethereum attempts to avoid some of the centralisation risks in mining by ensuring larger miners are unable to quickly change the gas limit and therefore exclude smaller miners. In effect, Ethereum deliberately makes the gas limit inflexible over shorter periods of time. This means that at any single point in time, Ethereum effectively has a constant maximum throughput. Below we define the maximum number of transactions that can be committed per minute in Ethereum.

We start by taking the median transaction size observed in the study: Median Transaction Gas = 90,000. We can then approximate the maximum number of transactions per block b in terms of the gas limit and median transaction size:

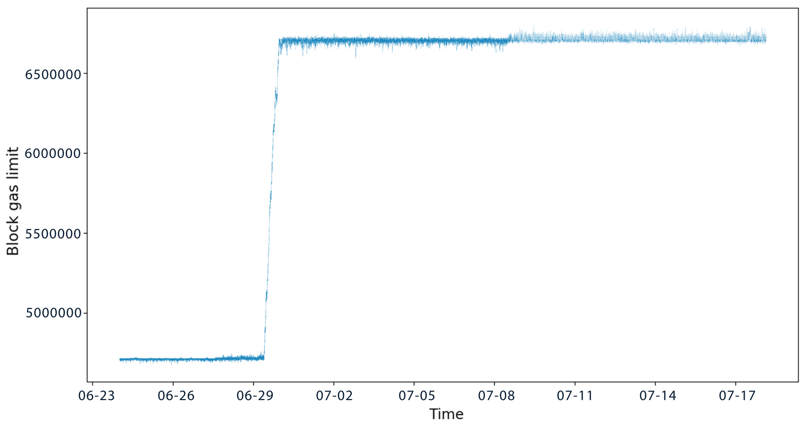
$$max\text{-}transactions_b = block\text{-}gas\text{-}limit/90000. \tag{1}$$

The average block interval throughout the study allows us to determine the Average Block Time = 17.63 seconds, and Blocks (Per Minute) = 3.40. Finally, we can derive an approximation for the maximum number of transactions that can be included per minute: $capacity\text{-}throughput = 3.4 \times gas\text{-}limit/90000$ transactions/minute.

5.2 Raising the gas limit

Before substituting the gas limit we need, however, to consider a significant event that occurred throughout the study.

Figure 3 Ethereum block gas limit



While it was explained that over the short term the gas limit can change only slightly, there was a significant shift observed during the study, shown in Figure 3. On the 29th of June 2017 miners in Ethereum began consistently voting up the gas limit to alleviate congestion from the increased demand being placed on the network. The result was that the gas limit increased by about 43%. Revising our earlier definition, we can include the average gas limit for each day in the study

$$\textit{block-gas-limit} = \begin{cases} 4711978 & \textit{if Date} < 29\ \textit{June}\ 2017, \\ 5348530 & \textit{if Date} = 29\ \textit{June}\ 2017, \\ 6711349 & \textit{if Date} > 30\ \textit{June}\ 2017. \end{cases}$$

Substituting these block gas limits into Eq. (1) yields the following bounds for the maximum number of transactions that can be committed per minute in Ethereum:

$$\textit{capacity-throughput} = \begin{cases} 178 & \textit{if Date} < 29\ \textit{June}\ 2017, \\ 202 & \textit{if Date} = 29\ \textit{June}\ 2017, \\ 253 & \textit{if Date} > 30\ \textit{June}\ 2017. \end{cases}$$

We now have approximated how many transactions can be included per minute given the relevant gas limit. Essentially these results mean that for the given day, if the number of transactions issued per minute is below the threshold, there should be a strong causal relationship between the transaction fee and the latency. However, it is now necessary to determine the demand placed on the network each minute in order to see the imbalances that occur between demand and supply.

