Learning Blockchain Delays: A Queueing Theory Approach

Saulo Ricci
CipherTrace
Los Gatos, CA, USA

Eduardo Ferreira
UFRJ
Rio de Janeiro, RJ, Brazil

Daniel Sadoc Menasché
UFRJ
Rio de Janeiro, RJ, Brazil

Artur Ziviani
LNCC
Petrópolis, RJ, Brazil

José Eduardo Souza
UFJF
Juiz de Fora, MG, Brazil

Alex Borges Vieira
UFJF
Juiz de Fora, MG, Brazil

ABSTRACT

Despite the growing interest in cryptocurrencies, the delays incurred to confirm transactions are one of the factors that hamper the wide adoption of systems such as Bitcoin. Bitcoin transactions usually are confirmed in short periods (minutes), but still much larger than conventional credit card systems (seconds). In this work, we propose a framework encompassing machine learning and a queueing theory model to (i) identify which transactions will be confirmed; and (ii) characterize the confirmation time of confirmed transactions. The proposed queueing theory model accounts for factors such as the activity time of blocks and the mean time between transactions. We parameterize the model for transactions that are confirmed within minutes, suggesting that its integration into a more general framework is a step towards building scalability to Bitcoin.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*Queueing Theory*

General Terms

Theory, Performance

Keywords

Bitcoin, queueing theory, delay

1. INTRODUCTION

Cryptocurrencies are in vogue. In particular, Bitcoin stands out as one of the *de facto* solutions for the digital market and is responsible for millions of US dollars that circulate across its nodes daily. Despite its worldwide adoption, some of Bitcoin's fundamentals have not been fully uncovered yet.

In this paper, we present a characterization of Bitcoin transaction properties primarily taking into account aspects related to transaction delays. This work sheds light on some features related to Bitcoin transactions that are harvested from public databases as well as from monitors deployed worldwide. Our monitors collect all transactions broadcasted through the Bitcoin network, and we register the timestamp when the transactions first hit one of our monitors. This data is then used to characterize the probability of confirmation of transactions and to assess features that tell whether a transaction is likely to be confirmed. In addition, they are also used to parameterize a queueing model that yields the confirmation time of transactions. In short, this work intends to answer the following questions:

1) Will a transaction be confirmed after seen by one of our monitors?

2) What are the important factors that impact the confirmation time of transactions?

We assume that a transaction that remains for at least 24 hours in the system without confirmation will remain unconfirmed. To answer the first question, we propose a classification framework, which relies on machine learning tools to classify transactions as soon as they are issued. Then, we rely on a queueing theoretic model to characterize the delay of transactions expected to be confirmed.

We summarize our contributions as follows:

Transaction classification framework. We propose a transaction classification framework to early distinguish between transactions that are likely to be accepted. We envision that this framework is instrumental in the use of Bitcoin in an online fashion. In particular, our framework relies on thresholds, which can be parameterized through queuing models.

Queueing theory model. We propose a queuing theory model to characterize the time to confirm transactions. Our model breaks down the confirmation time into multiple components and establishes the relationship among those. We parameterize our model using data collected using our monitors, over a period of three weeks.

2. TERMINOLOGY

Next, we introduce the basic terminology adopted through this work.

Transaction: records movement of cryptocurrency between users.

Block: a set of transactions. In Bitcoin, a block is created roughly every ten minutes.

Transaction fee: non-mandatory fee offered by the user who issued the transaction to the successful miner.

Blockchain: the longest chain of blocks of transactions. The blockchain is constantly evolving. In this work, we determine the reference blockchain *in retrospect*, defining it as the longest chain involving the transactions of interest one day after those transactions are issued.

Mined transaction: a transaction included in a block with a valid hash.

