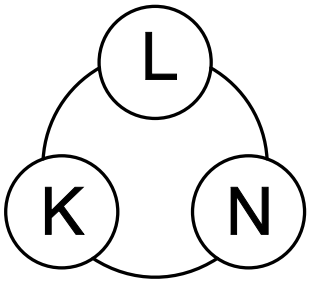



**Technische Universität München**
**Lehrstuhl für Kommunikationsnetze**
Prof. Dr.-Ing. Wolfgang Kellerer

# Forschungspraxis

Network Topology Analysis of the Lightning Network on the Bitcoin Mainnet


| | |
|---|---|
| Author: | Brüß, Claas |
| Address: | Heßstraße 77 |
| | 80797 München |
| | Germany |
| Matriculation Number: | 03610826 |
| Supervisor: | Marc Ablay & Prof. Dr.-Ing. Wolfgang Kellerer |
| Begin: | 12.11.2018 |
| End: | 31.03.2019 |

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

| München, 26.04.2020 | |
|---|---|
| Place, Date | Signature |



| München, 26.04.2020 | |
|---|---|
| Place, Date | Signature |

# Abstract

The Bitcoin Mainnet Lightning Network is a freely evolving payment network based on payment channels secured by the Bitcoin Blockchain and without a preset designed topology. To investigate the evolution of the network topology and to search for indicators of centralization the Network Size, Degree Distribution and the resulting Degree Exponent, Densification of the network as a whole as well as Clustering Coefficients and Betweenness Centrality for the 10 nodes with highest Degrees were measured over a 73 week period starting December 28th 2018. It was found that week 50 week marked a clear qualitative shift towards accelerated growth behavior and that the resulting densification was highly heterogeneous with rising Diameters and falling Average Shortest Path Lengths after week 50. Increasingly long tails of Degree Distributions and Degree Exponents below the value 2 categorize the network as anomalous. Highest degree nodes showed Clustering Coefficients falling below 0.1 while maintaining high Betweenness Centrality averaging over 0.06 in week 70 with multiple nodes surpassing values of 0.1 even after week 50. It is concluded that the network cannot be considered as decentralized since highest degree nodes are successfully maintaining high centrality even in stages of accelerated growth.

# Contents

# Chapter 1

# Introduction

With rise to fame of Bitcoin from 2009 on wards it has become clear that the blockchain infrastructure underpinning Bitcoin alone is not a viable solution for the realization of real time payment networks. It order to side step most of the issues arising from the Bitcoin blockchain architecture in the face of mass adoption for transaction processing of daily life purchases second layer architecture building on top of the Bitcoin blockchain have been proposed.

One of the most successful ones is called Lightning Networks. The goal generally is to preserve the security aspects of the blockchain but to enhance transaction speed and to drastically reduce transaction fees in order to make exchanging small amounts of funds economically viable for a main stream user. At the core of a lot of the development efforts and companies in this space is the believe that this should be achieved through an inherently trustless system with decentralized control rather than central authorities.

The Lightning Network is realized as a network of payment channels between nodes. This setup not only allows for transactions between users participating in the same payment channel, but for secure transactions between unconnected users through 3rd parties acting as intermediaries by routing payments through their channels in exchange for a small fee. The Lighting Network evolves freely without a predetermined topological design, giving way to complex behavior emergent in distributed systems.

To enable service availability in wide spread use Lightning Networks not only need be understood and verified from a perspective of cryptography but also from a perspective of network theory. This work seeks to take the first steps in this direction by evaluating some of the fundamental properties of the Bitcoin Mainnet Lightning Network topology over time and to search for behavior suggesting a push towards centralization within the network.

# Chapter 2

# Background

## 2.1 Bitcoin

When the Bitcoin white paper surfaced on Jan 3rd 2009[Nak08], it stated that it's aim was to provide a decentralized and verifiable digital cash equivalent, that does not require any trusted third parties in transactions and offers the possibility of very small micro transactions without overwhelming processing cost. In the following paragraphs I will outline why Bitcoin applies of the concept of a blockchain as in immutable ledger in a distributed ledger scheme with a consensus and inflation mechanism as well as the resulting limitations that motivate Lightning Networks.

### 2.1.1 Blockchain and Consensus

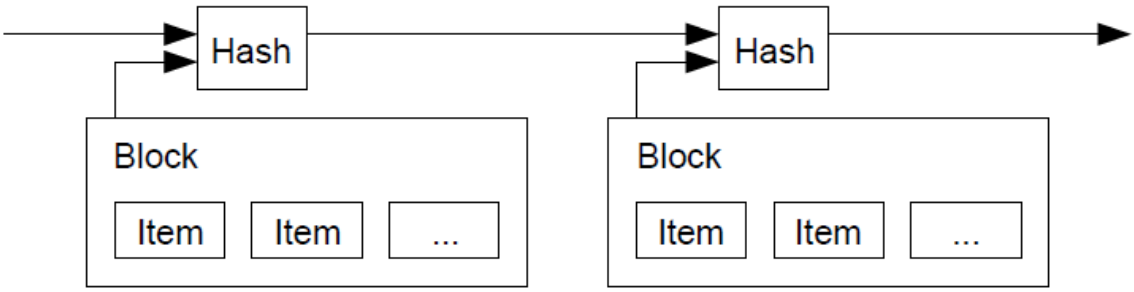



Figure 2.1: Blocks chained with hash pointers [Nak08]

Technically speaking a blockchain is a data format that could be described as linked list with hash pointers. The entries in the list are referred to as blocks. Hash pointers not only link a block to the previous block but also include a hash of the previous block. This

yields the interesting property that the data stored within a single block in the linked list cannot be changed alone, but all the following blocks up the list need to be changed as well in order to keep the integrity of the list, since each block contains the hash of the previous block. This logic interlinking of the blocks warrants the description of the list as a chain. In addition every block is timestamped which allows to proof the existence of specific data entries at a specified point in time.

In this case the blockchain is not kept and controlled by a superior entity, but by every participating entity in the system. To make any mediating entity obsolete a consensus mechanism is implemented to provide a global current state of the ledger throughout the system. The consensus is reached by a majority vote of the network participants confirming the integrity of the blockchain by validating the hash values in the block headers. The vote is expressed by the adoption of this new blockchain state by the majority of the network participants. Should there be two competing blockchains, preference is given to longer chain as it represents more data. Since Bitcoin is seeking to emulate currency of course the block entries cannot be allowed to be made arbitrarily as they reflect allocated funds that should not be double spent or be moved without explicit permission from their owners. For this the concepts of wallets and transactions validated through mining are introduced.

### 2.1.2 Mining

In order to introduce bitcoins into the system a mechanism of controlled and stable inflation is introduced. In this the newly to be minted bitcoins are given to the network participant who can solve a hash puzzle of a certain difficulty the fastest. The correctness of the solution is checked by all network participants and deemed correct through the adoption of the blockchain with the new entry stating that the winner has been rewarded with additional bitcoins. This process has been dubbed mining and the scheme itself proof-of-work, since the entity being rewarded with these new bitcoins had to invest computation power and therefore energy in order to compete in solving the puzzle.

