

# How to Charge Lightning\*

Simina Brânzei<sup>†</sup>

Erel Segal-Halevi<sup>‡</sup>

Aviv Zohar<sup>§</sup>

January 1, 2018

## Abstract

Off-chain transaction channels represent one of the leading techniques to scale the transaction throughput in cryptocurrencies. However, the economic effect of transaction channels on the system has not been explored much until now.

We study the economics of Bitcoin transaction channels, and present a framework for an economic analysis of the lightning network and its effect on transaction fees on the blockchain. Our framework allows us to reason about different patterns of demand for transactions and different topologies of the lightning network, and to derive the resulting fees for transacting both on and off the blockchain.

Our initial results indicate that while the lightning network does allow for a substantially higher number of transactions to pass through the system, it does not necessarily provide higher fees to miners, and as a result may in fact lead to lower participation in mining within the system.

## 1 Introduction

A main approach to solve the scalability problem in Bitcoin is to use off-chain transaction channels that allow parties to transfer funds while communicating directly, and only occasionally to settle on the blockchain. The recent deployment of SegWit, a solution to transaction malleability (among other benefits) opens the path for better constructions of off-chain transaction channels. While transaction channels themselves are limited to exchanges between pairs of individuals, further developments like the lightning network [PD16] allow to route payments over longer paths and thus can allow the construction of a well connected network of payment channels that can be used to transfer money quickly and with relatively little interaction with the blockchain. For additional discussion of micropayment channels and scalability, see, e.g., [HS, DW15, CDE<sup>+</sup>16].

One of the key unknowns regarding fast payment networks is the economic effect that they will have on the Bitcoin fee market. If the blockchain is used less often, fees to miners are paid less frequently and competition for space in blocks declines. Bitcoin’s security depends heavily on having a large amount of computational power invested in solving proof-of-work puzzles, making it hard for attackers to double spend or censor transactions in the currency. As the block reward in Bitcoin declines (halving every four years), the reliance on fees increases and these must suffice to pay for enough mining by honest participants.

---

\*The authors are in alphabetical order. This paper was presented at Scaling Bitcoin 2017: <https://stanford2017.scalingbitcoin.org/>.

<sup>†</sup>Hebrew University of Jerusalem, Israel. E-mail: [simina.branzei@gmail.com](mailto:simina.branzei@gmail.com).

<sup>‡</sup>Ariel University, Israel. E-mail: [erelsgl@gmail.com](mailto:erelsgl@gmail.com).

<sup>§</sup>Hebrew University of Jerusalem, Israel. E-mail: [avivz@cs.huji.ac.il](mailto:avivz@cs.huji.ac.il).

In this work we explore the economics of Bitcoin transaction channels, and in particular the economic equilibrium that results from the introduction of fast-payment networks to the Bitcoin ecosystem. Our main contribution is a theoretical framework in which one can reason about the usage of payment channels and the cost of committing records to the blockchain. We explore different topologies of payment channels and find the market equilibrium that dictates (among other things) the fees that will be collected by miners, the transactions that pass through the lightning network or directly through the blockchain, and the transactions that do not take place (e.g., micro transactions for which the fees are too high on both alternatives).

**Summary of our findings** While our findings strongly depend on the assumptions we have made regarding the distribution of payment sizes and willingness to pay fees, we generally find that the revenue obtained by miners can sometimes be lower when lightning networks are deployed compared to when they are not, unless extremely large numbers of participants take part in transacting (there too, results depend on the distributional assumptions). Naturally, the addition of payment channels does indeed result in very high transaction throughputs in the system overall.

The implications for Bitcoin are that the revenue from fee payments alone might be insufficient to support the security of the system. Still, we caution that our results should be taken with a grain of salt: we account for very simple topologies but hope our initial exploration will inspire further exploration of similar models.

## 1.1 An overview of our approach

The first step taken when modeling the effects of different transaction methods is to select a model for the demand for transactions — a model specifying which participants want to send money, to whom, what amount is transferred, and how much fee the sender is willing to pay to complete the transfer.

A second aspect that needs to be determined is the topology of payment channels that is set in the system: which pairs of participants choose to establish channels between them, and how much funding is dedicated to each such channel. This is crucial in determining the lifetime of the channel given different use patterns.

Finally, with this information in hand we set out to compute the demand for blockchain records. Such demand stems from two main sources: The establishment and settlement of existing lightning channels, and direct transfers that occur on the blockchain. Given that the block size is limited, the daily supply of new transaction records is fixed, and we are thus able to compute the market equilibrium fee for records.

**The mechanics of transaction channels** Transaction channels are typically established by locking funds using a single blockchain transaction. The channel state is then updated by the two participants by exchanging transactions that update the division of funds from the locked amount. These transactions are not typically transmitted to the blockchain. Each update represents a new division of funds and usually only the last transaction is committed to finalize the transfer. The transactions exchanged by the two participants on each channel are set up so that if one of the participants disappears or tries to take funds that are not his (e.g., by placing a transaction that represents an old state on the blockchain). The other participant can recover his funds or even punish his counterpart by taking all funds in the channel. For the purpose of this work, we assume all channels are established and settled cooperatively, as we aim to consider the expected behavior of the market under “normal” circumstances.

We additionally assume that channels can be settled and reopened with a single transaction, which may be of larger size in the network. We assume the parties that transfer money back and forth do so according to a random process, and that they therefore occasionally end up in a state where all funds in the channel are directed to one of the users. In this case, the channel can only be used to transfer money in one direction and the channel must be reset or re-funded to allow flow in the other direction. Clearly, if the amount of liquidity in the channel is high this event will occur rarely. It is therefore of paramount importance to establish the typical amount of funding in each channel. Since liquidity that is locked within the channel represents money that is not invested elsewhere, we consider the cost of holding liquidity in the channel as the lost income from interest payments on this sum. Stated differently: we allow participants to borrow as much money as they want to fund their transaction channels, and the cost for such payments is simply the interest rate in the economy. This cost is the de facto limiting factor for the lightning network.

**Models for the demand for transactions** We explore two primary models for the demand for transactions. The first is a model in which participants are paired and only transact with their direct partner. While this is not a realistic depiction of the flow of money in an economy, it is in some sense a best-case setting for transaction channels, as no routing of payments is required by the system. The topology of channels in this case is also simple: just create channels between transacting pairs if it is profitable to do so. In this setting we explore several variants: one in which transactions occur with equal probability in each direction, and one in which transfer is asymmetrically biased in one direction. Another axis along which we vary our analysis is regarding the size of payments. We assume in one case that payments come from a uniform distribution, and in another case that they are derived from a power-law distribution (as is typically the case in real life data).

Our second model assumes that all participants may pay each other, and that payments occur between participants that are chosen uniformly at random. Here we focus on an analysis of a payment network that includes a single payment hub. The hub needs to maintain additional liquidity, but allows participants that are connected to it to route payments to everyone else. We find surprisingly that the results bear great similarity to the pairs model (except for extra payments for the additional liquidity).

**Implementation** The code used for our simulations can be found at: <https://github.com/erelsgl/bitcoin-simulations>.

## 2 Model

We analyze the market for *records* on the blockchain. A record is a part of a block, in which a single transaction is recorded. Each record has a market-price  $\phi$  [bitcoins-per-record], which is the mining-fee for a blockchain transaction. The market-price  $\phi$  is determined as a price in which the *supply* of records equals the *demand* for records.

The *supply* in our market is quite simple: the bitcoin protocol ensures that the supply of records per day is fixed. We denote this parameter by  $\tau$ . As of this writing, the value of this parameter is  $\tau \approx 288,000$  [records-per-day]. The total revenue of the miners, which is an important factor in the security of bitcoin, will be  $\tau \cdot \phi$  [bitcoins-per-day].

The *demand* is driven by the need of users to transfer money to other users. The demand is determined by the following sets of parameters:

- The number of times that user  $i$  wants to transfer money to user  $j$  per day is a Poisson random variable with mean value  $\lambda_{i,j}$  [transfers-per-day].
- The size of transfers from user  $i$  to user  $j$  is  $z_{i,j}$  [bitcoins-per-transfer]. This models the fact that some users do micro-transfers while others do bulk transfers. We will sometimes assume that  $z_{i,j}$  is drawn from a probability distribution such as uniform or power-law. We assume that the transfer-size  $z_{i,j}$  is constant for each pair, i.e., it is drawn randomly once for each pair and then remains fixed.
- The utility that user  $i$  gains from each transfer to user  $j$  is  $v_{i,j}$  [bitcoins-per-transfer]. We will often assume that the utility is proportional to the transfer size, i.e.,  $v_{i,j} = \beta z_{i,j}$ , where  $\beta$  is some constant fraction in  $(0, 1)$ .

Each time user  $i$  considers to transfer money to user  $j$ , he compares three options: blockchain transfer, lightning transfer, or no transfer at all. The user selects the option with the highest net gain. In the case of a blockchain transfer, the net gain is  $v_{i,j}$  minus the blockchain fee  $\phi$ . In the case of a lightning transfer, the net gain is  $v_{i,j}$  minus the *lightning fee*, which is derived from the cost of maintaining the lightning channel. In case of no transfer, the net gain is zero.

The lightning fee is derived from several parameters which determine the cost of using lightning:

- The number of blockchain records required for a channel-reset transaction, denoted by  $a$ . A reset transaction is slightly larger than a standard transaction, so  $a$  is a number between 1 and 2 [records]. Therefore the cost of resetting a lightning channel is  $a \cdot \phi$  [bitcoins].
- The (fixed) interest rate  $r$  [per day]. A user who wants to use lightning has to lock money in channels, and thus he has to pay an economic cost determined by  $r$ . This means that in general a user will not want to lock all his money in lightning channels; he will look for the optimal amount to lock such that the total cost (economic cost plus channel-reset cost) is minimized.

We study several special cases for the transfer matrix  $\lambda_{i,j}$ :

**Pairs:** the users are divided in pairs, e.g.  $(1, 2)$ ,  $(3, 4)$ , etc. All transfers are only inside each pair, i.e., for every  $i$ , only  $\lambda_{2i,2i-1}$  and  $\lambda_{2i-1,2i}$  are non-zero. This is in some sense the “best case” for lightning, since we need a channel only between user  $i$  to user  $j$ .

