# Understanding the Lightning Network capability to route payments

Daniel Šatcs
University of Twente
PO Box 217, 7500 AE Enschede
The Netherlands
d.satcs@student.utwente.nl

## ABSTRACT

The Bitcoin adoption as a digital currency had been rising and reached its peak in 2017 when the network was processing half a million transactions daily. Such unprecedented load uncovered flaws in the way Bitcoin scales and a new solution was necessary. Bitcoin Lightning Network was proposed in 2016 and is aimed at fixing Bitcoin scalability problems by implementing second-layer payment channels on top of the blockchain. To avoid creating a channel between each pair of nodes, a routing algorithm had been introduced that could send payments via multiple hops to the destination. In this paper, experiments are carried out to understand how well the Lightning Network can route payments 2 years after its launch. Unlike other works, active probing is used, which reflects the real state of the network better in contrast to an analysis of static network snapshots. An experiment is performed by sending test payments of various volumes to every node in the network. Errors are analysed and separated into connectivity related and channel capacity related. Both Tor and IP based Lightning nodes are used as entry points and outcomes are compared. The results help to understand how well the nodes are connected and how likely the payments of various volumes are to succeed.

## Keywords

Bitcoin, Cryptocurrency, Payment, Lightning Network, Routing, Tor.

## 1. INTRODUCTION

The theoretical speed limit at which Bitcoin can process transactions is close to 7 transactions per second and is much lower than that of the conventional payment systems such as Visa [24]. There had been numerous proposed solutions to that problem in Bitcoin and other cryptocurrencies, some more scalable than the other. One of the most promising solutions are ones based on *Payment Channel Networks (PCNs)*.

The *Lightning Network (LN)* is a network from the class of PCNs that is aimed at resolving the scalability problem of Bitcoin. It introduces a notion of a peer-to-peer payment channel where parties can exchange a large number

*33$^{th}$ Twente Student Conference on IT* Febr. 2$^{nd}$, 2020, Enschede, The Netherlands.

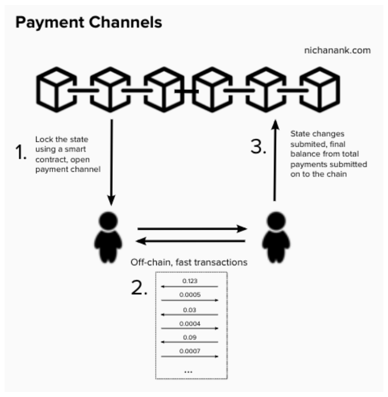
Payment Channels
nichanank.com
1.
Lock the state using a smart contract, open payment channel
3.
State changes submited, final balance from total payments submitted on to the chain
Off-chain, fast transactions
2.
0.123
0.0005
0.03
0.0004
0.09
0.0007
...

**Figure 1. Visualization of mircopayment channels as a second layer above the blockchain. Image from a third party blogpost [2].**

of payments without committing those to the blockchain. Only once all payments within the channel are completed, parties close the channel and a summarizing transaction is sent to the blockchain *(Figure 1)*.

If there is no channel open directly between two parties, in-between channels can be chained together to form a route along which payments are going to be sent.

The Lightning Network is important because it attempts to resolve many issues associated with on-chain Bitcoin payments, such as speed, fee size and privacy. It is the first and most well-known network of the PCN kind that is implemented and operating as of June 2020. A lot of people are interested in testing or using the LN but, perhaps because of its novelty, analysis has been mostly published in blog posts [4, 14]. In this paper, we measure the performance of the LN applying rigorous control to the experiments. Although there are problems with its current design (see below), it's a promising network that has the potential to revolutionize how we pay in day to day life.

Since the launch of the Lightning Network in 2018, previous research had largely focused on analysing payment channel graphs created by dumping a local view of the network from Lightning Nodes[12, 22, 24].These snapshots are built using the gossiping mechanism and may, therefore, be outdated and contain information about nodes that are not reachable anymore. While many interesting results are presented using this technique, it is unclear whether these findings represent the state of routing in the LN accurately.

Within any channel, both parties have a balance. Both balances added together form the channel capacity. Balances within channels can change with every payment. To route a payment through a channel, the balance in that direction should be at least equal to the payment amount. Consequently, the ability to route payments through that channel is changing rapidly. By design, balances are private. Although some studies attempt to conclude channel balances from the graph alone, they make some rough assumptions. It is impossible to accurately predict whether an LN payment would succeed without actually trying.

In this work, we make 2 main contributions.

First, a general connectivity experiment is performed. We attempt to establish a connection with a random sample of nodes that are visible from our Lightning node and reveal their network addresses. Properties of nodes such as types of network addresses they expose, number of open channels and latency are analyzed to reveal potential correlations and/or differences in behaviour. Measurements are repeated periodically for a few days to reveal how likely nodes are to close peer-to-peer connections.

Moreover, we probe the real network with fake payments of different volumes to see how well the routing performs under real circumstances. There are 2 main reasons most likely to cause a payment failure: either the route cannot be followed because one of the nodes went offline or a channel en route has insufficient capacity. These two reasons are distinguished by analysing errors that are returned from the LN. We attempt to establish links between payment volume and node reachability, as well as between payment volume and success rate. Additionally, we explore various node properties such as whether they expose Onion addresses and their channel count.