5.3 Demand side – dynamic fluctuations

The Ethereum network has been live since July 30 2015, but there has been a significant increase in popularity of blockchains and cryptocurrency in 2017. This can be seen on the Etherscan website that shows the number of transactions per day, growing from under 50,000 in early 2017 to over 300,000 during the study [9]. Compounding this increased demand has been the hundreds of ICO on Ethereum on the first half of 2017. Throughout the study there were several significant ICOs such as TenX on the 24th of June, which raised 200,000 Ether in around 7 minutes [17].

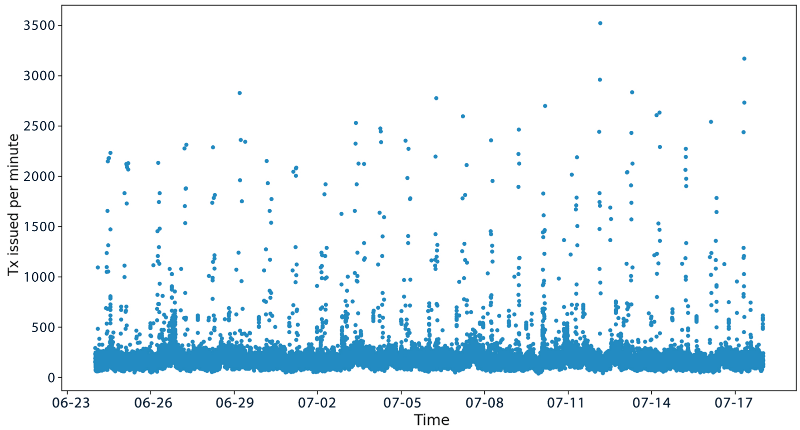
Figure 4 depicts the number of transactions issued per minute from June to the beginning of July 2017. It appears that there are many minutes where the number of transactions issued significantly exceeds the maximum throughput achievable at that point in time, as discussed in the previous section.

Table 4 Distribution of transactions per minute

Percentile	25	50	75	90	95	99
Transactions (Per Minute)	130	161	199	244	286	684

Table 4 shows the distribution of this data. Recall that 253 is the maximum possible number of transactions that could be committed per minute after the gas limit raise. This means that almost 10% of the time there were more transactions being issued than could be committed. This statistic is very closely aligned with the empirical cumulative distribution graph of latency in Figure 2 (right), with the shape of that graph changing sharply at the 90th percentile.

Figure 4 Ethereum transactions issued per minute



5.4 Median latency per minute

With the understanding of the relationship between demand and supply in Ethereum, we now revisit the latency results. Figure 5 shows the median transaction latency per minute in Ethereum during the period. This graph confirms our initial observation that the Ethereum blockchain typically confirms transactions within approximately 2 blocks. However, as a consequence of rapid fluctuations in demand and a relatively static supply, there are noticeable periods where median latency increases exponentially.