Confirmed transaction: a mined transaction that is part of the longest chain. We use the terms *confirmed*

Symposium on Cryptocurrency Analysis (SOCCA) 2018 Toulouse, France

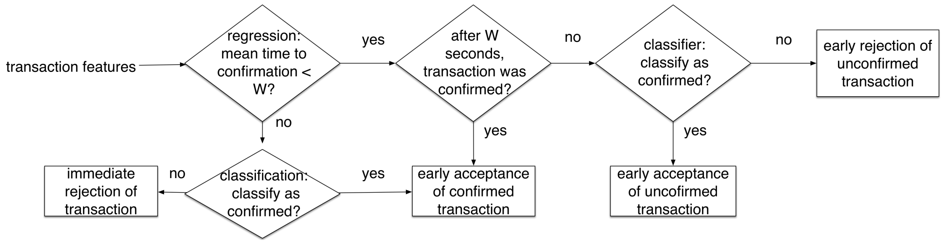

Figure 1: Framework for transaction confirmation.

transaction, *validated transaction* and *mined transaction* interchangeably when characterizing the delay, as we ignore orphan chains (see definition of blockchain above).

Confirmation delay: time between transaction is issued and its insertion into a block in the blockchain.

Additional confirmations: every time a new block is created, all transactions in the new blockchain are automatically reconfirmed. It is common for six (re)confirmations to be required to consider that a transaction is fully confirmed (i.e., chances of reverting the transaction are negligible). Then, a transaction is (fully) confirmed when it is six blocks deep in the blockchain, assuming no off-chain payment mechanism and no simplified payment verification (SPV). Nonetheless, in this note we ignore such additional (re)confirmations when defining the confirmation delay.

3. TRANSACTION CLASSIFICATION

The proposed framework for transaction classification is presented in Figure 1. Given the features of the new transaction, a regression yields the mean confirmation time for such transaction. If the predicted confirmation time is below a given threshold, it is worth waiting for that time to check if the transaction will be naturally validated. The threshold W must account for system requirements (e.g., users' patience) as well as for the sensitivity of the delay with respect to multiple parameters (which can be assessed, for instance, using the proposed queueing theory model). Then, if the transaction is confirmed by the expected time, it produces an early confirmation. Otherwise, a classifier is adopted to identify if the transaction will be confirmed, *given that it already lasted W seconds in the system.* Such a classifier must be trained using transactions that already experienced a system delay equal (or close) to the pre-specified delay. The output of the classifier may lead to an early rejection/acceptance.

Note that if the expected time to confirm the transaction was originally greater than W, the classifier, used to decide whether the transaction should be immediately rejected, must account for all the data available so far. In addition, the costs of rejecting or accepting a transaction should be balanced, as a false positive (classifying a transaction as acceptable, in case it is never confirmed) may incur different costs than a false negative (classifying a transaction as not-acceptable, in case it is eventually confirmed).

3.1 Why a queueing model?

A queueing model serves to determine a baseline (reference value) for transaction delays as a function of the time between block generations and other metrics that can be immediately assessed using the blockchain history. Such baseline is helpful to set thresholds needed to determine if a transaction should be early accepted or not. More broadly, a queueing model is helpful to get further insights into how different factors impact transaction delays.

Table 1: Notation.

variable	description
S	block active time
D	transaction delay
T	time between transactions
B	time between block confirmations
M	number of active blocks

4. QUEUEING MODEL

Next, we propose a simple queueing model to capture the relationship between different quantities that together impact delays in a blockchain system. Figure 2 illustrates the key quantities of interest in our queueing model.

4.1 Workload

We consider a flow of transactions that arrive at rate λ to the system. Each transaction is part of a block. Blocks are mined at rate λ_B blocks/s, and each block comprises an average of τ transactions. We assume a system under equilibrium, neglecting unconfirmed transactions in our queueing model. Therefore, there is a flow of $\lambda = \lambda_B \tau$ transactions served per time unit.

