Dauphine | PSL
CHAIRE FINTECH

WORKING PAPER SERIES

WP 2021-2

INCENTIVES ON THE LIGHTNING NETWORK : A BLOCKCHAIN-BASED PAYMENT NETWORK

LOUIS BERTUCCI

Incentives on the Lightning Network : A Blockchain-based Payment Network*

Louis Bertucci†

February 15, 2021

Abstract

The Lightning Network is a decentralized payment network built on top of a blockchain, in which intermediary nodes provide a trustless routing service for end users. We provide an overview of the current state of the network and show that it can be well approximated by a scale free generative model with a fitness parameter, which suggests that nodes behave strategically on the network. Those strategic interactions between nodes can be described by a Bertrand competition model with capacity constraints. We show that there is a unique equilibrium in which a centralized network is never optimal, and the routing fee is strictly greater than the marginal cost. When nodes are heterogeneous in their opportunity cost of capital only, the equilibrium network structure can match the current state of the network.

Keywords: Blockchain ; Lightning Network ; Decentralized Payment Network
JEL Codes: C72, D23, D85, G20

*The author would like to thank his PhD advisor Pr Gilles Chemla as well as Christophe Bisiere, Jerôme Dugast, Laurent Germain and Christine Parlour for valuable comments.

†Institut Louis Bachelier, Paris. louis.bertucci@institutlouisbachelier.org

1 Introduction

During the past 10 years, blockchains have gone from a short post on a cryptography mailing list to a global industry with a total market capitalisation of roughly USD 275 billions[1]. Introduced by Nakamoto (2008), a blockchain is a *communication protocol, a distributed database, and a consensus algorithm such that the communication protocol allows a group of untrusted agents to reach consensus over the content of the distributed database*. To each blockchain, there is an associated cryptocurrency, that is needed to provide the necessary incentives to the maintainers (i.e. miners). By extension, the database itself is also called the blockchain and it usually contains the list of transactions from the very first transaction in the denominated cryptocurrency. In the blockchain world a transaction can refer to *i*) a *pure* transaction (agent A sends some value to agent B), or *ii*) a *smart-contract* which is essentially a piece of software that can interact (send/receive value) with other agents (or other smart-contracts) the way it was meant at the contract creation[2]. For security reasons, the blockchain is slow and costly to use[3]. To overcome this issue, Poon and Dryja (2016) show that with the right combination of smart contracts, it is possible to create a payment network, which they called the *Lightning Network*, in which participants can transact among each other at a cost supposed to be a lot lower. Intermediary nodes on the network will charge a fee to route a transaction and in this paper we analyze the current state of the Lightning Network and investigate the incentives of nodes.

The term "blockchain" has been used to mean very different concepts. In this paper, we use the term "blockchain" as first intended by Nakamoto (2008), that is a "proof-of-work chain". *Proof-of-work* refers to the consensus algorithm used on the Bitcoin blockchain. Intuitively, *miners* need to brute force the solution of a hash based puzzle, that is designed to take a constant amont of time to be solved. If it is solved too quickly (slowly), the difficulty increases (decreases). The system does not rely on any trusted agent because miners prove their work by showing the solution of the puzzle, and anyone can validate this solution without the need to trust the miner who solved it. There are other consensus algorithms but, as of mid-2019, none of them is implemented at the scale of the bitcoin blockchain. There are other concepts named "blockchain" like "permissioned blockchains" that function like a proof-of-work chain but they rely on a group of trusted agents so there are very

[1]Total market capitalisation of all cryptocurrencies as observed on *coinmarketcap.com* on May 28th, 2019.

[2]Otherwise stated, once a contract is added to the blockchain, it cannot be removed or changed. It *will* execute as long as the blockchain is still growing.

[3]The speed is artificially lowered to reduce the probability of forks. See Biais, Bisiere, Bouvard and Casamatta (2019) for an analysis of forks in the blockchain. The monetary cost of using a blockchain has to stay high in order to incentivize the miners.

different from a trustless service. The real technological advance is the ability to generate consensus among a group of untrusted agents, so any other kind of so-called blockchain is outside the scope of this paper.

The Lightning Network is a network of bilateral *payment channels* allowing users to send/receive value (to/from anyone in the network) without all transactions being stored on the blockchain. In financial terms, this is a payment system. Users escrow some amount on the blockchain (that can be used only on the Lightning Network), then they send/receive value and when they need value out of the Lightning Network, the blockchain pays them the *net amount* given the history of bilateral transactions. At the blockchain level, the Lightning Network allows users to perform *net-settlement*, but there is an important difference with the Lightning Network and traditional payment systems. We sometimes refer to the blockchain as a *trustless* system, meaning users don't need to trust each other, as a consequence, there is no counterparty risk in the blockchain world[4]. The Lightning Network itself is very close to a *Real Time Gross Settlement* system. Because using the blockchain is costly, the Lightning Network allows users to move transactions to a less costly payment system, while keeping the absence of counterparty risk associated to a gross settlement system.

The contribution of this paper is twofold. First, we provide an overview of the current state of the Lightning Network. As the time of writing this paper, there are $2,865$ nodes and $36,982$ payment channels among them. Although the network is at its premises, we show that it already has good properties. From a static point of view, we show that the network is well interconnected in terms of clustering and centrality measures. A well interconnected network is necessary in order to be used as a global payment system. Indeed, if user A wants to pay user B, it needs to either *i*) have a direct payment channel with user B, or *ii*) find a route of payment channels within the network[5]. From a dynamic point of view, we estimate a generative model and we show that the node-specific component is an important determinant of the probability of getting a new edge (i.e. a payment channel with another node). This means that nodes are not connecting at random when they enter the network. The second contribution of this paper is to build a model to analyze nodes' incentives on the Lightning Network. Since nodes do not connect at random, we are interested in understanding how they decide which node to connect to, and what are the implications regarding the network structure.

To be able to route transactions, a node needs to have enough funds on his payment channels. If Alice wants to pay 1 unit to Bob through Charlie, Alice needs

[4]More precisely, there is no counterparty risk as long as the blockchain is well functioning. Users will systematically ask for proof of ownership of the amount they are about to receive.

[5]Note that no node can steal any funds when they are routing transactions.

to have at least 1 unit on her payment channel with Charlie and Charlie needs to have at least 1 unit in her payment channel with Bob. Payment channels are independent from one another, so if Charlie does not have enough funds on her payment channel with Bob, she won't be able to route Alice's payment. The amount of funds on a payment channel is called the capacity. When routing a transaction, Charlie will be able to collect a fee, paid by Alice, the original sender of the payment.

Our model features a set of profit maximizing nodes that are competing in a two-stages model to route a fixed number of transactions. In the first stage, nodes enter the routing market, each opens at least two payment channels and assign them a capacity. During the second stage, given the capacity and the graph of payment channels, nodes compete on price to attract the next transaction. A buyer on the product market will pick the cheapest available route (with enough capacity) to pay the seller of a good or service. We assume the marginal cost of routing a transaction to be zero[6].

In order to open a payment channel, nodes must send a transaction to the blockchain and therefore incur the associated cost. Also, when all the capacity of a payment channel is locked toward one side, say Charlie's side, Alice cannot pay Charlie anymore with this payment channel. There are then two solutions, Alice can either *i*) send a *refund* transaction to the blockchain (with the associated costs), or *ii*) if the network's graph allows it, she can pay herself within the Lightning Network and therefore rebalance her channels' capacity (for instance, Alice pays herself through David and Charlie, so that Charlie has to give back some capacity to Alice on the payment channel that was initially blocked). Note that in both cases, it is costly to maintain a well-funded channel, either because of the blockchain cost or the Lightning Network cost. It is supposed to be a lot cheaper on the Lightning Network though, so nodes will try to always have potential route to loop back to them in order to rebalance payment channels. We call this cost, *the cost of maintaining a given interconnectedness level.*

In our model, nodes are subjected to an opportunity cost of capital for the funds they lock inside payment channels. This is meant to represent the usual missed opportunities but also a cost linked to security. On the Lightning Network, nodes need to advertise their *ip-address*, as well as the capacity of each channel, in order for end users to be able to select them as part of the optimal route[7]. As a direct consequence, a node with a lot of funds locked inside payment channels will become very visible, most importantly to potential bad actors. Otherwise stated, it is likely to become the target of hackers, and will either have to incur additional security

[6]In practice, this is essentially zero, as it consists only of sending a few data packets over the internet.

[7]It is not mandatory for nodes to advertise those, but then they would be excluded from the public network, and no one could use them as intermediary node.

costs or be subject to the risk of loss. For this reason, we assumed the opportunity cost of capital to be quadratic.

We assume no barriers to entry in the routing market. This assumption is quite relevant for our use case, because the actual Lightning Network is completely open. This means that anyone can download the software, run it, and start to create payment channels with other nodes, and this does not require anything other than a computer and some funds. In particular, no barriers to entry means that if any outside agent sees a profitable opportunity, it will jump into the market and make a profit.

A key aspect of our model is that we identify 3 factors influencing the likelihood of a node to be on the optimal route. First, there has to exist an actual route (a set of payment channels) connecting the end-buyer with the end-seller through this node. Second, the route needs to have a sufficient capacity on all intermediary payment channels so that the transaction can take this route. And lastly, this route has to be the cheapest one. Among several routes that satisfy the first two conditions, the end-buyer would pick the cheapest one. In the first stage, nodes will individually invest in costly interconnectedness, and channels funding, to increase the probability that they route the next transaction. The probability of being selected as part of the optimal route is assumed to follow an exponential distribution. In the second stage, nodes compete on the third factor to attract transactions, the price. Note that this third factor is also costly because as a node decreases its price, to increase the probability of being selected, it lowers its revenue if it is selected.

We show that when the cost of maintaining a given interconnectedness level is linear, and the opportunity cost of capital is quadratic, there is a unique equilibrium for the two stages game. In this equilibrium, all nodes have the same optimal strategies when they are homogeneous, and the routing fee is strictly greater than zero, the marginal cost. Indeed, we show that there is an endogenous participation constraint for the nodes, that is if the routing fee is anticipated to be too low, nodes prefer not to participate in the routing market, and the Lightning Network collapses. Also, because of the price competition in the second stage, each node makes zero profit in equilibrium.

Our results indicate that when nodes are heterogeneous with respect to their opportunity cost of capital, the equilibrium is still unique but the heterogeneity is reflected in the level of interconnectedness and the amount of funds locked in payment channels. In particular, a high (low) opportunity cost of capital is associated with low (high) interconnection in the network, and low (high) funds in payment channels. Because we assumed there are no barriers to entry, no node makes a profit in equilibrium, even when they are heterogeneous.

Our model generates the equilibrium relationship between the opportunity cost of capital distribution, the distribution of centrality/clustering in the network, and the distribution of channels' capacity. This has important implications regarding the structure of the network. In particular, we show that a lower bound on the distribution for the opportunity cost of capital will produce an upper bound on node's centrality as well as on the amount a node will lock in the Lightning Network. The upper bound on node's centrality predicts that a completely centralized network is never optimal in equilibrium. Moreover, our model is able to consistently generate network topology given a distribution for the opportunity cost of capital.

It should be pointed out that the Lightning Network is still in an early stage of development. A lot of nodes operate on the Lightning Network to test the technology and the overall demand is relatively low for a global payment network[8]. Also, the fees are close to zero and no one is really trying to profit from routing transactions. Therefore, we should not expect our model to be tested on real data before some strategic agents enter the network, and the demand for this payment system is higher.

This paper contributes to the recent literature about blockchains. Although being an analysis of a so called second layer protocol, it relies on a blockchain to operate. The motivation behind Poon and Dryja (2016) is that the way Nakamoto (2008) designed the blockchain does not allow for a lot of scaling on the blockchain itself. Easley, O'Hara and Basu (2017) study how fees and delays are used as incentives for miners to operate the blockchain. Those *on-chain* costs are necessary for the security of the blockchain. As a consequence, small or micro payments are likely to be unfeasible if they have to be stored on a blockchain[9]. The Lightning Network, as presented by Poon and Dryja (2016) is called an *off-chain* scaling solution, and this paper is the first attempt to analyze such a solution from an economic point of view.

The first part of this paper provides a description of the current state of the Lightning Network. In line with Rohrer, Malliaris and Tschorsch (2019) and Seres, Gulyás, Nagy and Burcsi (2019) we find evidence of a well connected graph from a static point of view. From a dynamic point of view, our results also support a *scale-free* classification of the graph. In order to properly estimate the power law degree distribution resulting from a scale-free model, we use a dataset of channel openings and closings from mid-January 2018 (the early days of the Lightning Network) until mid-August 2018. Both paper cited above find a different coefficient for the power law exponent because they use a different, more recent period. However, contrary to those papers we also estimate the fitness distribution with a non parametric

[8]As we will see, the total capacity of the network is only around EUR 5 million and it is not completely used.

[9]The routing fee would be greater than the volume transacted.

estimation. The fitness distribution from Bianconi and Barabasi (2001) represents the idiosyncratic strength of each node. We find evidence of a significant non-constant fitness distribution which suggests that nodes behave in a strategic way. They do not simply connect at random to other nodes, they specifically choose who to connect to.

This paper also contributes to the payment systems literature. Rochet and Tirole (1996) proposes an overview of the payment system mechanisms, Freixas and Parigi (1998) and Kahn and Roberds (1998) study the difference between a *net* payment system and a *gross* payment system. They show that a payment system with *gross* settlement is generally preferred as it lowers the risk of default and contagion. Leitner (2005) shows that the risk of contagion might provide enough incentives for participants to bail out one another, even if transactions are net-settled. Usually in payment network literature the issue of net settlement is associated with risk and losses for participants. For instance, if the network nets all transactions at the end of the day, some participant might be unable to make the required payment. If, on the other hand, all transactions are settled separately and at the time they are made, there is no risk of loss. The problem is that the total amount exchanged is higher and therefore a *gross* payment system requires more money to operate. The Lightning Network, coupled with a blockchain, is the first kind of network that is able to take advantage of both *net* and *gross* payment system. From the blockchain point of view, the Lightning Network is a netting system. Participants can exchange value on this network, and when they want their funds back, they can exit the network and the blockchain pays them the net amount given the history of Lightning transactions. The transaction volume can be dramatically decreased with this system. However, because everything sits on a blockchain, there are no counterparty risk. Indeed, on the Lightning Network itself, everything is *gross* settled because, once Alice sent Bob 1 unit on the Lightning Network, she cannot use this unit to pay someone else. The unit is effectively transferred right away but the blockchain does not need to know about it. As long as the blockchain is secure, all the funds on the Lightning Network are at no risk[10]. This paper is the first attempt to analyze this particular network that separates the risk of the funds from the payment structure itself.