## 2. BACKGROUND

Bitcoin, launched in 2009, is the first digital currency in the world that can operate without a trusted third party. It relies on blockchain, a data structure that is essentially an unlimited sequence of blocks, where each block contains a list of transactions. A new block of limited size is appended to the chain once every 10 minutes. Because there is a limit on how many transactions can fit into one block, transactions that do not get added are put on hold and confirmed later.y network load, such as in December 2017, a very long queue of transactions can form leading to unacceptably slow perceived performance. The fees were increasing too because of the high competition for confirmations [25].

Joseph Poon and Thaddeus Dryja suggested using a network of *micropayment channels* that would run on top of the main blockchain and attempt to address the above problem [19]. They named it the Lightning Network. Their solution is aimed at resolving the above problem and had been running since 2018. The LN is only one representative of PCNs. An overview of other applications of PCN for improving the scalability of cryptocurrency is given by Gudgeon et al. [17].

A micropayment channel is a relationship between two parties letting them distribute bitcoins between each other. Distribution of value happens using so-called micropayments that are not broadcast to the blockchain. Once the two parties decide that all intended payments were completed, the channel is closed and one transaction summarizing all activity within the channel is saved to the blockchain *(Figure 1)*. This mechanism largely extends the number of transactions that Bitcoin can support since many transactions can be merged into one and sending to the ledger can be deferred [19].

Creating a micropayment channel requires making one

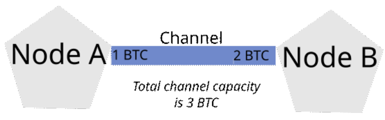

**Figure 2. Even though the channel has the capacity of 3 BTC, A can send at most 1 Bitcoin to B and B can send at most 2 Bitcoins to A.**

on-chain funding transaction to a multi-signature address that is controlled by both parties. The balance of that address represents the capacity of the channel created. Parties then make micropayments, distributing the channel capacity between each other. Both nodes have an in-channel balance, which limits how much money can be sent in each direction. An explanation is given in *Figure 2*.

It would be inefficient to create new payment channels for every pair of nodes that want to exchange payments since channel funding costs money in fees and takes time to confirm. Since doing that would defy the purpose of the Lightning Network, the LN supports routing micropayments via multiple existing in-between channels on the way to the receiver.

## 3. RESEARCH OBJECTIVES

To route a message through any kind of network, there must be a path to the destination. The same requirement holds also for payments in the Lightning Network. All nodes along the payment path have to be online and responding.

Moreover, to route a payment, all channels along the route must have a capacity of at least the payment size, otherwise the payment will fail. This requirement exists because of the way the LN is implemented. We are not going to go in-depth of reasons behind this since we consider it to be out of the scope of this work. An interested reader can refer to the original whitepaper [19].

In this work, we are going to carry out performance-related experiments on the current version of the Lightning Network to determine how well payment routing works. The following research questions are formulated.

- **Q1. How reliable are payments in the Lightning Network?** The primary goal of this work is to research how likely the payments sent into LN are to succeed (such that funds reach the intended destination). There had been previous suggestions that the Lightning Network is very unreliable and can not be used as an independent solution for real-life payments yet [14]. There are numerous reasons why a payment can fail and all of those need to be taken into account and explored when concluding general reliability of payment routing.

  **Q1.1. Network composition and stability of the LN nodes.**

  The first and the most straightforward failure reason is when one of the nodes on the route is offline and not accepting connections, either temporarily or permanently. Information about whether a node is offline or not is propagated to every network participant via the gossiping protocol. Information gathered this way is then used to build a so-called local view of the network's topology and choose a route along which a payment is going to be sent. Since inaccuracies in this information reduce the chances of payment for success, researching whether such information is always up to date is of interest. Because the Lightning Network is distributed, there is no single source of truth about the network's topology. Consequently, there is nothing to compare the information from gossip protocol against to verify its correctness. Therefore, an active connectivity experiment is carried out in which we attempt to connect to various nodes in our local network view and investigate whether they are reachable, how quick the connections are and how often they fail.

**Q1.2 Influence of payment amount on routing.**

Sending a payment in the LN also requires that all channels along the route have sufficient capacity in direction of the payment. Previous research suggests that insufficient channel capacity is a very frequent cause of routing failures [4]. There are previous suggestions that larger payments face this problem more often than smaller ones, with evidence of strong inverse correlation [14, 18]. The downside of the mentioned research is that it relies on static channel dumps obtained from the gossip protocol to construct and analyse a graph. Apart from the fact that such data may contain out-of-date information, it also does not hold any information about channel balances since those are hidden by design. As a result, conclusions about routing that are drawn this way require making certain assumptions and can only provide theoretical insight. Because of these limitations, we perform an active experiment by sending real payments of various amounts into the Lightning network. Such an experiment should help us understand how payment routing performs under real circumstances. Moreover, the correlation between payment amount and success rate can be confirmed.

- **Q2. Are there correlations between different node properties and performance metrics?** If we can experimentally establish some dependencies between node properties and their behavior in the network, separating nodes into groups based on their purpose may become possible. Nodes in the Lightning Network may be used for at least two clearly-defined purposes: some are used to pay for services anonymously, others exist for the sole purpose of collecting routing fees. We attempt to identify these use cases based on properties such as the number of open channels and types of exposed addresses. Then, comparing the results of both connectivity and payment routing experiments is possible to reveal potential differences between these groups.