The aforementioned hash puzzles are based on the fact that hash functions, in this case SHA-256, belong to the category of one way functions. Functions in this category cannot be reversed in the sense that the function inputs cannot be determined from specific outputs or even based on certain sought output attributes. This only leaves the computationally expense trial and error as the means of determining the inputs. In contrast, checking whether proposed inputs map to the desired outputs is trivial and inexpensive in computation. When a mining network node finds a solution to the puzzle and broadcasts it to the network, other nodes within the network can easily verify that the proposed solution actually yields a hash that fulfills the set requirements.

The difficulty of these trial and error based puzzles can be tuned through requirements set on the outputs of the hash functions. The more these requirements confine the targeted hash function's solution space the lower the chance to find the correct set of inputs that

yield a hash within that confined solution space. In Bitcoin the mining difficulty is tuned by requiring a set number of leading digits of the resulting hash to be zero. The difficulty is adjusted by taking into account how fast new valid blocks are being mined. The target inflation is specified at 2016 new blocks every two weeks or an average clearing time of 10 minutes per block. The speed at which new valid blocks can be found is clearly related to the overall computation power, also dubbed hash power, expended in the system. Every 2016 blocks the mining difficulty is adjusted to push the inflation closer to the target rate specified in the protocol.

Not all hash function inputs are iterated in the trial and error process. The only iterated input is called the nonce. Other inputs include the hash of previous block and multiple transactions. New blocks will only be accepted by other nodes in the network if the current hash puzzle is solved and all transactions included in the block are valid and unspent. The next section describes transactions and their validation in more detail.

### 2.1.3 Transactions and Validation

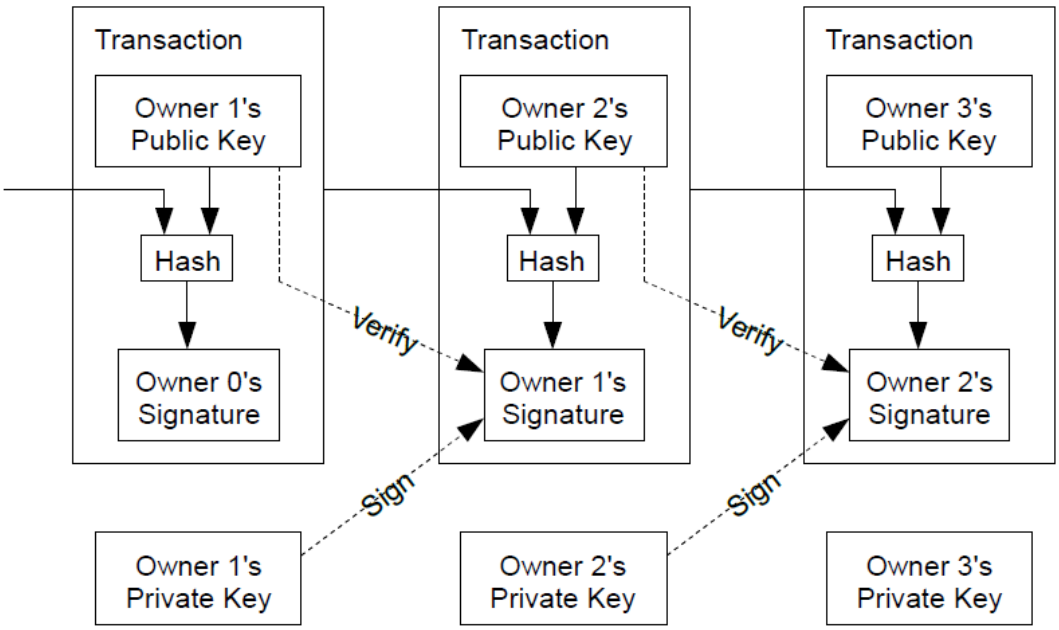



Figure 2.2: Transactions as chains of digital signatures[Nak08]

Funds in the Bitcoin system do not exist as a tally but rather as a chain of digital signatures. In the event of a transaction, the receiving party can verify that the sending party actually once controlled the specified funds. The signature the previous owner put on the transaction in question results from the encryption of the hash of the transaction

in question with the private key of the previous owner. The receiving party now verify the ownership by decrypting the signature with the previous owners public key and compare output with a hash of the transaction in question. If they match, the ownership is verified.

In order to prevent double spending with the network, each transaction is publicly announced and then as outlined in the previous section, timestamped, validated and included into a block by the miner that solved the hash puzzle for the current block. To prevent double spending of a transaction, membership proofs in Merkle trees are used to determine whether the inspected transaction has been used as an input in a previous verified block. These Merkle trees are constructed though pairwise hashing of the transaction ids of all transactions and their resulting hashes until a single hash value remains. This last singular value is called the Merkle root and is included in the block header of the block that includes all the transactions out which the Merkle root was calculated.

Miners have to validate each transaction they plan to include into the next block in order for it be accepted by the rest of the network. In return for this a miner that successfully mines the next block is not only rewarded with newly minted bitcoins, but is also granted a fee drawn from all transactions in the block. This fee is independent from the actual sum being transferred, but is not necessarily the same for all transactions. The fee is basically set by supply and demand and factors in how much of the block's limited data capacity the transaction requires. Entities seeking to increase the likelihood of their transaction being included into a block sooner rather than later can offer to pay a higher than usual fee to be more attractive for mines to include their transactions in the next block.

### 2.1.4 Limiting factors and issues in application

Due the block size being capped at a maximum of currently 1 *MB*, the number of transactions that can be validated per block is limited to approximate 4000. With an average block clearing time of 10 *minutes* this leads to a maximum of approximately 6.5 *transactions / second*. This is far too low to allow for high network liquidity in the case of wide spread adoption. Various other crypto-currencies such as Bitcoin Cash were established as hard forks of Bitcoin with larger block sizes and higher throughput. None have found wide spread adoption and benefits as well as risks of increased block sizes are very controversial to this day.

Besides this, 6 Blocks are by many deemed the necessary time to ensure that a transaction is genuine and that no longer valid blockchain will emerge to be adopted by the majority of the network effectively rolling back transactions that are not included in the dominant blockchain version.

Finally, as mentioned above, the transaction fees are independent of the amount of funds being transferred. As miners will prefer transactions from entities willing to pay higher than average fees, the costs associated with transactions are bound to rise in the scenarios

of wider adoption being faced with the limited transaction rate capacity of the Bitcoin network and rising Bitcoin to Dollar exchange rates.

The culmination of these aspects render transactions of small amounts of funds and therefore payment in everyday life situations, like buying a cup of coffee, economically not viable. This is especially the case when compared to other non crypto-currency based payment solutions that a sizable part of the worldwide crypto-currency community would like to see Bitcoin competing with.