Our work also relates to the industrial organization literature. Indeed, nodes on the Lightning Network are competing on the routing market in Bertrand style, that is competition on price. However, as explained above, nodes are constrained on the quantity of transactions they can route by *i*) their interconnectedness and *ii*) the ca-

[10]There could actually be an issue regarding adverserial close of payment channels, but there are some ways around it. See "Unilateral closing of channel" in section 2 and a discussion on this point in section 7.

pacity of their payment channels. A number of papers have analyzed such games of Bertrand competition with capacity constraints like Kreps and Scheinkman (1983), Davidson and Deneckere (1986), or Moreno and Ubeda (2006). The main findings of this literature is that when firms are constrained on their capacity while they compete on price, the equilibrium is actually the Cournot outcome. Otherwise stated, the equilibrium is the same as if firms were competing on quantity, and the price is higher than the Bertrand price (i.e. the marginal cost). The Lightning Network is different from the usual assumptions made when analyzing traditional product market competition. Indeed, nodes do not, strictly speaking, provide a homogeneous service, because depending on their position in the graph and the number of nodes they are connected to, they will be more or less useful for a given transaction. Also, we do not assume that the demand is expressed in terms of the equilibrium quantity. We argue that the quantity demanded by end users does not depend on the routing price as long as it is cheaper than an alternative payment system[11]. Our model features a two stages game very similar to Kreps and Scheinkman (1983), apart from the previous assumptions. Because we are less strict on the assumption we make, we do not have a Cournot outcome in equilibrium, but we are able to show that the equilibrium price is higher than the Bertrand price, which is zero (the marginal cost).

The remaining of this paper is organized as follows. Section 2 provides some generalities about the Lightning Network. Section 3 describes the current state of the network. Section 4 presents the main model with a homogeneous cohort of agents while section 5 introduces heterogeneity in the nodes' opportunity cost of capital. Section 6 provides a comparison between our model and the current state of the network. Section 7 sheds light on future potential research. Section 8 concludes.

2 General Facts about the Lightning Network

This section presents an overview of the technical aspect of the Lightning Network and how it is implemented. We assume basic knowledge of blockchain and cryptocurrencies.

The Lightning Network is a communication protocol, built on top of a blockchain, that allow users to send and receive value without the need for the underlying blockchain to store every transaction. Because the Lightning Network needs an underlying blockchain to operate, it is also called a second-layer protocol, with the

[11]A payment system has no value in itself. The value users get is from the physical product they are buying for instance. They only use the Lightning Network as a tool to pay for what they want. Therefore, it is important to note that everything is conditional on the outside option, that is to pay with another payment system. As long as the price on the Lightning Network is below the outside option, users don't care about the price itself, and the demand is constant.

blockchain being the first-layer protocol.

Value When we use the term value transfer and value storage for instance, it means value denominated in terms of a blockchain cryptocurrency. Whether a cryptocurrency constitues money is outside the scope of this paper and we assume it does.

Payment channel Payment channels are bilateral contracts in which both parties lock some amount $x \geq 0$ on the blockchain, of which the sum is called the *capacity*. To "open" the channel, they will have to broadcast a transaction (a contract) to the blockchain, hence incurring fees and delays inherent to blockchains[12], and they would broadcast another transaction to "close" the channel. Any transaction that changes the allocation of the channel capacity among both parties can be made bilaterally without the need to broadcast anything to the blockchain nodes. These transactions are fast (instantaneous) and a lot cheaper than transactions on the blockchain (the point of this paper is actually to study the cost structure on the Lightning Network).

Payment channels network A network of payment channels can be used to route transactions using intermediary nodes. Despite them being real intermediaries, thanks to some cryptographic tools, no intermediaries have the ability to steal any funds. Moreover, the routing algorithm currently implemented is based on *Onion Routing*. Onion Routing is the same routing algorithm used on the TOR Network for instance. With this algorithm, minimum information for the intermediaries is achieved: any intermediate node just knows that he received a transaction from address *0x.....4e12a4* and that it has to forward it to address *0x.....5834bc4* for instance, it does not even know whether any of those two nodes are the actual *sender/recipient* of the original transaction[13]. The sender does the routing selection based on the information it has from the network nodes. The most known implementation of such a network is the Lightning Network[14], and is currently (as of late 2018) being implemented on top of the bitcoin blockchain[15]. Otherwise stated, any mention of a payment channels network will refer to the Lightning Network on the Bitcoin blockchain.

[12] On the bitcoin blockchain for instance the block time interval is 10 minutes on average and one often expects around 5/6 confirmation blocks which then average to 1 hour. The fees are set by market forces using supply and demand and can be pretty high due to the limited space in each block. For instance on the bitcoin blockchain each block is limited to 1Mb.

[13] Also, for long payment routes, a given node does not know its position within the route.

[14] See Poon and Dryja (2016)

[15] There is another smaller implementation on the Litecoin blockchain.

Unilateral closing of channels If both parties want to close the channel, and if they agree on the history of bilateral transactions while the channel was open, they can each broadcast a closing transaction. The blockchain will "verify" that both transactions are correct and allocate the corresponding amount to each party. However, it is also possible to close the channel unilaterally. There has to be a way to do so because nothing prevents one party to disconnect from the network at any point in time, and the other party must still be able to retrieve his funds. However, unilateral close also means that one party might try to close the channel with a previous history of transactions. To prevent that, the current Lightning Network specifications work as follow. When a party tries to unilaterally close a channel, there is a 14-day window before the channel is actually closed by the blockchain itself. Within this window, if the other party broadcasts a more recent history to the blockchain[16], then the "cheater" loses all his funds on this channel, and the other party gets access to the full channel capacity. This is meant as an incentive not to unilaterally close channels with previous history. But the consequence is that if an agent has an active channel, he must monitor the network to make sure the other party does not cheat, and he cannot disconnect for more than 14 days, the safety window. Although not part of the current protocol specifications, it is also possible to delegate this monitoring to someone else, a *watchtower*, that would be paid to monitor channels for other agents. Of course, this is also achieved in a trustless fashion. Section 7 develops this idea.

3 Graph Analysis

Several implementations of the Lightning Network have been released in late 2017, and have been since then fully functional and growing. This section gives an overview of the current network structure.

3.1 Data

Since information about the network is needed by a node for finding a route across multiple channels, the state of the (public[17]) Lightning Network is available to any node. This information consists in a list of public nodes (public key, network address) and a list of edges or payment channels (source, destination, capacity, fees). Table 1 presents an overview of this dataset as of May, 2nd 2019.

Nodes with less than 2 edges are not very useful for routing as they essentially

[16]Thanks to digital signatures, the blockchain will be able to verify that the more recent history is actually more recent.

[17]Anyone can open a channel and not all channels are made public. Some nodes may be willing to run a parallel private network. We focus on the public network, available for routing.

	Full network	Nodes with more than 2 edges
Number of nodes	4,298	2,865
Number of edges	38,929	36,982
Total capacity	1,063.23 BTC	1,045.83 BTC

Table 1: State of the Lightning Network as of early May 2019

connect two nodes. We distinguish between them and the full network. Most analyses below will be on the subset of nodes than can be used for routing. Note that as of early May 2019 $1BTC \simeq 5,000EUR$, so this corresponds to a total capacity of around EUR 5 million.

In addition to this dataset, we have been able to get the history of opening and closing of payment channels almost from the first release (January 16th, 2018) until late August 2018. This has been privately saved by an individual who gave us access to this dataset. All channels opening and closing transactions are broadcasted to the blockchain but, from an external point of view, they cannot be differentiated from other transactions. Therefore it is easy to check if a channel-related transaction is indeed included in a bock, but we cannot make sure that all channel-related transactions are in this dataset. Table 2 presents some characteristics of this dataset.

	Value
Number of channel opened	35,809
Number of channel closed	24,368

Table 2: History of channels opening and closing from mid-January 2018 until late August 2018

This dataset will be used to analyze dynamic properties of the Lightning Network. In generative graph theory, the graph is obtained by starting with an initial set of nodes and adding successive node(s) and edges each round. In cases like *social network* or *citation network* this is generally not an issue but on the Lightning Network, edges (payment channels) can also be removed. Since the network is in a very early stage at the time of writing, some people are conducting some experiments by opening/closing channels. Table 3 separates channels by how long they stayed open.

	Value	Percentage
Number of channels closed	24,368	100%
Number of channels opened for less than 1 week	11,382	46,71%
Number of channels opened for less than 1 month	19,123	78,48%

Table 3: Dealing with closed channels

We see that around 78% of closed channels were opened for less than a month. At this stage it is likely that those channels were opened for testing purposes. For the remaining of this section we simply exclude the closed channels.

3.2 Static properties

A number of measures are used to describe graphs, but we can start by visualizing the current graph and qualitatively inspect its structure. Figure 1 and figure 2 show the state of the Lightning Network as of early May 2019. We can observe that a few nodes seem to be highly connected but most nodes seem to be well connected. On figure 2, we can clearly see all nodes with a single or a couple of payment channels surrounding the main network in the middle, the one that can be used for routing. For the remaining of this subsection, we remove nodes with 2 or less payment channels. Indeed, such nodes do not really bring any value regarding routing ability, therefore to measure and discuss some static properties of the network, we prefer to concentrate on the "useful" part of the network.

Figure 1: Vizualization of the Lightning Network (1) (Downloaded from *https://explorer.acinq.co* on May 4th, 2019)

3.2.1 Centrality

Centrality is a measure of a node's importance and can be expressed by different indicators. In this section we focus on degree centrality and the betweenness-centrality as defined by Freeman (1977).

The degree centrality of a node n is defined as the proportion of nodes it is connected to. Figure 3 presents the distribution of degree centrality (on a log-scale) in the Lightning Network. The distribution looks like a power law, which we will formally test in a later section. The highest value is 0.4613 which indicates that a node is connected to 46.13% of the graph.

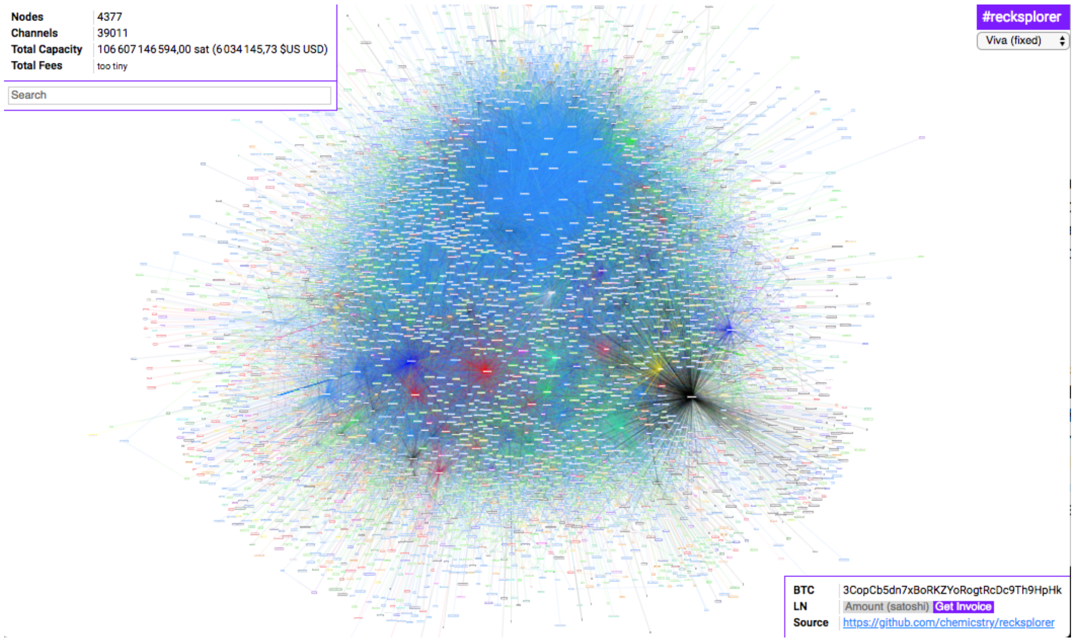

Figure 2: Vizualization of the Lightning Network (2) (Downloaded from *https://rompert.com/recksplorer/* on May 4th, 2019)

The betweenness-centrality of a node n measures how often the node n is part of the shortest path between two randomly selected nodes. This is a lot more relevant given the context of routing payment through the network. If a node were to be on the optimal route for too many transactions, it will start to have power over the network which obviously can be an issue. Here the shortest path is defined as the actual shortest path in terms of number of intermediaries but keep in mind that on the Lightning Network the "shortest path" will be computed regarding the fees associated with each intermediary. A rational user will select the cheapest route rather than the actual shortest one. However, the fees are quite low so far so the cheapest route will be very close to the shortest one. The betweenness-centrality, $c_B(v)$, is defined as

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s, t.

Figure 4 presents the distribution of node betweenness centrality (on a log-scale). It still looks like a power law even if the order of magnitude is much different for this measure. The highest value is 15.71% meaning that 15.71% of all of the shortest paths between nodes have the most important node on their path, this is also referred to as the Central Point Dominance. A low Central Point Dominance is preferable in this kind of network because it indicates that no node can suddenly break more

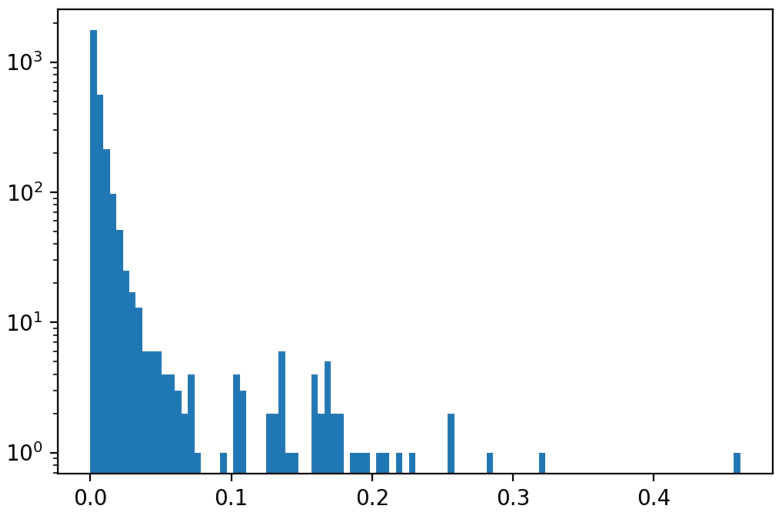

Figure 3: Distribution of the degree centrality (log scale)

than 15.71% of the optimal route on its own.