**Symmetric Uniform:**  $\lambda_{i,j} = \lambda$  for all  $i$  and  $j$ .

**Asymmetric Uniform:** for each pair  $i, j$ ,  $\lambda_{i,j} + \lambda_{j,i} = \ell$  and  $|\lambda_{i,j} - \lambda_{j,i}| = \Delta$ . I.e, the pairs are asymmetric, either user  $i$  transfers more money to user  $j$  or vice-versa.<sup>1</sup>

We also study several possible lightning topologies:

**Pairs:** each user  $2i$  has a channel only with user  $2i - 1$ .

**Star:** there is a single node (“bank”) which is connected to all users; all transfers go through the bank.

---

<sup>1</sup>note that in general, it is possible that some agents accumulate money endlessly while other agents spend money endlessly; this can be explained by assuming that they do some transfers outside bitcoin. Alternatively, we can assume a special topology in which all nodes have an even degree and have positive net transfer in exactly half their edges.

### 3 Analysis of a Single Channel

The basic building-block for our analyses is the analysis of a single channel between two users, Alice and Bob. We first analyze the expected channel lifetime for a given channel capacity and distribution of transfers, and then find the optimal channel capacity.

We checked two cases related to the transfer rates: in the symmetric case the transfer-rate from Alice to Bob equals the transfer-rate from Bob to Alice; in the asymmetric case the transfer-rates are different. For brevity, this paper presents only the results for the symmetric case; the results for the asymmetric case are qualitatively similar.

#### 3.1 Channel Lifetime

**Theorem 3.1** (Lifetime of Symmetric Channel). *Let Alice and Bob have a channel with  $w$  bitcoins, which they use to transfer single bitcoins to each other at a time. Each transfer consists of one coin sent by Alice with probability  $1/2$  and by Bob with probability  $1/2$ . Then the expected lifetime of the channel from the state where Alice has  $m$  bitcoins and Bob has  $w - m$  bitcoins is*

$$X_m = wm - m^2$$

*Proof.* Let  $X_m$  denote the expected channel lifetime from the state where Alice has  $m$  bitcoins. The recurrence relation can be derived from the asymmetric case, by substituting  $p = 0.5$ :

$$X_{m+1} = 2 \cdot X_m - X_{m-1} - 2$$

Note that  $X_0 = 0$ , since if Alice has zero in her account and wants to make a transfer, then the channel is reset. Similarly,  $X_w = 0$  corresponds to the case where Bob has zero in his account.

The associated homogeneous recurrence relation is  $X_{m+1}^h = 2 \cdot X_m^h - X_{m-1}^h$ , with characteristic equation  $x^2 - 2x + 1 = 0$ . The roots of the associated characteristic equation are  $x_1 = x_2 = 1$ , and so the general solution has the following form, where  $a, b$  are constants:

$$X_m^h = a + bm$$

Recall that in general, if  $s$  is a root with multiplicity  $q$  of the characteristic equation, there exists a general solution to the recurrence such that for some constants  $p_0 \dots p_t$ :

$$X_m^p = m^q \left( \sum_{k=0}^t p_k \cdot m^k \right) s^m \quad (1)$$

The particular solution, given by Equation 1, has  $q = 2$ ,  $s = 1$ , and  $t = 0$ , so  $X_m^p = vm^2$  for some constant  $v$ . Then

$$v \cdot (m+1)^2 = 2v \cdot m^2 - v \cdot (m-1)^2 - 2 \iff v = -1$$

The particular solution is

$$X_m^p = -m^2$$

Adding the homogeneous and non-homogeneous parts

$$X_m = X_m^h + X_m^p = a + bm - m^2$$

The boundary condition  $X_0 = 0$  gives  $a = 0$ , while  $X_w = 0$  gives  $b = w$ . Then for all  $n = 0 \dots w$ , the expected number of transfers made on the channel is:

$$X_m = wm - m^2$$

This completes the proof.  $\square$

**Corollary 3.2.** *Let Alice and Bob have a channel with  $w$  bitcoins, which they use to transfer single bitcoins to each other at a time. Both Alice and Bob transfer to each other using a Poisson process with mean  $\lambda_A = \lambda_B = \lambda$ . Then the expected number of days that the channel from the state where Alice has  $m$  bitcoins and Bob has  $w - m$  bitcoins is:*

$$\tilde{X}_m = \frac{wm - m^2}{2\lambda} \quad [days].$$

*Proof.* When the players send to each other with equal means, Alice sends to Bob a single unit with probability  $1/2$  and Bob sends to Alice with probability  $1/2$ . From Theorem 3.1, the expected lifetime from the state where Alice has  $m$  bitcoins is  $X_m = wm - m^2$ . To obtain a solution in days, we divide this by the mean number of transfers per day, which is  $2\lambda$ .  $\square$

### 3.2 Channel Optimal Initialization

Given the expressions for the channel lifetime, Alice and Bob can calculate the optimal way to initialize a channel with a fixed capacity  $w$ , i.e. the initial balance  $(a, b)$  that maximizes the expected lifetime of the channel.

**Lemma 3.3.** *The optimal initialization of a symmetric channel with capacity  $w$  is for both Alice and Bob to start with  $w/2$  bitcoins.*

*Proof.* Given that the expected channel lifetime from the state where Alice has  $m$  bitcoins is  $\frac{wm - m^2}{2\lambda}$ , the lifetime is maximized when  $m = w/2$ . Then an optimally initialized symmetric channel is expected to last for:  $T(w) = \frac{w^2}{8\lambda}$ .  $\square$

So far, we assumed that each transfer between Alice and Bob consists of a single coin. However, it is easy to generalize the results to a situation in which each transfer between Alice and Bob consists of  $z$  bitcoins, for some constant  $z$ . In this case, the channel funding  $w$  is given in *transfers*, i.e., the channel lifetime is  $T(w)$  when the channel is funded with  $z \cdot w$  bitcoins.

### 3.3 Channel Optimal Capacity

Alice and Bob have two options for performing a sequence of transactions, namely sending money to each other on the blockchain or through a channel in the lightning network. In the case of blockchain transactions, there will be a fixed fee  $\phi$  paid for each transaction, where  $\phi$  is equal to the price (in bitcoins) of a blockchain record.

If on the other hand Alice and Bob decide to use a lightning channel, they will incur an interest  $r$  on the money that is locked into the channel, in addition to the fee  $a \cdot \phi$  paid when the channel is reset and recorded on the blockchain. We assume that this cost is covered by charging a fixed fee per lightning-transaction. The fee is calculated such that the expected fee charged during the entire channel lifetime equals the total cost of maintaining the channel.

**Lemma 3.4.** *Let  $\phi$  be the blockchain mining fee and  $r$  the daily interest rate for locking money in a lightning channel. Suppose the transfers between Alice and Bob are Poisson processes with means  $\lambda_A, \lambda_B$  transfers-per-day respectively, with  $\ell := \lambda_A + \lambda_B$ . Each transfer is of  $z$  bitcoins, and they use a lightning channel with capacity  $w \cdot z$  bitcoins.*

*Then, to cover the maintenance costs of the channel, the fee per transaction should be:*

$$F(w) = \frac{wz \cdot (1+r)^{T(w)} + a\phi - wz}{T(w) \cdot \ell}, \quad [bitcoins/transaction]$$

*where  $T(w)$  is the expected channel lifetime (in days).*

*Proof.* The players pay interest on the quantity  $wz$  locked in the channel until the channel closes, i.e. for  $T(w)$  days in expectation. Since the interest is paid each day (rather than for each transaction), the cost due to the interest is:

$$\left( (1+r)^{T(w)} - 1 \right) \cdot wz$$

The users incur an additional cost of  $a\phi$  at the end for recording the channel balance on the blockchain.

The expected number of transactions during the channel lifetime is  $T(w) \cdot \ell$ , so the expected fee charged during the channel lifetime is  $F(w) \cdot T(w) \cdot \ell$ . This fee should cover the costs, so we should have:

$$\left( (1+r)^{T(w)} - 1 \right) \cdot wz + a \cdot w = F(w) \cdot T(w) \cdot \ell$$

from which the claim follows.  $\square$

**Lemma 3.5** (First-order approximation). *The first-order approximation of the function*

$$F(w) = \frac{\left( (1+r)^{T(w)} - 1 \right) \cdot wz + a\phi}{T(w) \cdot \ell},$$

that gives the expected fee per transaction, taken around  $r = 0$ , is:

$$\tilde{F}(w) = \frac{wzr}{\ell} + \frac{a\phi}{T(w) \cdot \ell},$$

where  $T(w)$  is the expected channel lifetime and is independent of the interest rate  $r$ .

*Proof.* Let  $g(r) = (1+r)^{T(w)}$ , where  $r \in \mathbb{R}^+$ . Then

$$F(w) = \frac{(g(r) - 1) \cdot wz + a\phi}{T(w) \cdot \ell}$$

Since  $T(w)$  is independent of  $r$ , the first derivative of  $g$  is  $g'(r) = T(w) \cdot (1+r)^{T(w)-1}$ . Then the first order approximation of  $g(r)$  around  $r_0 = 0$  is:

$$\tilde{g}(r) = g(0) + g'(0) \cdot (r - r_0) = 1 + r \cdot T(w)$$

Substituting  $\tilde{g}(r)$  for  $g(r)$  in  $F(w)$ , we obtain

$$\tilde{F}(w) = \frac{wzrT(w) + a\phi}{T(w) \cdot \ell}.$$

The error term for the estimation of  $g(r)$  is given by a function  $h(r)$  with the property that  $|h(r)| \leq \frac{Mr^2}{2}$ , where  $M$  is the maximum value taken by  $|g''(r)|$  on the interval  $[0, c]$ . We have that

$$g''(r) = T(w) (T(w) - 1) \cdot (1+r)^{T(w)-2}.$$

There are a few cases:

- $x \geq 2\sqrt{2\ell}$ :  $M_1 = \max_{r \in [0, c]} |g''(r)|$  is attained for  $r = 1$ , so we have that

$$M_1 = \left| T(w) \cdot (T(w) - 1) \cdot (1+c)^{T(w)-2} \right|.$$

- $x < 2\sqrt{2\ell}$ : the maximum is  $M_2 = |T(w)(T(w) - 1)|$ , attained for  $r = 0$ .

Take  $M = \max\{M_1, M_2\}$ .

Let  $R(w) = F(w) - \tilde{F}(w)$  be the error term in the approximation of  $F(w)$ . Then

$$|R(w)| = \left| \frac{wz}{T(w) \cdot \ell} (g(r) - \tilde{g}(r)) \right| = \frac{wz}{T(w) \cdot \ell} \cdot |h(r)| \leq \frac{2Mr^2}{x}$$

□

The cost of a transaction on the blockchain is  $\phi$ , thus performing transactions on the lightning network is profitable for Alice and Bob as long as  $F \leq \phi$ .