## 4. RELATED WORK

Bitcoin and other cryptocurrency blockchains have been subject of numerous measurement studies since their inception. [20, 16, 13]. Payment channel networks such as the LN, on the other hand, have received less attention, possibly because of their novelty.

There exist a few online services like 1ml.com and explorer.acinq.co that attempt to crawl the Lightning Network and are therefore similar to conventional blockchain explorers. These websites can provide anyone with a public snapshot of the LN obtained by running a Lightning node for a long time.

Such data can be conveniently represented as a channel graph (with nodes as vertices and channels as edges) and graph analysis algorithms can be applied. This approach had been used in some publications. [12, 22, 24].

Martinazzi and Flori report that the largest connected component included almost all nodes of the network, with exceptions of some small groups, mainly consisting of single pairs. They also find that in a year from 2018 to 2019 the average number of channels per node increased. This, along with other evidence made them conclude that the network is growing quickly and becoming more centralized (with a lot of highly-active nodes being deployed) [22].

Seres et al. also used publicly available snapshots at a later time. The authors simulated an attack where the most important (i.e. having the most channels with other nodes) nodes are brought offline to see how severely the network performance would drop. They found that routing strongly relies on several most important nodes, sometimes called hubs, that everybody else establishes channels with. These gateway nodes are likely there to gain profits from routing fees [24]. Other studies come to similar conclusions [18, 21].

Rohrer et al. researched the graph properties of the lightning network and detected small-world and scale-free properties. Once again, network snapshots were used, which are not known to be up-to-date. They also provide data showing that the Lightning Network is constantly growing, which is why repeating previous experiments is important [23].

A few studies about the profitability of running a Lightning node exist. These aim to give Lightning node owners indication about optimal routing fees to charge to maximize profits. Beres et al. provide a working transaction simulator that can help with that task, as well as understand the general topology of the Lightning Network [12, 15].

The downside of using this kind of snapshots is that it is not known whether failing nodes and out-of-date information were removed from these snapshots. As mentioned before, that makes conclusions made on such data strictly theoretical. In reality, Lightning nodes often go offline permanently or for a little while and these changes do not propagate to every node of the network immediately. Performing active measurements by attempting to send real payments into the network is of much greater value as it helps understand the actual state of the network and how it would perform whenever a payment is made in real life.

There are very little studies that perform active experiments on the network as opposed to analysing the snapshot channel graph.

At the end of 2019, an experiment was described in a Medium post, which tested reachability of nodes in the Lightning Network, showing that close to 80% all nodes in the network can be reached with a payment from a well-connected source node. The author also researched how quick the payments are routed and found out that more than 10% of payments required more than 4 attempts and longer than 30 seconds to be successful. The reason behind a large number of attempts is outdated gossip information, which is why it is important to understand how quickly nodes change to further improve the gossip protocol [4].

To our best knowledge, there had been no previous research regarding the stability of nodes within the network that would use active connection probing technique. There are no such studies for Tor nodes either.

There is however numerous related research regarding payment volume and how it influences routing.

In a study by Blockstream, it was experimentally shown that routing higher volume payments in Lightning Network is much less reliable compared to smaller payments with a clear inverse correlation between the payment amount and the success rate. According to the article, only half of 5 US dollar payments succeed, with success rate dropping to close to 1% with volumes of more than 300 dollars. [14]

At the beginning of 2019 Guo Yuwei, Tong Jinfeng and Feng Chen revealed that for transactions with high value (i.e. above network median channel capacity), the success rate of routing is low and is decreasing over time due to exhaustion of channel capacities [18].

While mentioned studies show that the issue with large payment exists, they were performed extremely soon after the Lightning Network was launched and so are likely outdated. Moreover, all of those analyse graph dumps and make several assumptions (e.g. about channel balances since they are not disclosed), so their results may not accurately represent how payments of different amounts are going to behave in real life.

A blogpost in Diar presents shows that the total node count in the LN grew at a rate of about 9% per month in mid 2018 (a few months after launch). The channel count grew too, so did the amount of funds invested into the network [14]. Other studies also concluded that the LN is growing [22]. Since the network is growing at such a high speed, repeating previous studies is important. Additionally, it is interesting to see whether the network is still growing 2 years later.

Tikhomirov et al. propose an attack to uncover channel balances by probing the network with invalid payment attempts. While the attack is of no interest for us, a similar technique can be used for our experiments [26]. The technique involves sending payments with invalid invoice IDs so distinguishing different error codes becomes possible. That technique is already partially implemented in c-lightning *probe* plugin [10] and is also mentioned in a blog post by Decker [4].

There appear to be no studies regarding the role of Tor nodes within the LN. Because a core-periphery structure of the LN has been mentioned before [24, 18, 21], it is curious to see whether Tor nodes are stable and belong to the core or disappear more frequently and belong to the periphery part of the network. Periphery nodes are primarily used for direct one-time transactions between two parties. We are going to run experiments to understand how stable Tor nodes are in the network (i.e. how often do they go offline) and whether they have connectivity to high-capacity channels that are required to serve as routers for payments of various payment amounts.

Running experiments in the real network, as proposed in some publications will also give us more insight into actual routing performance for both IP and Onion nodes [26, 4]. Because of that, we will perform the first experiment to establish the link between

In this work, snapshot analysis is not going to be used as it is already covered quite extensively in previous publications. We will focus on performing active experiments instead for all of the research questions.