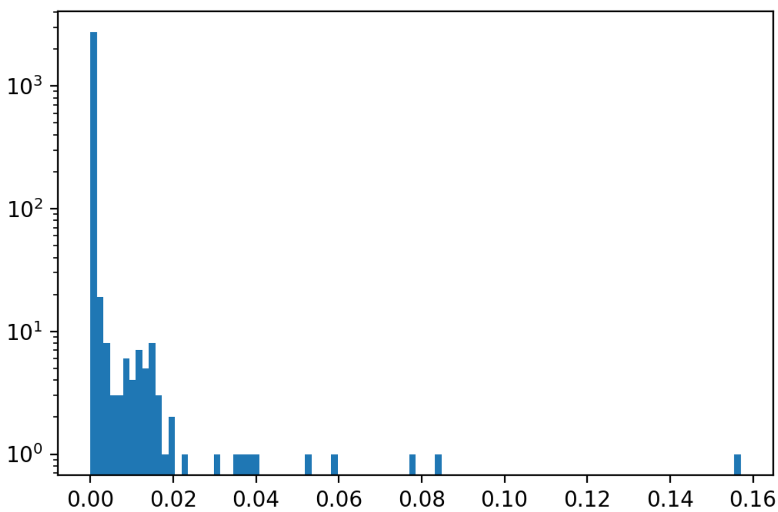

Figure 4: Distribution of the betweenness centrality (log scale)

3.2.2 Clustering

Another important measure for the Lightning Network is the level of interconnection within the graph. We will see that if the network is well interconnected, it will be easier for nodes to rebalance their channels and then keep routing payment across the network. It will be a key part of our model. In terms of measure, we can use the clustering coefficient measure as a proxy for interconnectedness. This measures the number of triangles through a specific node relative to the node's degree. Figure 5 presents the distribution of nodes clustering coefficients. We can see that the distribution of clustering coefficients is quite wide. Indeed, a lot of nodes have a quite high clustering coefficient. This is surprising because, as we previously saw in figure 4, the betweenness centrality is distributed like a power law. This suggests that nodes tend to prefer to be well interconnected within the graph even if they are not so often on the shortest path. The model will later show why nodes tend to favor such situations. This has to do with channel capacity rebalancing. It is cheaper for nodes to rebalance their channels if they are well interconnected (in the exact sense of the clustering coefficient).

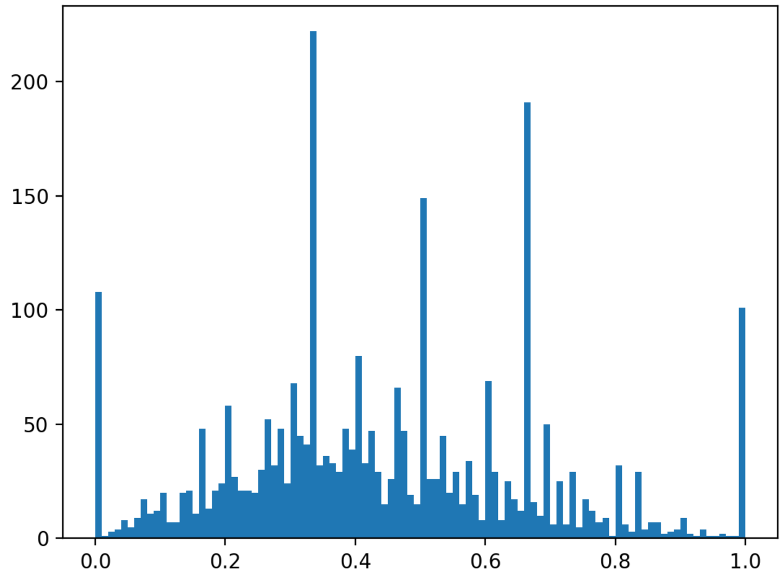

Figure 5: Distribution of clustering coefficients

3.2.3 Articulation points

Resiliency against the failure of one node is also a relevant concept. In a sufficiently high connected graph, no single node failure would have an impact on the ability of nodes to find a route to any other node. To measure that we use the notion of articulation point.

The Lightning Network is a connected graph, meaning that any node can reach any other node. A node n is an articulation point if the removal of n yields a disconnected graph, therefore preventing some node to reach the others. This is obviously very bad for the payment network. With the data we had we found that there are 0 articulation points.

From this point of view, the network seem well interconnected, and resilient to attack on one particular node.

3.2.4 Network capacity

A specific characteristic of the Lightning Network is the capacity. Each payment channel, edge in the graph, has a maximum capacity[18]. The distribution of total capacity per node gives an indication on how liquid the node is, or otherwise stated how likely it is to actually route transaction given a level of interconnectedness. Figure 6 presents the distribution of total capacity per node.

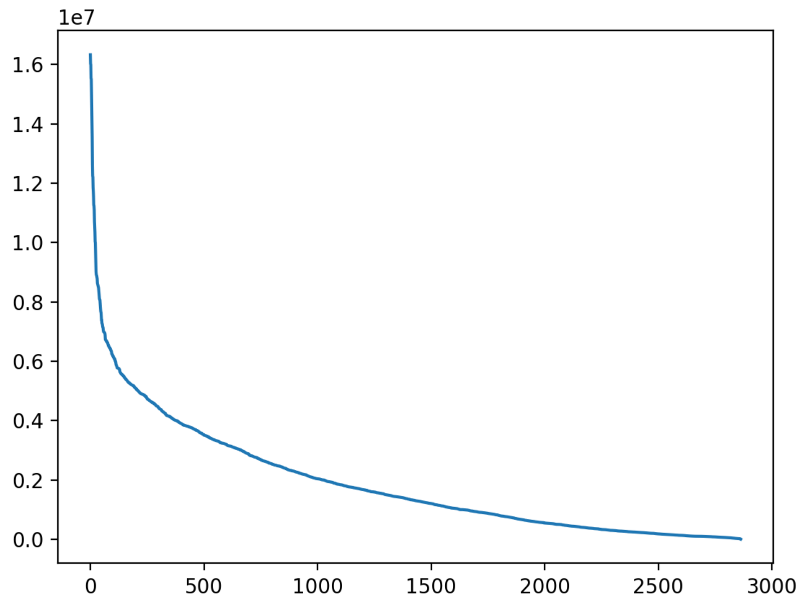

Figure 6: Distribution of average capacity per node

We can observe that a few nodes have a very high capacity and that the distribution looks like a power law. We previously saw that the distribution of centrality also looks like a power law. It turns out that the average capacity per channel is rather constant. Therefore the nodes with a high number of edges are also the ones that, on average, have a higher overall capacity. No nodes concentrate a lot of liquidity in a few payment channels. We see this graphically in figure 7 that presents a view of total capacity and number of payment channels on the same plot.

[18]There are actually capacities for both ways, *in* and *out*, but for privacy and security reasons there are hidden. In practice, a node select a route that it likely to have sufficient capacity, and if it fails, it tries another one, and so on.

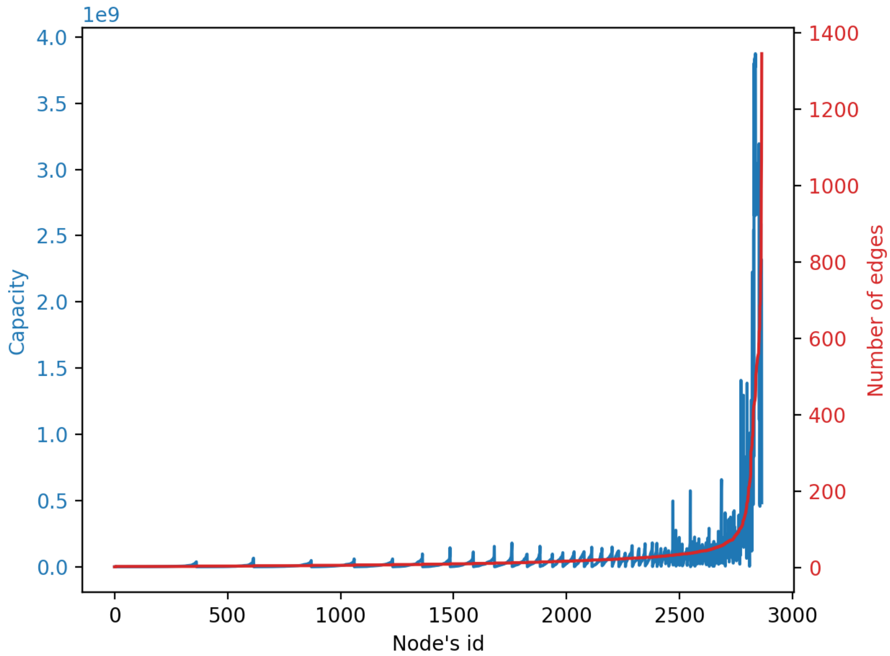

Figure 7: Distribution of total capacity with the associated number of edges

3.3 Dynamic properties

The static properties of the Lightning Network highlighted in the previous section tend to show that this network behaves like a scale-free network. This kind of complex network is best understood using generative models. Unlike random graphs, generative models feature the notion of node arrival as well as edges being added over time. We use the historical dataset of channel's openings and closings. Note that this dataset covers the period between January, 16th 2018 and August, 22nd 2018. This was the early stage of the Lightning Network, but we can still derive some interesting results as well as a method for later replications of this analysis.

3.3.1 Actual algorithm used in the Lightning Network

At this point, it is interesting to understand the actual behaviour of nodes on the network. The Lightning Network is actually no more than a standard of communication, a protocol, called *BOLT*. Therefore, anyone can, and is encourage to, develop his own application that will follow the *BOLT* standards[19]. There are currently 3 major implementations: *Eclair* (made by ACINQ), *c-lightning* (made by Blockstream) and *lnd* (made by Lightning Labs). Anyone willing to join the network can download any of those 3 clients and be sure about compatibility among BOLT-compliant clients.

Although not being part of the BOLT standards, interestingly all of these clients

[19] As defined in *https://github.com/lightningnetwork/lightning-rfc*

feature an *autopilot* mode when first starting the node. If the node is fired with that option, it will try to connect with some nodes in the network using a preferential attachment algorithm similar to Barabasi and Albert (2001). In the following, we estimate this preferential attachment model as well as Bianconi and Barabasi (2001), a more general model that introduces a node-specific fitness parameter in the edge probability between two nodes.

3.3.2 Models and method

The model developed by Barabasi and Albert (2001) allows for a node centrality to depend on the degree of this node, the more a node has edges the more likely it is for a new node to connect with it. This is called *preferential attachment* (PA) and is often used to model social network, or other scale-free networks.

The network begins with a set m_0 of initial connected nodes. Time is infinite, and at each point in time a new node gets added to the network and connects to $m < m_0$ existing nodes with a probability proportional to the current degree (number of edges) of each node. The probability p_i that the new node is connected to node i is

$$p_i \propto \frac{k_i}{\sum_j k_j}$$

where k_i is the degree of node i. Barabasi and Albert (2001) shows that under such circumstances the degree distribution across the network, $P(k)$, follows

$$P(k) \propto k^{-\gamma}.$$

It already looks like it should be the case regarding figure 3 or figure 4, but we will check that this is the case and estimate the value of γ, the parameter of the distribution.

As a more general version of this model, we can include another parameter in the probability of connecting to a node. Bianconi and Barabasi (2001) add a node-specific parameter, called the *fitness*, meant to represent the strength of the node. In the previous model, there is a natural tendency of earlier nodes to be the largest, because the older a node is, the more connection it has, which will trigger even more connections. To represent the ability of some nodes to enter the network at a later stage and still be able to be among the largest ones, a fitness parameter, η_i, is assigned at a new node i drawn from a distribution. The probability p_i that the new node is connected to node i is then modified to be

$$p_i \propto \frac{\eta_i k_i}{\sum_j \eta_j k_j}$$

Bianconi and Barabasi (2001) show that the network degree distribution depends on the distribution of the fitness parameter.

To estimate both models, we use a method called *PAFit* introduced by Pham, Sheridan and Shimodaira (2015) which consists in a non-parametric maximum likelihood estimation of the preferential attachment in temporal complex networks. The method goes as follows. Define the generic probability that a new node will connect to node i by

$$p_i(t) \propto A_{k_i(t)} \times \eta_i$$

where A_k is a function of the degree of node i at time t, $k_i(t)$. We briefly describe this method which is a Bayesian estimation, and we encourage the reader to dive into the *PAFit* paper for a more detailed explanation. The method is essentially a maximization of the log-likelihood of the most general model with suitably added regularization terms to avoid overfitting. Those regularization parameters control the amount of regularization for both the preferential attachment and the fitness. The dataset is split into a *learning sample* and *testing sample*, and a grid of *regularization parameters* is built. Then we compute a *Maximum-a-Posteriori* likelihood for each couple of *regularization parameters*. The estimates of the regularization parameters and the distribution are then the one that maximizes the overall likelihood.

3.3.3 Results

We first estimate, also with the *PAFit* method, both the Barabasi and Albert (2001) model with only preferential attachment, and the Bianconi and Barabasi (2001) with fitness. Table 4 presents the estimation of the preference attachment function when *i*) it is estimated alone, and *ii*) when estimated jointly with the fitness.

	$P(k)$	
	Pref. Attach. Alone (1)	Pref. Attach. & Fitness(2)
γ	0.973895	0.5730294
	[0.026]	[0.0068]

Table 4: Estimation of the attachment function (*standard deviation in brackets*)

For comparison purposes, table 5 presents different values for the power coefficient, γ, for several kinds of networks. Estimating the PA function alone yields a power coefficient close to 1, but when it is jointly estimated with the fitness distribution, the coefficient drops to 0.57. This tends to show that the fitness is actually important in the Lightning Network. In each case, the coefficient γ is smaller than most *real-life* networks, so it seems that the current degree of a node is less important, on average, to predict future links with new nodes.

Network	γ_{in}	γ_{out}	Reference
World Wide Web	2.72	2.1	Broder et al. (2000)
Movie actors	2.3	2.3	Barabasi and Albert (1999)
Silwood Park (food web)	1.13	1.13	Montoya and Sole (2000)

Table 5: Estimation of γ_{in} and γ_{out}, which are respectively the exponent of the degree *in* and *out* distribution for a directed graph

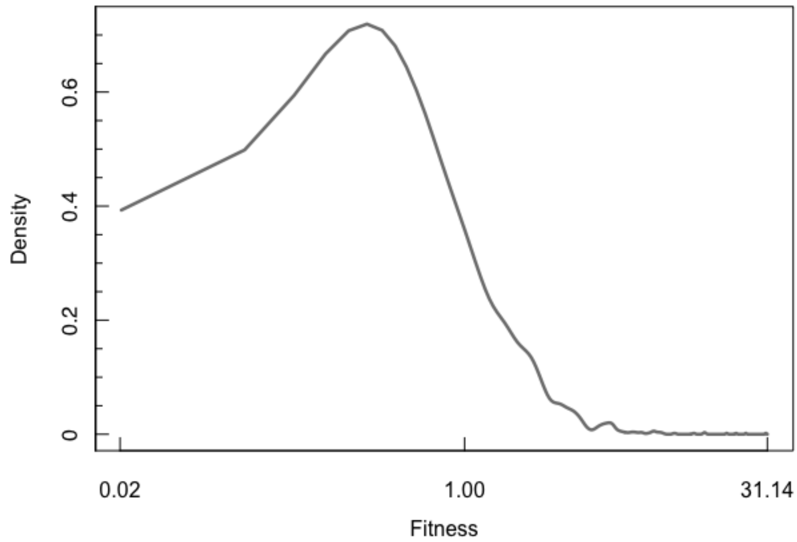

Figure 8: Distribution of Fitness parameter

The PAFit method also returns the fitness distribution, as presented in figure 8. We can see that most nodes have an estimated fitness below 1, and the distribution is quite wide. The wideness of the distribution means that the fitness is an important determinant of the network's topology. If all nodes had the same fitness, it would have suggested that all nodes behave the same, however we see that some nodes have a tendency to connect more, and to gain more importance in the network.

