Estimating the distribution of block sizes as a function of average transaction rates

Matt Simpson[1], Eli Dourado[2]

[1]M. Simpson is a postdoctoral fellow in statistics at the University of Missouri
[2]E. Dourado is a research fellow at the Mercatus Center at George Mason University and director of its Technology Policy Program, as well as a doctoral candidate in economics at George Mason University

December 23, 2015

Abstract

As Bitcoin's popularity rises, so does the number of transactions per block. Existing Bitcoin rules limit the number of transactions per block to 1 megabytes worth. Recent average transaction rates have been about 40 percent of that limit. But because Bitcoin blocks are discovered according to a stochastic Poisson process, some blocks may reach the limit even if the average is well below the limit. This paper models the percentage of blocks that reach the 1MB limit under various average transaction rates. In our most basic model, we derive a formula for that percentage as a function of constant transaction rates. We also estimate how that percentage changes when transaction rates exhibit minute-to-minute variability along specified, empirically defensible distributions.

1 Introduction

The optimal maximum size of Bitcoin blocks is an area of active controversy. When the maximum size is set at a high level, mining centralization will increase due to higher resource demands, and the network becomes vulnerable to selfish mining attacks (Eyal and Sirer, 2014).When the maximum size is too small, there is inadequate space to accommodate the volume of transactions that users demand, resulting in higher fees, wait times, and potentially lower rates of adoption, which affects security through the price of Bitcoin. The issue may be with us for some time; even assuming that current disagreements over raising the 1MB cap are resolved, if the Bitcoin network keeps growing and as computing power increases, the issue will need to be re-litigated from time to time. One proposal, BIP 100, would subject the decision to a continuous voting process by miners indefinitely into the future.

A better understanding of the relationship between the rate of Bitcoin transactions and block sizes is needed to inform this debate. Bitcoin blocks are discovered according to a Poisson process, directly implying that times between block discoveries are distributed exponentially. If transaction rates (in bytes per second) were constant, this would imply that block sizes too would be distributed exponentially. As the exponential distribution is well understood and has a closed-form cumulative density function, it would be trivial to evaluate the probability of blocks filling up as a function of the average transaction rate.

In fact, transaction rates are not constant. Block sizes are therefore not distributed exponentially, but according to some other distribution. In this paper, we empirically estimate the distribution of block sizes and calculate the probability that a block will reach the upper bound as a function of the average transaction rate. We find that the actual distribution of transaction data is more likely to result in a binding maximum than an exponential distribution would across the spectrum of average transaction rates, although the biggest effect is at the relatively low end.

2 Verifying the variability of transaction rates

Let τ index time in minutes, and $N(\tau)$ denote the number of blocks discovered by time τ. Then for an increment $\delta > 0$, the number of blocks discovered between time τ and time $\tau + \delta$ is Poisson distributed, that is $N(\tau + \delta) - N(\tau) \sim Poi(\delta/10)$. Let $k = 1, 2, \ldots$ denote each of the blocks, in order. Then the amount of time between blocks is exponentially distributed, i.e. $\delta_k \equiv \tau_k - \tau_{k-1} \overset{iid}{\sim} Exp(1/10)$ for $k = 1, 2, \ldots, K$ where τ_k is the arrival time of the k'th block. We use the rate parameterization of the exponential distribution so that if $x \sim Exp(\lambda)$ it has the density $p(x) = \lambda e^{-x\lambda}$ for $x > 0$ and $\lambda > 0$ with mean $1/\lambda$. The longer δ_k the more time for transactions to add up before being added to the block chain as part of block k.

There is always some chance that a given block will fill up since arrival times are stochastic. Specifically, for the k'th block this probability is

$$P\left(x_{\tau_k} - x_{\tau_{k-1}} > 1\right).$$

This probability depends crucially on the process that x_τ follows. If we assume that x_τ is deterministic with a constant transaction rate of γ MB/min, then

$$x_{\tau_k} - x_{\tau_{k-1}} = \gamma\delta_k \overset{ind}{\sim} Exp(1/10\gamma)$$

and the probability of a full block is available in closed form as

$$P\left(x_{\tau_k} - x_{\tau_{k-1}} > 1\right) = \int_1^\infty \frac{1}{10\gamma} e^{-x/10\gamma} dx = e^{-1/10\gamma}.$$

Clearly a constant transaction rate does not capture Bitcoin's behavior since it has been growing in popularity. With a non-constant and potentially stochastic transaction rate, the probability of a block filling up is no longer necessarily available in closed form. Using a Monte Carlo simulation of N draws of the block sizes from the process x_τ, we can approximate the probability of block k filling up with

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left\{x^{(i)}_{\tau^{(i)}_k} - x^{(i)}_{\tau^{(i)}_{k-1}} > 1\right\}$$

where i indexes the Monte Carlo simulations of each of the random quantities inside the indicator function. If we let γ denote the mean transaction rate, which is the same as the actual transaction rate in the constant, deterministic case, we can alter γ to see how the probability of a full block changes under a variety of regimes.