## 5. METHODOLOGY

### 5.1 Technical requirements

For the described experiments a Bitcoin node and a Lightning Node has to be set up and funded. Since the Lightning protocol is not delay sensitive, the geographical location of the node is not important as long as it has a stable internet connection. Additionally, a way to access Onion addresses has to be established. The server must have enough storage space to store the blockchain (275GB as of May 2020)[3]. The node will have to be online for a long while to conduct experiments so a server with good uptime is required. A good internet connection is a must.

### 5.2 Experiment environment

For this experiment, a VPS is created in a cloud location that satisfies all requirements mentioned in *Section 7*.

All experiments are carried out using a custom *c-lightning* Python plugin, based on the official *probe* plugin[10]. The plugin is organized in a way that makes it simple to keep track of experiments and their results by assigning unique identifiers to each experiment and organizing results accordingly. Raw results are stored in CSV files to keep it organized and to enable easy import into any database system.

### 5.3 Phase 1: Network composition and stability

The goal of the first phase is to research the composition of the Lightning Network and to understand the behaviour of nodes within it.

One of the lightning network node implementations, c-lightning [6] supports adding custom features using plugins written in Python. Additionally, the authors provide a collection of plugins for performing common tasks [9].

To achieve that goal, a custom *c-lightning* plugin was implemented in Python. This plugin attempts to establish a peer-to-peer connection to every node in a pre-configured node set, timing each connection attempt and keeping track of failures. A node may expose more than one address, usually because connectivity is required for both IP and Tor. In case the connection fails, other addresses are tried if any.

After connection to each node has either succeeded or failed, the plugin starts to periodically perform measurements to detect closed connections and whether our node has received any messages via the gossip protocol. These measurements are performed after these delays from the beginning of the experiment: 2, 4, 8, 12, 16, 20, 30 seconds, 1, 2, 3, 4, 5, 6, 10, 20, 30, 45, 60 minutes, 2, 4, 8, 12, 16, 20, 40 hours.

Lightning Nodes may be used to achieve various goals. Some may be trying to earn revenue by collecting routing fees, others may be using the Lightning Network to pay for services anonymously. The former have it in their best interest to stay as visible as possible to be able to route the maximum number of payments and earn the most.

It is therefore interesting to perform this experiment with nodes that have radically different connectivity in terms of the number of open peer-to-peer connections. That would let us confirm the above hypothesis and possibly find more differences in the behaviour of these nodes. Efficiently filtering out nodes by the number of connections they have open with other nodes proved difficult, so we chose the number of open payment channels as a metric that likely approximates the number of peers.

The technique described above has been used to carry out 3 experiments, with the only difference being the node set that the origin node establishes connections with:

- 50 nodes with exactly 1 channel open. Acquired from lightning RPC *listnodes* method.

- 50 nodes with 5-10 channels open. Acquired from lightning RPC *listnodes* method.
- 50 best-connected nodes: nodes with the highest number of open channels. Acquired from a public snapshot [8].

In each of the three cases, all nodes were used, regardless of whether they expose a Tor or an IP address.

Once the experiment is completed, the data is formatted and inserted into a relational database for analysis. Later, charts are plotted to help draw conclusions.

## 5.4 Phase 2: Payment routing performance

The goal of the second experiment to answer the question of whether the payment amount influences its chances to complete successfully. In contrast to previous research, in this work, we try to answer that question by actively probing the network with real payments from a c-lightning node hosted by us (the origin node).

Doing such an experiment with real payments can be very costly, so a way to probe the network without spending Bitcoin is needed. Payments in the Lightning Network are invoice-based: the payee sends an invoice identifier to the payer, who then issues a payment of the required amount, with the invoice identifier attached. Conveniently, whenever a node receives a payment with an invoice ID unknown to it, the payment is rejected with a distinct error and no money is spent.

This principle is exploited by the probe plugin[10] provided by creators of *c-lightning*. Largely inspired by that, we have created a similar plugin which can give us more control over the experiment.

Before starting, we must ensure that our node has been online for a while and has gathered some gossip information. After that, channels need to be established from our Lightning Node to a few other ones.

The more channels a node has, the larger part of the network it can in theory reach, so choosing nodes with a large number of channels is crucial for the success of our experiment. 3 nodes with the largest number of channels open were chosen, according to the LN explorer *1ml.com*[1].

After choosing the nodes, channels between them and the origin node have to be created and funded. We want to minimize the risk of payments failing during the first hop (i.e. at the channels we establish), so these channels need to have at least the capacity equal to the amount of 1 payment. In practice though, having a larger capacity can allow sending multiple probes at once. We chose to create 3 channels of around 230000 satoshis each (equal to USD 21 at the time of the experiment).

Once channels are funded and locked in, we attempt to send 3 payments of different volumes to each node in the visible network. The list of nodes is retrieved from *c-lightning's listnodes* RPC method. In total 3 payments are going to be sent to each node: USD 0.01, 1, and 10. Each payment attempt is timed and responses are recorded. Whenever a temporary channel error is received, the failing channel is added to the temporary blacklist, so the next attempt is going to use a different route, as recommended by the protocol. Attempts failing because of temporary channel errors are retried up to 25 times. We do not attempt more than 25 retries due to time constraints. Results obtained this way may help to confirm that the correlation between payment amount and success rate theoretically suggested in previous research exists in practice [14, 18]. Moreover, differences in routing performance between Onion and IP nodes are explored, if any. Results are going to be plotted and compared with previous work.