Overall, this suggests that most nodes are not running in *autopilot* mode for link (payment channel) creations. Some nodes have a natural tendency to be more important in the network, and decide with who and when to open a payment channel. The point of this paper is to study the behaviour of nodes on this market. We will see in later sections, why some nodes may be willing to connect more or less than with only the preferential attachment algorithm.

4 The Model : Homogeneous Nodes

This section presents the general version of the model featuring a homogeneous cohort of nodes. Because all nodes are the same, it is also a way to analyze the incentive of one node on the network.

4.1 General setup - Exogenous fee

The model features a network of payment channels, $G = (V, E)$, taken as exogenous, with V being the set of nodes with cardinal N and E being the set of payment channels among them, with cardinal M. A payment channel is defined as a bilateral contract in which both parties have to self-manage the allocation of some value "locked" on the contract. We assume that this network is backed by a secure enough blockchain, which means that agents (nodes) do not perceive any counterparty risk regarding payment channels since any amount deposited in the Lightning Network is escrowed on the secure blockchain. Recall that even if they are bilateral channels, thanks to some cryptographic tricks[20] it is possible to select a route of payment channels from the sender to the receiver so that no one in between has the ability to block the payment and steal value. We assume that no one will try to steal any funds, hence whatever fund an agent locks on the chain, there is no risk of loss. As explained in section 2, the closing of a payment channel can be made bilaterally or unilaterally, and here we abstract from any issues regarding channels monitoring, that is when a node wants to close a channel, it is done at no other cost than the fees and delay inherent to the blockchain itself, and it does not incur monitoring costs.

We assume that there is an exogenous cohort of buyers and sellers of goods and services that are willing to exchange value and, to simplify, have to use the payment network G for that. We are interested in the behaviour of a profit-maximizer node $i \in [1, N]$ among the V nodes[21]. In order to route payments, node i needs to open bilateral payment channels with at least 2 others nodes and fund those channels.

We call d the number of payment channels opened by node i, and $x \geq 0$ the aggregate amount of funds locked inside those d payment channels. We assume there is a quadratic opportunity cost of capital for node i to leave x inside the d channels, and denote it by $c(x)$. Recall that the Lightning Network is a public network, and in particular it may contain adversarial nodes or potential thieves might just be monitoring the network. Therefore by having a lot of money locked

[20] These are called *Hash TimeLock Contract*. If Alice has a payment channel with both Bob and Charlie, Bob can pay Charlie through Alice. For Charlie to claim the payment coming form Alice, she needs to make public the secret that Alice needs to claim the payment from Bob. Either both transactions are completed or nothing happens.

[21] Since we are only interested in this node, we might omit the subscript i.

inside those payment channels, a node will start to look like a potential target. Note that the security of the blockchain itself or the Lightning Network protocol is not an issue here, the risk we are referring to is an outside risk. Indeed, an attacker could use the network data to construct a list of public IP-address associated with the amount of funds locked in payment channels for each address, and therefore could use traditional hacking tools to try to steal the private keys associated with a given address in order to steal the funds. Therefore the node would have to invest more in cyber-security in order to protect its funds, hence the quadratic form. The following proposition summarizes this idea.

Assumption *The opportunity cost for node i to leave x in payments channels, called $c(x)$, is defined by*

$$c(x) = \frac{x^2}{2} \tag{1}$$

In order to route transactions, a node will have to open channels with existing nodes. Recall that the originator of the transaction will use the graph data to pick the optimal route. If a node wants to increase the volume of routed transactions, it will have to increase its interconnectedness in the sense of the probability to be on the optimal route for a random payment[22]. Instead of having to keep track of the whole graph structure, we define by $\lambda > 0$ the interconnectedness of node i. By assumptions, the case $\lambda = 0$ corresponds to node i not being connected to the graph, and the extreme case $\lambda \to \infty$ corresponds to node i being a "central point" on the graph. This notion is purposely loosely defined, but being a central point refers to situation in which all transactions would have to go through node i. The scale is arbitrary, and we will specify later how the expected revenue grows with λ but the idea is that node i will choose a level of interconnectedness λ and in order to maintain it through time, it will have to open and close channels depending on the evolution of the network structure. Indeed, if payments channels are closed in other parts of the graph, that would affect indirectly the interconnectedness of node i, hence the induced dynamic management of payment channels. We assume the cost of maintaining this interconnectedness level λ to be linear. The following proposition summarizes this idea.

Assumption *The cost of maintaining λ interconnectedness level, $\beta(\lambda)$ is defined by*

$$\beta(\lambda) = \lambda \tag{2}$$

[22] Recall that the optimal route is the cheapest one, but here we are referring to the interconnectedness everything else being equal, in particular the routing fee.

In practice, a node will need to fund all his channels independently. As transactions come through, channels balances will change. A channel balance is updated as soon as a transaction is routed. And because, from the blockchain point of view, the two legs of the routed transaction are 2 separate contracts, they cannot offset any funds from one another even if both channels are owned by the same agent[23]. Remember that from the blockchain point of view, the Lightning Network is just a bunch of bilateral contracts that are opening and closing with different allocations. Imagine a node i always makes a payment to node j and never the opposite, then regardless of the original allocation the capacity of the channel will be driven toward node j. At this point node i cannot make a payment anymore (not enough funds on his side of the channel), but node j can, if he wants, make a payment back to node i. Nodes ability to keep channels well balanced will be an important determinant of their profits (from the individual perspective) and even overall liquidity of the network (from the network perspective). There are 2 main ways to get around this issue. First, the option to broadcast a "refill" transaction to the blockchain (with fees and delays) is always available, at a cost. Second, if the network is "well connected" it might be possible for a node to pay itself over the Lightning Network choosing a route that decreases his highly loaded channels and increases his poorly loaded channels, we call this *rebalancing*. We assume this cost of *rebalancing* to be included in the cost of maintaining a given interconnectedness level, $\beta(\lambda)$. More specifically, the cost of rebalancing would be driven by the number of loops that are available to node i so that it can rebalance its channels by paying itself on the Lightning Network. We assume that this number of loops is sufficiently correlated to the interconnectedness level λ so that the cost savings induced by rebalancing is included in the linear form of the cost of maintaining a given interconnectedness level λ.

On the revenue side, the node essentially collects routing fees for forwarding payment across the Lightning Network. When a node initiates a payment, it selects a route based on the public information of the Lightning Network, namely the list of public payment channels and the fees that each node set. We abstract from some minor complications regarding routing[24] and focus on pure optimal routing based on fees minimization.

[23]Or at least it is costly. Recall that everything is always possible when you broadcast transactions to the blockchain, at a cost. We use the term "impossible" to mean impossible on the Lightning Network.

[24]Routing is not perfect on the Lightning Network. Take for instance a sender that selects a node i on a route, by the time the payment reach the actual node i the state of the payment channel might have changed. This would result in a payment rejection and the sender would have to find another route.

We are interested in what happens with the next transaction to be routed through the Lightning Network. An end-buyer contracts with an end-seller for a good or service and decide to pay through the Lightning Network. We define by $p(\lambda)$ the probability that node i is selected as part of the optimal route for this next transaction. We assume this probability is directly linked to the interconnectedness level of node i, λ, and follows an exponential distribution. This assumption implies that the term "optimal route" should be understood as the optimal route based only on the interconnectedness level λ and is therefore not amount-dependent. The real optimal route would be such a route in which all intermediary nodes have enough funding to actually route the transaction.

$$p(\lambda) = 1 - e^{-\lambda}$$

Once the node has been selected as part of this optimal route, he will be able to route the payment if and only if he has enough funds in his channels. We define the amount of the next transaction by y and assume it follows an exponential distribution, $F_y(y)$, with a probability density function $f_y(y) = F_y'(y)$.

$$f_y(y) = e^{-y}$$

When the node is selected as part of the optimal route, and when it has enough funds to route the payment, it will earn a routing fee, $R > 0$. This routing fee does not depend on the amount being routed[25]. We take the routing fee, R, as exogenous. Actual nodes on the Lightning Network are actually choosing fees but for now we abstract from that, in section 4.2 we will discuss implications of allowing the node to select the routing fee.

The node will maximize his expected revenue by solving for (x^*, λ^*).

$$(x^*, \lambda^*) = \underset{x \geq 0, \lambda \geq 0}{\arg\max}\ p(\lambda) \int_0^x R\ f_y(y)\ dy - \beta(\lambda) - c(x)$$

If we solve the integral and plug the cost functions and the probability, we can rewrite this maximization program as follows.

$$(x^*, \lambda^*) = \underset{x \geq 0, \lambda \geq 0}{\arg\max}\ \left(1 - e^{-\lambda}\right)\left(1 - e^{-x}\right) R - \frac{x^2}{2} - \lambda \tag{3}$$

The node solves this program and we have the following lemma.

[25] On the Lightning Network, the fee does not depend on the amount being routed. From a theoretical point of view, the fee could be made amount-dependent as transferring large amount of money could unbalance channels. Here we assume this cost is included in the cost of maintaining a given interconnectedness level, $\beta(\lambda)$.

Lemma 1 *(Unicity) On the Lightning Network, a profit-maximizer routing node chooses the optimal interconnectedness level,* λ^**, and the amount of funding to lock in payment channels,* x^**, and there is a unique solution* (x^*, λ^*) *that maximizes his expected payoff.*

By lemma 1, we know that the node's objective function behaves well. Intuitively this makes sense because the node's revenue is concave with respect to both control variables. The cost function for the interconnectedness level and the channel's funding however, are respectively linear and quadratic. The remaining of this section provides an analysis of the solution and its implications for the Lightning Network.

In this first general setup, the routing fee is taken as exogenous. This could represent a highly competitive market for which nodes are price-taker, or it could represent a situation of a new node entering the Lightning Network and using the network's current average routing fee. The next section will discuss implications of endogenous routing fees.

Here, both the cost of maintaining a given interconnectedness level and the opportunity cost of capital act like fixed costs, they essentially have to be incurred before the transaction is actually routed through the node. Therefore the node needs to make enough (expected revenue) otherwise it will be better off exiting the market and playing $(x^*, \lambda^*) = (0, 0)$. This idea is summarized in the following lemma.

Lemma 2 *(Exit condition) There is a minimum routing fees,* $\overline{R} > 1$*, under which the node exits the network, and chooses*

$$(x^*, \lambda^*) = (0, 0).$$

The very structure of the Lightning Network yields this exit condition in equilibrium. In equilibrium, and in order to operate, the nodes want a routing fee R greater than the minimum required routing fee $\overline{R}$. Compared to traditional production models, there is no variable cost and only fixed costs. Moreover, the revenue generated by routing a transaction is independent of its amount and there is no quality differentiation. This particular setup can be thought of as a lottery. A transaction is going to come through the network and all nodes are trying to get their hands on it[26]. They can do two things to increase the likelihood of being on the optimal route (*i.e. the selection likelihood*) : *i*) invest in interconnectedness,

[26]Note that this is not the usual type of lottery as there could be several "winners", i.e. all nodes selected as part of the optimal route.

through β, and *ii*) invest in payment channels through x. They do that by trading off the increase in selection likelihood and the associated cost. If the lottery prize, the fee R, is lower than the minimum required fee, $\overline{R}$, even increasing a little bit the selection likelihood will be too costly and the node will simply prefer not to participate in this lottery and keep its selection likelihood at zero. This lottery story accurately describes the Lightning Network from the point of view of a single node and considering the marginal transaction.

Because the routing fee is known to the node when it makes investment decisions, it is also interesting to study how the optimal investment is affected by different routing fees. By lemma 2, we already know that for a low enough fee, the investment in interconnectedness and in payment channels will be null. The following proposition describes the node's investment with respect to the current routing fee, R.

Proposition 1 *(Complete solution) Let's define by $f : R \mapsto x^*_R$ the function that maps possible values of the routing fee R to equilibrium values of the amount funded in payment channels x^*_R. We have that :*

- *The function $f(R)$ is null for $R < \overline{R}$;*
- *For $R \geq \overline{R}$, the function $f(R)$ is strictly increasing and concave ;*
- *Figure 9 is the graph of the function $f(R)$;*
- *The optimal interconnectedness level λ^* is given by*

$$\lambda^* = ln\left(1 + x^*\left(e^{x^*} - 1\right)\right). \tag{4}$$

There are no explicit solutions for the equilibrium quantities (x^*, λ^*) but fortunately we can describe the solution enough to analyze it. As previously stated, there is a minimum required fee under which $(x^*, \lambda^*) = (0, 0)$. Right above this threshold, the optimal funding amount is quite sensitive to changes in the routing fee. For larger values, because the function is concave, this sensitivity decreases. We will see in later sections that the routing fee is actually unlikely to be very high. This suggests that the network liquidity, as defined by the total amount locked inside all the network's payment channels, is likely to have a high sensitivity to short term (if any) variations of the average routing fee.

Note that the current state of the Lightning Network does not allow for any test for the funding-to-fee sensitivity. Indeed, as of early-2019, the network is essentially maintained by people trying to help the development of the network. As previously

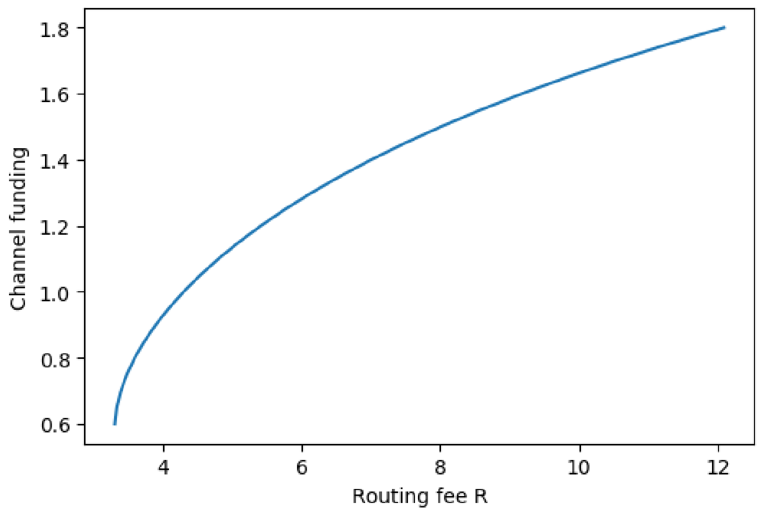

Figure 9: Graph of the function $f(R)$ representing the optimal amount invested in payment channels with respect to the current routing fee

stated, the average routing fee is very low, and no one is really trying to benefit from transactions' routing at the moment.