A key question is whether the assumption of a constant transaction rate is approximately correct on a short enough time-scale. In order to test this assumption we fit a gamma distribution to the block sizes of various pieces of the block chain. We use the shape-rate parameterization of the gamma distribution so that if $x \sim G(\alpha, \beta)$, it has the density

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-x\beta}$$

for $x > 0$, $\alpha > 0$, and $\beta > 0$ with mean α/β and variance α/β^2. The gamma function in the denominator is defined as $\Gamma(x) = \int_0^\infty x^{t-1} e^{-x} dx$. Since the $G(1, \beta)$ and $Exp(\beta)$ distributions are identical, if we find that $\alpha \neq 1$ then transaction rates cannot be constant. We focus on the most recent 5000 blocks in our database ending with block 303151 — we are in the process of downloading the entire chain. For priors we will assume $\log \alpha \sim N(0, 10)$ and $\beta \sim G(1, 1)$. This prior on α implies that the prior median is $\alpha = 1$ so that the null hypothesis is favored relatively strongly by the prior. We fit the model to all 5000 blocks, the first 1000 blocks, the second 1000 blocks, etc., and the first 500 blocks, the second 500 blocks, etc. The results are in Table 1.

For the full 5000 blocks, the posterior mean of α is 1.34 and the lower bound of the 95% credible interval is 1.29, so we have strong evidence that α is not one and that a constant transaction rate is not appropriate. This is not surprising since Bitcoin usage has gone up over time. A block is around 10 minutes on average, so 5000 blocks takes a little over a month on average. It is possible that on a short enough time-scale, transaction rates are approximately constant. So we fit the gamma distribution

	Mean	2.5%	97.5%
$n = 5000$	1.04	1.01	1.08
$n = 1000,\ g = 1$	0.93	0.86	1.00
$n = 1000,\ g = 2$	0.93	0.86	1.00
$n = 1000,\ g = 3$	1.00	0.92	1.08
$n = 1000,\ g = 4$	1.04	0.96	1.12
$n = 1000,\ g = 5$	1.52	1.41	1.64
$n = 500,\ g = 1$	0.89	0.80	0.99
$n = 500,\ g = 2$	0.97	0.86	1.08
$n = 500,\ g = 3$	0.92	0.83	1.02
$n = 500,\ g = 4$	0.94	0.84	1.04
$n = 500,\ g = 5$	1.26	1.12	1.40
$n = 500,\ g = 6$	0.83	0.74	0.92
$n = 500,\ g = 7$	1.04	0.93	1.16
$n = 500,\ g = 8$	1.04	0.92	1.15
$n = 500,\ g = 9$	1.40	1.25	1.57
$n = 500,\ g = 10$	1.67	1.49	1.86

Table 1: Posterior mean and 95% credible intervals for α from fitting a gamma distribution to block sizes of the last 5000 blocks in our database. For sample sizes less than 5000, blocks were broken into groups of equal size and the model was fit separately to each group. Sample size is indicated by n, group id by g. Each model was fit in RStan (Stan Development Team, 2015a,b) using 4 chains, each with 4000 iterations, 2000 of which used for warmup/burn in.

to 1000 blocks at a time — representing about a week — by breaking the original 5000 blocks into five groups. For each of these groups the 95% credible interval does not contain one — the lowest lower bound is 1.17. Similarly for groups of 500 blocks, each representing about 3.5 days, the lowest lower bound is about 1.08. So we can safely conclude that the transaction rate is not constant. A likely reason for this is seasonality in transaction rates — Bitcoin trades are probably more likely to happen during certain times of day than others, based on when relevant markets are open and when the bulk of Bitcoin users are awake. The *mean* transaction rate averaging over e.g. seasonality is probably approximately constant for time-scales this small, however. In any case, transaction rates are nonconstant even on short time-scales, which motivates modeling them as such.