## 2.2 The Lightning Network

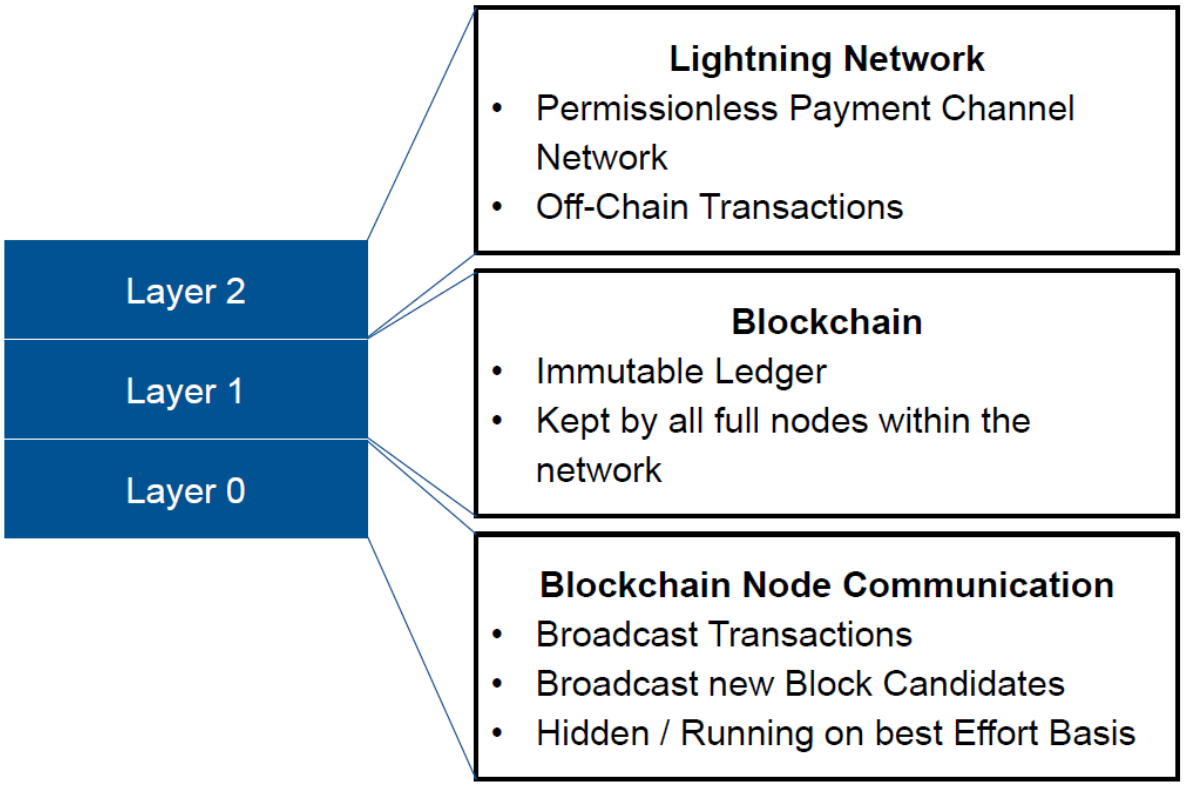



Figure 2.3: blockchain [BOLT - The Protocol Layermodel . Layer 0 Bitcoin Node Network (hidden) - Layer 1 Blockchain - Layer 2 Lightning Network

The shortcomings described in the previous section motivated the design and implementation of a secondary payment network layer. This new layer on top of the Bitcoin blockchain (Layer 1) and the Bitcoin Node Communication Layer (Layer 0) aims to provide a framework for off chain transactions in order to alleviate major bottlenecks in network liquidity and to drive wider real world adoption without giving up the security of the blockchain in

settling the reallocation of larger funds. In early 2015 Joseph Poon and Thaddeus Dryja published the first draft of their white paper[PD16] describing such a secondary layer with an system architecture based on a network of payment channels, which will be discussed in the following sections.

### 2.2.1 Payment Channels

In order to make secure off chain transactions possible Lightning Networks apply the concept of trust-less bidirectional payment channels. These channels are manifested as a 2 of 2 multi signature address on the blockchain under which both parties have allocated and locked funds. That means both parties have to sign any payments coming out of this address. To establish the payment channel an initial transaction of both parties to the multisig address is needed.

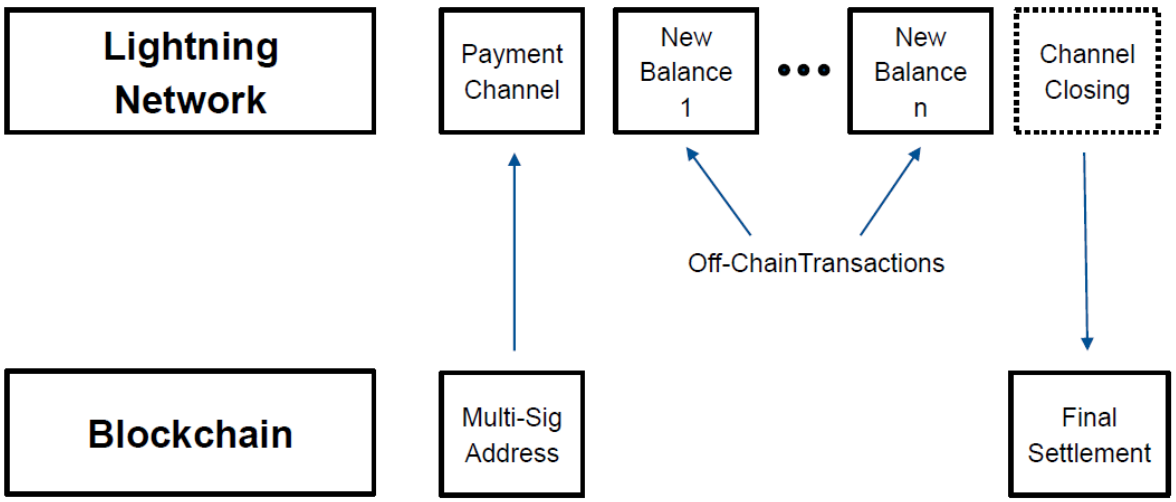



Figure 2.4: blockchain

Transactions within these payment channels can be seen as the two parties redistributing their partial claim to the collective funds locked inside the corresponding multi signature address. These changes are not directly recorded on the blockchain itself. Only with the closure of the channel the final state of the payment channel is manifested on the blockchain as an on-chain payment going through Level 1. This means only the participating parties are aware of the distribution of funds in the payment channel.

Transactions on the payment channel are recorded by both parties locally. Before each transaction both parties create a new secret to be stored locally and exchange the hash values of that secret. This means that each party can check the authenticity by hashing the secret of the counter party in case it is revealed. These secrets are only revealed to the counter party in the process of updating the distribution of funds in the payment channel.

The new state of the payment channel is represented by a pair of partially signed on-chain transactions that are exchanged between evolved parties but aren't being broadcast unless one party decides to close the channel and initiate the settling on-chain transaction.

The mentioned pairs of partially signed transactions are following a specific setup. Each party designs a transaction such that it own stake is transferred to itself and the counter party's stake is transferred to a new transactional multisig address. This multisig address provides two conditions under which funds can be spend from it. The first allows the counter party to spent from it 1000 blocks after the counter party broadcast this transaction on the blockchain. The second allows the party designing this transaction to immediately spend from the multisig address given it provides the counter party secret.