**Lemma 3.6** (First order approximation of symmetric channel capacity). *Let  $\phi$  be the blockchain mining fee and  $r$  the daily interest rate for locking money in a lightning channel. Suppose the transfers between Alice and Bob are Poisson processes with means  $\lambda_A = \lambda_B = \lambda$  transfers-per-day, with  $\ell := \lambda_A + \lambda_B = 2\lambda$ . Each transfer is of  $z$  bitcoins, and they use a lightning channel with capacity  $w \cdot z$  bitcoins.*

*Then, the first order approximation of the expected cost per transaction on the channel is  $\frac{zrw}{\ell} + \frac{4a\phi}{w^2}$ . The optimal channel capacity based on this approximation is  $w_{opt} = \sqrt[3]{\frac{8a\phi\ell}{zr}}$ .*

*Proof.* From Lemma 3.5, the first-order approximation of the lightning-fee  $F$  is:

$$\tilde{F}(w) = \frac{wzr}{\ell} + \frac{a\phi}{T(w) \cdot \ell}.$$

For a symmetric channel,  $T(w) = w^2/4\ell$  [days], so

$$\tilde{F}(w) = \frac{zrw}{\ell} + \frac{4a\phi}{w^2}$$

as claimed. The error term is given by:

$$|R(w)| = \frac{wz}{T(w) \cdot \ell} \cdot |h(r)| = \frac{4z}{w} \cdot |h(r)| \leq \frac{2Mr^2}{x}$$

Minimizing the fee given the interest rate  $r$ , blockchain fee  $\phi$ , and mean number  $\ell$  of daily transfers, is equivalent to minimizing the function

$$f(x) = \frac{4zx \cdot (1+r)^{\frac{x^2}{4\ell}} + 4a\phi - 4zx}{x^2}, \text{ where } x \in [1, \infty) \text{ is the channel capacity [in transfers].}$$

To find the optimal channel funding for a given  $r, \phi, \ell$ , we must find the minimum of  $f(x)$  on the interval  $[1, \infty)$ . As explained before, the first-order approximation of  $f$  is:

$$\tilde{f}(x) = \frac{zrx}{\ell} + \frac{4a\phi}{x^2}$$

The optimal channel funding based on this approximation can be determined by taking the minimum of the function  $\tilde{f}(x)$  on the range  $(0, \infty)$ . Note that

$$\tilde{f}'(x) = \frac{zr}{\ell} - \frac{8a\phi}{x^3} \quad \text{and} \quad \tilde{f}''(x) = \frac{24a\phi}{x^4} > 0.$$

Thus  $\tilde{f}$  is convex and the minimum is attained for  $x_{\min}$  with the property that  $\tilde{f}'(x_{\min}) = 0$ , that is,

$$x_{\min} = \sqrt[3]{\frac{8a\phi\ell}{zr}}$$

The lightning fee given the optimal funding is:

$$F^{opt} = F(x_{\min}) = 3\sqrt[3]{\frac{a\phi z^2 r^2}{\ell^2}}$$

This completes the proof.  $\square$

## 4 Pairs Topology

In this section we assume that the network of transfers has a very simple topology: the users are divided to pairs who only trade with each other.

Our goal is to calculate the market-equilibrium blockchain-fee  $\phi$ . To this end, we have to calculate the demand for each value of  $\phi$ , and find the value of  $\phi$  in which the demand equals the supply (which we assume is constant,  $\tau$  records per day).

### 4.1 Symbolic computations

We start by analyzing the demand of a single pair.

**Lemma 4.1** (Choice of a single pair). *Let  $\phi$  be the blockchain mining fee. Suppose the transfers between Alice and Bob are Poisson processes with means  $\lambda_A, \lambda_B$  transfers-per-day respectively, with  $\ell := \lambda_A + \lambda_B$  and  $\Delta := |\lambda_A - \lambda_B|$  and each transfer is of  $z$  bitcoins.*

*Then there exist thresholds  $\theta_{NL}, \theta_{NB}, \theta_{LB}$  such that Alice and Bob:*

- *do not transfer to each other when  $z < \min\{\theta_{NL}, \theta_{NB}\}$ .*
- *do all their transfers via lightning when  $\theta_{NL} < z < \theta_{LB}$ .*
- *do all their transfers via the blockchain when  $z > \max\{\theta_{NB}, \theta_{LB}\}$ .*

*The threshold values depend on the transfer-rates. With symmetric transfer-rates, the thresholds are:  $\theta_{NL} = \frac{27ar^2}{\ell^2\beta^3}\phi$ ,  $\theta_{NB} = \frac{\phi}{\beta}$ , and  $\theta_{LB} = \frac{\ell}{\sqrt{27ar^2}}\phi$ . (the thresholds for the asymmetric case are omitted).*

*Proof.* The users consider three options: (1) do not transfer at all, (2) transfer through the lightning channel between them, or (3) use direct blockchain transfers. They will choose the option with the lowest net utility. We assume that the users are quasi-linear in money, so their net utility is the value of transfer minus the fee.

The net utility of doing no transfer is obviously 0.

If the value of a transfer from Alice to Bob is  $v = \beta \cdot z$ , then the net utility of doing a blockchain transfer is:

$$u_B = v - \phi = \beta z - \phi$$

The net utility of a lightning transfer is  $v$  minus the lightning fee calculated in the previous subsection, assuming optimal channel funding:

$$u_L = \beta z - F^{opt}$$

$$= \beta z - 3\sqrt[3]{\frac{a\phi z^2 r^2}{\ell^2}} \quad [\text{symmetric case}]$$

The possible relations between the three net utilities, in the symmetric case, are given by:

$$\begin{aligned} u_L > 0 &\iff z > \frac{27ar^2}{\ell^2\beta^3}\phi = \theta_{NL} \\ u_B > 0 &\iff z > \frac{1}{\beta}\phi = \theta_{NB} \\ u_B > u_L &\iff z > \frac{\ell}{\sqrt{27ar^2}}\phi = \theta_{LB} \end{aligned}$$

The choice of the users depends on these parameters as follows. If  $0 > \max(u_L, u_B)$ , then the users choose no-transfer. If  $u_B > \max(0, u_L)$ , then the users use blockchain. If  $u_L > \max(0, u_B)$ , then the users use their lightning channel. The claim follows from the above by straightforward arithmetic manipulations.  $\square$

It is noteworthy that all three thresholds are linear functions of the blockchain-fee  $\phi$  (this holds both in the symmetric and the asymmetric case). This is so even though the channel-lifetime is a non-linear function of  $\phi$ .

Note that we ignore the case in which two of the three utilities are exactly equal (e.g.,  $u_L = 0$  or  $u_B = 0$  or  $u_L = u_B$ ). This is because we are going to assume that the transfer-size  $z$  is drawn from a continuous distribution, so the probability of an exact equality is zero.

The previous lemma assumed that the transfer-size  $z$  is fixed. Now, we let  $z$  be a random variable, and calculate the expected demand per pair. We calculate the demand function both with and without lightning.

**Lemma 4.2** (Demand of a single pair). *Let  $\phi$  be the blockchain mining fee. Suppose the transfers between Alice and Bob are Poisson processes with means  $\lambda_A, \lambda_B$  transfers-per-day respectively, with  $\ell := \lambda_A + \lambda_B$  and  $\Delta := |\lambda_A - \lambda_B|$ , and the transfer-size  $z$  is a random variable with probability-distribution-function  $f$ .*

*Then the expected demand of the pair [in records-per-day] is given by:*

$$D^0(\phi) = \ell \int_{\theta_{NB}}^{\infty} f(z) dz \quad (\text{without lightning})$$

$$D^s(\phi) = \int_{z=\theta_{NL}}^{\theta_{LB}} \sqrt[3]{\frac{\ell ar^2 z^2}{\phi^2}} f(z) dz + \ell \int_{z=\max(\theta_{NB}, \theta_{LB})}^{\infty} f(z) dz \quad (\text{with lightning, symmetric case})$$

*The transactions counts are:*

$$\text{Lightning: } \ell \int_{z=\theta_{NL}}^{\theta_{LB}} f(z) dz \quad \text{Blockchain: } \ell \int_{z=\max(\theta_{NB}, \theta_{LB})}^{\infty} f(z) dz$$

*and the transaction volumes are:*

$$\text{Lightning: } \ell \int_{z=\theta_{NL}}^{\theta_{LB}} z f(z) dz \quad \text{Blockchain: } \ell \int_{z=\max(\theta_{NB}, \theta_{LB})}^{\infty} z f(z) dz$$

where  $\theta_{NL}, \theta_{NB}, \theta_{LB}$  are as defined in Lemma 4.1.

*Proof.* By Lemma 4.1, when  $z > \max(\theta_{NB}, \theta_{LB})$ , the users use blockchain transfers. They do  $\ell$  transfers per day, each of which uses a single record, so their demand is  $\ell$  records-per-day. This explains the rightmost terms in all expressions.

When  $\theta_{NL} < z < \theta_{LB}$ , the users use their lightning channel, so their demand is  $a$  records each  $T$  days, where  $T$  is the channel-lifetime (in days) when it is funded optimally. So their daily demand is  $\sqrt[3]{\frac{\ell a r^2 z^2}{\phi^2}}$  in the symmetric case. This explains the leftmost term, and completes the proof.  $\square$

Note that, in theory, the range of the integral in the left-hand side might be empty (if  $\theta_{NL} > \theta_{LB}$ ). In this case, no users use lightning, and the value of this integral is zero. In practice, however, this option is very unlikely. In the Examples section below, we present some calculations with realistic numbers and show that usually  $\theta_{NL} \ll \theta_{NB} \ll \theta_{LB}$ .

Now we are ready to calculate the equilibrium blockchain-fee.

**Theorem 4.3** (Equilibrium blockchain-fee). *Suppose there are  $n$  users who only interact in pairs, so there are  $n/2$  pairs of users. In every pair, all parameters  $(\lambda_A, \lambda_B, \ell)$  are the same, except for the transfer-size  $z$ , which is drawn randomly for each pair from probability-distribution  $f$ . Then, the market-price  $\phi_{eq}$  is a solution to the following equation:*

$$\frac{n}{2} \cdot D(\phi) = \tau$$

where  $D(\phi)$  is given by one of the expressions in Lemma 4.2, depending on the case.

*Proof.* Assuming all pairs have the same parameters, the expected aggregate demand is  $\frac{n}{2} \cdot D(\phi)$  records-per-day. The supply is  $\tau$  records-per-day. The market-equilibrium price is the price in which the demand equals the supply.  $\square$

## 4.2 Power-law distribution

We now present a special case of Lemma 4.2 for the case in which the transfer-size  $z$  is distributed according to the following power-law:

$$f(z) = \begin{cases} 0 & z < z_{\min} \\ \frac{z_{\min}}{z^2} & z \geq z_{\min} \end{cases}$$

For simplicity we assume that  $\theta_{NB} < \theta_{LB}$ , in accordance with the numeric data below.