To analyze the sensitivity of the interconnectedness level to changes in the routing fee we need proposition 1 which provides the equilibrium relationship between the interconnectedness level, λ, and the channels' funding amount, x, with equation 4. Figure 10 represents the graph of the function $\lambda^*(x^*)$. Loosely speaking we can see that this function behaves like a linear function with slope higher than 1, except for small values of x. This suggests that the interconnectedness-to-fee sensitivity will be even greater than funding-to-fee sensitivity. However, note that it might be complicated to interpret the interconnectedness in a dynamic context. This analysis should be taken as sensitivity to initial conditions, whether the interconnectedness level will adjust dynamically to changes in the average routing fee is out of the scope of this section. By looking at the market from a higher perspective, the section 4.2 provides some insights regarding this issue.

4.2 Two stages model - Endogenous routing fee

So far we have taken the routing fee, R, as exogenous. In reality, nodes set the fee R and are using it as a competition device. In this subsection we endogenize the price by analyzing a more general game in which nodes also compete on price.

The routing activity on the Lightning Network can be thought of as a *Bertrand competition* model with capacity constraints. In line with Kreps and Scheinkman (1983), hereafter KS83, such a game can be described with a 2-stages competition model. In the first stage, firms invest in capacity, then at the second stage all firms capacity become public and they engage in a price competition. On the Lightning

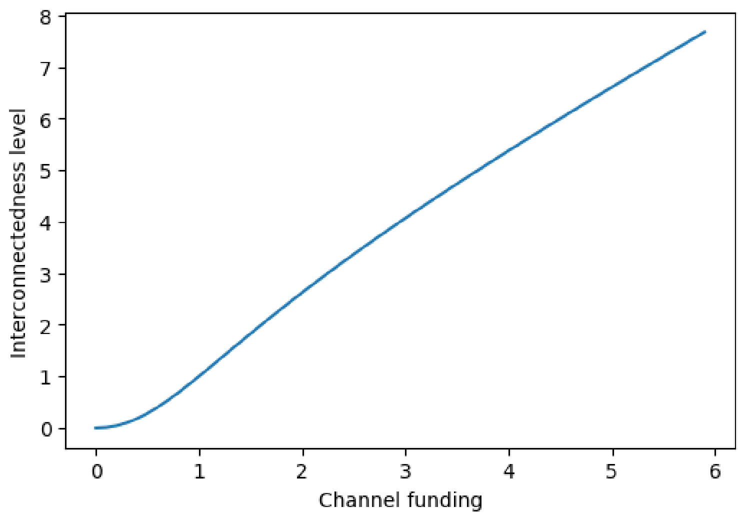

Figure 10: Equilibrium relationship between the interconnectedness level and the channels' funding

Network, the interconnectedness λ, and the amount funded in channels, x, can be seen as capacity constraints, let's call them the *network capacity* and the *channel capacity* respectively. Indeed, once a node has set (λ, x) it will only be able to route so much transactions[27]. The previous game can therefore be seen as the first stage of such a game. Nodes first compete on network and channel capacity, using the exact same setup as in the previous subsection, then they enter a price competition in *Bertrand* style.

There are some important differences with respect to a traditional product market. Like in KS83, the marginal production cost within the capacity is null[28], but the service sold is not homogeneous. Because we are analyzing a network, a specific node cannot be replaced by any other node, so the service they provide is not homogeneous strictly speaking. However, before a transaction is initiated, all nodes with a given interconnectedness level look the same, therefore we say that the routing service is homogeneous given a particular value of interconnectedness level. As soon as an end-buyer has been assigned an end-seller, and wants to transact, each node becomes different from all others, depending on how likely it is to be on the optimal route, but nodes cannot anticipate that. Moreover, due to onion routing, nodes cannot know their position within the optimal route, nor whether there is an alternative route, so they cannot differentiate themselves with different service apart from the interconnectedness level. We might be tempted to think that the

[27] Again, nodes can always update dynamically their interconnectedness level and payment channel funding by going through the blockchain, at a cost. However, for the sake of the argument, we assume this cost is infinite.

[28] Note that this assumption is a simplification in KS83 but it is actually true within the Lightning Network. More precisely, the marginal cost of routing a transaction is the cost a sending a few data packets on the internet, which is very close to zero.

capacity constraint works here a little different than in KS83 because we assumed that the end-buyer would select another route if a node's capacity does not meet the transaction amount. In KS83, the consumer demands first to the lowest-pricing firm up to this firm's constraint, and the residual demand is going to the other firm. However, it is actually very similar thanks to *Atomic-Multi-Payments*, with which the end-buyer is able to select several routes and split the amount among them so that the end-seller receives all payments or nothing. This is out of the scope of this paper but bear in mind that the channel capacity constraint behaves very much like a traditional capacity constraint like in KS83.

Another important difference is the interpretation of demand. Contrary to KS83 we do not make any assumptions regarding the demand for routing. We can focus only on the next transaction, because we only care about the initial decision of nodes, only the expected profit matters, which does not depend on the actual realisation of demand. On the Lightning Network, the profit of a node does not depend on the amount of the transactions routed, the fee is not amount-dependent. Also the volatility of the demand is already included in the channel and network capacity constraints, because with a given (x, λ) the node can only expect so much revenue.

Lastly, we assume the following for the price discrimination. For two nodes $i \neq j$, such that $x_i = x_j$ and $\lambda_i = \lambda_j$, then the node setting the lowest fee R is getting the next transaction, the other node expects no return at all. This represents the fact that nodes with similar capacity constraints are, *ex-ante*, the same. Because they are *ex-ante* homogeneous, Bertrand competition applies to them, and drives the price down in order to attract transactions. This idea is summarized in the following statement.

Assumption *(Price competition) Take any two nodes $i \neq j$, such that $x_i = x_j$ and $\lambda_i = \lambda_j$. Assume, without loss of generality, that they announce routing fees $R_i \geq R_i$, then*

- *If $R_i > R_j$: node j attracts the next transaction, and node i expects a null profit*
- *If $R_i = R_j$: node i and node j share the volume among each other, and both can expect a positive profit as defined in equation (3).*

Let's formally describe the 2-stages model. The first stage is exactly like the model in the last subsection, let's recall it briefly. There are N homogeneous nodes competing to route transactions. At the first stage, each of them sets an interconnectedness level and an amount to lock in payment channels. At the beginning of the second stage, all first stage choices are made public, and nodes announce a routing fee R_i. Then an end-buyer is matched with an end-seller and an optimal route is selected within the network. Each node still maximizes its profit at each stage.

At the second stage, the network topology is set and nodes are competing only on prices, taking what they know about the network as given. Let's define by Θ the information available at the end of the first stage.

$$\Theta = \left(\{x_i\}_{i\in I}\,;\{\lambda_i\}_{i\in I}\right)$$

To solve this game, we run backward by solving first the second stage subgame. For every outcome of the first stage game, we defined as the Θ-subgame, the price competition engaged by the nodes given the network state Θ, of which the resulting optimal routing fee is denoted by R^*. Since all nodes are homogeneous, they will take the same decision. Then we can restrict our attention to Θ-subgame in which all decisions are the same across nodes. Therefore, in this section, we can write $\Theta = (x, \lambda)$, and a (x, λ)-subgame.

In a (x, λ)-subgame, each node is maximizing its profit. Recall that nodes are solving this program under the *Price competition* assumption. In order for them to attract the next transaction (i.e. a non-zero volume), they need to take into account the fee sets by other nodes. The optimal fee set by node i, in the (x, λ)-subgame, denoted by $R_i^*(x, \lambda)$. We have the following lemma.

Lemma 3 *(Subgame) In the (x, λ)-subgame, price competition drives the equilibrium routing fee to low values and there is a continuum of Nash-equilibria regrouped into two categories :*

- *All nodes make zero profit and $\forall i$:*

$$R_i^*(x, \lambda) = \frac{x^2 + 2\lambda}{2\left(1 - e^{-\lambda}\right)\left(1 - e^{-x}\right)}$$

- *All nodes make a loss and $\forall i$:*

$$R_i^*(x, \lambda) \in \left[0; \frac{x^2 + 2\lambda}{2\left(1 - e^{-\lambda}\right)\left(1 - e^{-x}\right)}\right[$$

Note that this subgame can be viewed as a kind of prisoner's dilemma. The good outcome can be associated with zero profit for all nodes, and the bad outcome with a loss for all nodes. Cheating could be considered to be a price dumping strategy, to overtake the volume. However, a node does not benefit from "cheating", because its loss increases, so the "good" outcome is also a Nash-equilibrium. We will discuss the implication of having this continuum of equilibria once we solve the full game. To do this, we first restrict our attention to a subset of (x, λ)-subgame, and we establish the following proposition.

Proposition 2 *(Full game) There is only one Subgame Perfect Equilibrium (SPE) for the full game. The equilibrium is given by the following :*

- *Define* $\lambda(x) = ln\left(1 + x\left(e^x - 1\right)\right)$
- *All nodes make the same decision* (x, λ)
- *The optimal amount locked in payment channel,* x^**, is the unique solution to*

$$x^* = \left\{ x \in \mathbb{R}_+^* \ ; \ \frac{1 + x\left(e^x - 1\right)}{1 - e^{-x}} = \frac{x^2 + 2\lambda(x)}{2\left(1 - e^{-x}\right)\left(1 - e^{-\lambda(x)}\right)} \right\}$$

- *The optimal interconnectedness level,* λ^**, is given*

$$\lambda^* = \lambda(x^*)$$

- *The optimal routing fee,* R^**, is given by*

$$R^* = \frac{x^{*2} + 2\lambda^*}{2\left(1 - e^{-x^*}\right)\left(1 - e^{-\lambda^*}\right)}$$

- *No node makes a profit*

Given this cost structure, there is a single equilibrium outcome that consists in a strictly positive routing fee and no profit for the nodes. All nodes are homogeneous so they make the same decision. Section 5 explains the consequences of heterogeneous agents.

In KS83, they show that Bertrand Competition with capacity constraints yields the Cournot outcome. In our model, surely the equilibrium price is higher than the one with the Bertrand price without constraints, which would be zero, the marginal cost. However, the way the model is specified makes our analysis different. In particular, the demand is perfectly inelastic, which means that the users would demand the same amount regardless of the price. Also, the quantity demanded by the users does not impact the profit of the nodes (the routing fee is amount-independent). Those assumptions are far from the usual ones so the analysis of a pure Cournot game is not very relevant to the case of the Lightning Network.

The second subgame equilibrium is not an equilibrium of the full game, and we explain intuitively why this is the case. In the (x, λ)-subgame, setting the fee to zero is an equilibrium because nodes cannot change the capacity limit. In the full game, however, they are free to select the capacity they want, and because having a zero fee produces loss, the nodes would be tempted to lower their capacity investments to reduce their loss. This would yield the following $(x^*, \lambda^*, R^*) = (0, 0, 0)$. However, this point is not an equilibrium because the demand is completely unsatisfied and any node could jump into the network and start routing at any fee. The proof of proposition 2 goes on to demonstrate that any other point is not an equilibrium.

We've described a model to represent a homogeneous Lightning Network. The unicity of equilibrium basically ensures that the network is well functioning and can be used to route payments. Note that we are studying a payment network and in real life there would be other options for agents to transact. So any equilibrium in this "narrow" situation is contingent on the outside option of using another payment network. If the outside option is too low, the demand would be null and the network wouldn't exists. Whether this payment network would be stable and working in a more general setup, with other payment networks, is outside the scope of this paper.

The parameters we chose to model the network should be taken cautiously. Indeed, we model the interconnectedness of nodes with λ, which represents the probability that the node is on the optimal route between any two other nodes. With homogeneous agents, we saw that they all take the same decision in equilibrium, especially the same interconnectedness level. Such a network can have a very bizarre structure and, as seen in section 3, is not at all representative of the Lightning Network which is very concentrated in terms of node centrality (i.e. the interconnectedness level). With this simpler version in mind, the next section will establish results for a more general setup with heterogeneous agents.

5 Heterogeneous Agents and Network Structure

So far we have assumed a homogeneous cohort of nodes. In this subsection we analyze agents that are heterogeneous with respect to their opportunity cost of capital, $c_i(x_i)$.

On the Lightning Network, nodes could be run by very different kinds of agents. As previously discussed, one of the main risks of running a node is the payment channels funding security. We assumed that the network is powered by a secure enough blockchain so that nodes do not perceive counterparty risk. Although, on the blockchain, ownership is defined by private key ownership and nothing else, so nodes are subject to *private key* thief risk (or *security risk*). Indeed, the network status, including payment channel balances, is made public[29] so an external attacker could locate the *ip-address* of a node and deploy resources to hack into it and steal the amount locked in payment channels[30]. Having a large amount invested in payment channels would require to increase the security of the server itself, hence incurring another cost.

We assume that each node has a different ability to deal with security risk. Recall that we assumed that this risk is included in the node's opportunity cost of capital, hence its quadratic form. Therefore, we introduce some heterogeneity in the nodes' opportunity cost of capital to represent the difference in security risk. More specifically, let's assume that for each node $i \in [1, N]$, there is a value $\gamma_i > 0$ assigned by nature, and the node's opportunity cost of capital is now a function of γ (we shall omit the subscript i when we focus our attention only on node i).

$$c(x) = \frac{\gamma x^2}{2}$$

All the other assumptions remain. We describe how the equilibrium changes when γ changes like in the previous section, first the incentive of only one node with exogenous routing, and then the two stages game.

5.1 Exogenous fee

Let's first analyze the setup in which R is exogenously given. Nodes are maximizing their investment in channel and network capacity, (x, λ). The objective program of the node, i.e. equation (3), becomes the following.

$$(x^*, \lambda^*) = \underset{x \geq 0, \lambda \geq 0}{\arg\max} \left(1 - e^{-\lambda}\right)\left(1 - e^{-x}\right) R - \frac{\gamma x^2}{2} - \lambda \tag{5}$$

Note that the changes are small with respect to the last setup. Therefore, most of the results of the previous section still hold. What is interesting is to see where this γ parameter shows up in equilibrium. Existence and unicity still hold.

Recall that under a fee, $\overline{R}$, the node just prefers to exit the market and play $(x, \lambda) = (0, 0)$. This is still true but we can establish a link between the parameter γ and this fee, denoted by $\overline{R}(\gamma)$.

[29] Only part of the Lightning Network is actually made public. However, only this subgraph can be used to route payments for an anonymous user. Some private network can be deployed but will be used only by agents aware of this *sub-network*.

[30] The *ip-address* may not always be revealed. Some nodes are using TOR to basically hide their *ip-address* but still be able to route payments. However, the channel balance on the blockchain is assigned to a particular users. This is a way the node could be located, and it would require extra-caution for highly loaded payment channels.