Since the secrets associated with last transaction are exchanged before a new channel state is established, this setup ensures that a time window for intervention of 1000 blocks is available to a party should the counter party attempt to broadcast an transaction on the blockchain that represented a former channel state. This type of locking funds for specified number of blocks relative to the time of issue is called Check Sequence Verify or CSV. Another type of time lock is called Check Lock Time Verify or CLTV and rather specifies a dated specific point in time up until which the funds remain locked. This specific point in time can be a specific block height in the blockchain or actual date and time.

This intervention mechanism allows the intervening party to claim all funds in the channel, since it can claim the funds attributed to the transactional multisig address while the rest of funds already is attributed to itself immediately through the transaction itself.

### 2.2.2 A Network of Payment Channels

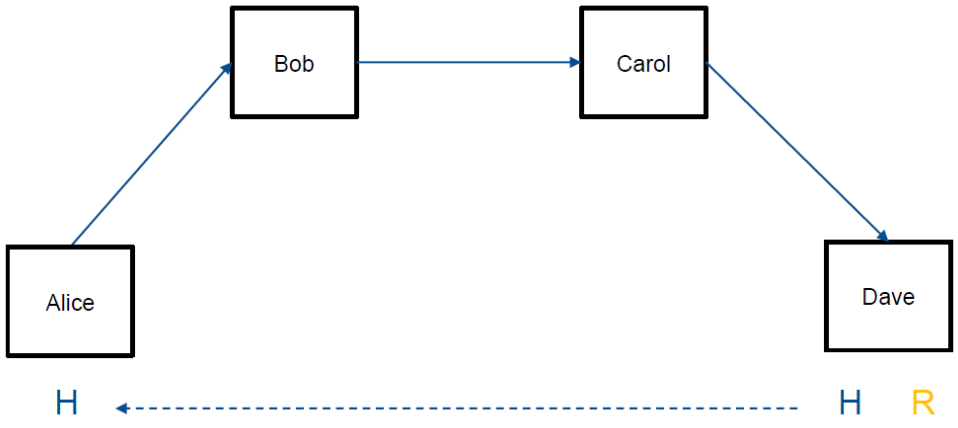



Figure 2.5: Setup of HTLCs with two intermediary parties

In the previous section we discussed payment channels and how funds are handled within

them. Based on this we discuss how these payment channels can be connected and turned into a network that allows for transfer of funds between two parties that are not participating in the same channel. Basically this can be achieved by finding a path of payment channels between the two primary and asking all third parties involved in these channels to move funds for them.

For this we consider the setup pictured above. Alice is looking to transfer funds to Dave but does not currently maintain a payment channel including Dave. The approach taken in Lightning Networks involves finding a path of payment channels between Alice and Dave, in this case through Bob and Carol. Two aspects need to be addressed for such a setup to be reliable, the transaction must save everybody involved without the need to trust any other party participating in this transaction. In order to incentivise users aka network nodes to support such transactions it has to be beneficial to all parties involved.

For this scheme to work it is paramount to ensure that none of the third parties simply keeps the incoming funds without passing anything along. To ensure this the concept of Hash Time Lock Contracts or HTLCs is introduced. To establish these contracts Alice tells Dave that she would like to transfer funds to him and asks him to generate a secret and to share the hash of this secret with her. This can secret - hash pair plays an instrumental role in setting up the HTLCs.

A HTLC is setup for each payment channel included in the payment path. The HTLC is based on a multisig address in which the sending party deposits the funds to be transferred. These addresses allow the funds to be spent on two conditions. Either the party looking to receive funds presents their signature as well as the secret value generated by Dave or the party look to send funds presents their signature and after a set CLTV is lifted. This limits the time that funds provided by the sending party can be stuck in escrow by putting pressure on the retrieval of the funds by the receiving party.

Thus for this setup of HTLCs to function properly as a cascade of transactions strating from Alice and finally reaching Dave with all payments being guaranteed the CLTVs need to be configured right, with the CLTV in the HTLC between Carol and Dave lifting first, between Bob and Carol second and between Bob and Alice last. CSVs are deliberately not applied in this context. Due to the HTLCs being created at slightly different times the use of CSV instead of CLTVs would provide ambiguity and therefore additional undesirable failure modes.

Secondly, nodes in the network need to be incentivized to participate in such transactions to provide enough available payments paths within the network. This is implemented by a paying a transaction fee to all intermediary parties in the transactions. These fees are not set in absolute but loosely proportional to the amount of funds being sent and usually lie in ranges double digit counts of Satoshi (10^8 Bitcoins).

## 2.3 Network Characteristics

The previous sections demonstrated how payment channels can be setup and how they function. We then discussed how these payment channels can be strung together into paths for funds to be sent on through third parties. As the name Lightning Network already suggests this now constitutes a network with parties as nodes and payment channels as edges. More precisely this can be abstracted as a temporal and topologically dynamic network, since channels can be established or dissolved at any time and with payment channel balances changing almost constantly.

The complete topology of the network can be seen as a graph of including all possible payment paths and therefore as the payment infrastructure. Since the system does not imply any central authority or dynamic regulation the current topology as well as dynamic behavior such as attachment behavior of new nodes are indicators for the level of functionality of the Lightning Network.

Security from a cryptographic standpoint is provided through the underlying blockchain and the second layer schemes of hashed secrets as well as multisig addresses with various forms of time locks. The clear reliance of the payment network on the network topology as explicit infrastructure makes it necessary to regard a wholly different set of failure modes, born out of the the fact that the topology is dynamic and the control of the network distributed.

Measures against these potential failures need to be developed in order to ensure continued reliable operational functionality. Only if these can be addressed appropriately, the wide spread use of Lightning Networks, which many of the protocol creators are seeking, will take place and will thus reach main stream users.

If actors controlling network nodes are interested in the participation of the wider network, they are incentivized to announce any new payment channels they establish. The information is usually broadcast to all parties that are sharing a payment channel with the node. After a short holding time these parties then re-transmit this information to nodes they are sharing payments channels with. This mode of information spreading is deemed the Gossip-Protocol and affords nodes a fairly update image of the existing payment infrastructure. It is important to note that updates in the funds distribution between parties involved in a channel are typically not communicated.

Thus it is possible to measure and observe the current network topology in terms of static network topology measures. Dynamics of the network topology are hard to model explicitly due to this missing information and are therefore sampled over time. The growth rates of the network are likely to be driven by external factors in economy and the alternative solutions to the Lightning Network that are available at that point in time. Services offering new ways to join the network through custodian exchanges for example are likely to have an impact on user behavior in the network as well. Nonetheless it is likely that the majority of events triggering channel closures are motivated by the distribution of

funds within payment channels. As mentioned before generally payment channel funding balance information is not openly available in the network making it hard to predict channel closures. Therefore it is hard to model dynamics explicitly or with high resolution.

Algorithmic and machine learning based approaches to modelling dynamic network topology through Node Embeddings [BKPB19] remain a current topic in research.