**Corollary 4.4.** *When the transfer-size has power-law distribution and  $\theta_{NB} < \theta_{LB}$ , the expected demand of the pair [in records-per-day] is given by:*

$$D^0(\phi) = \begin{cases} \ell & [\theta_{NB} < z_{\min}] \\ \ell \frac{z_{\min}}{\theta_{NB}} & [\theta_{NB} \geq z_{\min}] \end{cases} \quad (\text{without lightning})$$

$$D^s(\phi) = \begin{cases} \ell & [\theta_{LB} < z_{\min}] \\ 3z_{\min} \sqrt[3]{\frac{\ell a r^2}{\phi^2}} [z_{\min}^{-1/3} - \theta_{LB}^{-1/3}] + \ell \frac{z_{\min}}{\theta_{LB}} & [\theta_{NL} < z_{\min} \leq \theta_{LB}] \\ 3z_{\min} \sqrt[3]{\frac{\ell a r^2}{\phi^2}} [\theta_{NL}^{-1/3} - \theta_{LB}^{-1/3}] + \ell \frac{z_{\min}}{\theta_{LB}} & [\theta_{NL} \geq z_{\min}] \end{cases} \quad (\text{with lightning, symmetric case})$$

The transactions counts are:

$$\begin{cases} \text{Lightning : } 0, \text{ Blockchain : } \ell & [\theta_{LB} < z_{\min}] \\ \text{Lightning : } \ell(1 - \frac{z_{\min}}{\theta_{LB}}), \text{ Blockchain : } \ell(\frac{z_{\min}}{\theta_{LB}}) & [\theta_{NL} < z_{\min} \leq \theta_{LB}] \\ \text{Lightning : } \ell(\frac{z_{\min}}{\theta_{NL}} - \frac{z_{\min}}{\theta_{LB}}), \text{ Blockchain : } \ell(\frac{z_{\min}}{\theta_{LB}}) & [\theta_{NL} \geq z_{\min}] \end{cases}$$

and the expected transaction volumes are:

$$\begin{cases} \text{Lightning : } 0, \text{ Blockchain : } \infty & [\theta_{LB} < z_{\min}] \\ \text{Lightning : } \ell z_{\min} (\ln \theta_{LB} - \ln z_{\min}), \text{ Blockchain : } \infty & [\theta_{NL} < z_{\min} \leq \theta_{LB}] \\ \text{Lightning : } \ell z_{\min} (\ln \theta_{LB} - \ln \theta_{NL}), \text{ Blockchain : } \infty & [\theta_{NL} \geq z_{\min}] \end{cases}$$

where  $\theta_{NL}, \theta_{NB}, \theta_{LB}$  are as defined in Lemma 4.1.

### 4.3 Numeric examples

For the examples, below, we assume the following parameter values.

- Between each pair, there are  $\ell = 10$  transfers per day.
- The transfer-rate is symmetric, i.e, each of Alice and Bob makes an average of 5 transfers per day.
- $\beta = 0.01$ , i.e, the utility of a user from a money-transfer is 1% of the transfer-size.
- The interest rate is 4% per year. This implies that  $r \approx 4/365/100 \approx 0.0001096$  per day.
- The size of a channel-reset transaction is 1.1 times the size of a usual transaction, i.e,  $a = 1.1$  [records].
- The minimum transfer-size is  $z_{\min} = 0.001$  [bitcoins]:

These parameters give the following threshold values of  $z$  (in the symmetric case):

$$\begin{aligned} \theta_{NL} &= \frac{27ar^2}{\ell^2\beta^3}\phi \approx 0.0036\phi \\ \theta_{NB} &= \frac{1}{\beta}\phi = 100\phi \\ \theta_{LB} &= \frac{\ell}{\sqrt{27ar^2}}\phi \approx 16744\phi \end{aligned}$$

So, without lightning, users use blockchain iff their transfer-size is above  $100\phi$ ; with lightning, in the symmetric case:

- Users whose transfer-size is less than  $0.0036\phi$ , do not transfer at all;
- Users whose transfer-size is between  $0.0036\phi$  and  $16744\phi$  use lightning;
- Users whose transfer-size is above  $16744\phi$ , use direct blockchain transfers.

This has several interesting implications:

- In a world with lightning, whenever  $\phi$  is even slightly above zero (e.g.  $\phi > 10^{-7}$ ) almost all users user lightning — the number of direct blockchain transfers is negligible.

- Since  $\theta_{NL} \ll \theta_{LB}$ , the demand function for the range of high  $\phi$  (e.g.  $\phi > 0.3$ ) is dominated by the factor:

$$3z_{\min} \sqrt[3]{\frac{\ell ar^2}{\phi^2}} \theta_{NL}^{-1/3} = 3z_{\min} \sqrt[3]{\frac{\ell^3 \beta^3}{27 \phi^3}} = \ell \frac{z_{\min}}{\theta_{NB}}$$

which is the demand without lightning. So, for this  $\phi$  we expect the demand curves with and without lightning to be almost overlapping.

#### 4.3.1 Demand curves

Substituting in Corollary 4.4, the demand functions are:

$$D^0(\phi) = \begin{cases} 10 & [\phi < 10^{-5}] \\ \frac{10^{-4}}{\phi} & [10^{-5} \leq \phi] \end{cases} \quad (\text{without lightning})$$

$$D^s(\phi) = \begin{cases} 10 & [\phi < 6 \cdot 10^{-8}] \\ 1.5 \cdot 10^{-5} \left[ \frac{10}{\phi^{2/3}} - \frac{0.04}{\phi} \right] + \frac{6 \cdot 10^{-7}}{\phi} & [6 \cdot 10^{-8} \leq \phi < 0.3] \\ 1.5 \cdot 10^{-5} \left[ \frac{6.5}{\phi} - \frac{0.04}{\phi} \right] + \frac{6 \cdot 10^{-7}}{\phi} & [0.3 \leq \phi] \end{cases} \quad (\text{with lightning, symmetric case})$$

and the transactions counts are:

$$\begin{cases} \text{Lightning : } 0, \text{ Blockchain : } 10 & [\phi < 6 \cdot 10^{-8}] \\ \text{Lightning : } \ell \left( 1 - \frac{z_{\min}}{\theta_{LB}} \right), \text{ Blockchain : } \ell \left( \frac{z_{\min}}{\theta_{LB}} \right) & [6 \cdot 10^{-8} \leq \phi < 0.3] \\ \text{Lightning : } \ell \left( \frac{z_{\min}}{\theta_{NL}} - \frac{z_{\min}}{\theta_{LB}} \right), \text{ Blockchain : } \ell \left( \frac{z_{\min}}{\theta_{LB}} \right) & [0.3 \leq \phi] \end{cases}$$

The demand curves for the above parameters are shown in Figure 1. As  $\phi$  increases, we notice several interesting regions in the curves.

- When  $\phi$  is very low, all transfers above  $z_{\min}$  are profitable, so all demand curves start at  $\ell = 10$  (the maximum demand per day). In a world with lightning, a small increase in  $\phi$  induces many users to prefer a lightning channel, so the demand for blockchain transfers declines quickly. In a world without lightning, initially the demand remains high since users have no alternative, but then starts to decline as  $\phi$  becomes too high for blockchain transfers (Left).
- Both curves continue to decline with the price. In a world with lightning, almost all decline in demand comes from a decline in direct blockchain transfers; the number of lightning transfers remains at its maximum ( $\ell$ ) even with  $\phi$  becomes quite high. But when  $\phi$  becomes very high ( $\approx 0.3$ ), even the sparse channel-resets become too expensive. The two demand-curves meet and the number of lightning transfers starts to decline too (Right).

#### 4.3.2 Price curves

Based on the demand curves and Theorem 4.3, we calculate the market-equilibrium price as a function of  $n$  — the number of agents. Note that the number of pairs is  $n/2$ .

The price is 0 as long as the maximum demand  $\ell n/2$  is less than the fixed supply  $\tau$ . When the maximum demand hits the supply, the price starts to increase and the demand decreases.

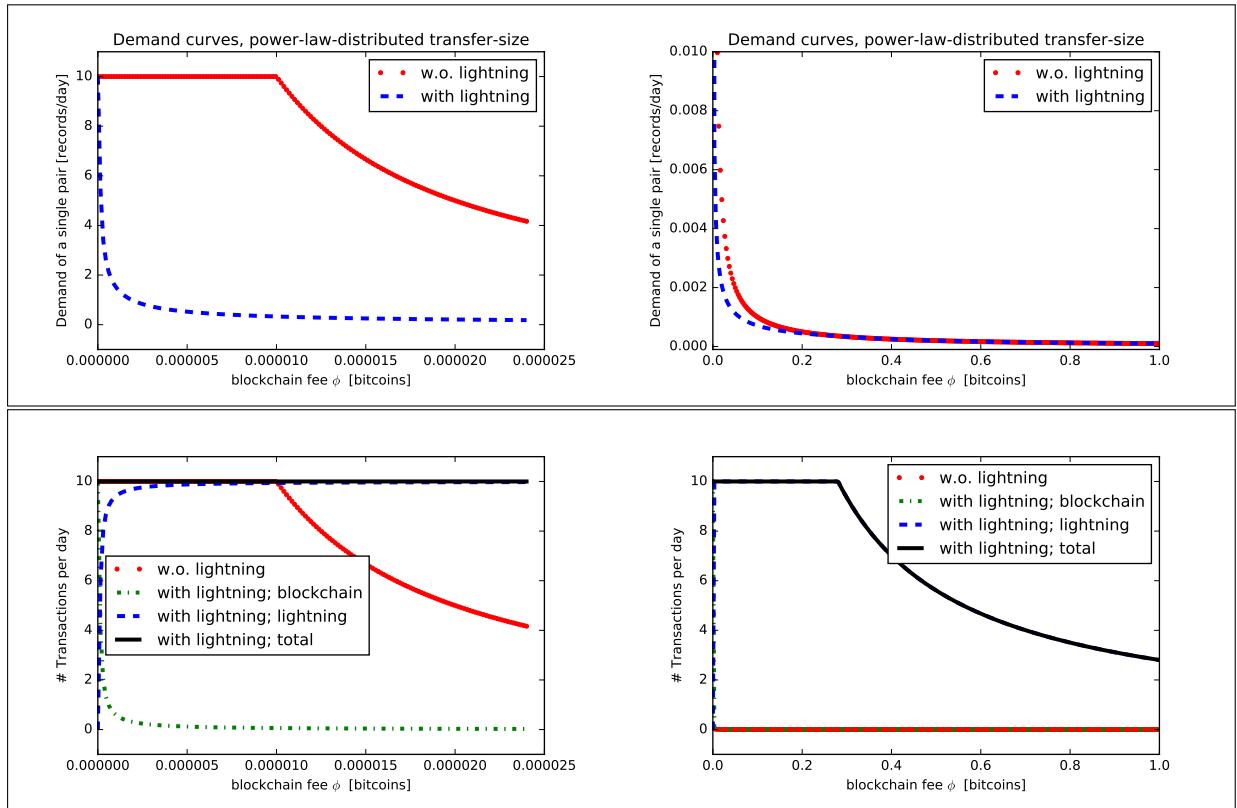


Figure 1: Demand curves when  $z$  has power-law distribution and the transfer-rate is symmetric.  
**Top:** Records-per-pair in a world with and without lightning (two different price-ranges).  
**Bottom:** Transfers-per-pair in a world with and without lightning (two different price-ranges).

