LUISS

Department of Business and Management

Chair of Blockchain & Cryptocurrencies

An Experimental Study of the Bitcoin Lightning Network Properties

Prof. Bernaschi Massimo

SUPERVISOR

Dr Giammusso Sara & Dr Galano Giuseppe

CO-SUPERVISORS

Giulia Iadisernia – 240001

CANDIDATE

Academic Year 2021/2022

Acknowledgements

I want to thank Prof. Bernaschi, my thesis supervisor, for kindly supporting me in the most important phase of my academic career.

A heartfelt thanks to Dr Giammusso Sara and Dr Galano Giuseppe, experts from the Applied Research Team (ART) in the IT directorate of the Bank of Italy and my thesis co-supervisors, for their guidance and tips that have strongly contributed to the realization of my thesis.

To mom and dad, for their moral and economic support and their teachings, without which today I would not be who I am. Thank you.

Table of Contents

CHAPTER 3

THE LIGHTNING NETWORK PERFORMANCE ANALYSIS

List of Tables

List of Figures

Introduction

The hype surrounding blockchain and cryptocurrencies has led to the development of many potentially life-changing new technologies. The rise of Bitcoin has made a huge impact on the digital world and revolutionized the concepts of transparency, anonymity, integrity and disintermediation. Despite the innovative viewpoint, Bitcoin has still many working points linked to the lack of scalability and slow transaction speed. Bitcoin believers developed and implemented some solutions to solve these problems. One of the solutions is the Lightning Network (LN), a payment layer-two protocol built on top of Bitcoin to improve its scalability. The LN works by transferring most of Bitcoin's transactions off-chain, exploiting payment channels and deferring broadcasting the transaction until channel closure. It sounds promising.

This paper aims at studying the properties of the Lightning Network and its evolution over the years. We want to verify whether the LN can resolve the previously stated issues, as Joseph Poon and Thaddeus Dryja affirmed in the White Paper.

The first chapter illustrates the main concerns of Proof-Of-Work, especially regarding the environment, the waste of energy and the scalability issues. Then, it focuses on some of the technical solutions such as alternative consensus mechanisms (POS), network partitioning (sharding) and second layer protocols.

The second chapter introduces the Lightning Network and briefly explains how participants forward transactions through direct payment channels and routing.

The third chapter describes a step-by-step practical application of the subject. We import the dataset, build the graphs and use several tools and programming languages such as R and Python to perform a few relevant analyses and extract some meaningful statistics.

Lastly, we interpret the results and conclude by answering the following questions: Is the current state of the Lightning Network adequate for supporting everyday life transactions? Specifically, if the entire world were to use the Lightning Network right now, could the LN process this many transactions successfully?

CHAPTER 1

Bitcoin's Scalability Problem and Some Technical Solutions

1.1 Proof-Of-Work Issues

The Bitcoin blockchain relies on Proof-Of-Work (POW) as the consensus mechanism. It consists in assigning miners a very computationally intensive puzzle, which costs participants high sums of money in terms of energy and computational resources. POW succeeds in addressing both the Byzantine General's problem, i.e. the problem of reaching the consensus in a decentralized network with potentially malicious participants and the double-spending problem, i.e. when a single unit of currency is spent simultaneously twice or more.

Although highly effective in assuring the integrity of transactions, POW has a significant impact on the environment. Indeed, annual electricity consumption for bitcoin production has been estimated to be equivalent to 32.56 tera-watts per hour (TWh), more than the aggregate consumption of Ireland or Denmark[1]. Moreover, as seen in Fig.2, one Bitcoin transaction consumes far more energy than 10,000 VISA transactions. However, it appears that producing one Apple iPad is even more energy-consuming compared to one Bitcoin transaction. Personally speaking, I agree with Stephen Diehl, who said in one of his most pessimistic articles about blockchain, from an environmental point of view, "mining […] is an irresponsible waste of energy in a world facing a dire climate crisis […]"[2]. Many studies suggest moving mining capacities to countries such as Germany or Denmark, which allow for the application of new and sustainable solutions to mitigate and potentially eliminate the environmental damage involved in cryptocurrency mining. The consumption of electricity is not the only waste associated with POW. The mining process generates also electronic waste

[1] Sergio Luis Náñez Alonso et al. (2021) "Cryptocurrency Mining from an Economic and Environmental Perspective. Analysis of the Most and Least Sustainable Countries", Catholic University of Avila
[2] Stephen Diehl (2021) "Web3 is Bullshit"

as the computers employed in mining factories tend to become obsolete after roughly 1.5 years.

Figure 1: Bitcoin's yearly energy consumption compared to countries' yearly energy consumption. Source: University of Cambridge Bitcoin Electricity Consumption Comparisons

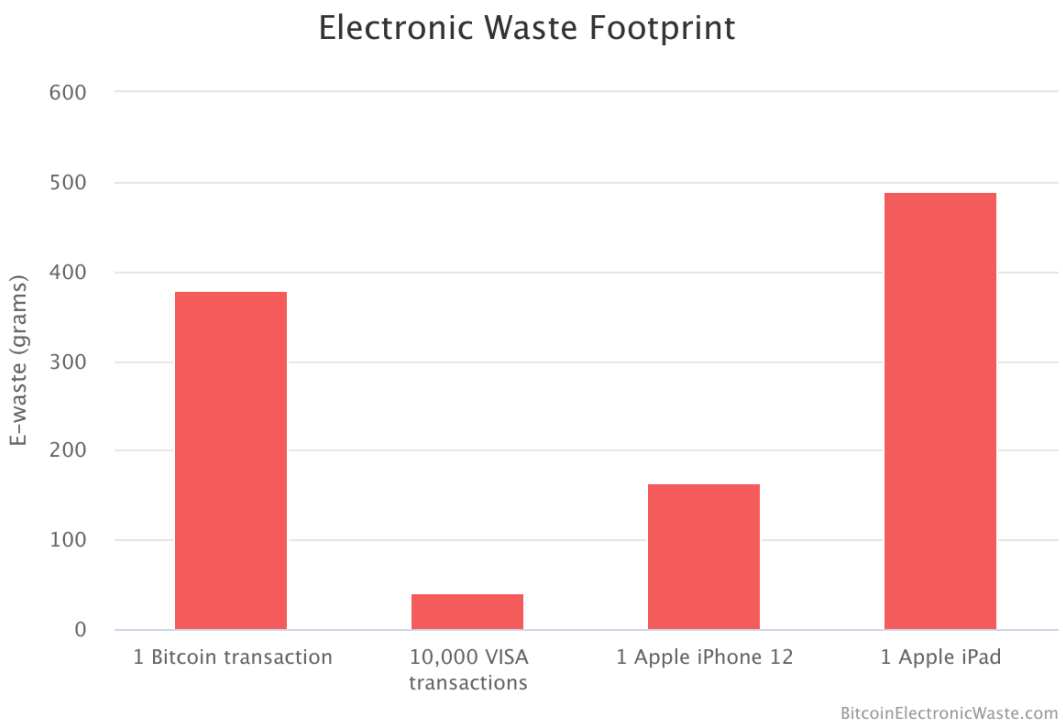

Figure 2: Electronic Waste of 1 Bitcoin transaction versus 10,000 VISA transactions. Source: BitcoinElectronicWaste.com

Another major issue of POW is its inability to scale. The Bitcoin blockchain can validate 3-7 transactions per second compared to Visa's peak rate of 47,000 per second[3]. Moreover, miners tend to prioritize transactions with higher fees, making it hard to sustain payments of small amounts. Thus, cryptocurrencies must find a solution for processing a higher number of transactions. Some[4] proposed increasing the size of the blocks. Shifting to bigger blocks would increase the number of transactions per time unit, hence, better scalability. However, larger blocks would favour mining pools leading to an increase in the centralization and, in turn, a decrease in the security. The latter is an example of what is known as Buterin's trilemma, which states that you cannot achieve scalability, decentralization and security at the same time without sacrificing one of those three aspects.

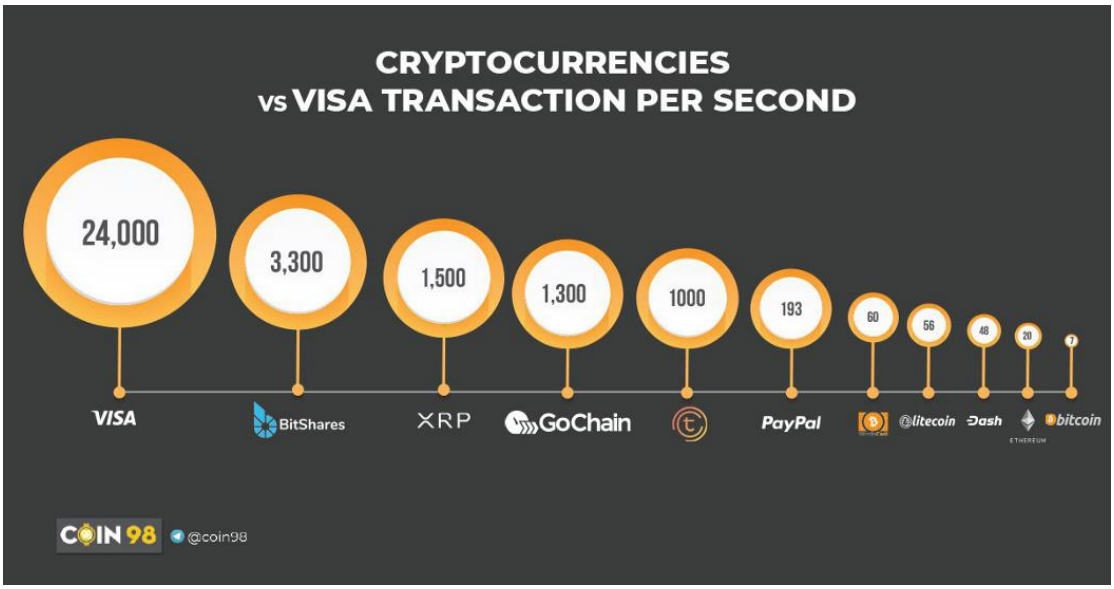

Figure 3: Amounts of transactions per second (VISA versus cryptocurrencies). Source: Coin98 Analytics, Medium (2018) "Compare the transaction speed of visa with that's cryptocurrency"

1.2 Technical Solutions to Increase Bitcoin's Scalability

1.2.1 Proof-Of-Stake

[3] Generally, Visa processes roughly 1700 transactions per second.
[4] Bitcoin Cash (fork of Bitcoin) increased the block size to 8 MB instead of 1 MB.

Many solutions have been proposed to address Bitcoin's scalability problem. One of them involves changing the consensus mechanism. *Proof-Of-Stake (POS)* is a well-known alternative for POW. For instance, Ethereum is already transitioning to POS to increase its scalability. POS, unlike POW, involves only a subset of nodes that decide to put at stake a certain amount of coins (generally above some threshold). The higher the invested sum, the higher the probability of being selected as a validator. Forgers[5] do not need to solve a puzzle or computationally-intensive problem. Instead, they need to freeze their coins into a specific wallet, meaning that they cannot be used for other purposes. This clause discourages any attempt to cheat. Indeed, no user will act maliciously if it risks losing all of its coins. Therefore, those who stake have an interest in keeping the network secure. By opting for POS in opposition to POW, we could potentially reduce the amount of waste associated with mining and increase the scalability. However, POS might lead to centralization if a not large enough subset of participants are chosen to validate transactions.

1.2.2 Sharding

Another possible solution for Bitcoin's scalability problem is resorting to *sharding*. Sharding is a form of horizontal database partitioning, like when a customer database is divided into geographical locations. The blockchain network is split into multiple smaller networks that can contemporarily process different transactions, shifting from a sequential execution model to a parallel execution model. For instance, if there are 100 partitions, the network can process 100 transactions simultaneously instead of one. Sharding allows to increase the scalability, efficiency and availability of the network. By storing the data into different shards, there is no need for the nodes (computers) who want to participate in validating transactions to store the full copy of the ledger. Therefore, "by storing the data across different computers, the computational burden on each can be reduced"[6].

Sharding requires inter-shard communication to avoid every shard acting as a separate blockchain network. Secondly, it is easier for malicious users to take over a single shard, as it requires fewer resources to succeed in the 51% attack. Ethereum

[5] Validators are not called miners but forgers.
[6] https://www.sofi.com/learn/content/what-is-sharding/

proposed to use a random sampling of notary nodes to address the security issues. By doing so, nodes are randomly assigned to different shards to validate blocks, decreasing the probability of a security attack.

1.2.3 Layer-Two Protocols

"We distinguish between four different kinds of layers within a blockchain system: the hardware, layer-zero, layer-one and layer-two"[7]. *Layer-zero* is a peer-to-peer network layer; *Layer-one* is the blockchain layer which hosts an append-only, timestamped ledger. The Latter guarantees consensus among the participants as well as the overall security of the network. *Layer-two protocols*, also known as *off-chain protocols,* are built on top of the existing blockchain layer from which they inherit two essential properties: the integrity of transactions and eventual synchronicity with an upper time-bound.