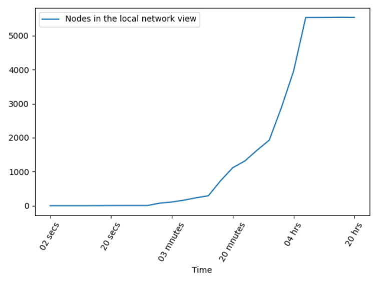

**Figure 3. Nodes in the local network view over time. The network view is completed each time a gossip message is received.**

## 6. RESULTS

## 6.1 Network composition

We first present our findings on the network composition and peer-to-peer connectivity, together with some other discovered properties of nodes in the network. We must emphasize that this section only covers the peer-to-peer network, to answer the research question **Q1.1**. Section 6.2 describes results regarding the payment channel graph, answering the question **Q1.2**.

### *6.1.1 Node count and network growth*

Between May 31 and June 20 2020, the total number of nodes reported by *c-lightning* grew from 5615 to 5743. In other words, the number of visible nodes in the network increased by 2% in 21 days. This shows that more people are getting involved with the Lightning Network.

That number of nodes roughly corresponds to the total network node count reported by the Lightning Acinq Explorer and other sources [8, 4]. It is, however, lower than that reported by 1ml.com explorer [1] because the latter does not implement pruning of stale nodes and nodes with no active channels.

A study in Diar reported node count growth to be 200 nodes in 21 days, or 9% in the same period of 2018 [14]. This difference in growth rate in 2018 and 2020 is expected because Diar made the measurements shortly after the LN's launch, when the public interest was much higher. Furthermore, their measurements immediately followed Bitcoin's popularity peak, which also could have contributed to LN's rapid growth.

Finally, out findings imply that the Lightning Network continues to grow. The interest is still there. As a consequence, we must stress that LN experiment results may quickly become outdated and have to be repeated.

### *6.1.2 Incoming gossip messages*

The local network view is a list of all nodes and channels known to a local node. It is built based on gossip messages received from peers, which are essentially node and channel announcements propagated through the network. While there are no peers connected, the local network view remains empty. Whenever a node announcement arrives via the gossip protocol, it is saved and the local network view grows.

After connecting to a total of 150 randomly chosen peers,

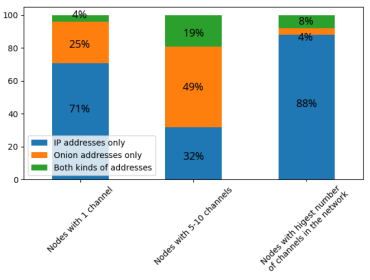

**Figure 4. Public addresses exposed by nodes.**

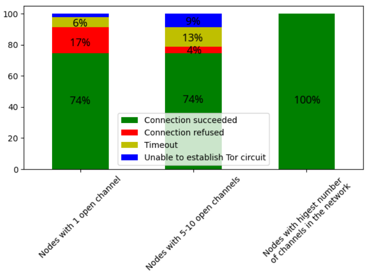

**Figure 5. Success rate and common errors when establishing peer-to-peer connections to Lightning nodes**

local network size measurements have been started using RPC listnodes method.

Figure 3 shows how quickly the local network view is filled up. There is a delay of around 1 minute before gossip messages begin to arrive. After 4 to 6 hours the node count reaches approximately 5600, which is around the total number of nodes in the network (Section 6.1.1). After that, the local network continues to grow slowly.

An important thing to understand is that having a complete local network view is vital for having optimal payment routing performance, because routes are generated based only on the locally available information. Consequently, a Lightning node is unusable for payment routing for the first 4-6 hours after coming online. We discuss whether this is a serious problem in *Section 7.*

Before continuing, we have collected some nodes from our local network view and divided them into three categories, based on the number of open channels they have.

- 1. Nodes that have exactly 1 visible channel
- 2. Nodes that have 5 to 10 visible channels
- 3. Nodes that have the largest number of channels in the visible network.

We had to limit our experiment to 150 nodes in total, or 50 nodes per category to complete the experiment quicker. Nodes for each category were chosen at random, so we expect it to be a representative sample.

It appears that nodes expose different kinds of network addresses depending on the number of open channels they have.

*Figure 4* shows that 68% of nodes with 5-10 open channels expose an Onion address either exclusively or alongside an IP address. The remaining 32% expose only IP addresses. More frequent use of Onion addresses may witness about their will to preserve anonymity. Prevalence of Onion addresses among nodes with only 1 channel (category 1) is significantly lower. Over 70% of nodes in this category only expose an IP address. Finally, nodes with the most channels in the network tend to only expose IP addresses with 88% of them not exposing any Onion addresses. This confirms that these nodes generally exist to route payments of others and that anonymity is not an important factor for these. Nodes in this category have it in their best interest to be reachable by as many nodes as possible, so almost all of those expose IP addresses only.

To confirm that the dependence between channel count and types of addresses is not accidental, a statistical test is performed. We represent the data categorically, with 3 categories for channel count and 3 for address types, similarly to Figure 4. Because the data is categorical, a Chi-squared test is used, with degree of freedom equal to 4. We receive the Chi-squared value equal to 41.28, which exceeds the critical Chi value 9.49 corresponding to our degree of freedom. This lets us reject the null hypothesis and confidently conclude that there is a correlation between channel count categories and address type categories.