Finally the aspect of centralization within the payment infrastructure is an important topic to large parts of the community. Starting from the white paper proposing Lightning Networks [PD16] the vision was to create and foster a decentralized payment network without the need for centralized control or oversight. But given the fact that each actor participating in the network can freely choose which payment channels to maintain and for which payments to act as an intermediary in payment routing, it cannot be guaranteed a priori that the network will stay decentralized. Clustering Coefficient as well as Betweeness Centrality of high degree nodes will be measured to investigate whether any centralizing behavior is prevalent in the network topology.

Snapshots of the Bitcoin Lightning Network at singular points in time have been analysed in the past [SGNB19]. Based on this work similar measurements have been conducted continuously over the time span of this data set to gain insights into the topological evolution of the Bitcoin Mainnet Lightning Network.

Here the network is modelled as an unweighted undirected multi graph. Besides trivial measures such as the Number of Nodes $N$, Number of Edges $L$, unweighted Average Node Degree $\langle k \rangle$ and Maximum Node Degree $k_{max}$ in the network various other coefficients are calculated.

### 2.3.1 Degree Exponent $\gamma$ of Degree Distributions and Network Topology Regimes

As Barabási suggests in seminal book "Network Science" [Bar16] networks can be classified into different network regimes based on the parameter $\gamma$ . This parameter is calculated based on the node degree distribution. This distribution is fitted with a power law distribution which expresses itself as a line on a log-log scale. The slope of this line is called the Degree Exponent $\gamma$.

Classifying networks into such regimes allows for the comparison of networks and determines how the network behavior can be understood since most network models are only precise within one of the regimes.

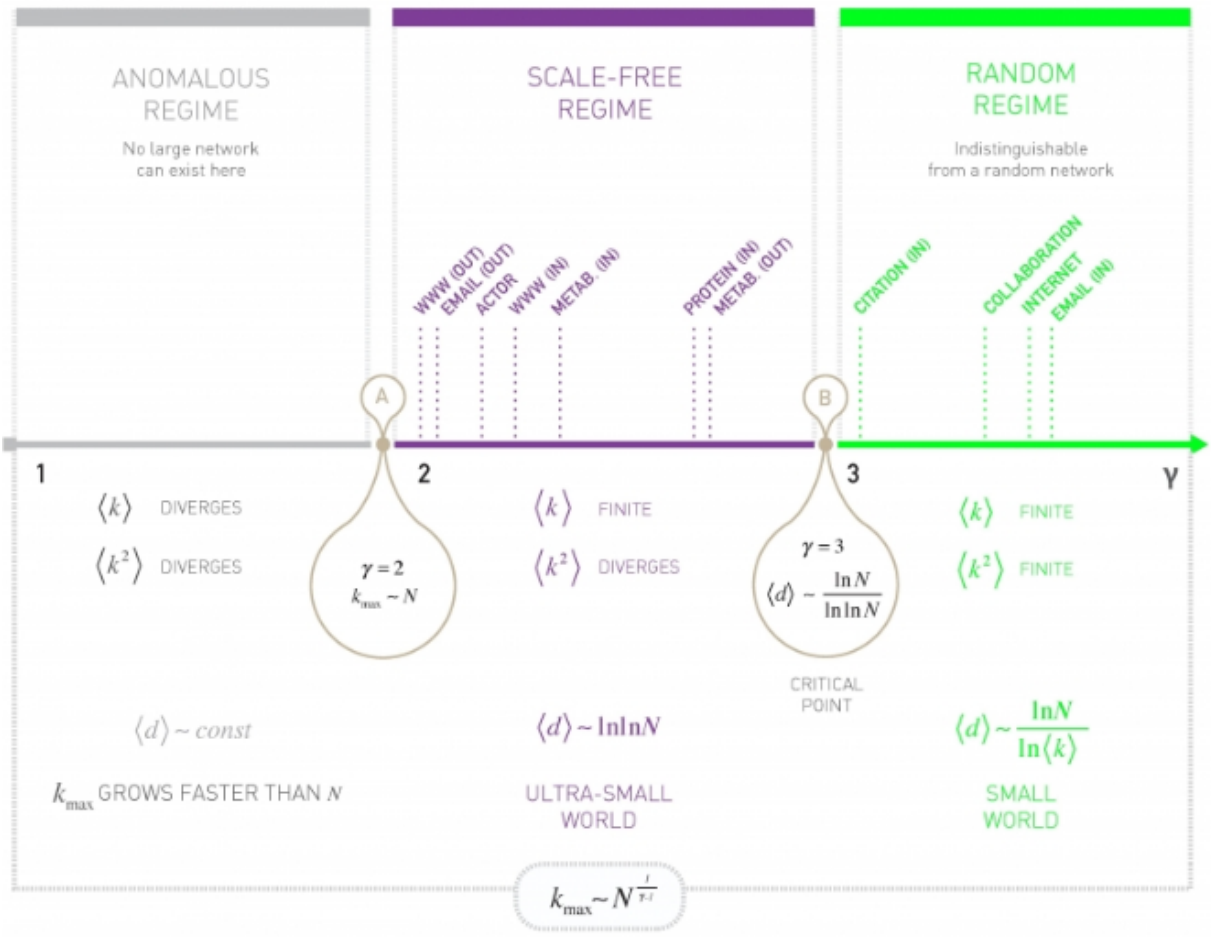



Figure 2.6: Network Regimes determined by $\gamma$ [Bar16]

### 2.3.2 Diameter and Average Shortest Paths

Average Shortest Path $a$ is calculated through:

$$a = \sum_{s,t \in V} \frac{d(s,t)}{N(N-1)} \tag{2.1}$$

$V$ is the set of nodes, in our case the whole network. $(s,t)$ is a pair of nodes in $V$ with $d(s,t)$ being the shortest path between $s$ and $t$.

With the Diameter $D$ of the network being:

$$D = \max d(s,t) \tag{2.2}$$

These two measures will provide a general indication whether the network is densifying over time.

### 2.3.3 Clustering Coefficient

The clustering coefficient is a measure of the connectivity between neighbours of a particular node. The clustering coefficient $C_i$ of the $i$-th node in the network with a degree of $k_i$ is the quotient of the number of realized edges $L_i$ between neighbours of the $i$-th node of the node and the number of all possible edges in the neighbourhood of the $i$-th node.

$$C_i = \frac{2L_i}{k_i\,(k_i - 1)} \tag{2.3}$$

Measuring this parameter throughout the network should give us a good indication, whether nodes with high degrees incentivize or deter their neighbours from setting up payment channels between them or to instead conduct transactions through them.

### 2.3.4 Betweenness Centrality

For open distributed systems without an explicitly designed network topology the centrality of nodes is highly important since nodes might become choking points by becoming hubs. Betweenness Centrality of a node measures the portion of shortest paths between all node pairs in the network which include the investigated node in relation to all shortest paths in the network .

In this context paths are payment routes. Betweenness Centrality is used to understand the impact of hubs on payment routing throughout the system, since users are economically incentivized to send funds through a minimal number of intermediary nodes ergo the shortest paths.