3 Modeling the distribution of block sizes

The first part of the model is determined for us since we know $\delta_k \overset{iid}{\sim} Exp(1/10)$ by the design of Bitcoin, but determining how transaction data is distributed requires more work. Crucial constraints on the models we can fit come from the data obtained from the block chain. Since Bitcoin is a distributed network without a home node, there is no official time-stamp for when each block arrives on the chain. Instead, the block miner's local time is recorded. Two constraints on the reported time that the block was mined are built into the Bitcoin protocol. First, any block submitted to the chain with a local time less than the median of local times of the last 11 blocks is rejected. Second, any block with a local time greater than network adjusted time plus two hours is rejected, where network adjusted time is defined as the local time plus the median difference between local time and the time at all nodes connected to the local node with a maximum offset of 70 minutes. Because of the nature of the blocks' time-stamps, occasionally a block has a time-stamp which is earlier than one or more previous blocks in the chain. Essentially, we have measurement error on the measurement times, and the lower and upper bounds for valid time-stamps define lower and upper bounds for the measurement error distribution. Unfortunately network adjusted time is not recorded in the block chain, so the upper bound is not observed, but the lower bound is at our disposal.

The second major constraint that the block chain imposes on us is that transaction data is only measured once a block arrives. So at random measurement times which we only observe with error, we observe the amount of transaction data since the previous measurement time — though we observe this without error. These two constraints motivate a class of models for modeling Bitcoin transaction data. Once again, let τ_k denote the actual time of the arrival of the k'th block, δ_k the amount of time between block k and block $k-1$ so that $\tau_k = \tau_{k-1} + \delta_k$, and now let t_k denote the observed arrival time in the block chain. Further, let x_τ denote the amount data in the entire chain at time τ so that $x_{\tau_k} - x_{\tau_{k-1}}$ denotes the amount of data in block k. Now suppose the amount of transaction data in MBs is distributed according to some nondecreasing stochastic process. Most of the interesting modeling choices come from modeling this stochastic process. Conventional Brownian motion is inappropriate since the increments should be nonnegative while geometric Brownian motion is a poor choice since it cannot start from zero. Instead, we consider nonnegative infinite activity Lévy processes, often called subordinators, with closed form solutions for x_τ. The following discussion is brief – see e.g. Barndorff-Nielsen and Shephard (2012) for a fuller treatment. The stochastic process $\{x_\tau : \tau > 0\}$ is a Lévy process if it is continuous time and has independent and stationary increments with $x_0 = 0$. Brownian motion is the most common example. A nondecreasing example is the gamma process where $x_{\tau+\delta} - x_\tau \sim G(\alpha\delta, \beta)$ for any time τ and positive increment δ. The gamma process is called an infinite activity process since for any time τ and increment $\delta > 0$, $x_{\tau+\delta} - x_\tau > 0$.

The gamma process is particularly convenient since the distribution of $x_{\tau+\delta} - x_\tau$ and x_τ are both known in closed form as a consequence of the fact that if $x_1 \sim G(\alpha_1, \beta)$ independent of $x_2 \sim G(\alpha_2, \beta)$, then $x_1 + x_2 \sim G(\alpha_1 + \alpha_2, \beta)$. More generally the gamma distribution is infinitely divisible so that if $x \sim G(\alpha, \beta)$ then for any integer $n > 0$ there exist iid random variables $y_1, y_2, \ldots, y_n$ such that $y_1 + y_2 + \cdots + y_n$ and x have the same distribution. Often Lévy processes are defined by considering an infinitely divisible distribution for x_1, but the distribution of x_τ for any $\tau > 0$ need not be from the same class as x_1. The inverse Gaussian process is another Lévy process for which the distribution of x_τ is in the same class as the distribution for x_1 for all $\tau > 0$. The inverse Gaussian density is

$$p(x) = \left[\frac{\lambda}{2\pi x^3}\right]^{1/2} \exp\left[\frac{-\lambda(x-\mu)^2}{2\mu^2 x}\right]$$

for $x > 0$ with mean parameter $\mu > 0$, shape parameter $\lambda > 0$, and variance μ^3/λ. Then increments of the inverse Gaussian process are inverse Gaussian distributed, viz. $x_{\tau+\delta} - x_\tau \sim IG(\mu\delta, \lambda\delta^2)$.

Since the observation times are observed with error, it is attractive that we can write down the distribution of $x_{\tau+\delta} - x_\tau$ in closed form for any τ and $\delta > 0$. This simplifies fitting the model because it allows us to write down a relatively simple exact likelihood function, facilitating both maximum likelihood estimation and Markov chain Monte Carlo methods. The parameters of the gamma or inverse Gaussian distributions can also be further modeled as stochastic processes in order to capture e.g. seasonal variation or stochastic volatility, though most reasonable models on these parameters will not allow for closed form distributions of the transaction data increments. We will use the gamma process since it is widely available in statistical packages and in particular is available in Stan (Stan Development Team, 2015b), unlike the main alternative, the inverse Gaussian.