### *6.1.3 IPv6 and Torv3*

We found that most of the nodes exposing an Onion address use Onion addresses of version 3. Only 15% of Onion nodes still expose older Onion 2 addresses. High preference for Onion 3 addresses is to be expected since migrating does not require any hardware updates. Onion addresses of the latest version are also now the default and feature many security improvements over the previous versions [5]. Simple migration and additional security benefits could be the driving force to migrate to the latest version for users with security in mind

Furthermore, our findings suggest that the adoption of IPv6 in the Lightning Network is lower than in the main Bitcoin network. According to our results, Just 3% of IP addresses are of version 6, while the rest are still version 4. A study from 2018 shows that at that time, 13% of mainnet Bitcoin nodes were using IPv6 [11]. The reasons behind such difference remain unclear.

### *6.1.4 Node connectivity*

We attempted to connect to every node from the above list and compare the outcome across the three categories. The results demonstrate two things.

First, nodes with the most number of channels are reliable and quick to respond to connection attempts: we have not witnessed one connection failure. Connections to smaller nodes on the other hand resulted in a failure in 27% of cases *(Figure 5).*

Second, nodes with a lower number of channels are much slower to respond to connection attempts: on average, it took 8 seconds to establish a connection with nodes from the first two categories, while most larger nodes replied in under a second. These findings support the theory that nodes in the last category strive to achieve the best connectivity.

Common reasons of failure are presented in *Figure 5*. It can be seen that almost 80% of failures are caused by either TCP connection refused or TCP timeout, which likely means that a node is behind a firewall, or resides in a private network, behind a NAT.

### *6.1.5 Disconnecting nodes*

Out of 150 nodes that we have established connections with, none terminated the connection within the first 40 hours, even though no channels were open by us. Peer to peer socket connections do not require a lot of resources nor any maintenance and so the fact that they are not actively terminated is not surprising.

On the other hand, a connection would get terminated whenever a node goes offline for a short while, e.g. when a personal computer is turned off at night. However, it appears that even nodes in the first two categories do not experience that at least in the first 40 hours. These results are unexpected and may imply that there are certain mechanisms in c-lightning that take care of silently restoring failed connections, if possible. In future works, running the same experiment longer may be of interest to find out when and how often nodes disconnect. It is also interesting to explore the *c-lightning* source code to explore the reasons behind our results and the future ones.

## 6.2 Payment routing

At the time of the experiment (mid-June 2020), there were 5721 Lightning nodes in the visible network. With 3 payment probes for each node, that would mean that 17163 probes have to be sent.

9510 probes were discarded as there was no suitable route to send them along: either there was no route at all or all existing routes did not have sufficient capacity to send a payment. Before starting with this experiment, the Lightning node has been allowed to run for a few hours (Section 6.1.2 explains why) to collect information about the network via the gossip protocol. As a result, the local network view should be fresh and the routes computed should be accurate.

Discarding 9510 probes left us with 7091 probes to send. This means that having a funded channel open to three of the largest nodes in the network still does not connect us to more than half of the network, contrary to conclusions made in previous works.

Martinazzi et al. concluded that the largest connected component in the LN contains most of the nodes in the network [22]. In our work, however, we found that half of the network is not reachable from our node, which implies that the largest connected component can not contain most of the nodes in the network.

There could be various reasons why our results differ. *First*, the mentioned study was performed in the first year of Lightning Network's existence. The structure of the LN likely changed and the number of users grew, as forecasted by the article in Diar and Martinazzi et al. [14, 22].

*Second*, the methodology of Martinazzi's study is solely theoretical. They use static network snapshots to calculate network properties, which can work in practice but fails to account for failures caused by insufficient channel capacity. Since channel capacities are not visible in snapshots, one can conclude that two nodes are reachable from each other if there is a path between them. In reality, however, a node is reachable with a payment from the other one if in the channel graph the maximum flow between those is greater or equal to the payment size. Consequently, an active experiment such as ours can give more accurate results.

A different study by Decker shows that 48% of nodes are not reachable (i.e. return a permanent error on the first attempt). In that study active experiment like we did was performed. Our results show that 57% of nodes are unreachable, which is very close to Decker's findings. The difference could be caused by different payment amounts used when probing or different years of experiment.

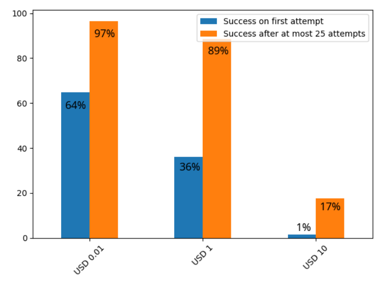

**Figure 6. Percentage of probes that succeeded with and without replies. Includes probes to reachable nodes only.**

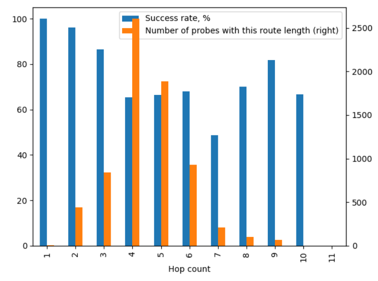

**Figure 7. Distribution of route length (orange) and corresponding success rates (blue).**

While 36% of all probes succeeded after the first attempt, there is a noticeable difference in success rate between probes of different sizes. As can be seen from *Figure 6*, over 60% of smaller payments reached the destination with just one attempt. On the other hand, only 1.38% of 10 dollar payments reached their destination without retries.