There are different layer-two protocols. One of the most known is the Lightning Network, a level-two payment protocol that aims at improving Bitcoin's scalability and transaction speed by processing instant transactions. A deeper explanation will be provided in the next chapter regarding, what it is, and how it works. The liquid network is another type of layer-two protocol. It has three main goals: promote faster transactions, improve the confidentiality of bitcoin transactions and allow for the trading of large amounts of bitcoins as well as other tokens such as stablecoins. Firstly, the Liquid Network[8] is optimised for processing medium to large-sized transactions, unlike the Lightning Network. Secondly, transactions need to be confirmed by producing blocks. Therefore, transactions are slower than in the Lightning Network but roughly ten times faster than the average Bitcoin transactions (one block every minute instead of ten minutes). Thirdly, the Liquid Network advocates privacy, meaning that the amount of the funds and the type of the transferred asset are not publicly revealed. Only the sender and the receiver are aware of this information.

[7] Lewis Gudgeon et al. (2020), "SoK: Layer-Two Blockchain Protocols"
[8] Liquid Federation (2022) "Six Differences Between Liquid and Lightning", The Liquid Blog.

CHAPTER 2

The Lightning Network

2.1 Introduction to the Lightning Network

One of the most common off-chain protocols is the *payment-channel network (PCN),* a network of bi-directional, weighted edges representing peer-to-peer payment channels. The two parties decide to defer broadcasting the transaction to the blockchain since they may decide to update the balance at a future date. If both parties in the channel agree on the current ledger state, the old transaction is invalidated in favour of the new one, and the channel balance is updated.

The Lightning Network (LN) is the primary PCN on top of Bitcoin. The LN is defined as "[…] a decentralized system for instant, high-volume micropayments that remove the risk of delegating custody of funds to trusted third parties"[9]. The LN aims at solving Bitcoin's scalability problem by shifting the majority of the payments off-chain. In this way, it allows minimizing the workload suffered by the blockchain and speed up the transactions, especially the ones of smaller value. Lightning transactions do not require any block confirmation like Bitcoin and the Liquid Network do. Instead, as long as the participants have a steady Internet connection, the network could potentially sustain thousands of instant payments.

There is also an anti-fraud mechanism which has been put in place to discourage any fund stealing attempts. If one party decides to cheat, the transaction is propagated to the blockchain where the dispute is resolved in the usual manner. The cheater is intercepted and all of the funds are transferred to the other party.

[9] Joseph Poon et al. (2016) "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments"

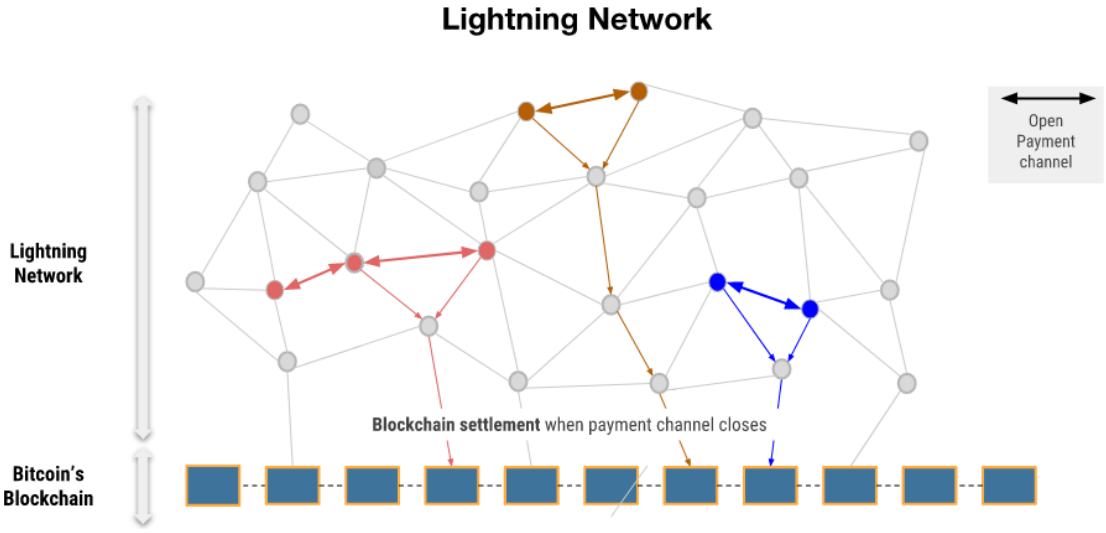

Figure 4: A visual representation of the Lightning Network. Source: TheLuWizz (2021) "How does Bitcoin get scalable with the Lightning Network?", Medium

2.2 Payment Channels

2.2.1 Definition of Payment Channel

Let's take a step back and focus on payment channels i.e., bi-directional payment connections between two nodes. A payment channel has three stages in its lifetime: opening, operating and closing. When two parties open a channel, they issue the so-called *funding transaction* which is registered in the blockchain ledger. The funds are stored in a *multi-signature address* which is accessible to both nodes in the channel. The sum of the individual balances of the parties amounts to the *capacity* of the channel, a number which stays fixed across the whole duration of the channel. Now, the two parties can directly send each other payments, also known *as commitment transactions*, as they both agree to update the state of the balance. The final balance is broadcasted to the blockchain at the closing stage of the channel.

2.2.2 Opening a Channel

Opening a lightning channel requires a *funding transaction*. This transaction is registered on the blockchain and marks the beginning of the financial relationship between two nodes. To create a channel there is the need for the *public keys* of both

participants, the *capacity* of the channel (which stays fixed throughout the life-time of the channel), a *multi-signature wallet* where the respective funds are kept and, finally, the signature of the counterparties, both the one who promoted the opening of the channel and deposited the funds, and the other. This address can be created either as a single-payer channel, or payments can be sent both directions.

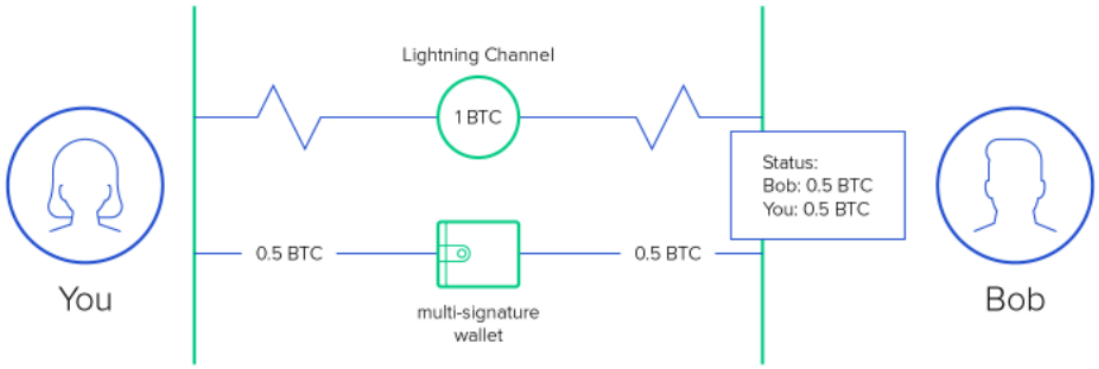

Figure 5: Creating a channel. Source: Amin Shah Gilani (2021) "Scale with Speed: The Bitcoin Lightning Network Explained", Toptal

2.2.3 Closing a channel

There are three ways of closing a channel: collaboratively, unilaterally, and breach remedy. Two are good, and one is bad.

- *Collaboratively*. Both parties agree on closing the lightning channel. There is no need for a time-lock, and the funds are almost instantly ready to be spent. Unarguably, it's the best way of closing a channel as there are no controversies and no delay.
- *Unilaterally*. One of the parties decides to close the channel without receiving the other party's approval. Unlike the previous scenario, this type of closure necessitates a time-lock in case one of the parties decides to cheat (see the point below). If both parties agree on the final state of the balance, the time-lock expires, and the funds are now free to use. This way of closing the channel is acceptable. However, the parties could have avoided the delay if they both agreed to close the channel.
- *Breach Remedy*. As previously said, there is a need for a time-lock. One of the parties may attempt to cheat by unilaterally closing the channel at a previous

state of the balance to gain more funds at the other party's back. The time-lock allows for the honest counterparty to call for Breach Remedy. The latter attributes all the capacity of the channel to the aggrieved party, while the malicious node loses all of its funds.

So, what happens if the honest party is off-line during the time-lock? Does he/she lose the possibility of using the Breach Remedy transaction? The short answer is no. When a node goes off-line, it can delegate a *watchtower* node that "acts on behalf of the users to secure their funds. […] Users can still verify the correct behaviour of watching services and punish them in the case of non-compliance"[10].

2.3 Types of Transactions in the Lightning Network

2.3.1 Hash-Time-Locked Contracts in the Lightning Network

A Hash-Time-Locked Contract (HTLC) is a smart contract that locks coins until a "secret passcode" is revealed. To initiate the payment, the receiver must generate a big random number R, a "secret passcode", and sends its hash H(R) to the sender. The sender can know the passcode only if the receiver reveals it. Calculating the preimage from the hash is (nearly) impossible, and there is no way of guessing it. HTLCs are often used to prevent any stealing of funds when carrying out a transaction that involves multiple intermediaries. The sender and the intermediaries must create a series of HTLCs using the same hash value, H(R). A chain of payments along the channels is created based on this hash value. The payments must be forwarded within a certain period of time t (hence the time factor in HTLCs), otherwise, inactive parties could leave the transaction forever pending, preventing the other parties from receiving their funds. This system allows the sender to detect the problematic actors and choose a different path to route his/her payment. Once the payment reaches its destination, the receiver reveals the preimage of H(R), and each node on the path redeems, in reverse order, the expected funds. Payments can only be delivered atomically, meaning they either succeed or fail entirely.

[10] Lewis Gudgeon et al. (2020) "SoK: Layer-Two Blockchain Protocols", pp 6-7

2.3.2 Single-Hop Transactions

Single-hop payments consist of transactions which are established between two parties that are directly connected through a payment channel. Let's make a practical example. You want to send 8000 satoshis to Bob. If both nodes agree, the individual balances are updated, and the old transaction is invalidated. You issue a new commitment transaction which requires the signature (approval) of both parties.

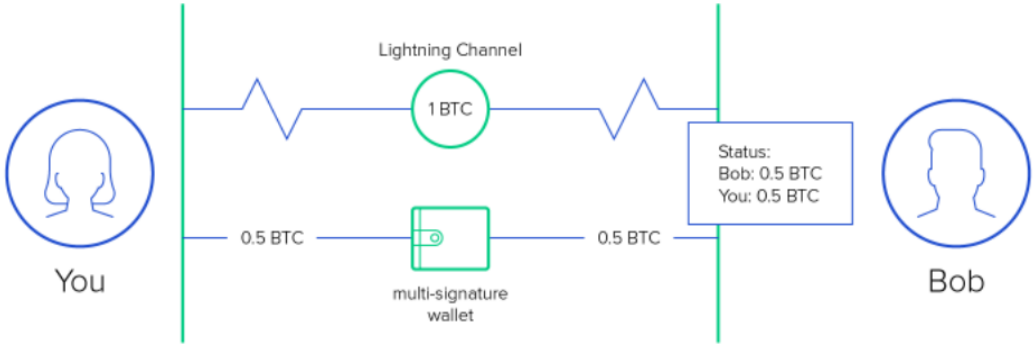

Figure 6: A commitment transaction part 1. Source: Amin Shah Gilani (2021) "Scale with Speed: The Bitcoin Lightning Network Explained", Toptal

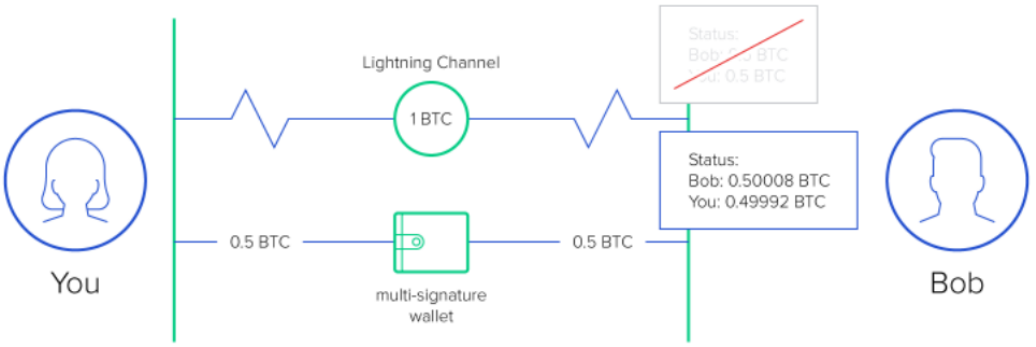

Figure 7: A commitment transaction part 2. Source: Amin Shah Gilani (2021) "Scale with Speed: The Bitcoin Lightning Network Explained", Toptal

2.3.3 Multi-Hop Transactions

Although single-hop transactions may be quick and efficient, opening a payment channel for every party with whom we exchange funds may not be wise. "Maintaining a payment channel has an opportunity cost since users must lock up their

funds while the channel is open, and funds are not redeemable until the channel is closed"[11]. The LN allows parties to exchange funds indirectly. Therefore, nodes that do not share a channel but share a common path can send and/or receive payments through a series of intermediary nodes in exchange for very small fees.