Without lightning, the equilibrium price is given by:

$$\left\{ \begin{array}{ll} \frac{n}{2}\ell z_{\min} \frac{\beta}{\phi} = \tau & [\frac{\phi}{\beta} < z_{\min}] \end{array} \right.$$

which leads to the following price-curve:

$$\phi_{eq}^0 = \begin{cases} 0 & \text{for } n/2 < \tau/\ell \\ z_{\min} \cdot \beta \ell \cdot \frac{n}{2\tau} & \text{otherwise} \end{cases}$$

Here, when the maximum demand hits the supply, the price jumps to  $\beta z_{\min}$  (see Figure 2, top left); the discontinuity in the price-curve comes from the discontinuity in the power-law distribution. The equilibrium price then increases linearly with  $n$  to keep the demand and supply equal (see Figure 2, top right).

With lightning, in the symmetric case the equilibrium price is:

$$\phi_{eq}^s = \begin{cases} 0 & \text{when } n/2 < \tau/\ell \\ z_{\min} \cdot \sqrt{27\ell a} \cdot r \cdot \left(\frac{n}{2\tau}\right)^{3/2} & \text{when } \tau/\ell < n/2 < \frac{\beta^2 \ell \tau}{27ar^2} \\ z_{\min} \cdot (\beta \ell - \sqrt{27ar}) \cdot \frac{n}{2\tau} & \text{otherwise} \end{cases}$$

Here, when the maximum demand hits the supply, the price jumps to  $z_{\min} \sqrt{\frac{27ar^2}{\ell^2}}$ , which is in general much lower than  $\beta z_{\min}$  (see Figure 2, top left); even a small price-increase is sufficient to induce users to use lightning, thus reducing the demand for blockchain records. The price-increase is initially super-linear ( $\Theta(n^{3/2})$ ), but eventually  $\phi$  reaches its critical point where even channel-resets are too expensive ( $\phi \approx 0.3$ ). At this point, the price starts to increase linearly with  $n$  and the two price-curves coincide (see Figure 2, top right).

The price with lightning network, hence also the miners' revenue, is always below the price without lightning.

Below each price-curve we show the total number of transactions in the market and how they split between lightning and the blockchain:

- When the number of users is small. In a world without lightning, the number of transactions quickly attains its maximum of  $\tau$ . In a world with lightning, while the number of blockchain transactions decreases, a small number of such transactions can support a much larger number of lightning transactions (Left).
- When the number of users grows, almost all transactions are done in lightning (Right). Eventually, all  $\tau$  transactions per day are used for reset transactions and the lightning network attains its maximum capacity, which is about  $8e9$  — more than 2 million times the blockchain capacity.

## 5 Simulations

In the theoretic analysis we assumed that the transfer-size  $z$  is randomized once per pair of agents. We found out that the case in which the transfer-size is randomized in each transfer is much more difficult to analyze theoretically. Therefore we study this case using simulations.

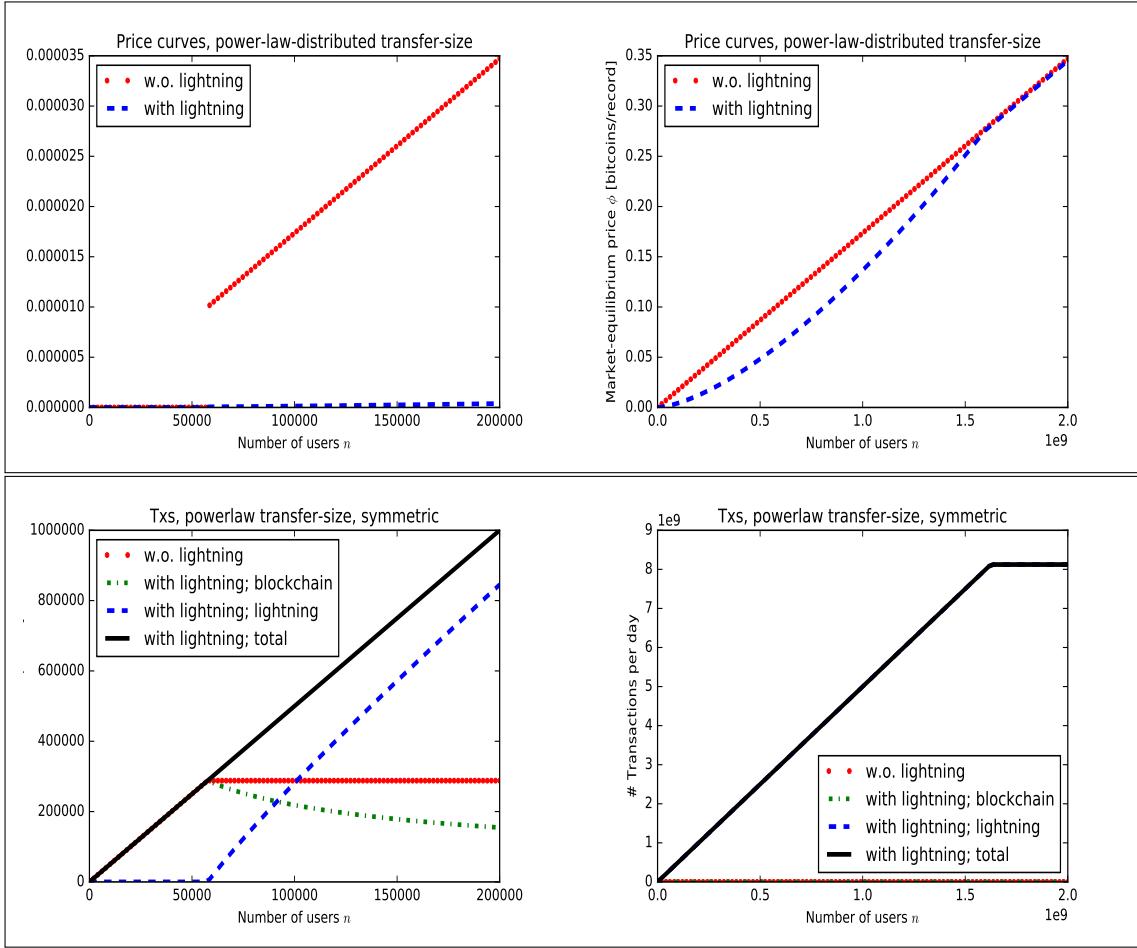


Figure 2: Price curves when  $z$  has power-law distribution.

**Top:** Equilibrium price in a world with and without lightning (two different scales)

**Bottom:** How the transfers split between lightning and blockchain (two different scales).

## 5.1 Channel operation and definitions

We consider a channel with a total capacity  $w$ . Since the transfer-rate from Alice to Bob is the same as from Bob to Alice, we assume that the channel is initialized symmetrically - each agent contributes  $w/2$ . The "channel state" is the balance of Alice in the channel; therefore the initial state is  $w/2$ .

Whenever a transfer has to be made, we determine its size  $z$  by randomly drawing from the power-law distribution  $f(z) = \frac{z_{\min}}{z^2} [z > z_{\min}]$ . Then, we check whether the transfer can be done in the current channel state (i.e, whether the agent making the transfer has a sufficiently high balance). If the transfer can be done in the channel, then it is done and the channel state is updated. Otherwise, we check whether it is worthwhile to do the transfer in the blockchain: if the transfer value  $\beta z$  is larger than the current  $\phi$ , then the transfer is done on the blockchain; otherwise, it is not done at all.

In the initial state ( $w/2$ ), the chances of channel failure are relatively small; as the channel drifts away from this initial state towards one of its endpoints, the chances of channel failure increase. Therefore, at some states it may be worthwhile to do a *channel reset* and return the channel to its initial state. We assume that there is some constant  $R$  such that, whenever the state after a channel transfer is smaller than  $R$  or larger than  $w - R$ , it is reset to  $w/2$ . We call  $R$  the *reset radius*.

## 5.2 Finding the optimal reset-radius

The first step in analyzing the channel performance is finding the optimal reset radius. The plots in Figure 3/Left illustrate the channel performance as a function of the reset radius, for a capacity of  $w = 10$ .

The first plot shows the number of *blockchain hits* — the number of blockchain records required for a typical sequence of transfers (this is the sum of the records used for out-of-channel transfers and the records used for reset transactions). The number of blockchain hits is minimized when the reset radius is about 0.5. The second plot shows the net utility of the agents, which is their value from making the transfers ( $\beta$  times the transfer sizes) minus the fee paid for blockchain hits. The utility is maximized at approximately the same point - 0.5. The third plot shows how many transfers are done in the blockchain and how many are done in the channel, as a function of the reset radius.

We now calculate the optimal reset radius. Naturally, the optimum depends on the capacity  $w$  and on the blockchain fee  $\phi$ . For simplicity we calculate and plot the optimal reset radius only as a function of  $w$ , for a fixed fee of 0.001. We make 50 experiments, each of which simulates transfers over 1000 days. Figure 3/Right shows the average optimal radius, as well as a linear regression line; it shows that the optimal reset radius is approximately 5 percent of the channel capacity.

## 5.3 Finding the optimal channel capacity

From now on, we assume that each channel with capacity  $w$  is operated with the optimal reset-radius for this capacity, using the linear regression formula shown in Figure 3/Right. Our next goal is to calculate the optimal channel capacity, given the blockchain fee (and the fixed interest rate  $r$ ).