Lemma 4 *When nodes are heterogeneous, the minimum required routing fee, $\overline{R}$, is an increasing function of the opportunity cost of capital, γ.*

$$\frac{d\overline{R}(\gamma)}{d\gamma} > 0$$

When a node is more constrained on its opportunity cost of capital, it makes sense that it requires a higher minimum fee to cover the increase in cost. The complete solution of proposition 1 is also still valid up to some parameter. Proposition 3 gives the main differences with respect to the previous section.

Proposition 3 *When nodes are heterogeneous, the function f that maps possible values of the routing fee R to equilibrium values of the amount funded in payment channel x^*, gets modified as follow.*

$$\forall \gamma_1 < \gamma_2 \;\; ; \;\; \forall R \geq \overline{R} \;\; ; \;\; \left.\frac{df(R)}{dR}\right|_{\gamma=\gamma_1} > \left.\frac{df(R)}{dR}\right|_{\gamma=\gamma_2} \tag{6}$$

If we denote by x^*_γ the optimal amount invested in payment channel for node i with opportunity cost parameter γ, we have

$$\forall \gamma_1 < \gamma_2 \;\; ; \;\; \forall R > \overline{R} \;\; ; \;\; x^*_{\gamma_1} > x^*_{\gamma_2} \tag{7}$$

Also, the equilibrium interconnectedness level, λ^*, is now given by

$$\lambda^* = ln\left(1 + \gamma x^* \left(e^{x^*} - 1\right)\right) \tag{8}$$

The second part of proposition 3, (7), says that for a given routing fee, R, a more constrained node will invest less in equilibrium, both in channel and network capacity, (x, λ). The first and last part of proposition 3, (6) and (8), tell us that when the nodes have a higher opportunity cost of capital, γ, the amount locked in payment channel is less sensitive to changes in the routing fee, and the interconnectedness level is less sensitive to changes in the funding amount. We can already see that the opportunity cost of capital will have a large impact on the structure of the network, through λ, and on its capacity, through x. We will discuss that once we solved the two stages game.

5.2 Two stages model - Endogenous routing fee

Let's now turn to the most interesting case, that is a two stages game with heterogeneous agents. In this section, we solve this two stages game and compare it with the homogeneous agents case. Again, most of the results still apply but we are interested in understanding where the opportunity cost parameter, γ, shows up.

We first have to clarify how the price competition works in a heterogeneous setup. We assumed in the previous section that for the same interconnectedness level, λ, a node displaying a higher price would not get any of the volume. It was enough for the homogeneous case because all nodes take the same decision in equilibrium. We assume that the service nodes are providing is differentiable with respect to the interconnectedness level. From the network point of view, nodes with a low interconnectedness level, λ, do not help the network as much as nodes with high λ. Therefore, they should be able to charge different price. And as a consequence, a node charging a higher routing fee could still attract some volume if it has a higher interconnectedness level. This idea is summarized in the assumption below.

Assumption *There is a constant exogenous number of transactions to be routed which, in equilibrium, is shared across all nodes with the lowest fee for a given interconnectedness level.*

We also need to define the notion of a *competitive network*. In this case, a *competitive network* is defined by a network in which barriers to entry are low, and a sufficiently big cohort of potential nodes is waiting outside the network in case there would be an opportunity. More precisely, a network with strictly more than one node for each interconnectedness level is also a *competitive network*.

Defintion 1 *A network is said to be competitive when either one of those 2 conditions are satisfied:*

- *There are always either zero or more than one nodes providing a given interconnectedness level*
- *There is an exogenous cohort of potential nodes outside the Lightning Network, that could jump in the network and if there is a profitable opportunity for any interconnectedness level.*

A competitive network ensures that the zero profit condition will hold in equilibrium. For the remaining we assume that the network is competitive. We will discuss in section 6 the potential consequence of a non competitive network.

Recall that we defined the Θ-subgame to be subgame in which nodes compete only on price, given all nodes channel and network capacity. Like previously, we first solve the last stage and then the full game. The setup is exactly like in the previous section except for heterogeneity in nodes' opportunity cost of capital. The routing fee is now a function of the complete information Θ, as nodes could take different decision in equilibrium. The following lemma provides the Θ-subgame equilibrium.

Lemma 5 *(subgame) In the Θ-subgame, with a competitive network, price competition yields low value for the routing fee. In particular, the same mechanics applies as in lemma 3, but for each different opportunity cost of capital. All nodes make at most zero profit, and in some case, they have a loss in equilibrium.*

The equilibrium continuum is a little more complex than in the homogeneous case, because there are a lot of different possible combinations, given all the possible opportunity cost of capital γ. Similarly to the previous section, we derive the equilibrium of the two stages game in the following proposition.

Proposition 4 *On the Lightning Network, when nodes are heterogeneous in their opportunity cost and in equilibrium, nodes use the price to discriminate among them for the interconnectedness they provide, and for each γ we have a unique equilibrium similar to the one in proposition 2. In particular, we have the following.*

- *Define $\lambda(x) = ln\left(1 + \gamma x\left(e^x - 1\right)\right)$*
- *All nodes with the same opportunity cost of capital make the same decision*

$$\forall i \neq j \quad if \quad \gamma_i = \gamma_j \quad then \quad (x_i^*, \lambda_i^*, R_i^*) = \left(x_j^*, \lambda_j^*, R_j^*\right)$$

- *The optimal amount locked in payment channel, x_i^*, is the unique solution to*

$$x_i^* = \left\{ x \in \mathbb{R}_+^* \quad ; \quad \frac{1 + \gamma_i x\left(e^x - 1\right)}{1 - e^{-x}} = \frac{\gamma_i x^2 + 2\lambda(x)}{2\left(1 - e^{-x}\right)\left(1 - e^{-\lambda(x)}\right)} \right\}$$

- *The optimal interconnectedness level, λ_i^*, is given*

$$\lambda^* = \lambda(x_i^*)$$

- *The optimal routing fee, R_i^*, is given by*

$$R_i^* = \frac{\gamma_i x_i^{*2} + 2\lambda_i^*}{2\left(1 - e^{-x_i^*}\right)\left(1 - e^{-\lambda_i^*}\right)}$$

- *No node makes a profit*

5.3 Implications for the network topology

Now that we have solved the full game, we interpret the equilibrium and derive some implications of this model for the network topology.

The main mechanics of the model can be expressed as follows. Nature assigns an opportunity cost parameter, γ_i, to each node i. Then each node sets capacity limits, in terms of interconnectedness level and funding capacity (x, λ), and a fee, R, it will charge for routing transactions through the network. This produces a network topology that is then used by end users (buyers and sellers) to select optimal routes for their transactions. The network topology will, of course, be an important determinant of the nodes' ability to make payment in an efficient manner. End users benefit from a well funded and well connected network, as they would be able to route payments at a lower cost[31].

In our model, the network topology is endogenous and fully depends on the nodes' optimal choices, (x^*, λ^*, R^*), and the only exogenous parameter is the opportunity cost of capital. By proposition 4 we know that the equilibrium is unique under the model assumptions, therefore there exists a unique function that maps the opportunity cost of capital γ to an equilibrium value for each control variable. Let's call those equilibrium functions $x^*(\gamma)$, $\lambda^*(\gamma)$ and $R^*(\gamma)$. Given the structure of our model, we cannot obtain closed form solution for those functions. However, it can be shown that the functions $x^*(\gamma)$, $\lambda^*(\gamma)$ and $R^*(\gamma)$ are monotonic[32]. At the network level, we can impose a distribution for the opportunity cost parameter, call it f_γ, that would give the equilibrium distribution for the channel capacity, f_x, the network capacity, f_λ, and the routing fee, f_R.

$$f_\gamma \rightarrow \begin{cases} x^*(\gamma) \\ \lambda^*(\gamma) \\ R^*(\gamma) \end{cases} \rightarrow \begin{cases} f_x \\ f_\lambda \\ f_R \end{cases}$$

Even without closed formulation for those equilibrium functions, we can provide some insights on the network topology. Note that we impose $\forall i \;\; \gamma_i = 0$. If the

[31] Note that the cost of routing depends on the capacity and the interconnectedness. It depends on the interconnectedness as end users would, on average, have to pick a longer route in a poorly connected network. If the capacity on a given route is too low, they can still use the network but at a higher price as they would need to split the payment in a multitude of payments that fit the capacity.

[32] We don't have any analytical proof, but simulations show that the functions are indeed monotonic.

opportunity cost were to be zero, the node would set an infinite channel capacity to maximizes its profit. Any $\gamma > 0$ will produce a finite equilibrium. Therefore, the support of the opportunity cost distribution, f_γ, will extend to the distribution of channel and network capacity, f_x and f_λ. The following proposition formalizes this idea.

Proposition 5 *(Upper/Lower bounds) Intrinsic characteristics of the Lightning Network create an upper and lower bound for nodes' the interconnectedness level, λ_i, and channels' funding, x_i, which corresponds to the minimum and maximum of γ_i.*

Because of the previous remark, we can show that there is an upper-bound on the channel and network capacity, (x, λ). It is complex to derive straight topology consequences but the fact that there is an upper-bound for the interconnectedness level and channel's funding already rule out some kind of network. For instance, a completely centralized network, in which a node (or a group of nodes)) gets to route all transactions, $\lambda \to \infty$, is never optimal given $\min_i \gamma_i > 0$. Similarly a completely fragmented network (i.e. low values of λ) would only come from a very high opportunity cost of capital and is therefore very unlikely.

It is complex to say much more given that we do not have any closed form for the equilibrium functions. However our model is able to consistently generate network topologies given an opportunity cost distribution. We leave a deeper analysis of the results from a network point of view to future research. Section 7 provides some other insights regarding future directions and extension of this model.

6 Bridge with the Real Lightning Network

In this section we present the main differences between our model and the actual Lightning Network, and discuss their implications on the equilibrium analysis.

First, the Lightning Network is still new. It was first introduced theoretically by Poon and Dryja (2016). After a couple of years, the first implementations were live from January 2018. Section 3 describes precisely the current state of the network, but note that it is overall still relatively small. Most active nodes are currently participating in the network for the sake of development and testing. The routing fees are overall close to zero, but the total volume is also low, no one is really trying to profit from the routing activity so far, as of early 2019. We have built a model in which heterogeneous nodes compete to route transactions. We shouldn't expect our model's results to hold with the current state of the market. Our model is more likely to predict nodes' behaviour in a highly competitive market in which the demand is also high. We leave for future research refinements of this model and empirical tests when the network would have grown sufficiently.

Second, our results are up to some proportional constant. In equilibrium, there is a direct relationship between the optimal routing fee and the investment in channel and network capacity, (x, λ), but to simplify the equilibrium analysis we didn't put any proportional constant in the cost functions. Recall that we specified $\beta(\lambda) = \lambda$ for instance. It is not likely that the real cost of maintaining λ as interconnectedness level is as such. A more realistic formulation could have been $\beta(\lambda) = \alpha\lambda$ with $\alpha > 0$ a parameter. The same would apply for the channel capacity cost, x. Such a formulation would have been able to get a more realistic equilibrium in terms of the variable's scale. However, we argue that it won't change any results of existence and unicity in equilibrium, under normal parameter conditions[33]. The global shape of the objective function won't change for instance, and therefore neither will the shape of *first order conditions*, etc. Equilibrium will prevail to a more general formulation. Note that this work of parametrizing the model would have to be done before any empirical work.

Lastly, our model relies on a fix exogenous demand, and we use traditional industrial organization literature to analyze this payment network. However, a payment network is different from a traditional product market. It would be more realistically represented by a needed service as it is a mean of payment allowing "real" transactions to happen on the physical market. Note that agents, end users, need to somehow pay the seller of any goods or services, but the Lightning Network is not the only option. Therefore, all our results are conditioned on the value of the outside option for end users. If the outside option is low enough, they would prefer to use it and the Lightning Network would probably die because of insufficient demand. Note that lemma 2 holds pretty much for the whole model. It says that if the routing fee R is too low, the nodes prefer not to participate in this market. The value of the routing fee could be generalized to the demand. If the demand is too low, nodes won't be able to extract enough value to pay for their expenses (channel and network capacity), therefore they would exit the network.

There can be two forms of outside options, the blockchain itself, or any other traditional payment networks (Visa, Mastercard, Paypal, etc.). Let's call the former *trustless* payment networks, while the latter would be the *trust based* payment networks as they require agents to trust a central party of any kind[34]. The first tradeoff

[33]Here, normal parameter conditions mean that the cost are not "too high". If the cost were too important, it will of course change the shape of the objective function, but those are unrealistic.

[34]Some might argue that high transaction costs are also inherent to those traditional payment network but there exists some service like Paypal that are essentially fee-less. The real difference between the Lightning Network and other payment networks is the trust that is required to use traditional payment networks.

is whether agents value the absence of trust in the network. If no one cares about the absence of trust, it is likely that the Lightning Network will be an expensive solution that another network could do charging less. The difference between the cost in a *trust based* network and a *trustless* one has to be lower that the benefit of not having to trust anyone. Note that this analysis does not take into account the demand for other reasons than pure transactional ones. If agents have other interest in holding cryptocurrency, they might be more likely to use such a *trustless* solution.

If it turns out that agents value the absence of trust, they still have another option to transact, namely the blockchain itself. Recall that the blockchain is always available to anyone to send and/or receive payments. The issue is that the fees on the blockchain are also dynamic and depend mostly on the blockchain characteristics like the time per block, or the size of each block. If the cost of transacting on the blockchain is low enough and if there are some issues on the Lightning Network, it could be cheaper to use the blockchain for *i*) a brief period of time, in such a case the Lightning Network would probably recover, or *ii*) an extended period of time, and in such a case the Lightning Network would probably collapse. However, this is quite unlikely, because even if the fees on the blockchain drop a lot, there is still the issue of confirmation time (it usually takes several confirmations - blocks - to make sure the payment is definitive) and this will never go below a few *inter-block* time, which is for instance 10 minutes on the Bitcoin blockchain[35].

7 Future Research

While computer scientists have been doing academic research on blockchain since the early days, it still remains a niche in other fields, especially economics and finance. This section provides a number of potential extensions of the model we presented and other future directions in general. Of course, there are also potential research on blockchain per se but we choose to focus on Lightning Network related issues. This is the first description and analysis of second layer protocols like the Lightning Network, so there are a large number of open questions, and we divided these remarks in two categories, the theoretical and the empirical side.