Figure 5 Ethereum transaction median latency per minute

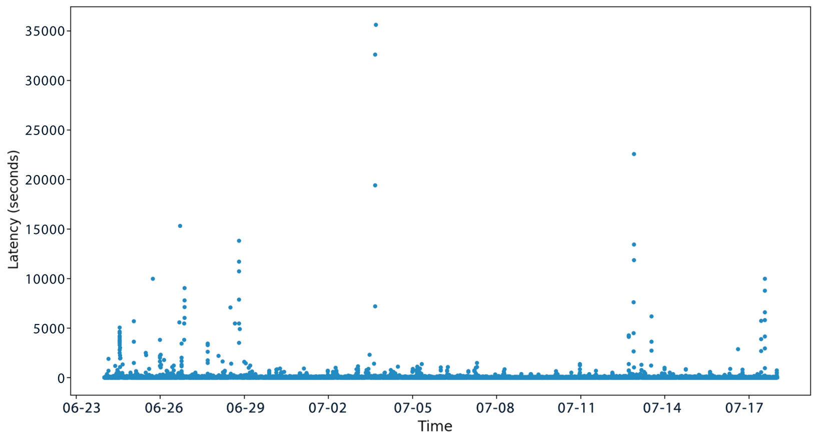


Table 5 Distribution of median latency per minute

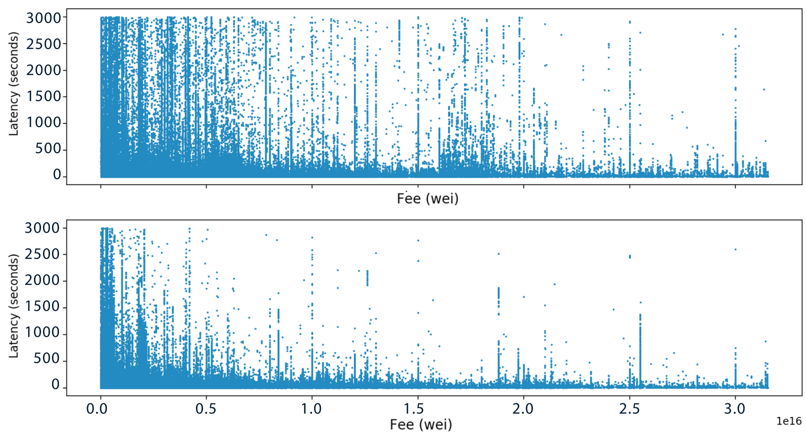
Percentile	25	50	75	90	95	99
Median Latency (Per Minute)	12.07	19.22	33.88	60.48	87.17	294.47

To recap, we have demonstrated that in Ethereum, the demand is constantly changing and hard to predict, but the supply (how many transactions can be included in the blockchain) is relatively static. This is one of the fundamental performance challenges for Ethereum. The effects of this problem on the users of the blockchain have been relatively insignificant until 2017 where the number of users participating in Ethereum started increasing rapidly.

5.5 Latency vs fee: before and after the gas limit raise

Our work so far has focused on identifying which factors are significant in determining latency in Ethereum. It appears that the transaction fee becomes insignificant in comparison to the overall demand and supply of Ethereum. Fortunately in our study there was a noticeable shift in the gas limit that allows us to analyse the latency vs fee relationship before and after this shift in supply. Figure 6 indicates that the relationship between fees and latency becomes more hyperbolic after the increase, representing the effect of increased Ethereum performance capabilities. The hyperbolic shape more accurately represent the relationship between fee and latency since our data is asymptotic, neither fee nor latency is ever equal to 0.

Figure 6 Latency vs fee: before and after the gas limit raise



6 Related Work

6.1 Proof-of-work and capacity throughput

In a first work [18], Sompolinsky and Zohar analysed the impact that high transaction throughput levels have on the level of security in the Bitcoin protocol. They analysed Bitcoin's longest chain selection method of reaching consensus. Their first contribution was to show that as transaction throughput increases, structural weaknesses in the longest-chain approach makes the network vulnerable to attackers with less computational resources [18].

In a subsequent work, Sompolinsky and Zohar [19] propose the Greedy Heaviest-Observed Sub-Tree (GHOST) consensus algorithm as a means of maintaining the security guarantees while increasing throughput. GHOST trades the longest-chain principle for selecting the heaviest subtree at each fork in the chain. This modification ensures that the work performed by honest nodes is incorporated by the network, even if their blocks do not appear in the final chain. This modification allows the network to deal with the inevitable increase in forks that increased throughput levels cause. They show that while GHOST can scale with the longest-chain algorithm in terms of block creation rate, the primary benefit is a constant security threshold as opposed to exponential decreases in the longest-chain algorithm [19].

6.2 Network and capacity throughput

GHOST helps reduce the security vulnerabilities that forks cause. However, forks still represent a significant weakness in the security of the blockchain as experimented by Natoli and Gramoli [16]. Fundamentally, the underlying network of the blockchain needs to be improved in order to reduce the chance of forks and the vulnerabilities that result from it.