Figure 4/Left shows some plots of the channel performance as a function of the channel capacity, for different blockchain fees. As  $w$  increases, the *blockchain cost* (the cost from blockchain

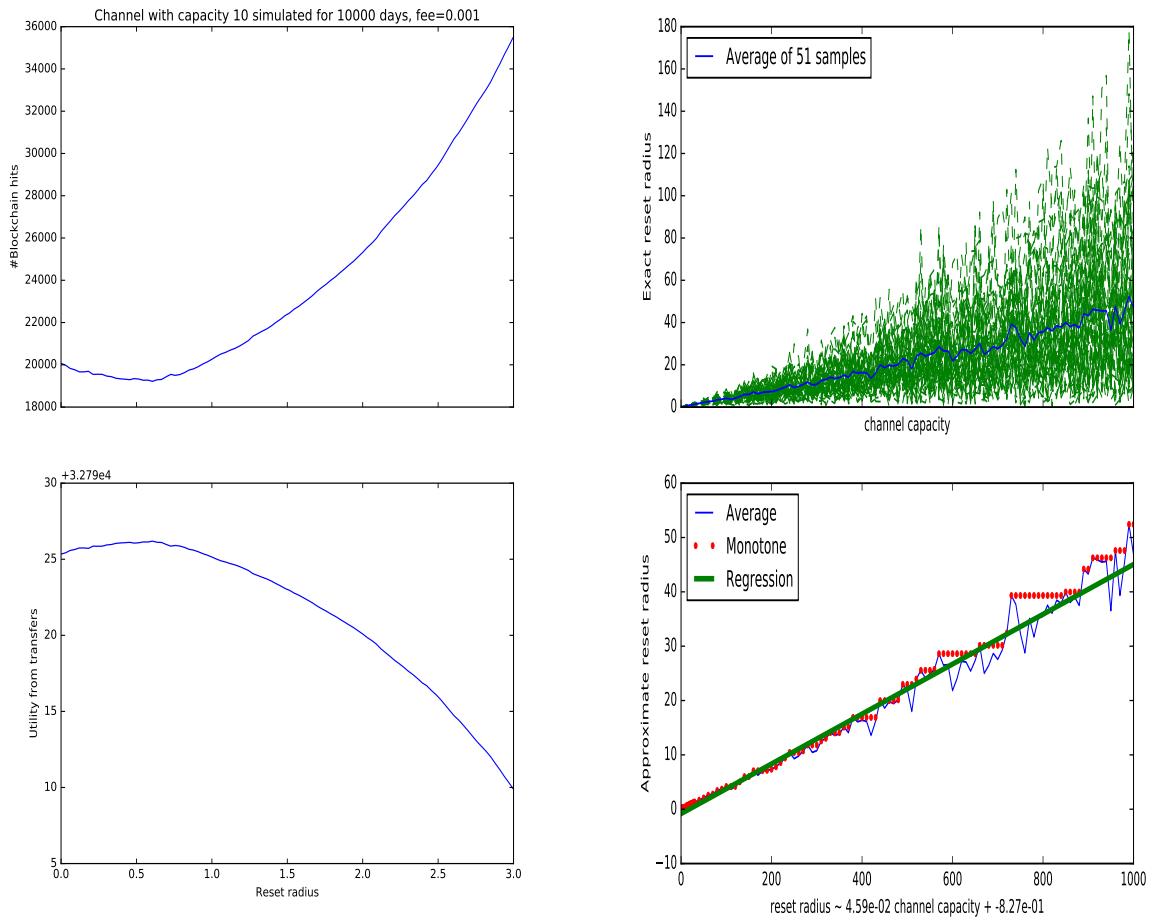


Figure 3: **Left:** channel performance as a function of the reset radius.  
**Right:** optimal reset radius as a function of the channel capacity.

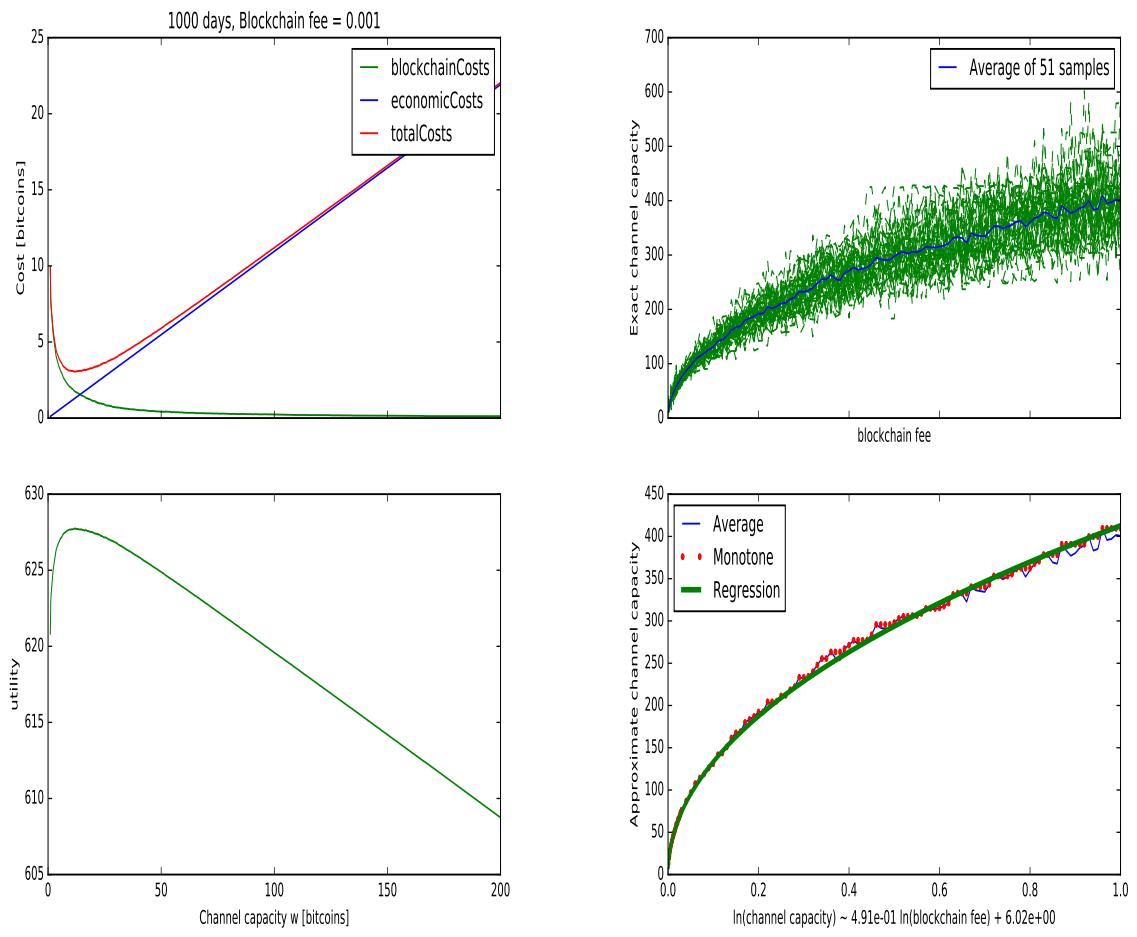


Figure 4: **Left:** channel performance as a function of the channel capacity.  
**Right:** optimal channel capacity as a function of the blockchain fee.

transfers and reset transactions) decreases, but the *economic cost* (the cost from interest) increases. The optimal utility is attained at a capacity in which these two costs are approximately equal.

We now calculate the optimal channel capacity as a function of the blockchain fee. We make 50 experiments, each of which simulates transfers over 1000 days. Figure 4/Right shows the average channel capacity, as well as a curve calculated by log-log regression; it shows that the optimal channel capacity is approximately a constant times  $\phi^{1/2}$ . The theoretic analysis in Lemma 3.6 gives a constant times  $\phi^{1/3}$ , which is qualitatively similar (recall that the situations are different since here we randomize the transfer-size each transfer rather than once per pair).

## 5.4 Demand curves

From now on, we assume that each channel is funded with the optimal  $w$  given the blockchain fee  $\phi$ , and operated with the optimal reset-radius given that  $w$ .

We calculate the curve of demand for blockchain records in the following way. For each value of  $\phi$  we calculate the number of records required to do the randomly-drawn transfers. This calculation is done in two steps: first, we calculate the total cost for doing all these transfers in a lightning channel (the blockchain cost plus the economic cost). We assume that this cost should be covered by putting a fee on each lightning transaction. The lightning fee is proportional to the transfer size, and the value of a transfer is also proportional to the transfer size. Therefore, if the total cost is smaller than the total value of all transfers, then all transfers will be done in the lightning channel. If the total cost is larger than the total value, then no transfers will be done in the lightning channel. In the latter case, the users will use only the blockchain, and they will do only the transfers with a value above the blockchain fee.

Figure 5/Top shows some plots of the daily demand for blockchain records (per pair of users) as a function of the blockchain fee. Log-log regression shows that the demand with lightning is approximately  $\Theta(1/\phi^{1/2})$  and the demand without lightning is approximately  $\Theta(1/\phi)$ . The theoretic analysis in Subsection 4.3 also gives a demand of  $\Theta(1/\phi)$  in a world without lightning. The demand with lightning is  $\Theta(1/\phi^{2/3})$ , which is qualitatively similar to the simulation results.

Figure 5/Bottom shows the division of transactions between lightning and blockchain. Interestingly, while almost all transfers are done in lightning (Left), most transfer-volume is done in the blockchain (Right).