The LN relies on a *gossip protocol* to spread information about nodes and channels across all participants. This mechanism turns out to be especially important when it comes to routing the payment, as it allows the sender to have a clear view of the network and its feasible paths to reach the receiver node. The most important gossip messages are the following.

— *Node announcement message*: it broadcasts node information like the Ip address, alias and timestamp.
— *Channel announcement/update message*: it informs the other nodes when a new channel is created and propagates channel parameters across the network.

Routing is needed every time a payer, and a payee are not directly connected. There are incentives for intermediary nodes for forwarding transactions i.e., they take a small fee.

The total transaction fee is equal to: *TransactionFee = baseFee + feeRate * TransactionAmount.*

- The *baseFee* is the fixed fee charged each time a payment is forwarded.
- *FeeRate * TransactionAmount* is the additional fee an intermediary node earns as a percentage (*FeeRate*) of the transaction it routs (*TransactionAmount*). "[…] The base fee and fee rate are set by individual users, thus forming a fee market for payment routing"[12].

Payments can only be delivered atomically, meaning they either succeed or fail. In the LN, Transactions are based on *Hash Time-Locked Contracts (HTLCs)*, a technique that allows payments to be securely routed across multiple payment channels. Let's make a practical example[13]. As seen in Fig. 8, Alice wants to send Eric 1 BTC.

[11] Ferenc Beres et al. (2020) "A Crypto-economic Traffic Analysis of Bitcoin's Lightning Network", Institute for Computer Science and Control (SZTAKI)
[12] Ferenc Béres et. al (2020), A Cryptoeconomic Traffic Analysis of Bitcoin's Lightning Network, Institute for Computer Science and Control (SZTAKI)
[13] Nikolaos Papadis (2020) "Blockchain-based Payment Channel Networks: Challenges and Recent Advances", IEEE Access

Alice could create a direct payment channel with Eric but instead opts for a multi-hop path with three intermediaries, Bob, Carol and Diana. First, Alice asks Eric to think of a secret R and send its hash H(R) to her. Alice uses this hash to create an HTLC with the amount being 1.003 BTC, 1 BTC for Eric and 0.001 BTC paid as fees to each intermediary. Bob takes his portion of the fees and creates an HTLC using the same hash as Alice, locking in 1.002 BTC. Carol and Diana proceed to take their portion of the fees and forward the payment until 1 BTC reaches its destination, Eric.

It's important to point out that the balances are still unchanged at this stage. The changes will only be applied if the payment completes successfully. For this to happen, Eric needs to share the secret number R with everyone else in reverse order of the path. By sharing R with Diana, Eric can unlock his coins. Diana then shares R with Carol and gets her 1.001 BTC back. Carol shares R with Bob, and she gets her 1.002 BTC back. Finally, Bob shares R with Alice, and he gets his 1.003 BTC back. These amounts are now reflected in an update in all the channel balances.

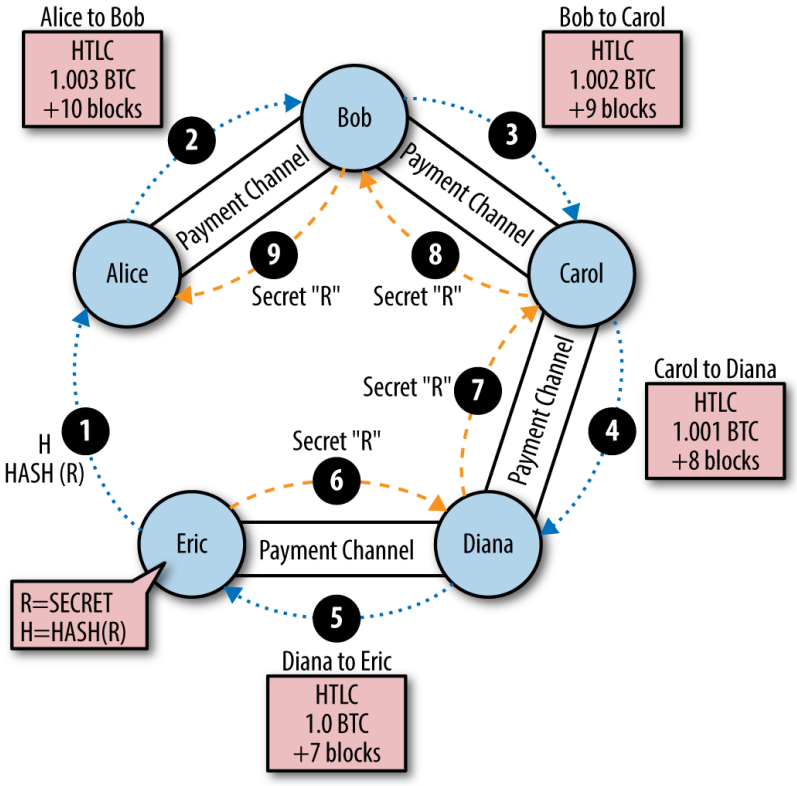

Figure 8: Routing payments. Source: Magomed Aliev (2018) "Lightning network in depth, part 2: HTLC and payment routing", Medium.

2.4 The Drawbacks

In the previous paragraph, we suppose that the transaction succeeds. However, it should not be taken for granted. One of the main drawbacks of the Lightning Network is the lack of *payment reliability*. The latter means that "[…] a user with a sufficient balance can quickly send payments with a high probability"[14]. If the intermediaries have an individual balance smaller than the payment threshold, they cannot successfully forward the payment, and the transaction is aborted. The sender may have to search for an alternative route until he/she finds one that finally succeeds. Only channel capacities are publicly revealed, while node balances are kept secret for privacy reasons, and since routing nodes are not forwarding money but are loaning money instead, payment success cannot be guaranteed. Moreover, the higher the transaction size, the lower payment success will be[15], and the harder will it be to find a complete pathway from buyer to seller. We can conclude that the Lightning Network is less reliable for large transactions.

Another issue of the Lightning Network is linked to centrality and routing. For the sender to compute the best route to reach the receiver may be expensive, especially with large networks, as he/she needs to calculate all the possible shortest paths. Not all nodes may necessarily have sufficient processing power to find the most efficient route.

An increase in the centralization of the network may lead to a decrease in *security* and, in turn, a decrease in the overall efficiency of the network. The most central nodes are more prone to being attacked by malicious parties who want to disrupt the entire balance of the network. If the top 1% of its most central nodes were to be attacked, both network and routing efficiency would drastically decrease. If the attack were random, the network would not suffer as much. We can conclude that the more centralized the network becomes, the riskier it is for the network.

The Lightning network tries to do its best in protecting confidentiality by keeping individual balances secret and adopting *onion routing*. The latter allows for anonymous routing of payments in the Lightning Network. The intermediary nodes can

[14] Rene Pickhardt et al. (2021) "Security and Privacy of Lightning Network Payments with Uncertain Channel Balances".
[15] Look at paragraph 3.7.

only view the positions of their immediately preceding node and their immediately following node in the path. However, they are not aware of how many nodes are present in the payment route, and they cannot identify the sending and receiving nodes. Despite onion routing, some "[…] statistical evidence can be gathered about the sender and receiver of the LN payments, since a substantial portion of payments involves only a single routing intermediary, who can easily de-anonymize participants"[16]. However, plausible deniability still holds, meaning that the sender node can deny being the payer node and state that the payment was initiated by one of its neighbours instead.

[16] Ferenc Béres et al. (2020) "A Cryptoeconomic Traffic Analysis of Bitcoin's Lightning Network", Institute for Computer Science and Control (SZTAKI).

CHAPTER 3

The Lightning Network Performance Analysis

3.1 Dataset Explanation

The dataset consists of *six snapshots* of the Lightning Network in 2018.12.08, 2019.02.21, 2019.04.10, 2020.12.27, 2021.05.30 and 2022.02.07, extracted using the *describegraph* command of the lightning node called LND[17]. Graph construction started from a JSON file which contains information about nodes and channels. Node information includes node public key and node alias.

- The *public key* is a 66-character "cryptographic code used to facilitate transactions between parties"[18]. It allows users to send/receive payments and enables channel opening. An example of Lightning ID is "02004c625d62-2245606a1ea2c1c69cfb4516b703b47945a3647713c05fe4aaeb1c".
- Node *alias* is just a user-defined creative name that uniquely identifies a node in the network, such as "LivingRoomOfSatoshi.com (LND) 2". The alias can be changed over time.

Edge/Channel information includes channel id, channel nodes, channel capacity, node policy, time lock delta and min_htlc.

- *Channel id* is a 16-number code which uniquely identifies a channel. An example of channel id is "599774796939788289".
- *Channel node* refers to the public keys of the two participants involved in the channel. In the JSON file, it's defined as Node1_pub and Node2_pub, , where Node1 is the node who took the initiative to open the channel.
- *Capacity* is the total balance (in satoshi) of a channel, meaning that it's given by the sum of the individual balances of the participants. Channel capacity is set at the opening of the channel and cannot be changed throughout its lifetime without closing the channel.

[17] https://github.com/lightningnetwork/lnd
[18] Jake FrankenField (2021) "Public Key", Investopedia.

- *Node1_policy* and *Node2_policy* refer to the fees that node1 and node2 require for forwarding a payment in which they are involved as intermediaries. The *base fee* (fee_base_msat) is the fee required in forwarding every payment regardless of the payment size. The *fee rate* (fee_rate_milli_msat) refers to the fee that is charged as a percentage of the value of the payment. The larger the payment, the higher the requested fee will be.
- The *Time lock delta*[19] enables to give an "expiration date" to a transaction which otherwise could remain forever pending. It's measured in number of blocks. The maximum time lock value is equal to 14*144 = 2016 which is the number of blocks that are expected to be mined in 14 days.
- *Min_htlc* is the smallest value htlc a node will accept. It's a static parameter set during channel opening and remains fixed until channel closure.

3.2 Graph Construction

To study the properties of the Lightning Network, we need to construct the graph starting with the data. A graph G = (N, E) is a data structure consisting of a finite set of nodes N and a set of edges E connecting them, where each edge is defined by a pair of nodes (n_i, n_k). In the Lightning Network, an edge is a channel which can be opened by any two nodes (n_i, n_k) ∈ N such that $n_i \neq n_k$. G does not admit self-loops, meaning that no node in the network can create a channel with itself.

The first step consists in constructing the graph for each snapshot of the network. We collect the list of nodes and the corresponding edge list then, we build the graph using the *graph_from_data_frame* function in the igraph package in R. Theoretically, a channel is directed from user A to user B if user A has a balance[20]. Therefore, a channel can be bidirectional if both participants own a positive balance. For the sake of simplicity, we assume that the network is undirected and unweighted[21]

[19] Look at paragraph 2.3.1.
[20] Yuwei Guo (2019) "A Measurement Study of Bitcoin Lightning Network", Beihang University.
[21] For the time being, we do not add weights to the graph. In one of the following analyses the network will be weighted according to channel capacities.

as the purpose of this study focuses on analysing the evolution of the properties of the network, especially the ones linked to connectivity, and not on payment direction.

3.3 The evolution of the Lightning Network's topology

In studying the evolution of the Lightning Network, we focused on the following metrics:

- Number of nodes
- Number of edges
- Number of (weakly) connected components
- Average path length and diameter
- Average degree, maximum degree and minimum degree
- Assortativity coefficient

As shown in Table 1, the network has grown impressively from 2018.12.08 to 2022.02.07. The *number of nodes*[22] increased by approximately 1060% while the *number of edges* increased by approximately 661%. Specifically, the number of nodes and edges almost doubled[23] from 2021.05.30 to 2022.02.07.

Dates	# Nodes	# Edges	# CC (Weak)	Diam.	Assort. Degree	Avg. len.	Max Degree	Degree Sd
2018.12.08	1881	12951	3	8	-0.294	2.83	625	37.00
2019.02.21	+76.5% 3321	+122.3% 28793	3	7	-0.289	2.77	625	37.00
2019.04.10	+23.4% 4099	+34.6% 38751	2	7	-0.309	2.71	1249	67.58
2020.12.27	+103.5% 8343	-4.5% 36998	62	12	-0.222	3.60	1307	40.52
2021.05.30	+41.5% 11809	+29.3% 47858	88	12	-0.207	3.66	2014	42.38
2022.02.07	+68.9% 19946	+78.9% 85632	112	12	-0.186	3.64	2795	47.08

Table 1: Topological Metrics Evolution

[22] With the expression "number of nodes" I refer to all of the nodes in the network, including the isolated nodes, if present, i.e. the nodes having no connections in the graph.
[23] In 2021.05.30 the number of nodes is 11809 and the number of edges is 47854, while in 2022.02.07 the number of nodes is 19946 and the number of edges is 85632.