7.1 Theory

Let's begin with open questions on theory. This paper is an attempt to model a network of payment channels. We characterize some aspects of the network topology

[35]To fully confirm a transaction, it is considered good practice to wait around 5 to 6 blocks after the block containing this transaction. This significantly lowers the likelihood of *double-spending*.

but we do not have a proper endogenous network structure. Using graph and network theories should prove useful in filling this gap. However, it seems that graph theory needs to progress a little as, to the best of our knowledge, the Lightning Network properties do not seem to fit any usual graph frameworks. Indeed, mathematicians have worked extensively on graph with constraint on edges but the capacity constraint[36] work differently here. In traditional capacity constrained network (like the electricity network for which the constraint is the wire's size), constraints are fix and constant. In other words, the capacity between two nodes is constant and it is either full of partially used. In particular, the capacity does not depend on the actual quantity flowing through this edge. On the Lightning Network, this is not true. The capacity between two nodes changes dynamically depending on the transactions volume. At some point the capacity would have to be reset (either to the blockchain itself or with rebalancing), or the edge is essentially cut off. Taking those differences in consideration could yield interesting results on the network's structure like : "What conditions are needed for all channels to remain well balanced?", "Is there any possible attack scenario that would significantly harm the network liquidity?", or "Can such a model be used to analyze anything else than the Lightning Network?"

Another way to better approximate the equilibrium in such a payment network could be to use the Mean Field Game theory (MFG hereafter). Indeed, if we do not want to use complex network and graph models, we could extend the framework of this paper to a MFG-like setup, as introduced by Lasry and Lions (2007). MFG is basically an extension of the Nash-equilibrium concept with an infinite number of players. MFG is proven to be very handy to model dynamic situations, or otherwise stated it could allow us to study the network properties dynamically. It could be a way to model attack scenarii and other dynamic issues related to the Lightning Network.

At the micro level, a transaction is very different from a bank transaction for instance. There is no central authority so the nodes have to, instead, do a lot of work to properly use the network. First, the sender needs to work its way through the network. There are a number of optimal routing algorithms but they all fail at some point. Indeed, the sender will request the network status and then pick the optimal route but by the time the payment is actually sent, the network status (and especially the capacity on the optimal route) might have changed. If the selected route is not able to handle the payment, the transaction is rejected and the sender tries to find another route. True optimal routing is therefore unlikely to be available but some algorithms can do *best effort* routing. However, depending on how those

[36]We mean the capacity constraint from the point of view of end users, that is the amount funded in payment channels.

algorithms are implemented and the *best effort* route is chosen, there might be some ways for nodes to increase the probability of being on the *best effort* route without necessarily incurring the same cost as for the optimal route. Second, as explained in section 2, network nodes need to stay connected to prevent the payment channel counterparty from stealing the funds. In a payment channel, funds are secured to the extent that each node stays online to monitor the channel state[37]. The incentive for a node to cheat could have important consequences on the network formation.

Both previous issues could be externalized. Channel monitoring can be trustlessly outsourced to some specific nodes called *watchtowers*. Such nodes would be in charge of monitoring payment channels for other nodes, and if they detect a cheating attempt, they would provide the blockchain with the cheating proof and be compensated with a percentage of the recovered funds. Externalization of optimal route computation is done for another reason though. In the future it might be impractical for a regular node to compute the optimal route as it would have to download the full state of the network each time he wants to send a payment, of which the size can be quite high[38]. The state of the network could be downloaded on a regular basis by another node which will be paid by the sender to send back the optimal route. The incentives of each node and the optimal contract for both externalized services, channel monitoring and optimal route computation, remain largely unknown. Note that both services yield in somehow a loss in privacy for the original sender[39].

There are higher order protocols that allow to generalize the idea of the Lightning Network. The main idea of the Lightning Network is to lock some funds on the blockchain, tell all participants about this transaction, and then use those funds bilaterally without the need to tell anyone but the nodes on the payment route. The same idea can be taken further with *channel factories*. With *channel factories*, a group of nodes holds some amount on the blockchain that allows them to open and close payment channels among each other without telling anyone but the nodes in the same *channel factory*. Each of those payment channels can then be used as a regular payment channel on the Lightning Network. In principle, it is possible to push this idea further and create *factories of factory* and so on. Recall that those factories or even payment channel are simply a set of contracts between several agents. Those contracts and their implications for end users were never studied

[37]Recall that, a node can attempt to close unilaterally a payment channel with any previous history of transactions (one in which he has more money for instance). To prevent that the other node has to monitor those attempts. If a node detects a false closing he provides the most recent history of transactions to the blockchain and, if valid, the blockchain allocates the whole channel funding to the non-cheating node. This is meant as a punishment for the cheating node.

[38]As of early 2019, it is around 25Mb.

[39]*Watchtowers* would need to hold the current state of the payment channels, and nodes that compute optimal route need to know the recipient of the transaction.

from an economic and financial perspective.

Lastly, let's shed light on *atomic swaps*. So far, we presented the Lightning Network as a payment channel running on a single blockchain. However, several blockchains are currently running and it is possible to use the Lightning Network as a way to communicate value across blockchains. Note that this also works on the blockchains themselves, without the need of the Lightning Network. On the blockchain, *atomic swaps* can be used as follows. On blockchain A, user A sends some amount of A-coin to user C, who then sends some amount of B-coin on blockchain B to user B. This *atomic swap* is done trustlessly[40]. The same idea can be implemented on the Lightning Network so that users do not need to even know what type of coin someone else is accepting, the swap could be made automatically in the background. Therefore even with one payment network, there could be several options, in terms of possible coin, for a buyer to pay the seller. Whether those swap contracts have an impact on the structure of the Lightning Network is unknown.

7.2 Empirical

As explained in section 6, we should not expect any empirical results as long as the network is still mostly experimenting. However, note that most of the data is publicly available as anyone should be able to use the Lightning Network to route payments. Historical data might be hard to get but the current state of the network like the channels, their capacity, the fees, etc. can be downloaded by anyone.

In sub-section 5.3, we briefly discuss the relation between the equilibrium variables. Recall that in this model, there are unique equilibrium relationships between the interconnectedness level, the channel total capacity and the fee rate. Once the network would have grown, it would be interesting to see whether those relations hold in the data.

On the liquidity side, there are also many questions that need to be answered empirically. Does the network produce a stable capacity? Can nodes rebalance channels in an efficient way? Are there liquidity traps in which all funds get locked in payment channels such that no rebalancing can occur? Liquidity will be an important determinant of the usefulness of the Lightning Network as end users should be able to send and receive payments without caring much about the underlying details. If those issues turn out to be unsolvable or too complex to solve, the network is likely to collapse.

Lastly, we want to emphasize the link between the payment channel network and the underlying blockchain. Since the Lightning Network can be reduced to a set of contracts on the blockchain, the cost of underwriting and managing those

[40] It is called "atomic" because either both users receive their value or no one does. Otherwise stated, counterparty risk is fully eliminated.

contracts will be directly impacted by the transaction cost on the blockchain[41]. It will therefore be interesting to know if and how shocks on the blockchain parameters (transaction cost, security) propagate to the Lightning Network.

8 Conclusion

The Lightning Network is a communication protocol allowing the creation of payment network on top of a blockchain. The payment network is a set of bilateral contracts, called payment channels, registered on the blockchain. Once the blockchain usual transaction costs (monetary cost and delay) have been incurred to open a channel, value can move instantly at a fee supposed to be smaller than the blockchain fees. Some nodes in this network will have for sole purpose the routing of transaction across the network. This paper *i*) presents an overview of the current state of the Bitcoin Lightning Network, *ii*) builds a model for analyzing routing nodes' incentives and *iii*) derives some implications of the model regarding network characteristics.

There are currently 2,865 nodes and 36,982 payment channels in the network[42]. The total capacity of the network is 1,045.8298 BTC (around USD 6 millions) as early 2019. The network centrality, as measured by degree centrality and betweenness-centrality, has the shape of a power law. We show that the Lightning Network is already quite interconnected. We also estimate two dynamic models, Barabasi-Albert (2001) and Bianconi-Barabasi (2001), and both indicate that the network's dynamic structure is similar to the one of social networks. Essentially, this means that older node tends to have more connections which suggests that no one is really trying to profit from the routing activity so far. Note also that average routing fees are quite low (close to zero). Those results could have been expected as the network is too knew and there is still a lot of testing happening.

We propose a theoretical framework to analyze nodes' incentives and the resulting structure of the network. We feature a model in which several nodes are competing on price to route transactions, however, they must beforehand set costly channel capacity and interconnectedness level. Indeed, due to the network mechanics a node can only route a transaction if it has enough funds available in both the *in* and *out* payment channel. The marginal cost of routing a transaction is assumed to be zero. This creates a game similar to a Bertrand competition game with capacity constraints.

We show that there is a single equilibrium in which all nodes make no profit and the routing is strictly positive. If the routing is expected to be too low, nodes

[41] A *contract* is simply a particular kind of transaction.

[42] These figures do not take into account the nodes with only 1 or 2 open channels, as they are not very useful to route transactions.

prefer to not participate in the market as they cannot break even. This endogenous participation constraint provides a rationale for positive routing fees even with zero marginal cost. Otherwise stated, the capacity constraints prevent the equilibrium fee to reach marginal cost. Moreover, we show that, in equilibrium, the investment in both capacities (channel and network) are linked with a positive relationship. The more a node will be interconnected, the more it will have invested in its payment channels.

We show that heterogeneity in the nodes' opportunity cost of capital is enough to generate a wide variety of network topology and overall capacity. The only exogenous parameter is the opportunity cost of capital. We show that in equilibrium, there exists a mapping between the opportunity cost of capital and the optimal values for the control variables. In particular, we show that the bounds of the opportunity cost of capital will extend to the control variables including interconnectedness level and funding amount. For instance, it means that there is an upper-bound on the centrality of a node, which corresponds to the node with the lowest opportunity cost of capital. Another way to put it is that a completely centralized network is never optimal as long as the lowest opportunity cost is higher than zero.

The Lightning Network might very well become a widely used payment network. This paper is a first attempt to formalize decisions in such a network. Our results suggest that there are stable equilibrium outputs in which the network is liquid, well connected and with low fees (in the sense that no nodes make a profit). However, there are a number of issues that should not be overlooked, like dynamics properties, network formation, etc. We believe that this model can be used as a reference for any further extensions and improvements. The specifics of the Lightning Network (mainly capacity constraints that adjust dynamically) makes it also interesting from a pure mathematical point of view. The study of this particular network has the potential to significantly improve what we understand about networks in general.

The very nature of the Lightning Network and the blockchain itself could raise the question of the goal of this research. Indeed, those systems were built to be self regulated, i.e. regulated by the software itself. The blockchain builds consensus, therefore everything that is recorded on the blockchain is the ground truth, and the Lightning Network is essentially a set of contracts lying on the blockchain. The consequence is that any disagreement on the channel balances would ultimately be resolved by the blockchain and the rule written within it source code. No regulator is needed for the blockchain or the Lightning Network and everyone is free to use it however they want. Also, it can be very hard to change actual rules, because every user need to agree on protocol changes. It is important to understand that the blockchain and the Lightning Network cannot be regulated in a traditional way. Ultimately the developers are the ones who decide the rules and write them inside the source code, after what they cannot be changed[43]. What each government can do is to allow or forbid the use of those technologies. Therefore, the goal should be to understand the mechanics and see in what cases it produces a stable output that could eventually and safely be used by end users across the world.

[43]We refer here as changes in consensus protocol and communication standards. Small changes like the algorithm used to optimally route transactions is not part of consensus and, therefore, it can be updated easily.

References

Albert, Réka, and Albert-László Barabási, 2002, Statistical mechanics of complex networks, *Reviews of modern physics* 74, 47.

Biais, Bruno, Christophe Bisiere, Matthieu Bouvard, and Catherine Casamatta, 2019, The blockchain folk theorem, *The Review of Financial Studies* 32, 1662–1715.

Bianconi, Ginestra, and A-L Barabási, 2001, Competition and multiscaling in evolving networks, *EPL (Europhysics Letters)* 54, 436.

Davidson, Carl, and Raymond Deneckere, 1986, Long-run competition in capacity, short-run competition in price, and the cournot model, *The Rand Journal of Economics* pp. 404–415.

Easley, David, Maureen O'Hara, and Soumya Basu, 2019, From mining to markets: The evolution of bitcoin transaction fees, *Journal of Financial Economics*.

Freeman, Linton C, 1977, A set of measures of centrality based on betweenness, *Sociometry* pp. 35–41.

Freixas, Xavier, and Bruno Parigi, 1998, Contagion and efficiency in gross and net interbank payment systems, *Journal of Financial Intermediation* 7, 3–31.

Kahn, Charles M, and William Roberds, 1998, Payment system settlement and bank incentives, *The review of financial studies* 11, 845–870.

Kreps, David M, and Jose A Scheinkman, 1983, Quantity precommitment and bertrand competition yield cournot outcomes, *The Bell Journal of Economics* pp. 326–337.

Lasry, Jean-Michel, and Pierre-Louis Lions, 2007, Mean field games, *Japanese journal of mathematics* 2, 229–260.

Leitner, Yaron, 2005, Financial networks: Contagion, commitment, and private sector bailouts, *The Journal of Finance* 60, 2925–2953.

Moore, Cristopher, Gourab Ghoshal, and Mark EJ Newman, 2006, Exact solutions for models of evolving networks with addition and deletion of nodes, *Physical Review E* 74, 036121.

Moreno, Diego, and Luis Ubeda, 2006, Capacity precommitment and price competition yield the cournot outcome, *Games and Economic Behavior* 56, 323–332.

Nakamoto, Satoshi, et al., 2008, Bitcoin: A peer-to-peer electronic cash system, .

Pham, Thong, Paul Sheridan, and Hidetoshi Shimodaira, 2015, Pafit: A statistical method for measuring preferential attachment in temporal complex networks, *PloS one* 10, e0137796.

Poon, Joseph, and Thaddeus Dryja, 2016, The bitcoin lightning network: Scalable off-chain instant payments, .

Rochet, Jean-Charles, and Jean Tirole, 1996, Controlling risk in payment systems, *Journal of Money, Credit and Banking* 28, 832–862.

Rohrer, Elias, Julian Malliaris, and Florian Tschorsch, 2019, Discharged payment channels: Quantifying the lightning network's resilience to topology-based attacks, *arXiv preprint arXiv:1904.10253*.

Seres, István András, László Gulyás, Dániel A. Nagy, and Péter Burcsi, 2019, Topological analysis of bitcoin's lightning network, *CoRR* abs/1901.04972.

9 Appendix

9.1 Proofs of Lemmas

Proof of Lemma 1

The proof of this lemma is provided within the proof of proposition 1.

Proof of Lemma 2

The proof of this lemma is provided within the proof of proposition 1.

Proof of Lemma 3

In this subgame, nodes must find the routing fee that maximizes their profit, R^*, taking the channel and network constraints, (x, λ), as given. With the assumption we made, nodes get to routes payment only if the routing fee they set is lower or equal to all other nodes' routing fee. Otherwise stated, nodes are competing on price to attract volume, and a price too high would yield zero volume as end-users would select nodes with lower price.