4.2 Active blocks and block confirmation times

Then, we introduce the notion of the active time of a block. The active time of a block initiates when the first transaction to eventually be confirmed in a block is issued, and finishes when that block is confirmed. We denote by S the active time of a block, and by M the number of active blocks in the system. The active time of block i is denoted by S_i. It follows from Little's law that,

$$E(M) = \lambda_B E(S). \tag{1}$$

Let B denote the time between block confirmations. It follows that $E(B) = 1/\lambda_B$. Let also T be the inter-transaction time. We denote by $T_{i,j}$ the time until the j-th transaction arrival, following the $(j-1)$-th arrival of a transaction served in the i-th block.

4.3 Transaction delays and delay model

We denote by D the delay incurred by a typical user transaction. In particular, $D_{i,j}$ is the delay incurred by the j-th transaction that was confirmed in block i. The delay is assumed to be assessed by a user that samples the network at an arbitrary point in time, chosen uniformly at random (allowing us to use renewal theory arguments).

We assume that system inspection occurs uniformly at random. Then, our model for the delay experienced by a user is given by

$$E(D) = \alpha E(B) + E(B_r), \tag{2}$$

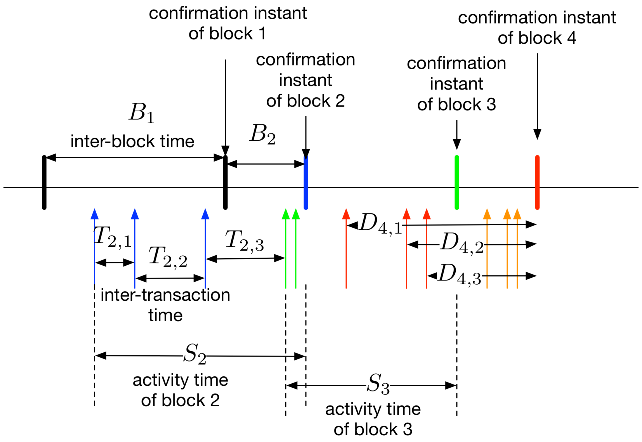

Figure 2: Basic quantities of queueing theory model.

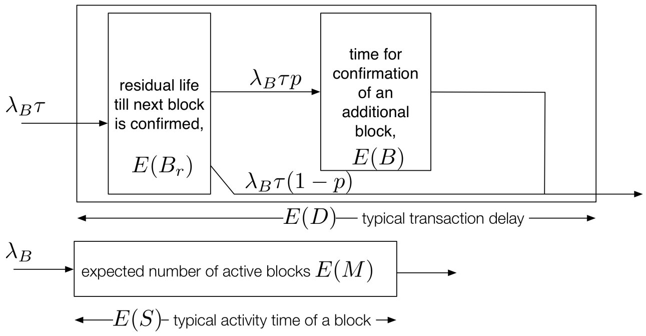

Figure 3: Dissecting the transaction delay.

where

$$E(B_r) = \frac{E(B^2)}{2E(B)}. \tag{3}$$

$E(B_r)$ stands for the residual time of the inter-block time. Parameter α denotes the expected number of blocks that a user needs to wait before having a transaction confirmed. Recall that we assume that a transaction is confirmed whenever its block is added into the blockchain (see Section 2). Note that $\alpha E(B)$ is the time that a user needs to wait in addition to the residual life of the block encountered in the system when the user first arrives.

Equation (2) is inspired by the standard M/G/1 model. It is similar, for instance, to equation (23.2) in [3]. In the M/G/1 model, the coefficient of the residual service time equals the system utilization, which captures the probability that the system is busy at the instant of a new arrival. In our model, in contrast, a block is always being mined at the system (i.e., the system is always busy and stability is an open research challenge [10]).

In our measurements, we found that α is typically smaller than 1. Therefore, we provide an alternative interpretation of the model as shown in Figure 3. According to this figure, all users need to wait for the residual life of a block. In addition, with probability p, users need to wait for an additional block to be confirmed. Then, $E(D) = pE(B) + E(B_r)$, where $p = \alpha$.