After retrying probes that failed due to transient channel errors, success rate across all three probe categories increases significantly, however, the inverse correlation between payment size and success rate is preserved, which correlates with theoretical conclusions of previous studies [14, 18]. While almost all payments of tiny amounts succeeded when probing with retries, only 17% of 10 dollar payments reached the payee successfully. 1 in 10 payments of 1 dollar never reached the destination, even after 25 retries.

On average, every second payment of one USD cent had to be retried to succeed. Payments had to be retried on average 1.6 and 9.2 times for amounts of 1 and 10 US dollars respectively. Since each new attempt takes time, this implies that among successful payments, larger payments took more time than smaller ones.

As expected, payments that are routed over fewer hops have a higher chance to reach the destination successfully (See *Figure 7*).

*Figure 5* gives a summary of different error messages that were received while probing the network. It appears that even after 25 retries, temporary errors are still the most common. Apart from that, some payment attempts timed out after 2 minutes, while others failed because one of the channels on the route had insufficient capacity.

## 7. DISCUSSION

In section 6 we presented a few insights into the performance of payment routing and the composition of the publicly visible peer-to-peer network. In this section, we discuss our findings with the goals of the network in mind and try to highlight the problems most detrimental to LN's success.

## 7.1 Ability to route payments

According to our results, payments via routing are very unreliable, especially for larger amounts. Payments of USD 10 are close to 6 times less likely to succeed than payments of USD 0.01, with 83% of USD10 payments failing to reach the destination. Moreover, 53% of probes were never performed because there was no route at all. Our findings demonstrate a strong inverse correlation between payment size and its chances to reach the destination, which have been predicted by numerous research [14, 18]. Transient channel error is the most frequent reason of failure especially for larger probes, which means that channels in the network are not funded sufficiently. Failure to route larger payments corresponds to limitations of routing imposed by the original whitepaper: the capacity of all channels along the route must be no less than the size of the payment [19]. We believe that attracting more users willing to invest more money into channels can resolve that problem. How that can be done is out of scope, but increasing routing fees to provide a good incentive for users can be one of the solutions.

## 7.2 Outdated network information

More than half of all probes were discarded because there was no route to the destination. One may ask why the gossip protocol does not purge these nodes from the local network view in the first place. The answer is, only nodes without any open channels attached directly to them are purged. That is because users may still want to establish a direct channel to an unreachable node, for which the other party needs to be known.

The majority of all probes had to be retried multiple times to succeed. The reason behind this is that information about network topology that is received via the gossip protocol and used to construct a payment route is outdated. Whenever a transient error is received, the routing table is updated so a better route can be found and next attempts can succeed. Consequently, payment attempts are repeated many times and take a long time before completing, which defies one of the purposes of the Lightning Network.

## 7.3 Centralization and Lightning usage scenarios

By looking at the three categories of nodes we used in our experiments and comparing their performance, we can perhaps make conclusions about goals these nodes try to achieve by using the Lightning Network. Connecting to the top 50 nodes by channel count is many times faster, on average, than connecting to smaller nodes. Figure 5 also shows that there is a much larger chance that these nodes fail the connection.

The reason behind this might be that nodes with fewer channels reside in private networks, behind a firewall or NAT since these kinds of systems may actively drop packets or send TCP RST packets in response, for example, because they deem the Lightning port numbers to be malicious. This, together with the fact that nodes with fewer channels are interested in staying anonymous *(see Figure 4)* suggests that these nodes are likely used by individuals to pay for services rather than collecting routing fees for profit.

The top 50 nodes, on the other hand, are probably focused on collecting as much routing fees as possible. These nodes have a lot of channels of high capacity, which makes them good candidates as intermediate hub nodes. These nodes are also much more reliable in terms of connectivity. Establishing a connection to these nodes never failed and took less than a second on average. This, along with the fact that they also almost exclusively use IP addresses suggests that for these hubs, being accessible to a maximum number of nodes is more important than preserving anonymity.

The whole network relies on hub nodes to route payments, which makes the LN *de facto* centralized. Centralization in the Lightning Network can lead to problems in performance and privacy. For example, hub nodes may collect information about huge numbers of peers (e.g. Lightning explorers [1, 7]) and exploit their key position in the network to raise fees or block some transactions [22]. Since the basic goals of the Lightning Network are undermined, the centralization problem needs to be fixed.

On average, a successful probe followed a route of only 3 hops, excluding the channel connecting our node to the hub, with a standard deviation of 1.3, implying that the channel graph of the Lightning Network may have some characteristics of scale-free networks, as have been previously noticed. [23].

## 7.4 General reliability of the LN

As can be seen from the above conclusions, the Lightning Network is not ready to fully address all issues of Bitcoin yet. Sending payments to nodes without a direct channel remains unreliable. Amounts most common in everyday life like 1 to 10 dollars have a very low chance of arriving at the destination. As mentioned above, this is mostly because the network lacks enough nodes that can invest sufficient amounts of money into channels. Although that is expected since the Lightning Network is quite new and constantly improving, an emphasis should be put on attracting more investors to make the Lightning Network live up to its full potential.

## 8. SUMMARY

Payment routing performance, node properties and goals, network size and growth were researched in this work. Active measurements were used where possible as opposed to snapshot analysis, which is dominating in the previous work.