It also appears that the graph has become progressively disconnected. A graph is said to be *connected* if there exists a path between every pair of nodes or equivalently if the *distance*(n_i, n_k) $\neq \infty$ for every (n_i, n_k) $\in$ E. Ideally, we would want a connected network, as isolated nodes and small components do not serve the network and the network does not serve isolated nodes and small components. The number of *weakly connected components*[24] grew from 3 to 112. Among them, there exists one main component which constitutes most of the network. The number of nodes excluded from the largest component is 4 in 2018.12.08 and 234 in 2022.02.07, with an average of 2 nodes per weakly connected component in the last snapshot of the graph. Unfortunately, the participants belonging to the smaller components miss out on the full benefits (especially the ones linked to routing) of the Lightning Network. Here, the percentage of nodes belonging to the smaller components grew from 0.2% in 2018.12.08 to 2.2% in 2020.12.27 and declined to 1.17% in 2022.02.07. Therefore, in 2022.02.07, 1.17% of nodes in the network lost the benefits of the main component.

The *average path length* increased in three years by approximately 1 hop (or 1 intermediary), where the average path length is defined as the average number of steps along all shortest paths between every pair of nodes. In 2022.02.07 it takes twelve steps to connect the two most distant vertices in the network compared to the eight steps of 2018.12.08. Ergo, the *diameter*[25] has increased by 4 hops.

Formally, the *degree* of a node is the number of edges that are incident to the vertex. In the Lightning Network, it can be viewed as the number of channels each node has currently open. While the *maximum degree* has increased from 625 to 2795, the *mean degree* has grown from 13.8 in 2018.12.08 to 18.9 in 2019.04.10 and then suddenly decreased by more than 50% in the following years until reaching 8.6 in 2022.02.07. This behaviour could be explained by an increase in the degree centralization, meaning that a subset of nodes has a much higher degree than the

[24] "A weakly connected component is a subgraph of the original graph where all vertices are connected to each other by some path, ignoring the direction of edges". Source: "Find Weakly Connected Components in a Directed Graph", https://www.geeksforgeeks.org/find-weakly-connected-components-in-a-directed-graph/, 2021.
[25] The diameter is defined as the longest shortest path between any two nodes in the network.

average one and therefore enjoys a higher centrality in the network. Lastly, we define the *assortativity coefficient*[26] as a correlation measure of the degree between pairs of nodes in the graph. This metric assumes values between –1 and 1. In all six snapshots of the network, the assortativity coefficient is negative, indicating that there are relationships between nodes of different degrees. More precisely, the assortativity degree has progressively increased from –0.294 in 2018.12.08 to –0.186 in 2022.02.07 showing growth in the number of relationships between nodes of similar degrees.

3.4 The Evolution of Centrality in the Lightning Network

In principle, the Lightning Network is decentralized, meaning that every participant has the same level of importance and control over the network. This layer-2 protocol claims to be able to find a way "[…] to encompass all transactions in a way that doesn't sacrifice the decentralization and security that [Bitcoin] provides"[27]. An increase in centralization would represent an issue in many aspects. The Bitcoin Lightning Network (BLN) is supposed to solve Bitcoin's scalability problem by processing most transactions off-chain and deferring broadcasting the balance until channel closure. This layer-2 protocol "aims at breaking the trade-off between block size and centralization"[28] enabling the Bitcoin blockchain to potentially achieve scalability, decentralization, and security at the same time proving the famous scalability trilemma[29]. If the BLN were to become increasingly centralized, it would sacrifice one of the main features of the Bitcoin blockchain. Moreover, high centrality nodes may render the network more susceptible to security attacks. Therefore, when

[26] The assortativity degree coefficient is defined as:

$$r = \frac{\sum_{i=1}^{n}(Xi - \bar{X})(Yi - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(Xi - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Yi - \bar{Y})^2}}$$

Where X and Y refer to the degree of any two nodes in the network.

[27] Joseph Poon (2016), The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, p.2 – The BLN Whitepaper

[28] Jian-Hong Lin et. al (2020), Lightning network: a second path towards centralisation of the Bitcoin economy, New Journal of Physics, p.2

[29] https://vitalik.ca/general/2021/04/07/sharding.html

studying the Lightning Network, it is crucial to focus on centrality, particularly on *betweenness centrality.*

Betweenness centrality is a node-level measure that quantifies the number of times a node lies on the shortest path between any pair of nodes in the graph. Let σ_{st} denote the number of shortest paths from s ∈ V to t ∈ V and let $\sigma_{st}(v)$ denote the number of shortest paths from s to t that v lies on, betweenness centrality is defined as:

$$C(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Equation 1: Betweenness centrality

A high betweenness centrality node lies on many shortest paths and can play the role of intermediary in many transactions, exercising a great influence on the network. It is therefore vital to study and carefully observe the behaviour of these nodes, which, if inactive or malicious, could hinder the natural flow of payments. In addition, these nodes are the most popular targets for those wishing to attack the network and, as a result, represent a vulnerability.

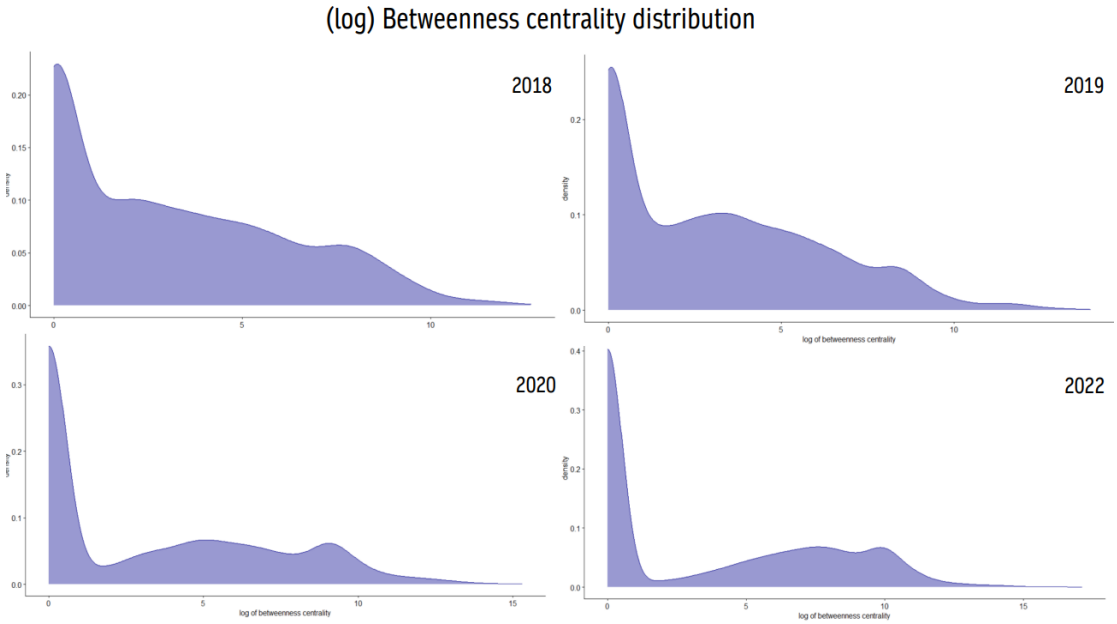

Figure 9: The evolution of the distribution of the log(betweenness centrality +1)

I plotted the distribution of the betweenness centrality for each snapshot of the network. The data resulted heavily skewed and graphing it as-is rendered it difficult to interpret the results. Ergo, I plotted the distribution of the log(betweenness centrality + 1). By taking the logarithm, it's easier to see all the data, even the smallest patterns hiding in it. I opted for a density plot rather than a histogram in representing the distribution of the variable as density plots are a smoothed version of histograms, not affected by the number of bins and are more accurate in determining the distribution shape. The peaks of the density plot can help display where there is a concentration of values over the interval. As shown in Fig. 9, there is a peak in zero in all plots and the proportion of zero betweenness centrality nodes rises across the years. Furthermore, the density plot becomes progressively skewed due to an increase in higher betweenness centrality nodes (e^{10}) and a decrease in the number of mid-range betweenness centrality nodes.

Date	Mean BC		Median BC		Maximum BC	
2018.12.08		1713.12		9.94		314,414
2019.02.21	+72%	2939.95	+37%	13.62	+134%	748,253
2019.04.10	+19%	3496.45	-9%	12.41	+51%	1,130,985
2020.12.27	+197%	10389.83	-86%	1.72	+289%	4,398,357
2021.05.30	+47%	15235.40	-100%	0	+162%	11,541,595
2022.07.02	+69%	25749.79	+0%	0	+134%	27,031,246

Table 2: The evolution of mean, median and maximum betweenness centrality

The empirical cumulative distribution function (ecdf) can help us deep dive into the evolution of the betweenness centrality across the years. As seen in Fig.10, in 2018.12.08, 97% of nodes have a betweenness centrality lower than 1900, and 75% of nodes have a betweenness centrality score below 85. Even though most nodes have such low betweenness centrality scores, its mean appears to be equal to 1713.12. Hence, there must be a subset of nodes with a much higher betweenness centrality which pushes the mean way higher. In 2022.02.07, 95% of nodes have betweenness centrality below 19700, and 75% have betweenness centrality below 9000. Again, the mean is

higher than most of the betweenness centrality values, indicating that some highly central nodes in the network are extremely above average. We can conclude that the mean is not representative of the average betweenness centrality for all years. Moreover, the median shows an increase in the disparity between the centrality of nodes. For instance, in 2021.05.30 and 2022.02.07, at least half of the nodes have betweenness centrality equal to zero, despite the increase in the average.

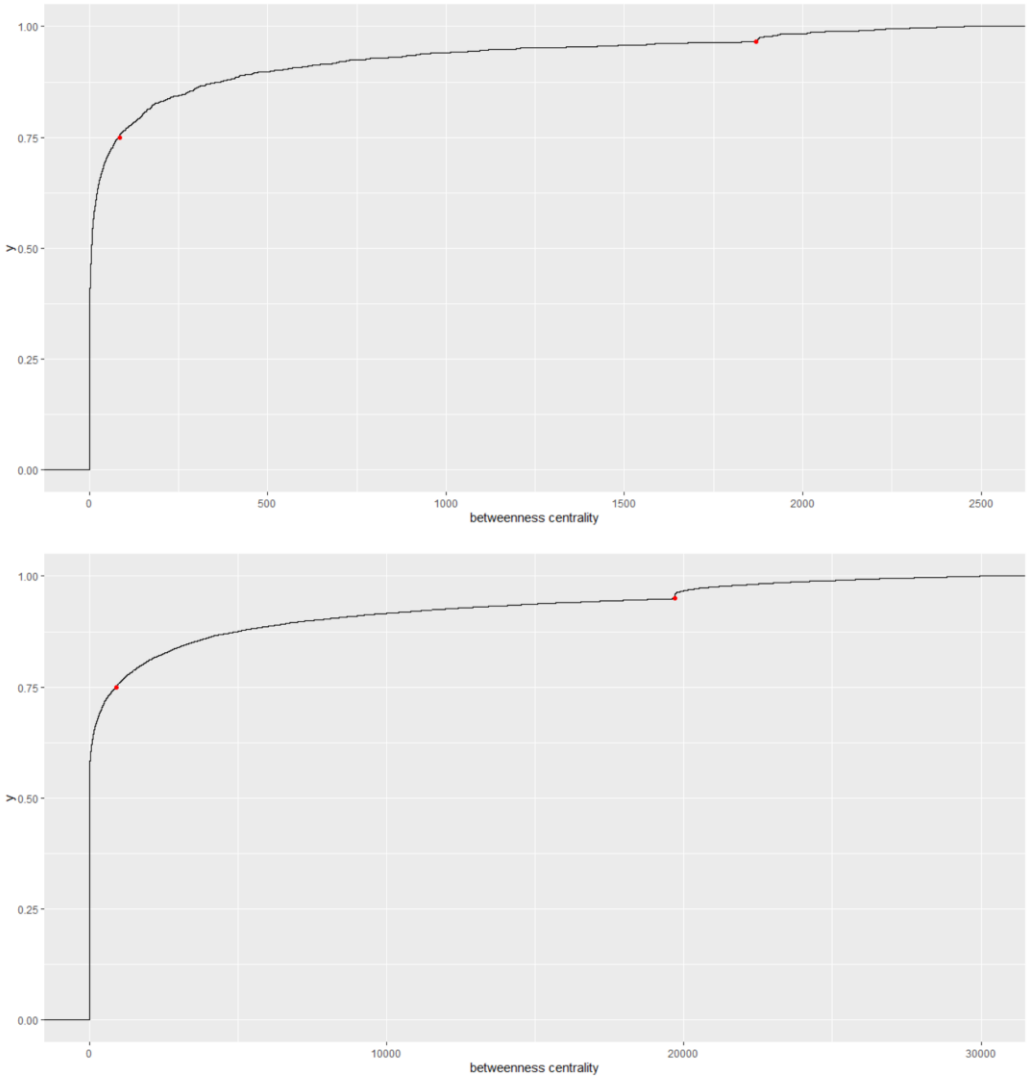

Figure 10: Two plots of the ecdf of the betweenness centrality in 2018.12.08 (top) and 2022.07.02 (bottom).