Decker and Wattenhofer [6] analysed information propagation throughout the Bitcoin network in an attempt to determine the primary cause of forks in the blockchain. Their research focuses primarily on identifying improvements in the way the network communicates, by modifying the logical structure of the Bitcoin network. Their motivation is to reduce the number of forks in the blockchain and therefore reduce the decrease in network efficiency that is caused by forks. They identify three significant improvements to the current method of propagation in Bitcoin.

They contend that by dividing the block verification process (that occurs when a node receives a new block) into two components, the initial difficulty check and the validation of transactions, allows for a significant increase in propagation speed. After a node completes the difficulty check and hence verifies the proof-of-work, it can retransmit the block to its peers, before attempting transaction validation. The proposed gain is significant because the majority of work resides in the validation of transactions, whereas verifying proof-of-work is a trivial process [1]. They assert that this modification does not increase the risk of malicious behaviour because producing an invalid block with proof-of-work is just as hard as producing a valid block.

They also suggest that nodes can immediately forward all incoming messages to other nodes, even before actually receiving the block, in an attempt to reduce the round trip time between nodes. While they admit that this does allow an attacker to arbitrarily announce non-existent blocks, attackers are already able to flood the network with fake transactions, and therefore there is no reduction in security. Finally, they suggest that the most significant improvement can be gained by minimising the distance between any two nodes. This can be done by increasing the number of connections that each node maintains, effectively reducing the number of times messages need to be relayed between nodes [6].

6.3 Other approaches to increase the capacity throughput

Decker and Wattenhofer note that the above improvements, while valuable, do little to address what they contend are fundamental structural problems with the network. In a more recent paper, they put forward an entirely different network structure with duplex micropayment channels. They claim that this structure allows vastly superior scalability by deferring to the blockchain for initial setup of a payment channel and conflict resolution, while handling all transactions through the channel itself [7]. Another piece of work by Lewenberg, Sompolinsky and Zohar introduces the inclusion of off-chain blocks into the Bitcoin network. The consequences are similar to that of the GHOST algorithm whereby increased throughput can be achieved, however they also prove that they payoff for weak miners is increased [12].

Kiayias and Panagiotakos [11] also consider the tradeoffs between security and speed, however they extend on the above work by considering multiple blockchains. They introduce a new generic blockchain property, called chain growth, in order to express the minimum rate at which chains of honest parties grow. They derive this property as an extension of their previous work in which they isolate the backbone of the Bitcoin protocol, a useful framework for analysing blockchain fundamentals [10].

The underlying issue we identify here is the security-performance tradeoff that has left blockchains incapable of providing both high security and throughput to its users. Crain et al. [3] recently designed DBFT, a leader-less consensus algorithm to cope with this tradeoff. The algorithm is deterministic and does not assume synchrony, hence guaranteeing that no disagreements can occur, even when the network is behaving badly due to misconfigurations, natural disasters or attacks. The algorithm is also democratic in that it leverages the bandwidth of multiple links rather than relying on the classic leader-based design that is subject to bottlenecks at the leader network interface. The Red Belly Blockchain builds upon this algorithm and an efficient verification sharding protocol to offer a throughput that keeps increasing when increasing the number of consensus participants, typically to hundreds of low-end consensus participant machines [4].

6.4 Transaction fees

Möser and Böhme analyse transaction fees in the Bitcoin blockchain in an attempt to understand the economic and technical components [13]. They examine transactions empirically, in order to determine how fees change over time, and how impatient users incentivise miners to include their transactions. They suggest that the instability of fees over time is a consequence of the protocol failing to provide a mechanism by which users and miners can coordinate to set fair prices. Interestingly, the paper suggests that this issue is not necessarily dangerous as long as mining rewards still dominate the composition of income for miners. This statement raises an interesting question in relation to high throughput which is generally associated with decreasing block rewards [5]. Some information regarding the relation between gas price and confirmation time in Ethereum can be found on the publicly available Eth gas station website[3], however, it does not relate this information to the latency of transactions.