On the data level we have measurement error with a lower bound, motivating a class of models of the form

$$t_{k+1}|t_{1:k}, \tau_{1:K} \overset{ind}{\sim} p(t|\tau_{k+1}, \phi)\mathbb{1}\left\{t > m_{k+1})\right\}$$

where $p(t|\tau, \phi)$ is a class of probability densities on $t > 0$ with centrality parameter τ and dispersion parameter ϕ, and $m_k = \text{median}(t_{k-0:10})$. For example τ could be the mean or the mode and ϕ the variance or precision. This interpretation of these parameter applies for the *untruncated* distribution $p(t|\tau, \phi)$, but not necessarily for the conditional distribution of t_k. A natural choice here is another gamma distribution, since the untruncated distribution still must be non-negative. However the gamma causes numerical problems at times since it forces the latent measurement times, τ_k to enter the gamma function. A similar distribution that avoids numerical problems is the lognormal, which we will use.

Combining these pieces we obtain the following model. For block $k = 1, 2, \ldots, K$

$$\begin{aligned}
t_k|t_{1:(k-1)}, \tau_{1:K} &\sim LN(\log(\tau_k) - \sigma^2/2, \sigma)\mathbb{1}(t_k > m_k) \\
x_{\tau_{k-1}+\delta_k} - x_{\tau_{k-1}}|\tau_{1:K} &\overset{ind}{\sim} G(\beta\gamma\delta_k, \beta) \\
\delta_k = \tau_k - \tau_{k-1} &\overset{iid}{\sim} Exp(1/10)
\end{aligned}$$

where $x \sim LN(\mu, \sigma)$ means $\log(x) \sim N(\mu, \sigma)$. To complete the model we assume σ, β, and γ are independent in the prior with $\sigma \sim half - Cauchy(0, 2.5)$, $\beta \sim G(1, 1)$, and $\gamma \sim LN(0, 10)$. Note that the half-Cauchy distribution is the Cauchy truncated to the positive real line. Table 2 contains the results of fitting the model.

	Mean	SD	SE(Mean)	2.5%	25%	50%	75%	97.5%
γ	0.0240	0.0004	0.0000	0.0232	0.0237	0.0240	0.0243	0.0248
β	15.0951	0.9228	0.0103	13.3725	14.4461	15.0760	15.6962	16.9591
σ	0.0024	0.0001	0.0000	0.0023	0.0023	0.0024	0.0024	0.0025

Table 2: Results of fitting the measurement error model with stochastic measurement times. We use RStan (Stan Development Team, 2015a,b) to fit the model, obtaining four chains each of 4000 iterations, 2000 of which used for warmup/burn in.

Given an approximate simulation of the posterior distribution of the model parameters, we can use Monte Carlo simulation to estimate the probability of a block filling up in a number of ways. The simplest way is to compute parameter estimates based on the posterior, e.g. posterior means or posterior medians, then estimate the probability of a block filling up by simulating the model conditional on these parameter values. In order to take into account increased Bitcoin usage, we can manually increase the mean transaction data rate parameter, γ, while holding other parameters constant. In order to take into account seasonality, we can make γ change in a predictable, seasonal pattern. This approach directly specifies the mean transaction rate at γ, but because of the mean-variance relationship of the gamma distribution it also indirectly specifies the variance of the transaction rate at $\gamma/\hat{\beta}$ where $\hat{\beta}$ is an estimate of β.

This approach has two major shortfalls: 1) it ignores parameter uncertainty, and 2) it assumes that the other parameters will remain constant as γ increases. In particular, 2) implies that the variance of transaction rates is always proportional to the mean, i.e. it is γ/β. We can mitigate 1) while making 2) if not worse, different. Specifically, given access to an approximate simulation of the posterior of the model parameters, we can simulate the model for each draw from the posterior in order to estimate the probability of a block filling up — this is the posterior predictive distribution (Gelman et al., 2014). In order to allow for different transaction rates in this simulation, we can set γ at a particular value while using the full posterior of the other model parameters, or even better we can shift the posterior of γ so that the posterior mean of the shifted posterior is the desired value. This approach takes into account parameter uncertainty but assumes that our uncertainty about the other parameters would be unchanged if the location of γ's distribution were different. In other words, it still assumes that the variance of transaction rates is still γ/β, but now γ and β are also stochastic with β distributed according to its marginal posterior distribution and $\gamma|\beta$ distributed according to a shifted version of its conditional posterior given β. Essentially this assumption requires that if we were in a higher transaction rate regime, our priors would have been different such that the posterior distribution we would have obtained would be the shifted posterior we construct. Explicitly constructing this prior would be involved and maybe intractable or even theoretically impossible. Instead we estimate the probability of a block filling using all three approaches, but acknowledge that the estimates become less credible for values of γ or $\mathrm{E}[\gamma]$ further away from their estimated values.