As shown in Table 2, there is a tendency for a single node (or a subset of nodes) of being more central with respect to all the other nodes in the network. Moreover,

maximum betweenness centrality grew impressively from 314,414 in 2018.12.08 to 27,031,246 in 2022.02.07. Thus, looking at graph centrality may help us in understanding whether the network has become increasingly centralized over the years or not. Let's introduce *Freeman centralization*, a graph-level centrality score formulated on a node-level centrality measure. This metric is based on "the differences between the centrality of the most central point and that of all others"[30]. Let u ∈ V be the node with the highest betweenness centrality and $C(v_i)$ be the betweenness centrality of all nodes v_i " i ∈ [0, N], we define Freeman centralization as the ratio of the actual sum of differences between C(u) and C(vi) and the maximum possible sum of differences, or equivalently:

$$\frac{\sum_{i=1}^{N}\big(C(u) - C(v_i)\big)}{max\ \sum_{i=1}^{N}\big(C(u) - C(v_i)\big)}$$

Equation 2: Freeman (degree) Centralization

This coefficient assumes values between 0 and 1, with 1 being the most centralized network of N nodes (i.e., the star graph) and 0 being a completely decentralized network. The graph-level centrality was calculated using the *centralization.betweenness* function from the igraph package in R. According to this measurement, the network in 2018.12.08 had centralization equal to 17.73%, while in 2022.02.07 it had centralization equal to 13.58%. Therefore, we can conclude that despite the increase in node-level centrality, the network in 2022.02.07 is no more centralized than in 2018.12.08.

Date	2018.12.08	2019.02.21	2019.04.10	2020.12.27	2021.05.30	2022.02.07
Freeman Centralization	17.73%	13.53%	13.43%	12.61%	16.54%	13.58%

Table 3: Freeman Centralization measured in the six snapshots of the Lightning Network. The highest centralization was registered in 2018.12.08 and the lowest in 2019.04.10.

[30] Linton C. Freeman (1979) "Centrality in Social Networks Conceptual Clarification", Lehigh University, p. 227

3.5 The Roles of Hubs in Longest Shortest Paths

Another way of thinking about centralization is counting the number of hubs[31]. The emergence of *Lightning Hubs* might suggest that the Bitcoin Lightning network is indeed centralized, or at least not decentralized to the core. As seen in Fig. 11 there are three types of networks. The presence of solely one hub indicates a maximally centralized network, meaning that every single payment must be forwarded by that central node or central institution. Having this level of centralization in the Lightning Network would not make any sense and would lead to the dangerous single-point-of-failure problem. Instead, when there are many connected hubs, where each one is, in turn, connected to the other nodes, we are referring to a moderately de(centralized) network. This argues that the functioning of the network is somewhat controlled by a subset of participants who act as hubs or intermediaries. Lastly, if every node in the network is strongly connected, every node is a hub, or more precisely, no node in the network is really a hub. The latter represents the case of a maximally decentralized network which is how Thaddeus Dryja and Joseph Poon initially proposed in their whitepaper the Lightning Network.

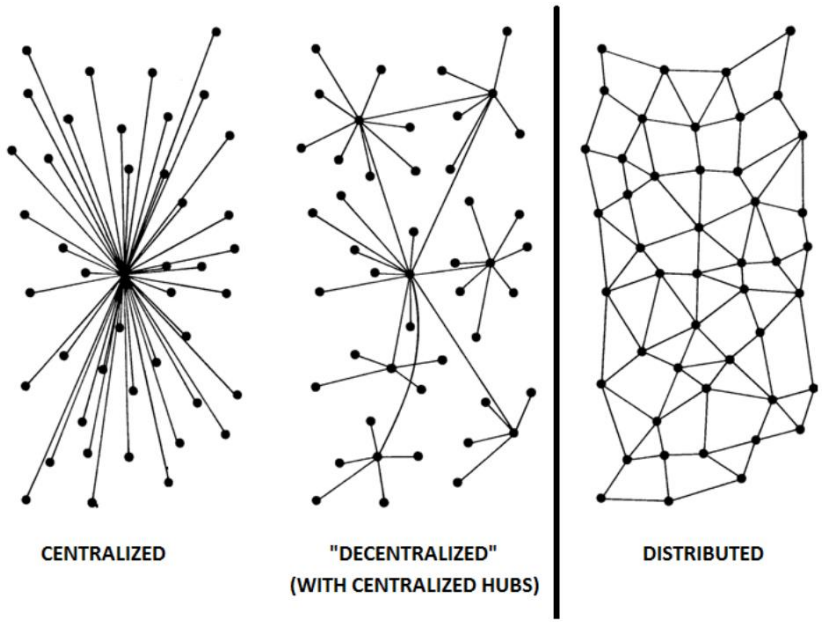

Figure 11: Hubs in the Lightning Network, "Don't Buy Into the FUD", Medium

[31] "[…] A hub is a node with a number of links that greatly exceeds the average". Source: "Hub (network science)", Wikipedia.

These hubs are also referred to as "*routing nodes*"[32], meaning that they are actively involved in forwarding payments, facilitating the connectivity of many participants in the network. Due to the emergence of hubs, the network may appear centralized to some extent. However, users can choose among several routing paths[33] (if there are any) which involve different routing nodes, like in a proper decentralized system. As seen in paragraph 2.3.3, nodes gain a fee for their routing services. This can be viewed as an opportunity to make profits and may lead in the future to the creation of "routing businesses", just like mining rewards led to the advent of mining pools in the Bitcoin blockchain.

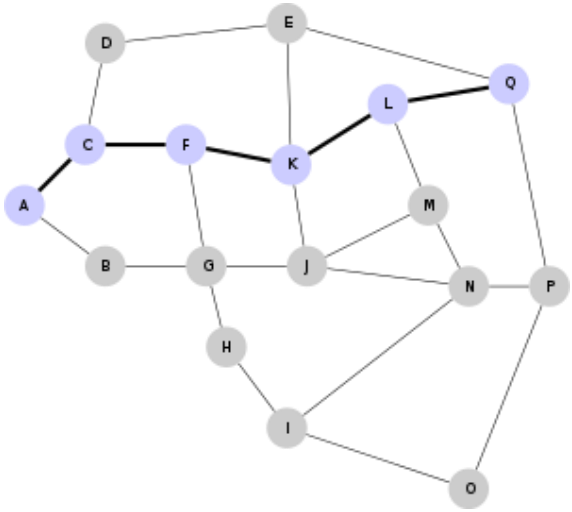

Figure 12: An example of route linking node A and node Q. Source: Wikipedia

In graph theory, the shortest path between a pair of nodes in the graph is the route or path that links those two vertices such that it minimizes the sum of edge costs (weights). For instance, in Fig. 13, the path which minimizes the costs is ACEDF. In an unweighted graph, the term shortest path only refers to the route involving the minimum number of edges. If the graph represented in Fig. 13 had no weights, the shortest path would be ABDF, as we no longer care about edge weights but only about the number of steps (or edges) required to connect nodes A and F.

[32] Roy Sheinfeld (2018) "Mitigating the Risk of Running Lightning Network Hubs", Medium, https://medium.com/breez-technology/the-risk-of-running-lightning-network-hubs-23ef333c07a4

[33] In some cases, participants may find themselves in choosing among a great number of possible routing paths of the same length. This issue is discussed further in paragraph 2.6.

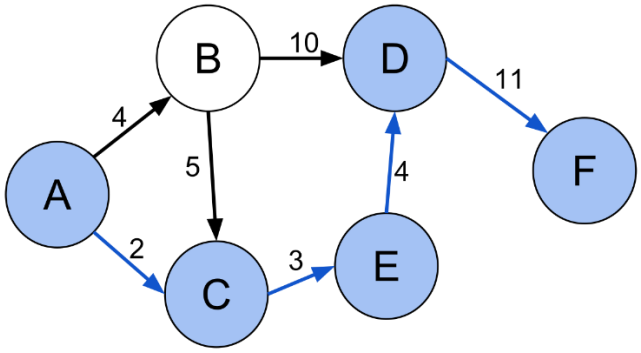

Figure 13: An example of shortest path. To reach node F from node A we must traverse at least four edges. Equivalently, it can be said that the shortest path linking node A and node F has length four. In the LN context, participants C, E, and D, are intermediaries in the transaction, and the weights are the fees requested by the respective intermediaries for forwarding the payment to node F. Alternatively, node A must find another path to reach node F such as ABDF, which, however, is not as efficient.

With multi-hop transactions, the payer may be faced with a wide choice of routes to reach the payee. In our simplified model, which does not take into account the node policies and payment channel states, we would like to choose the shortest path, namely, the one which minimizes the length of the route linking the sender and recipient of the payment, thereby involving the least number of intermediaries possible. Why? Because the fewer the intermediaries, the lesser the fees and the greater the success probability of the payment. As a matter of fact, we cannot be sure that the transaction will be successful as we do not know the state of the intermediaries, which may be inactive.

This part of the work focuses on the roles of hubs in longest shortest paths. Specifically, it concentrates on how important (in terms of centrality) are the nodes involved in routing and what positions high betweenness centrality nodes tend to occupy in the path. Many studies[34] demonstrate how the removal of those central nodes negatively impacts the Lightning Network's efficiency[35] as opposed to the removal of random nodes. Therefore, it is crucial to analyse those nodes and their behaviour.

[34] Stefano Martinazzi et al. (2020), "The evolving topology of the Lightning Network: Centralization, efficiency, robustness, synchronization, and anonymity", Politecnico di Milano
Yuwei Guo (2019), A Measurement Study of Bitcoin Lightning Network, Beihang University

[35] Efficiency between two vertices is defined as the reciprocal of the distance between the two nodes. Or equivalently, $E(i,j) = \frac{1}{d(i,j)}$ where i ≠ j. The global (network) efficiency is defined as the average of all

We start by finding the list containing all possible shortest paths in the network. From this list, we derive two summary statistics on those shortest paths. The first consists of obtaining the number of possible shortest paths between every pair of nodes (which we will study more in detail in the next paragraph). The second consists in identifying the longest shortest path connecting every pair of nodes. In this paragraph, we concentrate on the second measure. In particular, the focus is on the diameter. Let's take into consideration the snapshot of the Lightning Network in 2018.12.08. The diameter[36] of the network has a length of eight. Once we have identified the nodes at the extremes of the diameter (1225 and 1404)[37], we can recover all the intermediary nodes and print the list of paths having maximum length. As can be seen in Fig.6, there are nine different shortest paths connecting the source and target nodes.

```
[1225, 391, 1422, 1425, 408, 838, 959, 1155, 1404]
[1225, 391, 1422, 1425, 1565, 968, 528, 1155, 1404]
[1225, 391, 1422, 1425, 1654, 968, 528, 1155, 1404]
[1225, 391, 1422, 1425, 1654, 968, 528, 1155, 1404]
[1225, 391, 1422, 1425, 1710, 968, 528, 1155, 1404]
[1225, 391, 1422, 1748, 314, 968, 528, 1155, 1404]
[1225, 391, 1422, 1748, 314, 968, 528, 1155, 1404]
```

Figure 14: Output lists of nodes connecting the two most distant participants in the network.

The next step consists in calculating the betweenness centrality of those nodes. The main intuition is that the nodes which occupy a central position in the shortest path tend to have higher betweenness centrality values, unlike the more peripheric positions. To verify this assumption, we plot the distribution of the betweenness centrality for each position occupied in the shortest path. Ergo, if the diameter has a length of eight, there are nine positions where a node can find itself along the shortest path. Now, we can check what positions the highest betweenness centrality nodes occupy. I decided not to plot the first to third, eighth and ninth positions as the nodes involved are always the same in all paths. Therefore, the distribution of those positions wouldn't have made

efficiencies over all pairs of vertices in the graph. Or equivalently, $E_{glob}(G) = \frac{1}{n(n-1)}\sum_{i \neq j}(E(i,j))$. In R, there exists the Efficiency function of the "BrainGraph" package.

[36] Look at paragraph 2.3.

[37] 1225 and 1404 are just numbers that uniquely identify each node in the graph. These numbers substitute the public key for the sake of simplicity.

much sense. Fig. 15 shows how the betweenness centrality on average increases as we approach the most central positions in the route (red line). In fact, as can also be seen in Table 3, the mean betweenness centrality gradually increases from zero in positions one and nine to the value of 80717.50 in position 5.