7 Conclusion

In this paper, we analysed the parameters that impact transaction latency in Ethereum. The popularity of ICOs in 2017 has created a competitive environment for users wishing to purchase tokens. While buyers generously incentivised miners to include their transactions, the supply and demand of the service was the predominant factor determining latency and inclusion. For future work, we would like to reproduce the analysis for more recent periods as the Ethereum protocol and network keep evolving.

Acknowledgements

We wish to thank our colleagues, Ralph Holz and Nate Wiggins, for sharing the Geth Data. We also benefited from the publicly available datasets of Infura and Etherscan. This research is in part supported under Australian Research Council Discovery Projects funding scheme (project number 180104030) entitled “Taipan: A Blockchain with Democratic Consensus and Validated Contracts”. Vincent Gramoli is a Future Fellow of the Australian Research Council.

[3] `https://ethgasstation.info/`.

References

1. T Bamert, C Decker, L Elsen, R Wattenhofer, and S Welten. Have a snack, pay with bitcoins. In *IEEE International Conference on Peer-to-Peer Computing*, 2013.
2. V Buterin. *A Next-Generation Smart Contract and Decentralized Application Platform*, 2013. URL: `https://github.com/ethereum/wiki/wiki/White-Paper`.
3. T Crain, V Gramoli, M Larrea, and M Raynal. DBFT: Efficient leaderless byzantine consensus and its applications to blockchains. In *Proceedings of the 17th IEEE International Symposium on Network Computing and Applications (NCA'18)*, pages 1–8, 2018.
4. T Crain, C Natoli, and V Gramoli. Evaluating the Red Belly Blockchain. Technical Report 1812.11747, arXiv, 2018.
5. K Croman, C Decker, I Eyal, A. E Gencer, A Juels, A Kosba, and D Song. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125, 2016.
6. C Decker and R Wattenhofer. Information propagation in the bitcoin network. In *IEEE International Conference on Peer-to-Peer Computing*, pages 1–10, 2013.
7. C Decker and R Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *Symposium on Self-Stabilizing Systems*, pages 3–18, 2015.
8. Etherchain. Etherchain application programming interface block data, 2017. URL: `https://etherchain.org/documentation/api/`.
9. Etherscan. Etherscan chart library, 2017. URL: `https://etherscan.io/charts`.
10. J Garay, A Kiayias, and N Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310, 2015.
11. A Kiayias and G Panagiotakos. Speed-security tradeoffs in blockchain protocols. *IACR Cryptology ePrint Archive*, 2015.
12. Y Lewenberg, Y Sompolinsky, and A Zohar. Inclusive blockchain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 528–547, 2015.
13. M Möser and R Böhme. Trends, tips, tolls: A longitudinal study of bitcoin transaction fees. In *International Conference on Financial Cryptography and Data Security*, pages 19–33, 2015.
14. S Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. `doi:https://bitcoin.org/bitcoin.pdf`.
15. C Natoli and V Gramoli. The blockchain anomaly. In *Proceedings of the 15th IEEE Int'l Symp. on Network Computing and Applications*, pages 310–317, 2016.
16. C Natoli and V Gramoli. The balance attack or why forkable blockchains are ill-suited for consortium. In *IEEE/IFIP DSN'17*, Jun 2017.
17. Smith and Crown. ICOs, token sales, crowdsales, 2017. URL: `https://www.smithandcrown.com/icos/`.
18. Y Sompolinsky and A Zohar. Accelerating bitcoin's transaction processing. fast money grows on trees, not chains. *IACR Cryptology ePrint Archive*, pages 507–527, 2015.
19. Y Sompolinsky and A Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527, 2015.
20. I Weber, V Gramoli, M Staples, A Ponomarev, R Holz, A B. Tran, and P Rimba. On availability for blockchain-based systems. In *Proceedings of the 36th International Symposium on Reliable Distributed Systems (SRDS'17)*, pages 64–73. IEEE, Sep 2017.
21. G Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, 2014. `doi:http://gavwood.com/paper.pdf`.