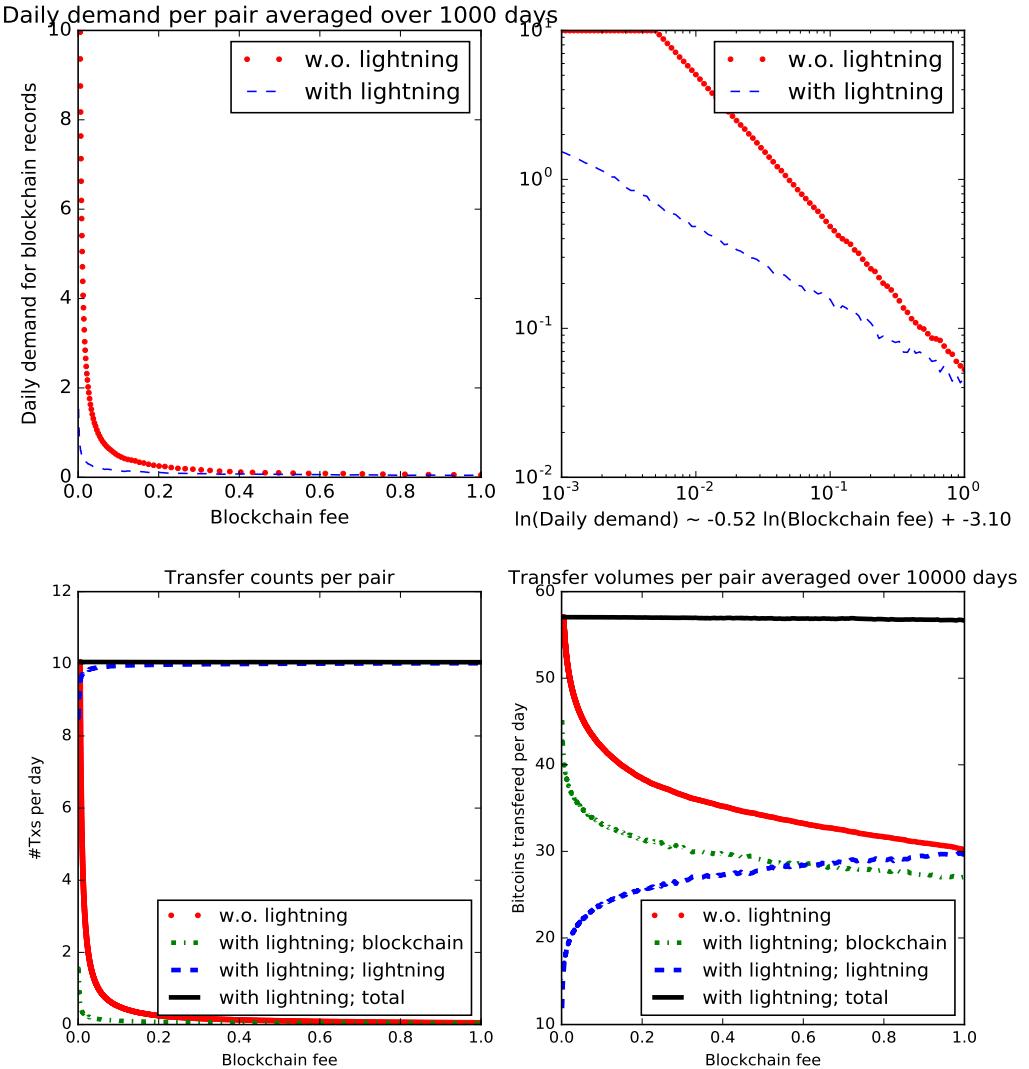


Figure 5: **Top:** Demand curves with and without lightning. *Left:* linear-linear plots. Compare to Figure 1 top-right. *Right:* log-log plots, and regression formula for the lightning curve. **Bottom:** transactions with and without lightning. *Left:* transaction-counts. Compare to Figure 1 bottom-right. *Right:* transaction volumes.

## 5.5 Equilibrium fee

Our next goal is to calculate the price-curves — describing the equilibrium blockchain fee as a function of the number of users. Given the number of users  $n$ , we estimate the equilibrium fee  $\phi$  in the following way.

- Draw a random sequence of transfer-sizes for 1000 days (an average of  $1000\ell = 10000$  transfers).
- Define the *demand function*  $D(\phi)$  as the number of blockchain-hits when the above sequence is performed in a lightning channel with optimal capacity and optimal reset-radius.
- Define the *surplus function*  $s(\phi)$  as the difference between the demand and the supply per day, i.e:  $s(\phi) := \frac{n}{2} \cdot D(\phi)/1000 - \tau$ .
- Find a  $\phi^*$  for which  $s(\phi^*) = 0$  (we used `scipy.optimize.brentq`, which uses a method by Brent (1973) to find a zero).

We executed the above procedure for 100 values of  $n$  between  $10^5$  and  $10^8$  (logarithmically spaced). For each value, we repeated the random calculation 500 times. We calculated both the average  $\phi$  for each  $n$  (averaged over the 500 samples), and a regression curve of  $\phi(n)$ . The results are shown in Figure 6.

The blockchain fee grows super-linearly with the number of users. The data exhibits a very good fit with a polynomial of degree 2, which shows that the blockchain fee grows like  $\Theta(n^2)$ , where  $n$  is the number of users. In the theoretic calculation (Sub. 4.3.2), the price grows like  $\Theta(n^{3/2})$ , which is qualitatively similar.

Without lightning, the price grows linearly with  $n$ , which is also similar to the theoretic results.

## 5.6 Overall network performance

Finally, we simulated the overall network performance as a function of the number of users. We simulated random transfers over 100,000 days, and 1000 values of  $n$  between  $10^5$  and  $10^8$  (logarithmically spaced). For each value, we calculated the equilibrium fee with and without lightning, the number and the volume of transfers done in blockchain vs. lightning.

We did this experiment once for the current block size (a supply of 288000 records per day) and another time for a double block size (a supply of 576000 records per day). The results are shown in Figure 8.

The results are qualitatively similar with single or double block size. As the number of users grows, the equilibrium fee grows super-linearly (1st line) and with it, the miners' revenue (5th line). However, the utility per user decreases (6th line). All the supply of records is used (2nd line). Almost all transfers are done in lightning (3rd line), but the volume of transfers is split approximately equally between lightning and blockchain (4th line), since the largest transfers are done in the blockchain.

Doubling the block size (with lightning) has some minor quantitative differences: the blockchain fee is smaller by a factor of about 3, the miners' revenue is smaller by a factor of about 1.5, and the utility per user is higher by factor 1.5.

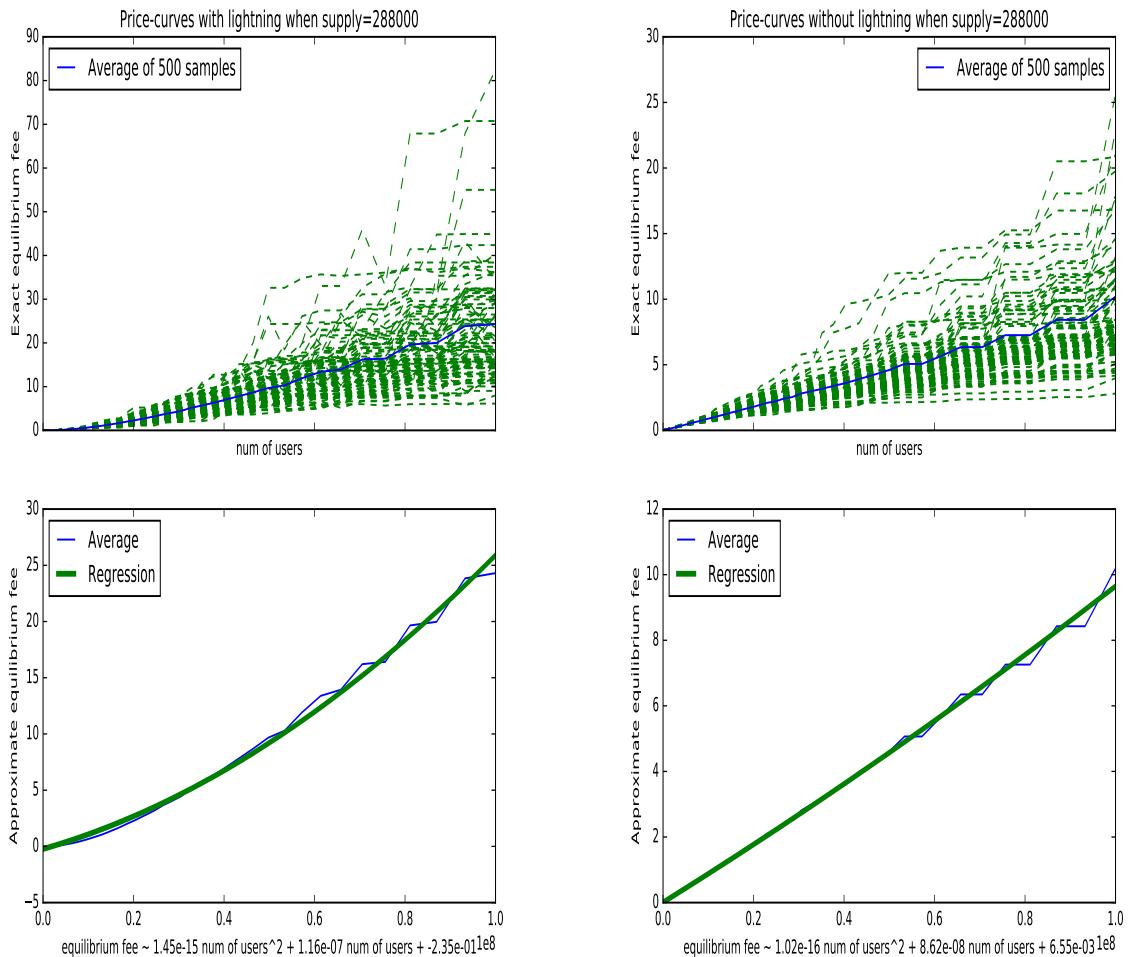


Figure 6: **Left:** Price curves with lightning. **Right:** Price curves with no lightning.