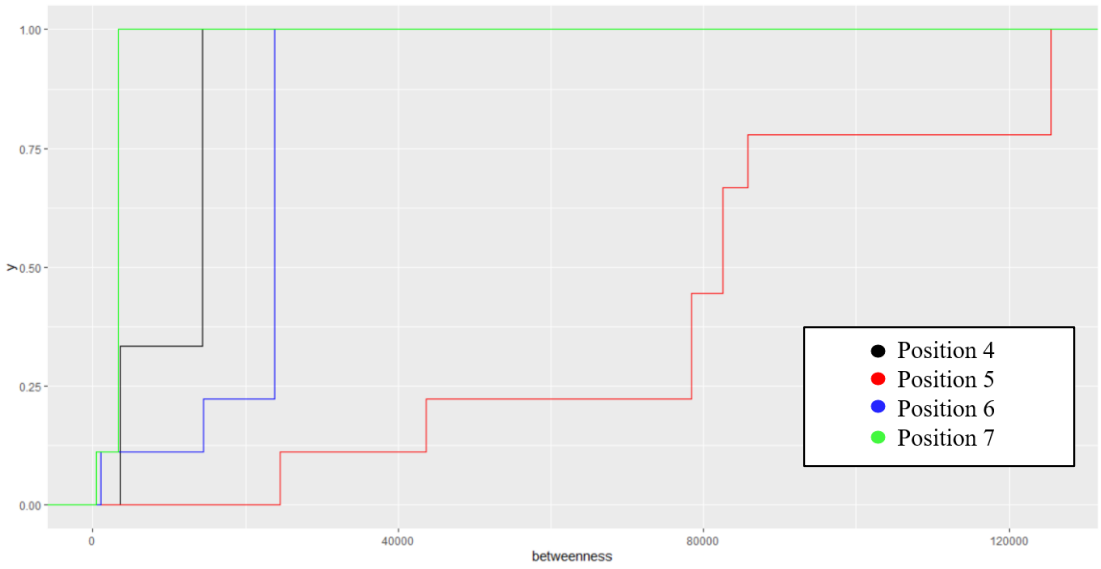

Figure 15: The ecdf of the betweenness centrality in positions 4 (black), 5 (red), 6 (blue), 7 (green).

Positions	Mean BC	BC Sd
Position 1	0	0
Position 2	1874	0
Position 3	3756.65	0
Position 4	10780.47	5384.88
Position 5	80717.50	32740.21
Position 6	20230.45	7802.36
Position 7	3028.62	987.92
Position 8	1876.96	0
Position 9	0	0

Table 4: Mean and standard deviation of the betweenness centrality across all positions in the diameter.

For each of the nine maximum-length shortest paths, there exists at least one hub (position 5) which facilitates the connectivity among the nodes in the network, even

the most distant ones. It would be interesting to verify how long would the diameter become if we deleted the hubs in the shortest path. More generally, we can ask ourselves the following questions: how long would the diameter become by removing the top 1%, 5% or 10% of the most central nodes in the network? How disconnected would the network become measured in the number of (weakly) connected components? Future studies and additional research might address these questions.

3.6 Some Statistics on Shortest Paths in 2022.02.07

Shortest paths play an essential role in guaranteeing the correct functioning of the Lightning Network. Ergo, this study focuses on determining some interesting statistics on shortest paths, in particular concerning the snapshot of the graph in 2022.07.02. As seen in the previous paragraph, there may exist multiple shortest paths between each pair of nodes in the network, which would complicate the choice of the "best" route for delivering a payment to another participant in the network. These shortest paths can be seen as different combinations of intermediary nodes involved in forwarding a transaction. Of course, we must choose the most efficient one. Personally speaking, this subject is worthy of being further analysed in detail.

Let's start by collecting for each node the number of possible shortest paths to reach any other node in the network together with the length of that shortest path. Then, we can focus on the extremes of these two statistics by gathering for each node:

1) The maximum number of shortest paths that each node can choose from to reach another node in the network. For instance, as seen in Table 5, node_1 can pick among 760 possible shortest paths to reach node_507, meaning that node_1 has to select, at most, among 760 routes to deliver a payment to any other node in the network.
2) The longest shortest path that each node has to travel to reach the most distant node. For instance, as seen in Table 6, the length of the shortest path between node_1 and node_9632[38] is equal to 9, meaning that node_1's shortest paths to

[38] Where node_9632 is the most distant node for node_1.

deliver a payment to any other node in the network have a length smaller than or equal to 9. Interestingly, node_9632 seems to be involved in many longest shortest paths concluding that it may be a node generally distant from the majority of the other nodes in the network.

ID of the 1st node	ID of the 2nd node	# Shortest paths
1	507	760
10	13205	1964
100	6427	956

Table 5: An extract of the table containing the maximum number of shortest paths for each node.

ID of the 1st node	ID of the 2nd node	Length of the longest SP
1	9632	9
10	9632	9
100	9632	8

Table 6: An extract of the table containing the longest shortest paths for each node

Table 7 shows some summary statistics on the previous findings. On average, each node has a longest shortest path length of 8.82, meaning that the sender must involve at least eight intermediaries to deliver a payment to its most distant node. As previously seen in paragraph 2.3, the diameter has length twelve compared to the one of the graph in 2018.12.08 which had a length equal to eight. Therefore, in 2022.02.07, it takes four intermediaries more to connect the two most distant nodes in the network. Shockingly, on average, each node has 1924 equally valid routes to choose from in the worst case. In particular, the greatest number of shortest paths among which a node must pick is equal to 21,696 compared to the 1860 in 2018.12.08. This means that one of the nodes in the network has 21,696 possible combinations of intermediaries to reach another participant. We can conclude that, from this point of view, the Lightning Network has become increasingly complex concerning routing and the choice behind routing. Table 7 shows that both variables have outliers, especially “longest shortest

path length". As seen in Fig. 17, most observations are equal to eight, nine or ten. However, there are some shortest paths with lengths of five to seven, eleven and twelve, which are infrequent and extreme values. Also "maximum number of shortest paths" has a significant amount of outliers. Fig. 16 displays the distribution (histogram) of the variable. As can be seen, the plot is strongly skewed to the left, implying that there are a few higher values which we classify as outliers. These extreme values are generally not representative of the graph. Instead, it's better to look at the mean of each of these variables.

Statistic	Longest shortest path length	Maximum number of shortest paths
Mean	8.82	1,924.2
Maximum	12	21,696
Minimum	5	236
St. Dev	0.62	1,459.8
#Outliers	769	101

Table 7: Summary statistics on shortest paths in 2022.02.07

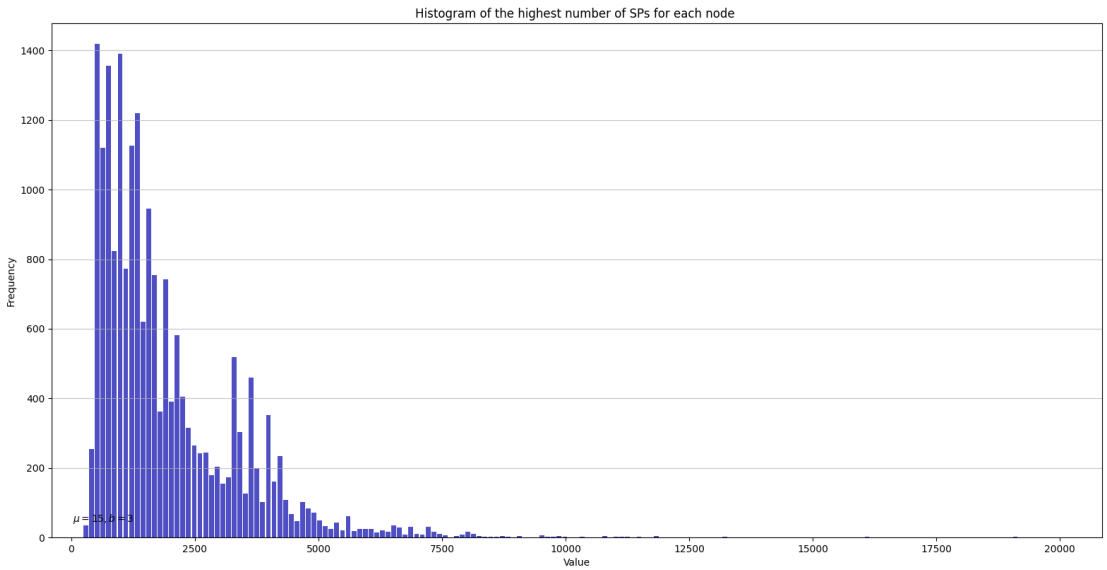

Figure 16: The distribution (histogram) of the maximum number of shortest paths for each node of the network in 2022.02.07

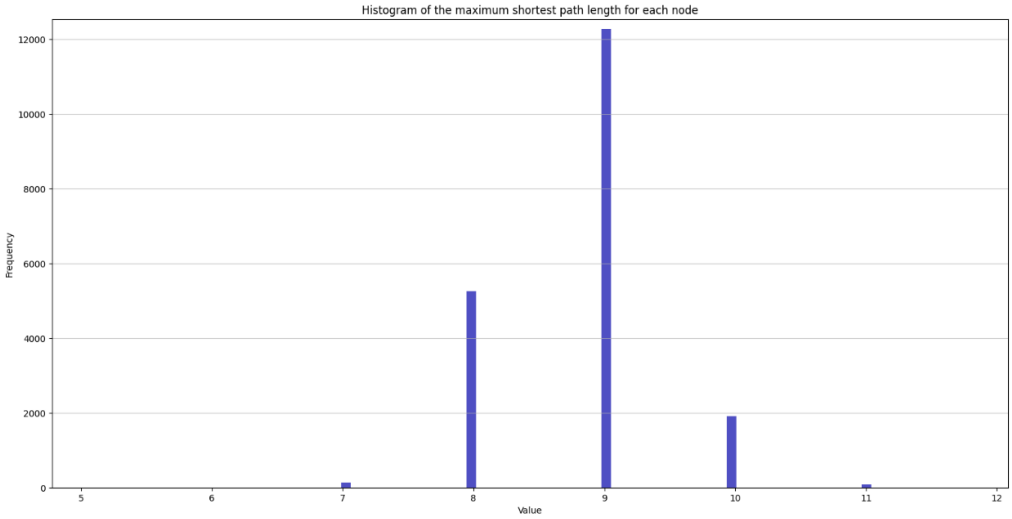

Figure 17: The distribution of the longest shortest path for each node of the network in 2022.02.07

3.7 Network Routing Performance

Routing is the process of selecting a path in the network to perform a transaction. Various factors may affect payment success in the Lightning Network. The first is related to the capacity of the channels involved in routing. If the capacity is below the amount of the transaction, there is no way that node can forward the payment. Not only the capacity of the channel must be larger than the payment amount, but also the individual balances of the participants involved in the routing process must be sufficient to forward the transaction. However, the individual balances are not known for privacy reasons. Therefore, payment success may be trial and error, as it depends on the reliability and availability of the intermediaries. For the sake of simplicity, let's focus solely on capacity. Network capacity refers to the sum of the capacities of all channels in the network. As seen in table 8, network capacity increased by 652% from 2018.12.08 to 2022.02.07. Specifically, the total amount of bitcoins involved in payments grew from 456.27 in 2018.12.08 to 3431.55 in 2022.02.07. In 2020.12.27, network capacity decreased by 1.9% compared to the previous snapshot. It's crucial to address that the value of bitcoin increased by 356% from 2019.04.10 to 2020.12.27. This sudden appreciation in value could explain the decrease in network capacity. As we know, bitcoin is highly volatile, meaning that its value is subject to sudden increases/decreases due to external factors such as supply and demand or even media hype. Table 9, instead, shows the correspondent values in Euro. The conversions

capture the value of bitcoin in all snapshots and help give an idea of the amount of money involved in the Lightning Network.

Some additional statistics on channels are displayed in Table 8. Minimum capacity stayed constant in all snapshots, while maximum capacity increased by approximately 8233%. More precisely, it recorded an increase of 2876.2% in 2020.12.27 and a further increase of 180% in 2022.02.07, where the largest capacity is equal to 14 BTC (536,429.74 Euro). Median and mean capacity did not register substantial growth. However, it can be seen that the mean capacity is larger than the median capacity in all snapshots of the network. Looking at the empirical distribution function may help us achieve a better understanding of the distribution of channel capacity and how it evolved across the years.