Should a small group of nodes process a sizable portion all payment routing, the network would be reliant on these hubs. This reliance on singular entities in the system goes against the idea of a decentralized payment network, which is a core motivation for a large portion of cryptocurrency advocates.

Betweenness Centrality $c_B(v)$ of node $v$ is calculated as follows:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{2.4}$$

$V$ is the set of nodes, in our case the whole network. $(s,t)$ is a pair of nodes in $V$ with $\sigma(s,t)$ being the number of shortest $(s,t)$-paths. With $v$ representing the node under investigation, $\sigma(s,t|v)$ is the number of those paths passing through $v$.
If $s = t \rightarrow \sigma(s,t) = 1$. If $v \in s$, $t \rightarrow \sigma(s,t|v) = 0$.

## 2.4 Tools and Implementation

The project initially aimed to build a new full stack from the ground up consisting of multiple Bitcoin full nodes and Lightning Network nodes. To orchestrate this better the nodes as well as the node clients were containerized through Docker, built through GitLab builders to minimize the spread of secrets and run on local servers of the company Blockchain Consulting GmbH, which hosted the project until end of March 2019.

Due to internal restructuring of the company the initial tech stack was scrapped and the node stack required for data acquisition was moved to a RaspberryPi 3B+ computer running the openly available RaspiBlitz [Rot13] implementation of a full Bitcoin and Lightning node with SSH capabilities. This setup was used for initial testing. Due to time constraints and the lack of backlog the approach was abandoned and replaced by data from a preexisting dataset.

The data processing was implemented in Python 3.6 with the packages SciPy[VGO+20], NumPy [Num] and matplotlib[CDL+20]. More specialized packages included NetworkX[Net] to calculate all network topology related me metrics and powerlaw[ABP14] to fit distributions and estimate the $\gamma$ values.

Please resort to Appendix A for the further detail on the code base and the project repository.

## 2.5 Dataset

The dataset used in this analysis here spans the time of 28th December 2017 until 15th May 2019 and covers the blocks on the Bitcoin blockchain between the block height of 501337 and block height of 576140. This dataset is taking into account every new block on the Bitcoin blockchain on which channels openings and channel closures are announced. It was sub sampled with a frequency of 1008 blocks on the Bitcoin Blockchain which equates approximately one week in real time. Some measurement required further sub sampling with a factor of 10 due the limited computation power of the personal computer used for this project.

The dataset covers all visible channel openings and closures on the Bitcoin Mainnet Lightning Network in this block height range. This allows us to observe and study all changes to the network topology. It also allows to study the statistics of channel lifetimes in future work.

Besides the topological information the dataset also state the initial funding level of the channel. It is important to note that it does not include the changes in channel

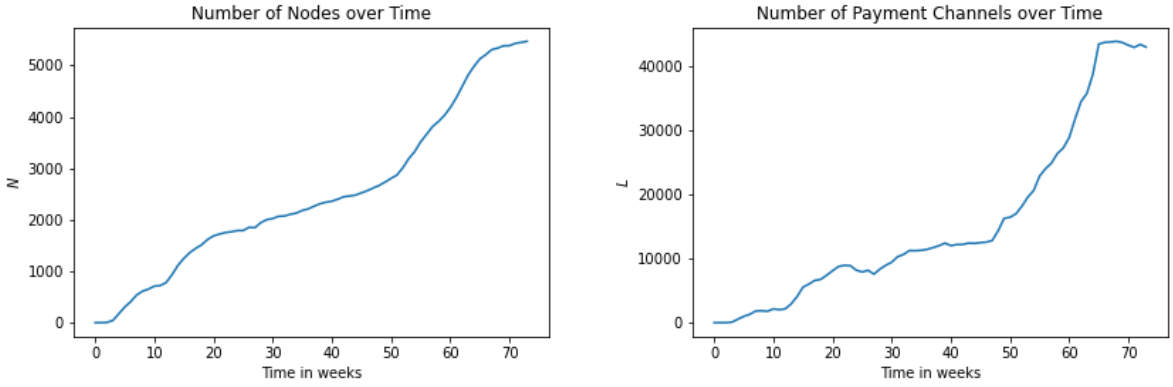
balances.

The dataset was kindly provided by István András Seres from ELTE in Budapest and his collaborators, who supported me after I had to move the project outside the company and was forced to rebuild the data pipeline.

# Chapter 3

# Results

## 3.1 Size of the Network

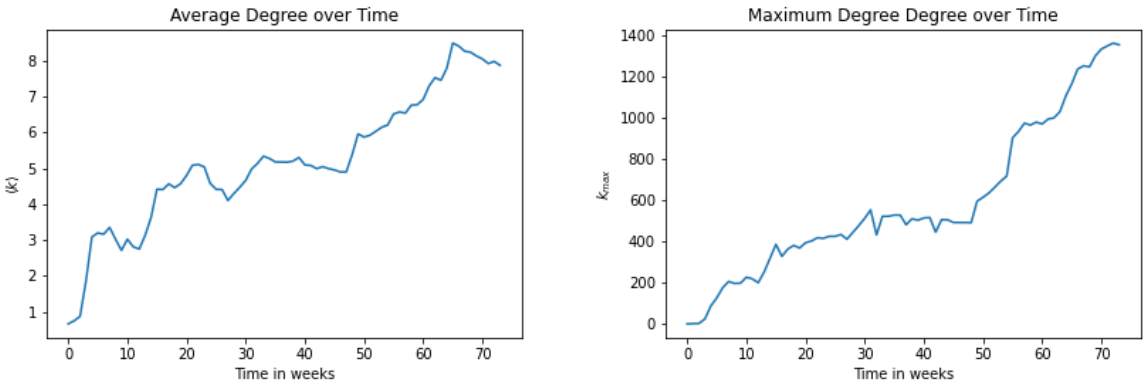



(a) (b)

Figure 3.1: (a) Number of active Nodes and (b) open Payment Channels in the Network

The growth of the network is evident with a shift in growth rate for the number of nodes as well as the number of payment channels around week 50. This roughly coincides with the release of Eclair and Eclair Wallet app which at that time offered a faster and simpler way to join the Lightning Network than the solutions offered previously.

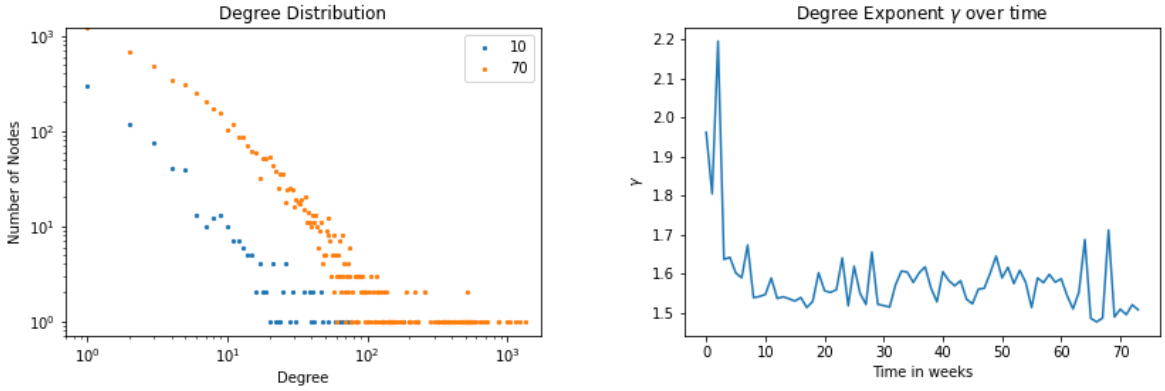
Due to the rising average and maximum node degrees the network seems to be densifying. The maximum degree in the network is growing at higher rate after week 50 compared to the average degree, which is growing far more steadily. This suggests that a select few nodes are opening large number of payment channels and are potentially becoming hubs.