Inverse correlation between payment amount and success rate forecast by previous research, was confirmed in practice. The majority of 10 USD payments never reached the destination, even the smallest payments required multiple attempts. More than a half of the network is not reachable at all. The Lightning Network in its current state can not reliably route payments of the most common amounts. The issue is that the network lacks users that are willing to invest more money into channels.

Network growth of 2% recorded in a period of 3 weeks shows that the network is still attracting new users. That means that there is still a chance that a critical user mass is going to accumulate by itself (without making any network adjustments) after some more time.

Additionally, it seems that goals of nodes can be deduced by analysing node properties. We found that nodes with large number of channels usually do not expose Onion addresses and respond to connection attempts very quickly. Nodes with fewer channels are much less reliable and often expose Onion addresses. We conclude that these differences are caused by differences in what node owners want to achieve in the network. Separating nodes into categories this way can be helpful in future studies.

## 9. REFERENCES

[1] 1ML - Lightning Network Search and Analysis Engine - Bitcoin mainnet. https://1ml.com/.

[2] Bitcoin Scaling, Lightning Network & The Future with Micropayments — Nichanan Kesonpat. https://www.nichanank.com/blog/2019/1/5/bitcoin-scaling-lightning-network-micropayments.

[3] Blockchain Charts. https://www.blockchain.com/charts/blocks-size.

[4] c-lightning Plugins 05: Probe Plugin Results - Blockstream Engineering Blog - Medium. https://medium.com/blockstream/c-lightning-plugins-05-probe-plugin-results-984a18745e93.

[5] doc/NextGenOnions – Tor Bug Tracker & Wiki. https://trac.torproject.org/projects/tor/wiki/doc/NextGenOnions.

[6] ElementsProject/lightning: c-lightning — a Lightning Network implementation in C. https://github.com/ElementsProject/lightning.

[7] Lightning Network Statistics - Bitcoin Visuals. https://bitcoinvisuals.com/lightning.

[8] Lightning Nodes - Top Channel Count. https://1ml.com/node?order=channelcount&json=true.

[9] lightningd/plugins: Community curated plugins for c-lightning. https://github.com/lightningd/plugins.

[10] plugins/probe at master · lightningd/plugins. https://github.com/lightningd/plugins/tree/master/probe.

[11] Ben Mariem, S., Casas, P., and Donnet, B. Vivisecting Blockchain P2P Networks: Unveiling the Bitcoin IP Network. *ACM CoNEXT Student Workshop* (2018).

[12] Beres, F., Seres, I. A., and Benczur, A. A. A Cryptoeconomic Traffic Analysis of Bitcoin's Lightning Network. *http://arxiv.org/abs/1911.09432* (11 2019).

[13] Decker, C., and Wattenhofer, R. Information propagation in the Bitcoin network. In *13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013 - Proceedings* (2013), IEEE Computer Society.

[14] Dior. Lightning Strikes, But Select Hubs Dominate Network Funds. *Volume.2 Issue.25 https://diar.co/volume-2-issue-25/* (2018).

[15] Ersoy, O., Roos, S., and Erkin, Z. How to profit from payments channels. *Proceedings of the 24th Financial Cryptography and Data Security, Kota Kinabalu, Sabah, Malaysia* (11 2019).

[16] Gencer, A. E., Basu, S., Eyal, I., van Renesse, R., and Sirer, E. G. Decentralization in Bitcoin and Ethereum Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10957 LNCS* (1 2018), 439–457.

[17] Gudgeon, L., Moreno-sanchez, P., Roos, S., and Mccorry, P. SoK : Layer-Two Blockchain Protocols. *Financial Cryptography and Data Security*, i (2020), 1–44.

[18] Guo, Y., Tong, J., and Feng, C. A measurement study of bitcoin lightning network. In *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019* (2019).

[19] Joseph Poon, T. D. The Bitcoin Lightning Network: Scalable Off-Chain Payments, http://lightning.network/lightning-network-paper.pdf.
1–59.

[20] Kim, S. K., Mason, J., Ma, Z., Miller, A., Murali, S., and Bailey, M. Measuring ethereum network peers. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC* (New York, NY, USA, 10 2018), Association for Computing Machinery, pp. 91–104.

[21] Lee, S., and Kim, H. On the robustness of Lightning Network in Bitcoin. *Pervasive and Mobile Computing 61* (1 2020), 101108.

[22] Martinazzi, S., and Flori, A. The evolving topology of the Lightning Network: Centralization, efficiency, robustness, synchronization, and anonymity. *PLoS ONE 15*, 1 (2020).

[23] Rohrer, E., Malliaris, J., and Tschorsch, F. Discharged Payment Channels: Quantifying the Lightning Network's Resilience to Topology-Based Attacks. In *S&B '19: Proceedings of IEEE Security & Privacy on the Blockchain* (2019).

[24] Seres, I. A., Gulyás, L., Nagy, D. A., and Burcsi, P. Topological Analysis of Bitcoin's Lightning Network. In *1st International Conference on Mathematical Research for Blockchain Economy.* 2019.

[25] Thorsrud, E. Long-term Bitcoin Scalability. *NTNU*, June (2018).

[26] Tikhomirov, S., Pickhardt, R., Biryukov, A., and Nowostawski, M. Probing Channel Balances in the Lightning Network.