Table 3 contains estimates of the probability of a filled block under a variety of assumptions, using parameter estimates of the posterior distribution of the model using 1000 contiguous blocks ending with block 303151. Probabilities in the first row were computed exactly using the cdf of the exponential distribution. Probabilities in the second and third rows were computed by simulating 800,000 blocks from the model at the specified values for γ and β, where $\beta^{(0.5)}$ denotes the posterior median for β. Probabilities in the fourth row were computed by simulating 100 blocks from the model for each of the 8000 draws of from the posterior of β and with γ fixed at the specified values. Probabilities in the fifth and final row were computed by simulating 100 blocks for each of the 8000 draws from the posterior of β and the shifted posterior of γ, where a draw from the shifted posterior of γ is a draw from the posterior plus the difference between the specified $\mathrm{E}[\gamma]$ and γ's posterior mean.

Under a fixed mean transaction rate, the probability of a full block increases as β decreases, especially for smaller mean transaction rates. For larger mean transaction rates there is little effect on the probability the next block is full from decreasing β. This is expected since increasing the spread of

$E[\gamma] = \hat{\gamma} =$	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
$\beta = \infty$	0.0067	0.0357	0.0821	0.1353	0.1889	0.2397	0.2865	0.3292
$\beta = 3 \times \beta^{(0.5)}$	0.0082	0.0387	0.0853	0.1383	0.1915	0.2420	0.2874	0.3293
$\beta = \beta^{(0.5)}$	0.0117	0.0446	0.0917	0.1435	0.1958	0.2446	0.2881	0.3300
$\beta \sim p(\beta \vert x, t)$	0.0119	0.0452	0.0919	0.1437	0.1954	0.2438	0.2892	0.3304
$(\beta, \gamma) \sim p(\beta, \gamma \vert x, t)$	0.0118	0.0448	0.0916	0.1433	0.1955	0.2442	0.2896	0.3301

Table 3: Probability of a full block under a variety of parameter values — γ is the mean transaction rate and β is a dispersion parameter so that the variance of transaction rates is γ/β. In the first row the transaction rate is constant at $\hat{\gamma}$, in the second through fourth rows the transaction rate is stochastic with fixed mean $\hat{\gamma}$, and in the final row the transaction rate is stochastic and unknown with mean $\hat{\gamma}$.

transaction rates while holding the mean constant increases the variance of block sizes. When block sizes are on average much smaller than 1 MB, increasing the variance of block sizes puts significantly more probability mass above 1 MB, but when the mean is near but still below 1 MB it has a relatively small effect. What is unexpected is that no matter what the expected mean transaction rate is, the probability of a block filling up is largely unaffected by taking into account parameter uncertainty in β or in both β and γ, which can be seen from the final three rows of Table 3.

4 Discussion

Our analysis does not resolve the contentious block size issue, but it is an important input into weighing the tradeoff between supporting higher transaction rates and the security issues created by larger blocks. For example, at the existing 1 MB maximum, as average transaction rates increase from 500 KB/block to 600 KB/block, an additional five percent or so of blocks will hit the limit. These full blocks signal that not all transactions are being included in the first block after their transmission. In addition, a greater understanding of the dynamics of full blocks versus the average transaction rates could be used to help clients set fees appropriately, as a function of the observed rate of transactions over the last several blocks and the existing maximum block size. Creating a more dynamic fee market is another way, besides simply increasing the maximum block size, to scale the network effectively.

References

Barndorff-Nielsen, O. E. and Shephard, N. (2012). Basics of Lévy processes. *Draft Chapter from a book by the authors on Lévy Driven Volatility Models.*

Eyal, I. and Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. Taylor & Francis.

Stan Development Team (2015a). RStan: the R interface to Stan, version 2.8.0.

Stan Development Team (2015b). Stan: A C++ library for probability and sampling, version 2.8.0.