(a) (b)

Figure 3.2: (a) Average and (b) Maximum Node Degree in the Network over Time

## 3.2 Degree Distributions and Degree Exponent $\gamma$




(a) (b)

Figure 3.3: (a) Degree Distributions for different weeks and (b) $\gamma$ over Time

We can confirm this through the observation of the degree distributions developing longer tails over time as evident in figure 3.3a. A small set of nodes with the highest degrees seems to be seeking to maximize their number of payment channels. In the most extreme cases the entities in control of this high degree group seem to be trying to reach the high percentage of nodes in the network with the number of payment channels having the same order of magnitude as the number of active nodes in the network. Even though it is technically possible for two parties to maintain multiple payment channels between them, this seems unlikely to strongly contribute to the count of open payment channels in the high degree nodes since it offers no technical or security advantages beside partial settling.

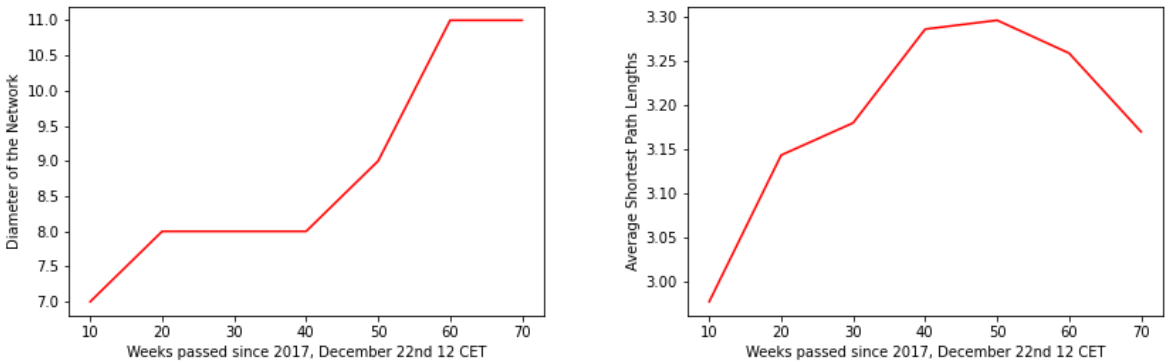
As illustrated by 3.3b $\gamma$ rapidly drops just under the value of 2 within the first 10 weeks.

From this point in time onward we observe fluctuation between two seemingly discrete values. After week 25 these fluctuation become more episodic. Even though further more detailed investigation would be needed to identify the cause of these fluctuations, it is likely that larger hubs or groups of hubs under control of a single entity are periodically going through cycles of closing and reopening payment channels at large scale to secure funds on the blockchain. Since $\gamma$ is calculated as the slope of a linear fit of the degree distribution the powerlaw package[ABP14] might contribute to the characteristic by shifting the fit due to shifts in the increasingly long tail of the degree distributions.

In reference to figure 2.6 showing the classification of networks into regimes based on the $\gamma$ parameter, the Bitcoin Mainnet Lightning Network is just outside the regime of scale free networks. It is not surprising that this network does not reside in the regime of random networks simply due to the fact that funds need to be kept in escrow to fund a payment channel and that the opening of payment channels is strongly motivated by the intend of repeated transactions between the participating parties. On the other hand it is surprising that the network cannot be clearly classified as Scale-Free. As figure 2.6 suggests this network cannot grow in this fashion arbitrarily.

For $\gamma$ to remain below the threshold for Scale-Free Networks, Barabási suggests that $k_{max}$ should be growing faster than $N$ which is consistent with our observations in 3.1a and 3.1b. Again it is important to note that opening payment channel requires the allocation of resources which limits the capability of entities to participate in an ever rising number of payment channels. Should future growth of the network be driven through services acting as a custodian to funds this will likely continue, since they are not limited through their own funds but allocate the funds of the funds of their users on their behalf instead.

## 3.3 Densification of the Network




(a) (b)

Figure 3.4: (a)Network Diameter over Time (b) Average Shortest Path over Time

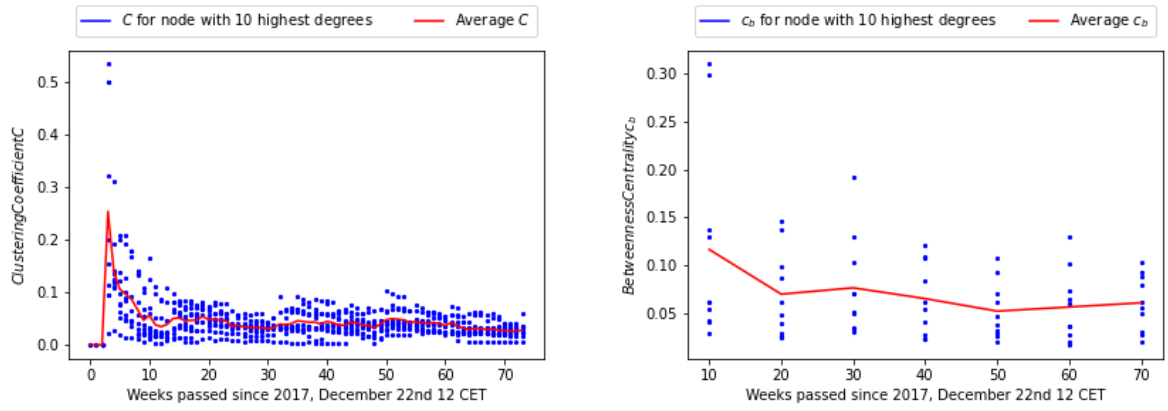
To judge whether the network is densifying the Diameter $D$ of the network and Average Shortest Path $a$ are measured over time. In this all connected component with less 100 active nodes are disregarded. In the first weeks connected component with more that 100 active nodes existed and thus all data week 10 omitted for this measurement.

Similar to 3.1a and 3.1b the diameter of the network is rising overall with the strongest growth around week 50 and a plateauing progression after week 60.

In contrast to this the Average Shortest Path Length within the network peaks around week 50 at $a \approx 3.295$ and then fall in the proceeding 20 weeks which is consistent with the indications for a stronger centralization in the network as seen in the comparison of 3.2a and 3.2b.