Discussion and extensions. Note that this model captures the typical time to confirm a transaction, which is on the order of minutes. Certain transactions take hours to be confirmed (and certain transactions are never confirmed). The time to confirm a transaction depends on its fee, priorities, and byte size. In this simplified and preliminary modeling attempt, we neglected all such aspects, which we leave as subject for future work.

In addition, note that we established a relationship between $E(B)$, $E(M)$, and $E(S)$ through Equation (1) and a relationship between $E(D)$ and $E(B)$ through Equation (2). As next steps, we plan to relate these two expressions, looking at the interplay between block-delays and transaction-delays.

5. MEASUREMENT RESULTS

We conducted three measurement campaigns, during three –almost stable– consecutive months of Bitcoin operation. Each measurement campaigns corresponds to an interval:[1]

- Interval 0: **[794 blocks]** Fri, 19 Aug 2016 04:55:20 GMT – Wed, 24 Aug 2016 08:39:57 GMT (5d 3h 44m 37s)
- Interval 1: **[2020 blocks]** Fri, 23 Sep 2016 04:33:36 GMT – Thu, 06 Oct 2016 06:58:28 GMT (13d 2h 24m 52s)
- Interval 2: **[295 blocks]** Sun, 30 Oct 2016 18:07:44 GMT – Tue, 01 Nov 2016 16:36:55 GMT (1d 22h 29m 11s)

5.1 Transaction confirmation: features

Table 2 shows the number of observed and unconfirmed transactions per measurement interval. It also shows that some blocks contained transactions not observed by any of our monitors (either due to glitches in our measurement infrastructure, or because such transactions were broadcasted before our measurements started).

Table 2: Transaction confirmations.

	not included in blocks (observed by monitors)	not observed by monitors (observed in blocks only)	transactions observed by monitors
0	10,186	39,754	1,103,800
1	27,374	99,015	2,961,400
2	7,938	17,318	451,000

Table 3 reports the fees offered by confirmed and not confirmed transactions. Clearly, the average fee of not confirmed transactions is much smaller than the fees corresponding to confirmed transactions. This well known fact, combined with other transaction features, may be used to infer if a transaction is likely to be confirmed.

We found that the size of transactions (in bytes) is not a good discriminator to determine whether a transaction will be accepted. In contrast, in addition to fees, we also found that the value of the transaction itself is another important factor: the larger the value of a transaction, the higher the chances of it being confirmed (in part, due to a correlation between fees and transaction value). For confirmed transactions, the queueing model presented in the next section characterizes the mean confirmation time.

5.2 Delay characterization: queueing model

We have observed that roughly 90% of all transactions are confirmed within 33 minutes. Table 4 shows the time between transactions, measured in seconds, accounting for all

---

[1] Scripts are available contacting the authors or through Github: `https://github.com/edufgf/BitcoinInspector`.

Table 3: Transaction fees.

	confirmed	not confirmed
0	23848.33	14597.13
1	21950.87	16479.83
2	23974.65	16750.34

transactions in the network, $E(T)$, and only for transactions in the same block. The latter is larger than the former, as there may be multiple arrivals inbetween transactions that end up being confirmed in a given block. Such transactions are not taken into account when computing the mean time between transactions in that given block.

Table 4: Time between transactions/active blocks.

	$E(T)$	intra-block transaction time	$E(M)$	λ_B	$E(S)$
0	0.38	1.16	18.89	0.00175	10799.16
1	0.36	1.45	18.51	0.00177	10459.12
2	0.35	1.27	15.53	0.00173	8930.34

Table 4 also shows the quantities in Equation (1). Note that λ_B is very similar across the three intervals ($1/\lambda_B \approx$ 570 seconds, which is the average time to mine a block, taking the difficulty adjustment into account). Nonetheless, the mean active time of blocks varied significantly. It is worth also pointing that to generate this table we neglected certain blocks (around 15 blocks per interval) for which we did not collect any transaction through our monitors (due to glitches in the measurements). As future work, we plan to investigate if the average number of active blocks in the system converges to an asymptotic value if we consider long enough intervals.