Date	Network Capacity (BTC)	Mean Capacity[39]	Median Capacity	Max Capacity	Min Capacity
2018.12.08	456.27	0.0352	0.005	0.168	0.0000105
2019.02.21	+55.5% 709.61	0.0246	0.006	+0% 0.168	0.0000105
2019.04.10	+51.7% 1076.27	0.0278	0.008	+0% 0.168	0.0000105
2020.12.27	-1.9% 1055.64	0.0285	0.005	+2876% 5	0.0000105
2021.05.30	+31.5% 1388.67	0.0290	0.005	+0% 5	0.0000105
2022.02.07	+147.1% 3431.55	0.0401	0.01	+180% 14	0.0000105

Table 8: Statistics on channel capacity in bitcoin

Date	Network Capacity (euro)[40]	Mean Capacity[41]	Median Capacity	Max Capacity	Min Capacity
2018.12.08	2,532,129.68	195.35	27.75	93.23	0.06
2019.02.21	2,474,197.19	85.77	20.92	585.77	0.04
2019.04.10	5,082,211.52	132.66	37.78	801.71	0.05
2020.12.27	22,725,395.66	613.54	107.64	107,638.00	0.23
2021.05.30	40,645,704.34	848.82	146.35	146,347.60	0.31
2022.02.07	131,484,676.74	1,536.49	383.16	536,429.74	0.40

Table 9: Statistics on channel capacity in euro

[39] Capacity (BTC) refers to channel capacity and not network capacity.
[40] The conversion from bitcoin to euro was done looking at the historical data (Yahoo Finance). In this way, we keep track of the appreciation of the value of bitcoin. In 2018.12.08, 1 BTC = 5,549.63 euro. In 2019.02.21, 1 BTC = 3,486.70 euro. In 2019.04.10, 1 BTC = 4,722.06 euro. In 2020.12.27, 1 BTC = 21,527.60 euro. In 2021.05.20, 1 BTC = 29,269.52 euro. In 2022.07.02, 1 BTC = 38,316.41 euro.
[41] Capacity (euro) refers to channel capacity and not network capacity.

Fig. 18 displays the ecdf of channel capacity in 2018.12.08 and 2022.07.02. In 2018.12.08, all channels have capacities below or equal to 0.168 BTC, with 88% of capacities under 0.168 BTC (and 12% equal to 0.168 BTC) and 80% under 0.09 BTC In 2022.02.07, 95% of channels have capacities smaller than 0.1 BTC compared to 87.5% in 2018.12.08, and 87.5% are under 0.05 BTC compared to 75% in 2018.12.08. Channel capacity has, on average, increased. But, also the disparity between capacity levels has increased. Even though 95% of channel capacities are under 0.1 BTC, there is at least one channel with a capacity equal to 14 BTC. That channel can potentially forward many transactions, even ones with the highest payment amounts. However, for amounts greater than 0.1 BTC, routing is unlikely to happen with only 5% of channels available.

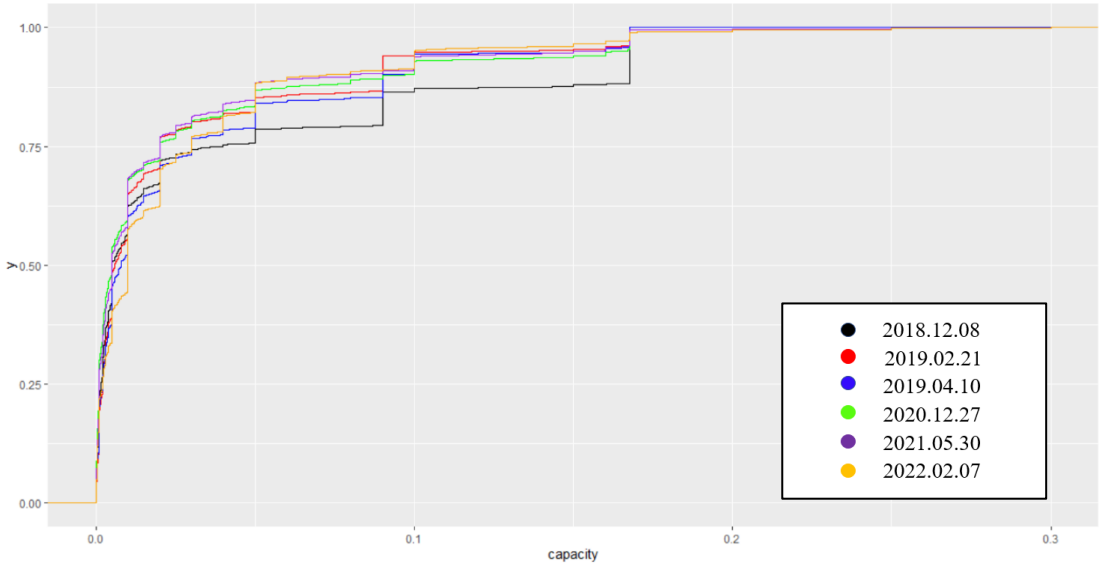

Figure 18: Ecdf of channel capacity in all snapshots of the network. Black: 2018.12.08, Red: 2019.02.21, Blue: 2019.04.10, Green: 2020.12.27, Purple: 2021.05.30, Orange: 2022.02.07

Let's define routing efficiency as the ratio of the non-isolated nodes over the total number of nodes in the network. This metric indicates how able is the network in routing or forwarding payments. This measure is highly dependent on the channel capacity and the amount of the transaction. For each amount, we proceed by deleting the edges which have a weight below the payment amount. Then, we individuate the isolated nodes and compute the routing efficiency. The chosen payment thresholds are 0.0001 BTC, 0.001 BTC, 0.005 BTC, 0.01 BTC and 0.05 BTC.

As seen in Table 10, the routing efficiency of the network decreases as the amount of the transaction increases. For amounts greater than 0.0001 BTC, the network seems less efficient in the latest snapshots or equivalently from 2020.12.27 onwards. For instance, routing a payment of 0.05 BTC has a success rate of 37.29% in 2018.12.08 and 13.61% in 2022.02.07. The appreciation in the value of bitcoin and an increase in the centralization of the network may be the causes of this decrease in routing efficiency for higher payment thresholds.

RE[42]	2018.12.08	2019.02.21	2019.04.10	2020.12.27	2021.05.30	2022.02.07
0.0001 BTC	98.94%	99.85%	99.88%	96.69%	97.59%	98.57%
0.001 BTC	86.70%	92.68%	93.31%	75.82%	73.95%	66.73%
0.005 BTC	65.53%	73.08%	72.55%	42.42%	41.12%	43.44%
0.01 BTC	52.07%	61.82%	62.97%	33.61%	31.04%	35.19%
0.05 BTC	37.29%	34.96%	40.52%	15.74%	12.45%	13.61%

Table 10: Routing Efficiency

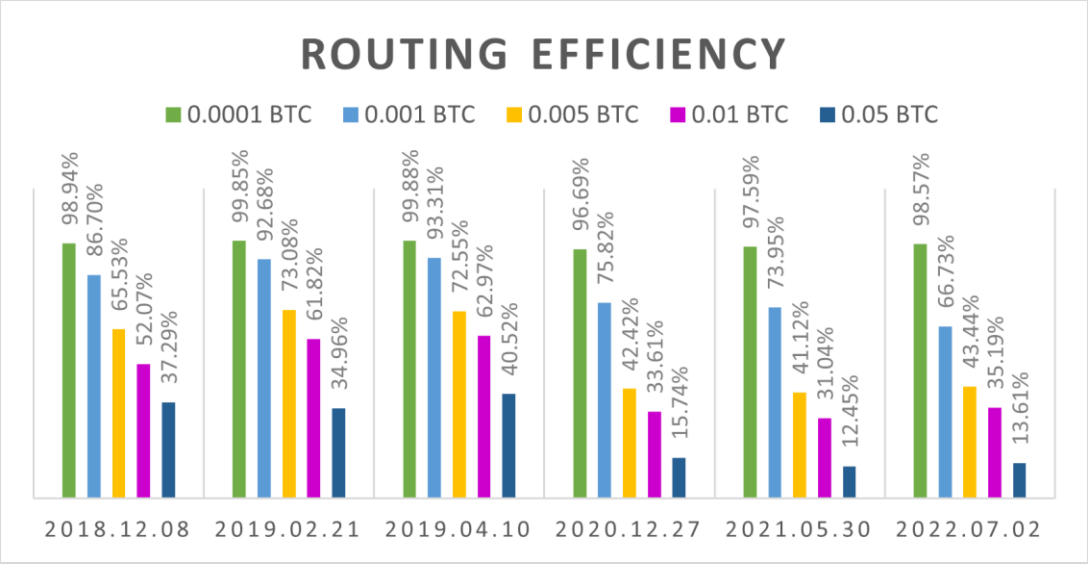

Figure 19: Routing Efficiency

[42] RE refers to routing efficiency. In 2022.07.02, 1 BTC = 38,316.41 Eur (0.0001 BTC = 3.83 Eur, 0.001 BTC = 38.32 Eur, 0.005 BTC = 191.58 Eur, 0.01 BTC = 383.16 Eur, 0.05 BTC = 1915.82 Eur).

Even though routing efficiency decreased across the years, the percentage of channels with capacities larger than the five selected thresholds appears to be larger or, at least, not significantly smaller. For instance, the percentage of channels with insufficient capacity to forward payments of 0.005 BTC is 42.27% in 2018.12.08 and 33.47% in 2022.07.02, even though the percentage of isolated nodes increased from 62.17% in 2018.12.08 to 86.39% in 2022.07.02. Therefore, although routing efficiency is smaller in 2022.07.02, the percentage of channels available for routing is larger compared to 2018.12.08.

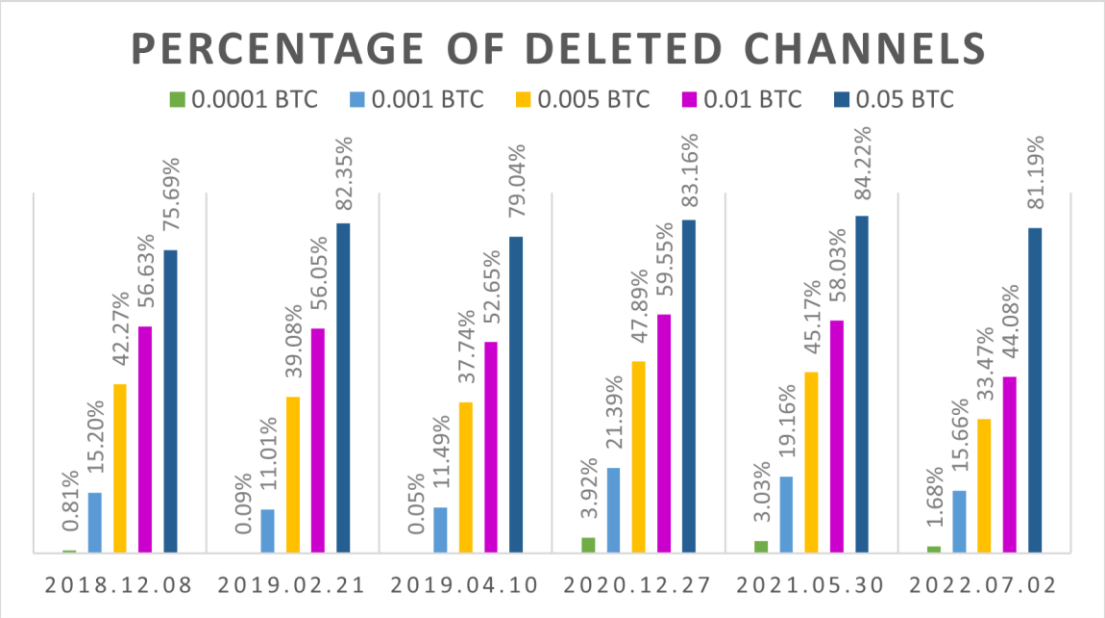

Figure 20: Percentage of deleted edges (channels with capacity smaller than the payment amount)

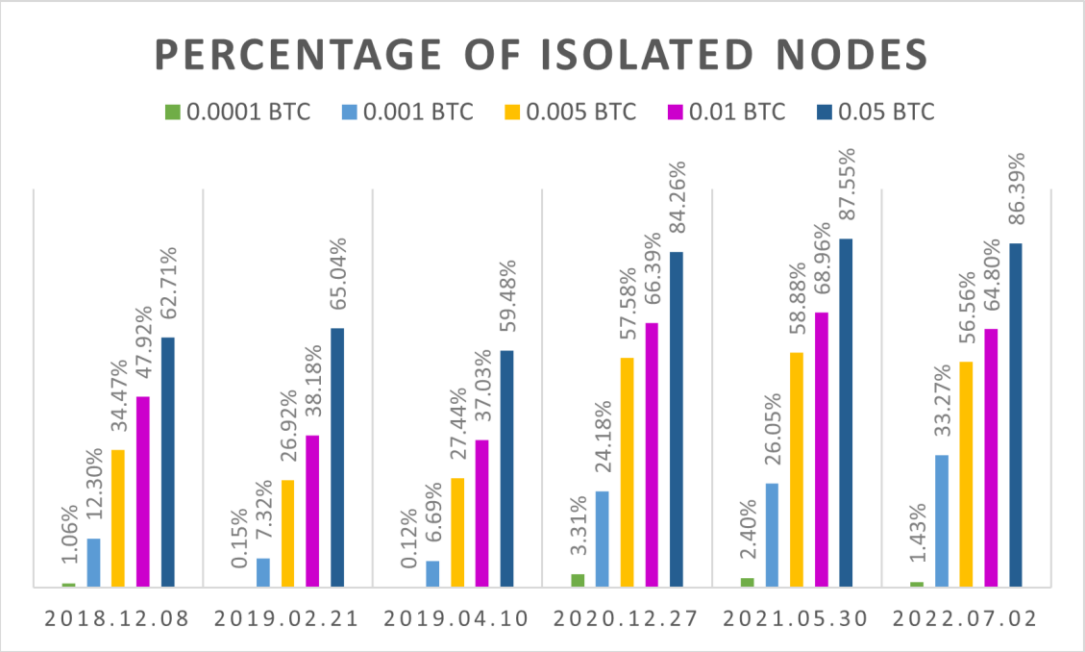

Figure 21: Percentage of isolated nodes after channel deletion

To conclude, it appears that the Lightning Network has become increasingly less efficient in routing, especially for higher payment thresholds. Yet, the Lightning Network has been proposed to enable small and instant bitcoin payments in large numbers, and it wasn't initially suggested to sustain large payment amounts such as 0.05 BTC. Future studies could further address payment success. Instead of focusing solely on channel capacity, it would be interesting to merge the role of hubs or the critical node detection problem (CNDP)[43] with routing efficiency to answer the following questions: How is routing efficiency affected by removing the most central (or important) nodes in the network? How would it differ from a random removal of nodes? To what extent does the reliability and availability of nodes affect payment success?