This marks a clear qualitative change of the densification of the network. The Diameter growing fastest in the time when the Average Shortest Path Length is beginning to decline suggests that the densification of the network is not uniform throughout the network but heterogeneous and focused. This is further supported by the fat tailed degree distributions seen in 3.3a and thus $\gamma$ falling below the value 2.

## 3.4 Clustering Coefficients and Betweenness Centrality




(a) (b)

Figure 3.5: (a)Clustering Coefficient $C$ and (b)Betweenness Centrality $c_B$ of the top 10 highest degree nodes in blue - (a)Average $C$ and (b)Average $c_B$ of the top 10 highest degree nodes in red

To further investigate this densification the 10 nodes with the highest degrees in the network were selected for each time point in time represented in the data. Clustering Coefficients as well as Betweeness Centrality were measured for the selected nodes and then averaged.

The spread of Clustering Coefficients is quickly declining in the first 20 weeks of measurements with values remaining below 0.11 after week 20. Given the fact that some of these nodes have degrees of over 1000 after week 60 motivates the measurement of Betweenness Centrality to gain insights into how impactful these nodes are for payment routing in the network.

The average of the Betweenness Centrality is falling approximately 50 percent over the course of the 60 weeks between week 10 and week 70 with the Number of Nodes 3.1a increasing with a factor of larger than 5 it is clear that the highest degree nodes within the network are committed to maintaining a high centrality within the network. From this measurement alone it is unclear how strong the clustering of high degree nodes is and therefore how cumulative the centrality of these nodes is.

With some nodes reaching Betweenness Centralities of $c_b > 0.1$ this network cannot be considered as fully decentralized. The mentioned steep rise in Maximum Degree and high Betweenness Centrality values suggest a clear commitment of the entities controlling nodes with high degrees to continue investment in the Bitcoin Mainnet Lightning Network to retain a central position within the network infrastructure.

# Chapter 4

# Conclusions and Outlook

This project sought to measure the topology of the Bitcoin Mainnet Lightning Network over time to investigate the evolution of the network and to search for indicators for centralization in the network.

After multiple iterations of data acquisition pipelines the approach of independent data collection was abandoned in the interest of time and due to the lack of a backlog with sufficient length to investigate the network evolution over a meaningful time span. The dataset was kindly provided by supporters from ELTE Budapest and included channel openings, closures and initial funding level.

Based on previous work lead by Isteván Seres from ELTE[SGNB19] which looked at a snapshot of the network on January 3rd 2019 and evaluated its topology as well as resistance to attacks, the measurements were extended over 73 weeks starting on December 28th 2018 in order to investigate the evolution of the network.

The Bitcoin Mainnet Lightning Network showed accelerated growth after week 50 with Maximum Degree strongly diverging from the Average Degree. Degree Distributions developed longer tails over time with Degree Exponent $\gamma$ falling below the value 2 before week 10 quickly moving the network out of the Scale-Free into the Anomalous regime. To investigate the densification of the network both Diameter $D$ and Average Shortest Path $a$ where measured between week 10 and week 70. With the rise of $D$ accelerating after week 50, $a$ is peaking at week 50 moving into decline from this point on. The further investigate this clearly heterogeneous densification of the network Clustering Coefficients $C$ and Betweenness Centrality $c_b$ are measured over time for the top 10 highest degree nodes in the network. With $C$ falling below 0.11 for all nodes after week 20 and some nodes continuously reaching $c_b$ values above 0.1 the network cannot be considered as decentralized, since there are clear indicators that the entities behind the highest degree nodes are seeking to maintain high centrality within the growing network.

Further investigation is needed to sharpen the understanding of the network dynamics at play in Lightning Networks. The immediate next step would be to dissect whether highly clustered communities of high degree nodes provide critical infrastructure for this network. For this the evolution of the percolation resistance under scenarios such as an attack on nodes withe highest Degrees or highest Betweenness Centrality could give critical insights into how users need to be incentivized to set up payment channels that contributed to stabilizing decentralization similar to Seres work in [SGNB19] but conducted over long time spans. Finally the explicit modeling of dynamic payment networks through the mentioned work by Béres [BKPB19] could inform new design guidelines for future payment networks.

# List of Figures

# Appendix A

# Code Base

The complete code used for data analysis can be found under this repository:

`https://github.com/T4B4/LKN_research_intership_report.git`

Intermediary results were ever possible have already been included in the repository as .json files. Further instructions can be found in the NetworkAnalysis.py Python file as comments.

Raw data can be shared upon request. For requests please reach out claas.bruess@tum.de

The folder containing the Raw data needs to be called (LNdata) and should be placed on the same folder hierarchy level as the folder containing the NetworkAnalysis.py file .

# Appendix B

# Symbols

| | |
|---|---|
| $c_B(v)$ | Betweenness Centrality of node $v$ |
| $N$ | Number of Nodes |
| $L$ | Number of Edges |
| $k$ | Node Degree |
| $\langle k \rangle$ | Average Node Degree |
| $k_{max}$ | Maximum Node Degree |
| $\gamma$ | Degree Exponent |
| $a$ | Average Shortest Path |
| $d(s,t)$ | Distance between Nodes $s$ and $t$ |
| $V$ | Set of nodes |
| $D$ | Diameter |
| $C$ | Clustering Coefficient |
| $\sigma$ | Number of Shortest Paths |

# Appendix C

# Abbreviations

| | |
|---|---|
| CLTV | Check Lock Time Verify |
| CSV | Check Sequence Verify |
| HTLC | Hash Time Lock Contract |
| SSH | Secure Shell |

# Bibliography


[ABP14] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. Powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS One*, 9(1):1–18, 2014.

[Bar16] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.

[BKPB19] Ferenc Béres, Domokos M. Kelen, Róbert Pálovics, and András A. Benczúr. Node embeddings in dynamic graphs. *Appl. Netw. Sci.*, 4(1), 2019.

[CDL+20] Thomas A Caswell, Michael Droettboom, Antony Lee, John Hunter, Eric Firing, David Stansby, Jody Klymak, Tim Hoffmann, Elliott Sales de Andrade, Nelle Varoquaux, Jens Hedegaard Nielsen, Benjamin Root, Phil Elson, Ryan May, Darren Dale, Jae-Joon Lee, Jouni K. Seppänen, Damon McDougall, Andrew Straw, Paul Hobson, Christoph Gohlke, Tony S Yu, Eric Ma, Adrien F. Vincent, Steven Silvester, Charlie Moad, Nikita Kniazev, Paul Ivanov, Elan Ernest, and Jan Katins. matplotlib/matplotlib: Rel: v3.2.1, March 2020.

[Nak08] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *www.Bitcoin.Org*, page 9, 2008.

[Net] NetworkX developers. Networkx.

[Num] NumPy Developers. Numpy.

[PD16] Joseph Poon and Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. *Tech. Rep.*, page 59, 2016.

[Rot13] Christian Rotzoll. Raspiblitz. `https://github.com/rootzoll/raspiblitz`, 2013.

[SGNB19] István András Seres, László Gulyás, Dániel A. Nagy, and Péter Burcsi. Topological Analysis of Bitcoin's Lightning Network. pages 1–12, 2019.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore,

Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.