Let's define by $\hat{R}(x, \lambda)$ the routing fee that produces a zero profit given (x, λ). Conditioned on attracting volume, if $R > \hat{R}$, the node will make a positive profit and a loss otherwise.

$$\hat{R}(x, \lambda) = \frac{x^2 + 2\lambda}{2\left(1 - e^{-x}\right)\left(1 - e^{-\lambda}\right)} \tag{9}$$

Let's consider node i, and check when he benefits from deviating in the candidate equilibrium. Assume node i expects all other nodes to play $R_{-i} > \hat{R}$. Then node i can increase its profit by attracting all the volume and playing R^* such that $\hat{R} < R^* < R_{-i}$. The profit would still be positive and the node gets to route more volume. The same mechanics can be applied to any node to show that the routing fee will be driven toward $R^* = \hat{R}$.

If node i expects all nodes to play $R_{-i} = \hat{R}$, he does not benefit from increasing or decreasing the fee. If he increases it, he would get no volume, and if he decreases it, he would take all the volume but at a loss.

If node i expects all nodes to play $R_{-i} < \hat{R}$, he cannot benefit from playing something other than $R^* = R_{-i}$. If it plays higher it won't attract any volume, and if it plays less, it would increase the loss. Therefore all prices $R^* \in [0; \hat{R}[$ are also Nash equilibrium of the subgame.

Q.E.D

Proof of Lemma 4

The proof of this lemma is provided within the proof of proposition 3.

Proof of Lemma 5

In this subgame, nodes must find the routing fee that maximizes their profit, R^*, taking the channel and network constraints, $(\{x_i\}_i; \{\lambda_i\}_i)$, as given.

The argument is exactly the same as for lemma 3, except that the mechanisms in lemma 3 applies for any interconnectedness level, λ_i, because of the assumption we made regarding the demand (infinite with any interconnectedness level).

Define $\hat{R}(x, \lambda)$ by

$$\hat{R}_i(x, \lambda) = \frac{\gamma_i x^2 + 2\lambda}{2\left(1 - e^{-x}\right)\left(1 - e^{-\lambda}\right)}$$

We can show that if $R_i^* \geq \hat{R}_i(x_i, \lambda_i)$, there will be an arbitrage opportunity. However, once again, for each value of γ_i, any price between 0 and $\hat{R}(x_i, \lambda_i)$ will be a Nash equilibrium, in which nodes make a loss.

Q.E.D

9.2 Proofs of Propositions

Proof of Proposition 1

Recall that we are in the one-shot version of the model, where R is taken as exogenous. The program the node needs to solve is the following.

$$(x^*, \lambda^*) = \underset{x \geq 0, \lambda \geq 0}{\arg\max}\ \Pi(x, \lambda).$$

with

$$\Pi(x, \lambda) = \left(1 - e^{-\lambda}\right)\left(1 - e^{-x}\right) R - \frac{x^2}{2} - \lambda$$

The first order condition gives

$$\begin{cases} \frac{\partial}{\partial x}\Pi(x, \lambda) & = & e^{-x}(1 - e^{-\lambda})R - x & = & 0 \\ \frac{\partial}{\partial \lambda}\Pi(x, \lambda) & = & e^{-\lambda}(1 - e^{-x})R - 1 & = & 0 \end{cases}$$

Combining the 2 equations and rearranging the terms yields the following FOC.

$$\lambda^* = ln\left(1 + x^*(e^{x^*} - 1)\right) \tag{10}$$

We can now plug that relationship into the first derivative of the objective function with respect to λ.

$$\frac{1 + x^*(e^{-x^*} - 1)}{1 - e^{x^*}} = R \tag{11}$$

There is no closed form solution for this equation. However, we can numerically demonstrate some important properties. Let's define $g(x)$ by

$$g(x) := \frac{1 + x(e^{-x} - 1)}{1 - e^{x}}.$$

Figure 11 presents the graph of the function g. Given the definition of the function g, x^* is such that $g(x^*) = R$. The function g admits a minimum, denoted by $\overline{R} > 0$.

$$\overline{R} = \min_{x>0} g(x)$$

This tells us that for $R < \overline{R}$ there are no value of x that can satisfy equation 11. For values of R such that $R > \overline{R}$, there are two solutions to the first order conditions, x_1 and x_2 with $x_1 < x_2$. However it can be shown that x_1 is actually a saddle point. The second order condition tells us that $x^* = x_2$, when $R > \overline{R}$, is a maximum of the function $\Pi(x, \lambda)$ on the domain $x \geq 0$ and $\lambda \geq 0$, with λ^* given by (10).

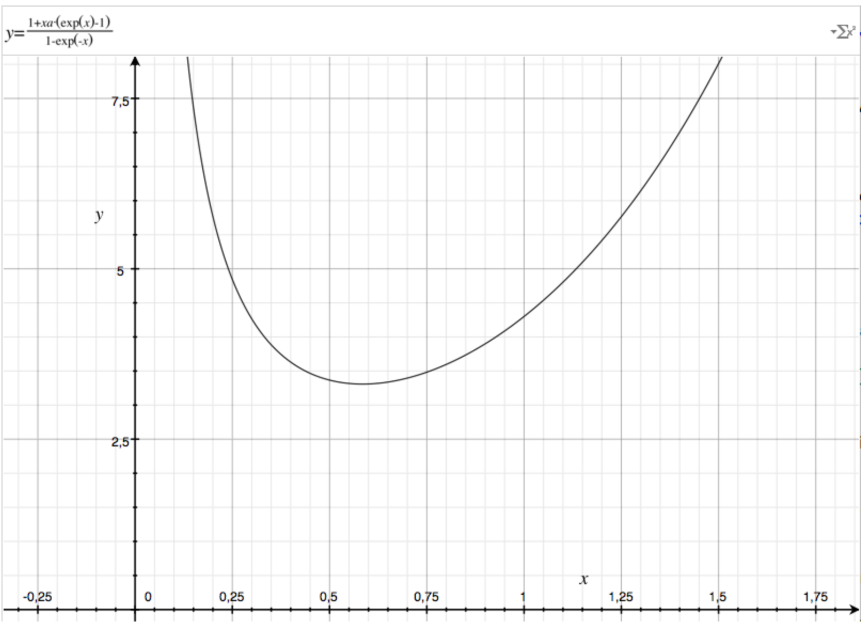

Figure 11: Graph of the function $g(x)$

The full solution is therefore characterised by the following.

$$x^* = \begin{cases} 0 & \text{when } R < \overline{R} \\ x_2 & \text{when } R \geq \overline{R} \end{cases}$$

The function f defined in the proposition is the inverse of the g function for values of x greater than the $\arg\min$ of $\Pi(x,\lambda)$.

Q.E.D

Proof of Proposition 2

We start by focusing our attention on equilibrium such that $\forall i \neq j; x_i^* = x_j^*$. The argument bellow could be used to rule out those cases. So keep the notation (x,λ) for the subgames. Equilibria of the full game are defined as (x^*,λ^*,R^*).

First, let's consider the following points

$$\forall R; (x,\lambda,R) = (0,0,R).$$

For the same reason as in proposition 1, those points cannot be equilibrium because the demand is completely unsatisfied and any node could jump into the network with any arbitrary $(x,\lambda) \in \mathbb{R}_+^* \times \mathbb{R}_+^*$, charge a corresponding fee and make a positive profit.

With proposition 1 we know that x^* and λ^* are linked in equilibrium with the following relation.

$$\lambda^* = \lambda(x^*) = ln\left(1 + x\left(e^x - 1\right)\right)$$

therefore we can restrict our notation to

$$(x^*,\lambda^*,R^*) = (x^*,\lambda(x^*),R^*) = (x^*,R^*).$$

Let's now define, $\overline{x}$, a critical point for our study as follows.

$$\overline{x} := \left\{ x \in \mathbb{R}_+^* \quad ; \quad \frac{1 + x\left(e^x - 1\right)}{1 - e^{-x}} = \frac{x^2 + 2\lambda(x)}{2\left(1 - e^{-x}\right)\left(1 - e^{-\lambda(x)}\right)} \right\}$$

It is possible to show that $\overline{x}$ has a single possible value. We cannot prove this analytically but, first we can rewrite the equation defining $\overline{x}$ as $h(x) = 0$.

$$h(x) = x^2 + 2ln\left(1 + x\left(e^x - 1\right)\right) - 2x\left(e^x - 1\right) = 0$$

Then, when we plot the graph of $h(x)$ in figure 12, we can see that it has a single solution. The point $x = 0$ is also a solution of this equation but we can safely discard it for the reason used to prove that $(0,0,R)$ is not an equilibrium. Note that the model is up to some proportional constant, but changing the scale of variables should not change the curvature of the function and therefore the existence of the solution should extend to any different scale.

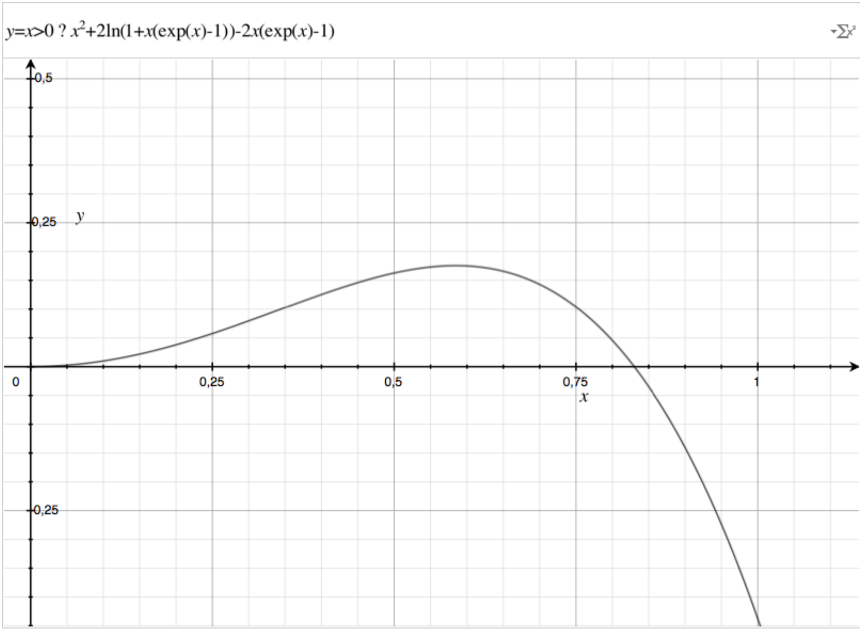

Figure 12: Graph of the function $h(x)$

Note that $\overline{x}$ and the corresponding fee $\overline{R}(\overline{x}, \lambda(\overline{x}))$, as defined by (9), define the point for which the profit is maximized but equal to zero. At this point the node cannot increase its profit by playing around with capacity constraints (x, λ). Also the fee is such that the node makes zero profit. Decreasing this fee would increase the loss, and increasing the fee would not attract any volume, so resulting in a loss as well. Now consider possible values of x compared to $\overline{x}$.

Case 1 : When $x > \overline{x}$. By lemma 3 we know that either $R = \hat{R}$ as defined by (9), or $R \in [0; \hat{R}[$. Recall that $\hat{R}$ yields zero profit. Values such that $R \neq \hat{R}$ cannot be equilibrium because all nodes would make a loss and therefore prefer a subgame such as $(x, \lambda, R) = (0, 0, R)$ which, as previously seen, cannot be an equilibrium. If $R = \hat{R}$, it cannot be considered an equilibrium either because node i could play an x' such that

$$\overline{x} < x' < x$$

which would increase its profit. Indeed, at $x > \overline{x}$ it is profitable for the nodes to play a smaller value of x, it will make the profit increase to its maximum as the node approaches $x = \overline{x}$.

Case 2 : When $x < \overline{x}$. For the same reason, the points $(x, \lambda(x), 0)$ cannot be considered equilibrium. If $R = \hat{R}$, it cannot be considered an equilibrium as well because node i could play an x' such that

$$x < x' < \overline{x}$$

which would increase its profit. Indeed, at $x < \overline{x}$ it is profitable for the nodes to play a larger value of x, it will make the profit increases to its maximum as the node approaches $x = \overline{x}$.

Case 3 : When $x = \overline{x}$. For the same reason as above, $R < \hat{R}$ cannot be an equilibrium. When $R = \hat{R}$, no node can benefit from playing something else. If they move x, keeping the same R, they would start to make a loss, as $\overline{x}$ is defined to be the objective function maximum, holding R constant. And moving R would either increase their loss or make them unable to attract any volume. Therefore this point is an equilibrium.

Only the last case is an equilibrium, so without loss of generalities we can say that the only equilibrium is the following

$$(x^*, \lambda^*, R^*) = \left(\overline{x}, \lambda(\overline{x}), \hat{R}(\overline{x}, \lambda(\overline{x}))\right).$$

Q.E.D

Proof of Proposition 3

This proof is very similar to the proof of proposition 1. We don't detail everything, only the main difference compared to the homogeneous agents' case.

First of all, the first order conditions give

$$\begin{cases} \frac{\partial}{\partial x}\Pi(x,\lambda) & = & e^{-x}(1-e^{-\lambda})R - \gamma x & = & 0 \\ \frac{\partial}{\partial \lambda}\Pi(x,\lambda) & = & e^{-\lambda}(1-e^{-x})R - 1 & = & 0 \end{cases}$$

and it can be re written as

$$\lambda^* = ln\left(1 + \gamma x^*(e^{x^*} - 1)\right).$$

We can now plug that relationship into the first derivative of the objective function with respect to λ.

$$\frac{1 + \gamma x^*(e^{-x^*} - 1)}{1 - e^{x^*}} = R \tag{12}$$

Again, there is no closed form solution for this equation. However, we can numerically demonstrate some important properties. Let's define $g(x)$ by

$$g(x) := \frac{1 + \gamma x(e^{-x} - 1)}{1 - e^{x}}.$$

We can graphically see that the minimum of this function for positive values of x, increases with γ. Given the definition of the function g, x^* is such that $g(x^*) = R$. The function g admits a minimum, denoted by $\overline{R} > 0$.

$$\overline{R} = \min_{x>0} g(x)$$

Relation (6) and (7) also come from graphical interpretation of the function $g(x)$.

Q.E.D

Proof of Proposition 4

The proof is very similar to the proof of proposition 2. The equilibrium mechanisms are exactly the same as for the homogeneous case except everything happens within each cohort that shares the same opportunity cost of capital, γ. Note that this is fine even if the opportunity cost is continuous because we assumed that a continuum of potential nodes can jump into the network with any γ if they see an arbitrage opportunity. This will drive the price further down until the no profit condition is satisfied, for each γ.

The rest of the proof is exactly the same as in proposition 2.

Q.E.D

Proof of Proposition 5

It can be shown that all the equilibrium applications are monotonic and continuous. Also, as long as $\gamma \in]0; \gamma_{max}]$ (and $\gamma_{max} < +\infty$), the equilibrium applications, $x(\gamma)$ and $\lambda(\gamma)$, will converge to finite values.

Therefore, if the support of the γ-distribution is finite, the equilibrium value of (x, λ) will also be finite, bounded by the γ-distribution's bound.

Q.E.D