[43] Mohammed Lalou et. al (2018) "The Critical Node Detection Problem in networks: A survey": The Critical Node Detection Problem (CNDP) is the optimization problem that consists in finding the set of nodes, the deletion of which maximally degrades network connectivity according to some predefined connectivity metrics."

Conclusions

The Lightning Network was proposed to solve Bitcoin's scalability problem by handling most transactions off-chain. Theoretically, through direct payment channels and routing, the LN should be able to sustain up to 25 million transactions per second (compared to the seven transactions per second of Bitcoin). We carried out different analyses to check the actual state of the Lightning Network and how it evolved in the six snapshots taken into account in paragraph 3.1. In the light of our findings, let's try to answer the following questions: is the current state of the Lightning Network adequate for supporting everyday life transactions? Specifically, if the entire world were to use the Lightning Network right now, could the LN process this many transactions successfully? On the bases of the results obtained, the short answer is no. The Lightning Network could not replace Visa's scalability. As seen in paragraph 3.7, the network has become less efficient in routing payments, with a significant decline in 2020.12.27 and onwards. We can conclude that especially for higher sums of money routing is unlikely to happen. Despite the rapid growth of the network, routing efficiency did not improve. The functioning of the network is "controlled" by the same restricted subset of nodes with high betweenness centrality values and very high capacities. While, as seen in paragraph 3.4, at least 50% of participants in 2021.05.30 and 2022.02.07 have betweenness centrality zero, meaning that they lie on nobody's shortest paths. Hence, they have no role in routing and little to no role in increasing the connectivity of the network. Today, the Lightning Network is not adequate for solving the problems it was initially proposed to solve. With no efficient routing, Bitcoin's scalability and transaction speed cannot be drastically improved. I think that the LN has some great potential, but some limits need to be overcome. For now, its use is restricted to "sporadic" and smaller transactions and not for world use.

Future Works

In this study, we assume that the Lightning Network is an undirected and unweighted graph. Theoretically, we know that the Lightning Network is bidirectional as payments can be potentially forwarded both ways and weighted, as each node has a fee policy and each edge has a cost. Travelling across a channel in one direction is not (generally) equivalent to travelling across the same channel in the other direction. The parties owning the channel request different fees for forwarding payments. Crossing the channel in one direction may require a very high fee, and crossing it in the other may require a very low fee. Future studies could replicate calculating the same metrics and compare them to the ones obtained with an undirected and unweighted graph. For instance, how do the shortest paths change? And how does the diameter change accordingly? As seen in paragraph 3.5, shortest paths on weighted graphs are not found by minimizing the number of edges needed to connect any two nodes in the network. Instead, they are found by minimizing the costs of travelling across those edges. Similarly, when we need to choose the fastest route to reach our destination by car. Each road in the route is weighted according to the time needed to travel across it. We must pick the combination of streets that compose the less-time consuming route, i.e. we want to minimize the weights of the path linking the starting point to the destination. Moreover, as seen in paragraph 2.3.3, the fee is given by the sum of the baseFee and an additional fee. The latter is calculated as a percentage of the payment amount (feeRate * amount). The higher the amount to be forwarded, the higher the fee requested by the intermediary. Therefore, the cost of travelling across an edge is not the same for every payment threshold. The shortest paths must be calculated for different payment thresholds.

Another focal point of this study is the concept of centrality and, particularly, of betweenness centrality. As said in paragraph 3.4, the latter is computed on shortest path statistics. Hence, also the betweenness centrality values could drastically change. It would be interesting to compare the values calculated on an undirected, unweighted graph versus a bidirectional, weighted graph.

Paragraph 3.6 illustrates how many nodes face a ridiculously large number of possible shortest paths. The main question one could ask him or herself is the following: how can participants choose among all the possible equally-long routes? It's not a simple question to address. It involves developing a heuristic which could allow us to solve this problem quickly and efficiently. We want to pick the shortest path. So, we can automatically exclude the longer routes. However, we know that each node may have to select between multiple shortest paths. There are other valid aspects to consider when choosing the most performing route. Firstly, we want to avoid traffic as networks can become congested. If, for instance, every node were to pick the same shortest paths and the same intermediary nodes, those nodes would saturate their capacity even if they had a very high budget. Therefore, it would be best to distribute routing across the whole network and avoid congestion. Secondly, we should consider the Lightning Network as a bidirectional graph with the respective routing fees as weights. Clearly, every participant wants to minimize the fees it has to pay to the intermediaries for forwarding the payment. Lastly, we do not want to choose the shortest paths containing channels having a capacity smaller than the amount of the transaction. We could suggest a service based on an algorithm that considers all the previously cited aspects of routing to minimize the costs associated with routing and the time required for the transaction to be completed. Ideally, it should calculate the optimal and most efficient route in a similar fashion to Google Maps, that tries to avoid traffic by proposing the shortest or less time-consuming path to reach your destination.

Lastly, as seen in paragraph 3.7, routing efficiency seems to have worsened from 2018.12.08 to 2022.02.07. We could ask ourselves the following question: how can we optimize and improve the routing efficiency?

Unfortunately, most of these proposals involve using advanced computational and storing resources and cannot be executed on a simple laptop.

Appendix

In paragraphs 3.4, 3.5 and 3.6, we focused on the concept of shortest path. "Recall that the shortest path between two nodes A and B is the path that has the minimum length among all possible paths between A and B"[44]. Calculating shortest paths and their length are not trivial. When computing the shortest path length in R, we obtain an (n x n) matrix where each cell assumes a value between zero and infinity. Two nodes have a distance equal to infinity when no shortest path links them. Assuming that the LN is an undirected graph, those nodes would belong to two different weakly connected components. The matrix containing the shortest path (length) values is symmetric, meaning that the shortest path (length) from node A to node B is equivalent to the shortest path (length) from node B to node A (in an undirected graph). Hence, if every node had only one shortest path to reach any other node in the network, the number of shortest paths would be equal to (n*n)/2. Let's make an example. In 2018.12.08, there are 1880 nodes, meaning that the network has at least (1880*1880)/2 shortest paths (including the ones having a length equal to infinity). Therefore, in 2018.12.08, the network has at least 1,767,200 shortest paths. Following the same reasoning, in 2022.02.07 there should exist at least (19,946*19,946)/2 = 198,921,458 shortest paths. Ergo, we can deduce that as the network becomes larger, the number of nodes and edges increases, making calculating the shortest paths much harder.

But, how many shortest paths does every network have? We can start by excluding the ones having a length equal to infinity, as they do not exist. We want to output the complete list of feasible shortest paths. As seen in paragraphs 3.5 and 3.6, every pair of nodes in an undirected network may share several shortest paths that involve different intermediary nodes. Hence, every graph has at least (n*n)/2 shortest paths, but usually many more. Various algorithms allow finding all the shortest paths in a graph, such as Dijkstra's algorithm, BFS and DFS[45]. In paragraph 3.5, we calculated all the possible shortest paths for the network in 2018.12.08. The output list

[44] https://www.baeldung.com/cs/graph-number-of-shortest-paths
[45] Look at the previous link.

contained 20,288,892 different shortest paths, much larger than the minimum requirement of 1,758,754 feasible shortest paths (excluding the ones having a length equal to infinity). From these computations, we can deduce that, on average, there exist approximately 11 shortest paths between every pair of nodes in the network. The analysis carried out in paragraph 3.5 relies on a much smaller graph than the one observed in 2022.02.07, where the number of nodes is 19,946, and the number of edges is 85,632. Calculating all the shortest paths of that graph would have required a much greater computing power than what my pc can handle.

Computing the betweenness centrality of all nodes in the network is not trivial also. As seen in paragraph 3.4, betweenness centrality is a node-level measure that quantifies the number of times a node lies on the shortest path between any pair of nodes in the graph. Ergo, as the network grows, the number of shortest paths increases, and, in turn, calculating the betweenness centrality becomes harder. It requires calculating the number of all shortest paths and the number of shortest paths each node lies on.

In paragraph 3.6, we started from n (19,946) zip files (1.3 GB) containing all the shortest paths each node can travel to reach all the other nodes in the network calculated in the same manner as seen in paragraph 3.5. One file for each node. Again, the number of shortest paths in the network is much greater than 198,921,458 (considering only one shortest path for each pair of nodes), as most pairs of nodes share more than one shortest path. For each node, we calculated the maximum number of shortest paths and the shortest path length to reach its most distant node. The output file contained 19,946 rows, one for each node.

Given these premises, it may be complicated for a Lightning network participant owning a pc with standard computing power to calculate the shortest path(s) to reach any other node in the network. Moreover, dropping the assumption under which the LN is undirected and considering the LN as bidirectional and weighted (fees), participants would need to use much stronger computational and storing resources.

References

[1] Sergio Luis Náñez Alonso et al. “Cryptocurrency Mining from an Economic and Environmental Perspective. Analysis of the Most and Least Sustainable Countries”, Catholic University of Avila, 2021.

[2] Stephen Diehl “Web3 is Bullshit”, https://www.stephendiehl.com/blog/web3-bullshit.html, 2021.

[3] Lewis Gudgeon et al. “SoK: Layer-Two Blockchain Protocols”, 2020.

[4] Liquid Federation “Six Differences Between Liquid and Lightning”, The Liquid Blog, https://blog.liquid.net/six-differences-between-liquid-and-lightning/, 2022.

[5] Joseph Poon et al. “The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments”, 2016.

[6] Ferenc Béres et al. “A Crypto-economic Traffic Analysis of Bitcoin’s Lightning Network”, Institute for Computer Science and Control (SZTAKI), 2020.

[7] Nikolaos Papadis “Blockchain-based Payment Channel Networks: Challenges and Recent Advances”, IEEE Access, 2020.

[8] Rene Pickhardt et al. “Security and Privacy of Lightning Network Payments with Uncertain Channel Balances”, 2021.

[9] Jake FrankenField “Public Key”, https://www.investopedia.com/terms/p/public-key.asp, Investopedia, 2021.

[10] Yuwei Guo “A Measurement Study of Bitcoin Lightning Network”, Beihang University, 2019.

[11] Jian-Hong Lin et. al “Lightning network: a second path towards centralisation of the Bitcoin economy”, New Journal of Physics, 2020.

[12] Linton C. Freeman “Centrality in Social Networks Conceptual Clarification”, Lehigh University, 1979.

[13] Roy Sheinfeld “Mitigating the Risk of Running Lightning Network Hubs”, Medium, 2018.

[14] Stefano Martinazzi et al. “The evolving topology of the Lightning Network: Centralization, efficiency, robustness, synchronization, and anonymity”, Politecnico di Milano, 2020.

[15] Mohammed Lalou et. al “The Critical Node Detection Problem in networks: A survey”, 2018.

[16] “Find Weakly Connected Components in a Directed Graph”, https://www.geeksforgeeks.org/find-weakly-connected-components-in-a-directed-graph/, 2021.

[17] Brian Nibley “A Guide to Sharding in Crypto”, SoFi Learn, https://www.sofi.com/learn/content/what-is-sharding/, 2022.

[18] Said Sryheni “Number of Shortest Paths in a Graph”, https://www.baeldung.com/cs/graph-number-of-shortest-paths, 2021.

[19] Carlo Cantamaglia “Sistemi Peer-To-Peer Analisi della Lightning Network”, https://mirkozichichi.me/assets/tutor/slides/repcantamaglia.pdf