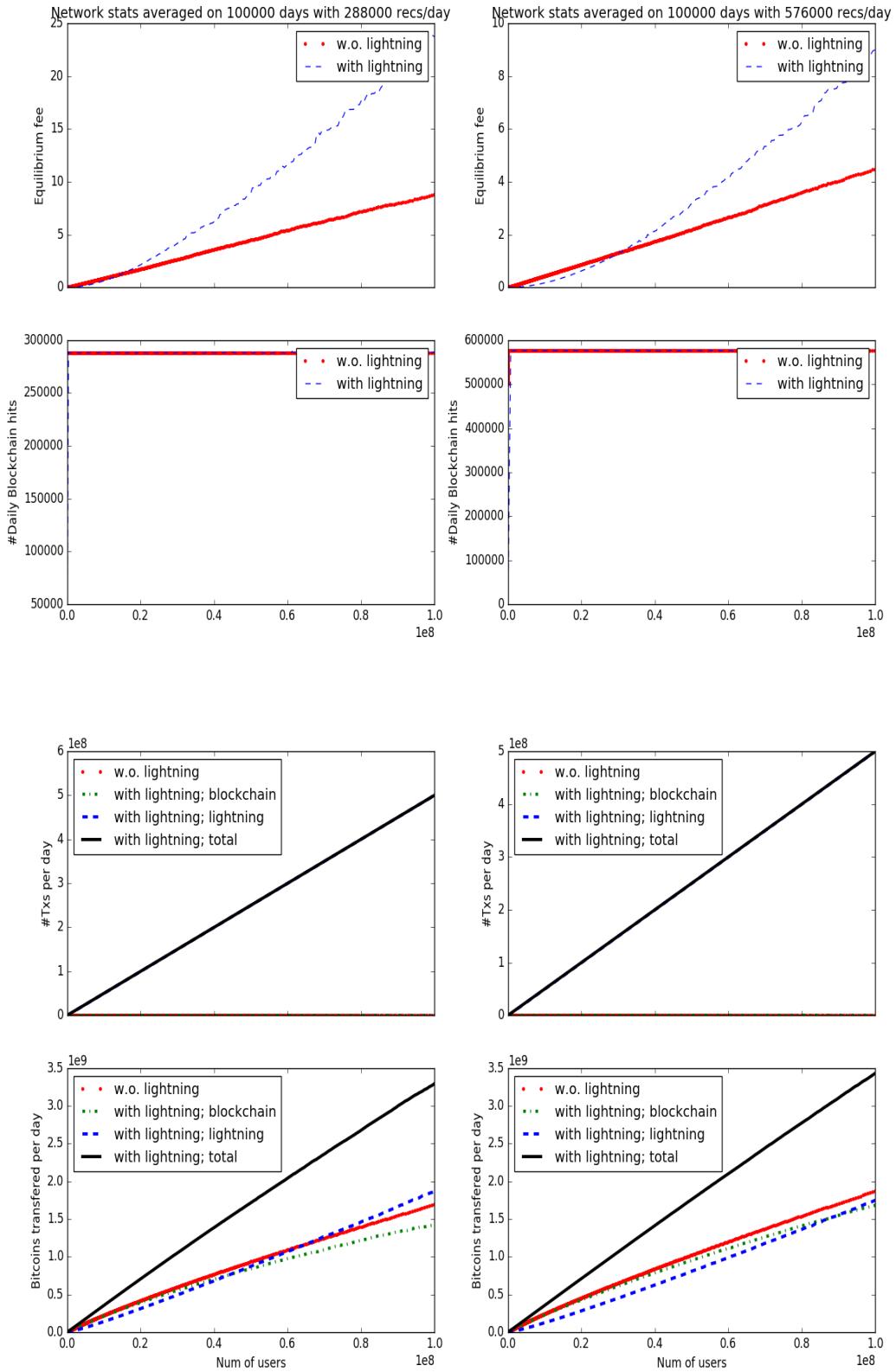


Figure 7: **Right:** Network stats with current block size. **Left:** Network stats with double block size.

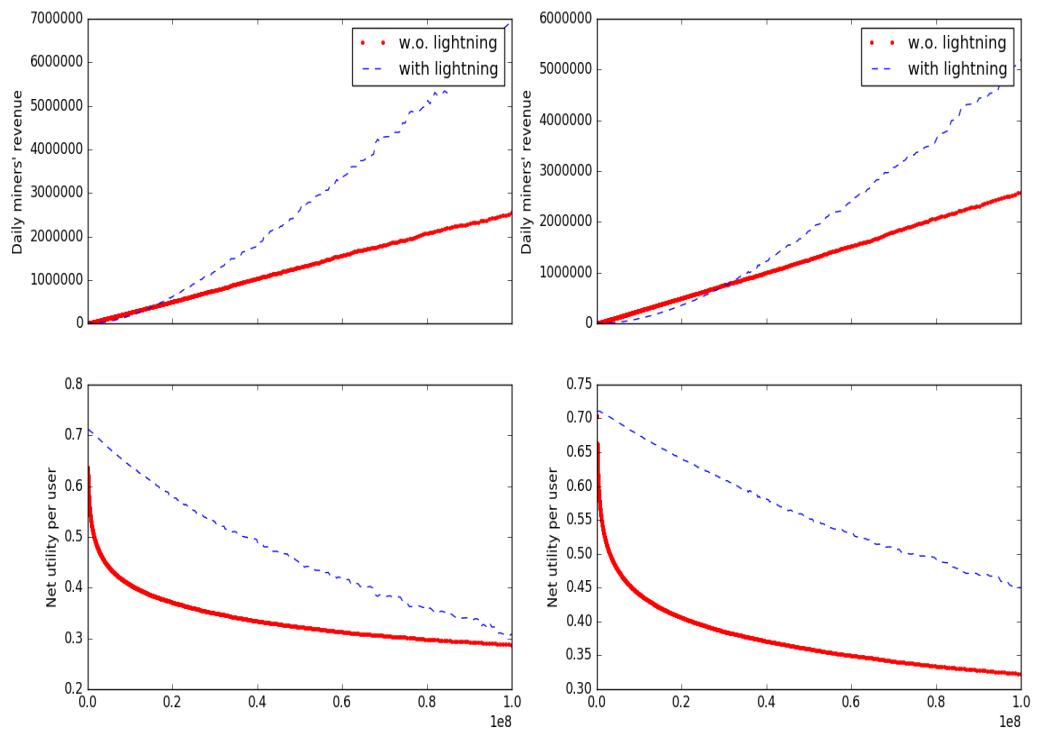


Figure 8: **Right:** Network stats with current block size. **Left:** Network stats with double block size.

## 6 Star Topology

We now show how to solve networks where there is a central bank that each user is connected to. For each pair of  $i, j$ , let  $\lambda_{i,j}$  denote the flow of player  $i$  towards player  $j$ . Then we can define  $\lambda_i^+ = \sum_{j \in N} \lambda_{i,j}$  and  $\lambda_i^- = \sum_{j \in N} \lambda_{j,i}$ . Set  $p_i = \frac{\lambda_i^+}{\lambda_i^- + \lambda_i^+}$  and  $q_i = \frac{\lambda_i^-}{\lambda_i^- + \lambda_i^+}$ . Then  $p_i + q_i = 1$ , so the expected lifetime of the channel between user  $i$  and the bank can be obtained as a corollary from the single channel analysis.

**Corollary 6.1** (Lifetime of Channel between User and Bank). *Suppose there is a lightning star network, where the transfers from each user  $i$  to any other user  $j$  are given by a Poisson process with mean  $\lambda_{i,j}$  transfers per day, and the channel between user  $i$  and the bank has capacity  $w$ . Let  $\lambda_i^+ = \sum_{j \in N} \lambda_{i,j}$  and  $\lambda_i^- = \sum_{j \in N} \lambda_{j,i}$ . Then, the expected lifetime of the channel between user  $i$  and the bank from the state where user  $i$  has  $m$  bitcoins and the bank has  $w - m$  bitcoins on the channel with user  $i$  is:*

(a) when  $\lambda_i^+ \neq \lambda_i^-$ :

$$\tilde{X}_m^i = \frac{m}{\lambda_i^+ - \lambda_i^-} - \frac{w}{\lambda_i^+ - \lambda_i^-} \left( \frac{1 - \left( \frac{\lambda_i^+}{\lambda_i^-} \right)^m}{1 - \left( \frac{\lambda_i^+}{\lambda_i^-} \right)^w} \right) \quad [\text{days}]$$

(b) when  $\lambda_i^+ = \lambda_i^-$ :

$$\tilde{X}_m^i = \frac{wm - m^2}{2\lambda_i^+} \quad [\text{days}]$$

**Corollary 6.2.** *The optimal initialization of a star network with channels of capacity  $w$  is such that for each user  $i$*

- If  $\lambda_i^+ = \lambda_i^-$ , both the bank and user  $i$  start with  $w/2$  bitcoins. The expected lifetime of user  $i$ 's channel with the bank when initialized this way is  $T_i(w) = \frac{w^2}{8\lambda_i^+}$ .
- If  $\lambda_i^+ > \lambda_i^-$ , user  $i$  starts with  $w$  bitcoins and the bank with 0 bitcoins, while if  $\lambda_i^+ < \lambda_i^-$ . User  $i$  starts with 0 bitcoins and the bank with  $w$  bitcoins. The approximate expected lifetime of user  $i$ 's channel is  $\tilde{T}_i(w) = \frac{w}{|\lambda_i^+ - \lambda_i^-|}$ .

Next we calculate the optimal channel funding for the star topology, finding that the fee per transaction for any user in the star network is twice as high compared to the pairs topology.

**Theorem 6.3.** *Consider a star network with users  $1 \dots n$  connected through a bank, such that the transfer from any user  $i$  to another user  $j$  is a Poisson process with mean  $\lambda_{i,j}$ , with  $\ell_i = \sum_{k \in N} \lambda_{i,k} + \lambda_{k,i}$ . Let  $\phi$  be the blockchain mining fee and  $r$  the daily interest rate for locking money in a lightning channel. Each transfer between any pair of users is of  $z$  bitcoins, and each user  $i$  has a channel with the bank of capacity  $w_i \cdot z$  bitcoins.*

*Then the expected cost of any transfer that a user  $i$  is involved in is:*

$$F(w_i) = \frac{w_i z \cdot (1+r)^{T(w_i)} + a\phi - w_i z}{T(w_i) \cdot \ell_i}, \quad [\text{bitcoins/transfer}]$$

*where  $T(w)$  is the expected lifetime (in days) of a channel of capacity  $w$ .*

*The expected fee paid by each user  $i$  is twice as high in the star network compared with the model where user  $i$  transacts only with a fixed user  $j$  without a bank.*

*Proof.* This follows from the single channel analysis (Lemma 3.4), by noting that the expected (total) cost of a transaction that passes through user  $i$ 's channel with the bank is

$$\frac{w_i z \cdot (1+r)^{T(w_i)} + a\phi - w_i z}{T(w_i) \cdot \ell_i}.$$

Since the bank does not support any cost that user  $i$  incurs, the fee for user  $i$  is twice as high compared to the case where  $i$  has transactions with only one other user  $j$  (with no intermediary).  $\square$

The analysis from the pairs topology applies here too, except that the cost of using the lightning network for each user is multiplied by a factor of two.

## 7 Future Work

Instead of Poisson transfers, we can assume that there is a fixed sequence of transfers:  $(i_1, j_1)$ ,  $(i_2, j_2), \dots$ , where transfer number  $k$  is from user  $i_k$  to user  $j_k$ . Each user  $i$  has a fixed initial budget  $K_i$ . The sequence is feasible, i.e., at time  $k$ , user  $i_k$  always has a sufficient amount of money for transferring to user  $j_k$ . We assume that at time  $k$ , user  $i$  considers only two options: blockchain transfer or bitcoin transfer, and selects the cheaper option. There is an interesting algorithmic question: given a fixed sequence of transfers, what is the optimal configuration of lightning channels?

## 8 Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 740282), from the Israel Science Foundation (grant 616/13 and grant 1083/13) and from the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Bureau (grant 039-9230). Erel was supported by the ISF grant 1083/13.

The symbolic calculations were done with Sympy [MSP<sup>+</sup>17]. The plots were done using Desmos [des] and Matplotlib [Hun07].

## References

- [CDE<sup>+</sup>16] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. On scaling decentralized blockchains. In Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, pages 106–125, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
- [des] Desmos. <http://www.desmos.com>. Online; accessed Dec 2017.
- [DW15] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *Proceedings of the 17th International Symposium on Stabilization, Safety, and Security of Distributed Systems - Volume 9212*, pages 3–18, 2015.

- [HS] Mike Hearn and Jeremy Spilman. Bitcoin contracts. <https://en.bitcoin.it/wiki/Contracts>. Online; accessed Dec 2017.
- [Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.
- [MSP<sup>+</sup>17] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017.
- [NBF<sup>+</sup>16] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.
- [PD16] J. Poon and T. Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2016. Technical report.