Table 5 reports the delays experienced at the three considered intervals. We note that the value of α is different across the three time segments. This is because the mean time to include a transaction in a block was larger in the second interval, but similar in the first and last intervals. We are currently investigating the reasons for the different transaction inclusion times. It is worth noting that on average transactions do not need to wait much more than the residual time of a block confirmation to be inserted into the blockchain.

Table 5: Parameterization of delay.

	$E(D)$	$E(B)$	$E(B^2)$	$E(B_r)$	$V(B)$	α
0	797.40	535.79	547193.96	510.64	260121.88	0.53
1	937.15	530.89	534901.08	503.78	253057.54	0.81
2	798.81	546.70	521800.65	477.22	222922.39	0.58

6. RELATED WORK

There is a growing literature on queuing models for blockchain systems [1, 2, 4–9]. Nonetheless, none of those works analyzed the active time of blocks, or characterized the residual time of times between blocks and its connections to confirmation delays.

In [6] the authors propose a priority queue to analyze Bitcoin priority mechanisms. In our work, in contrast, we have not accounted for priorities. Still, we were able to show that our model can explain some of the basic dynamics of Bitcoin.

In [2] the authors emphasize the problems caused by the fact that delays are typically not used to prioritize transactions in the Bitcoin pool of unconfirmed transactions (mempool). Then, the authors argue in favor of new prioritization mechanisms. Extending the models proposed in this paper, we believe that complementary insights can be derived about how priorities impact delays, and we leave that as subject for future work.

In [7, 9] the authors also adopt a queueing theory methodology to study transaction delays. Such works are similar in spirit to ours, as the broad goal is to characterize delays. Nonetheless, the tools used by [7, 9] involve matrix geometric methods, whereas in this paper our focus is on simple models, e.g., using renewal arguments, and their connection to broader machine learning frameworks.

7. CONCLUSION

In this note we have introduced a simple queueing theory model to characterize the delay experienced by Bitcoin transactions. The proposed model relates the delay of transactions with the time between block confirmations, and can be easily parameterized using real measurements. Our measurements indicate that typical users experience a delay which is slightly larger than the residual lifetime of the time between block generations.

Acknowledgements: We acknowledge the partial support of CNPq, FAPEMIG, FAPERJ, and FAPESP.

8. REFERENCES

[1] Biais, B., Bisiere, C., Bouvard, M., and Casamatta, C. The blockchain folk theorem, 2018. Toulouse School of Economics.

[2] Easley, D., O'Hara, M., and Basu, S. From mining to markets: The evolution of Bitcoin transaction fees. *SSRN* (2017).

[3] Harchol-Balter, M. *Performance modeling and design of computer systems: queueing theory in action.* Cambridge University Press, 2013.

[4] Hu, Y., Manzoor, A., Ekparinya, P., Liyanage, M., Thilakarathna, K., Jourjon, G., Seneviratne, A., and Ylianttila, M. E. A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain. *arXiv:1801.10295* (2018).

[5] Huberman, G., Leshno, J. D., and Moallemi, C. C. Monopoly without a monopolist. *SSRN* (2017).

[6] Kasahara, S., and Kawahara, J. Priority mechanism of Bitcoin and its effect on transaction-confirmation process. *arXiv* (2016).

[7] Kawase, Y., and Kasahara, S. Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism. In *Queueing Theory and Network Applications* (2017), Springer, pp. 75–88.

[8] Koops, D. Predicting the confirmation time of bitcoin transactions. *arXiv preprint arXiv:1809.10596* (2018).

[9] Li, Q.-L., Ma, J.-Y., and Chang, Y.-X. Blockchain queueing theory. *arXiv:1808.01795* (2018).

[10] Pontiveros, B. B. F., Norvill, R., and State, R. Monitoring the transaction selection policy of Bitcoin mining pools. In *NOMS* (2018), IEEE, pp. 1–6.