University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Effectivenes simulation of a rebalancing algorithm for the Lightning Network under partial participation

Bachelor Thesis

August 1, 2020

Organization	FHNW, School of Business, Basel
Study program	Business Information Technology
Author	Tobias Koller
Supervisor	Prof. Dr. Kaspar Riesen
Project sponsor	Prof. Dr. Thomas Hanne
Expert	René Pickhardt

Abstract

Here follows my abstract Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.1-2

Keywords: Lightning, Bitcoin, path finding

Contents

DECLARATION OF HONOR

I the undersigned declare that all material presented in this paper is my own work or fully and specifically acknowledged wherever adapted from other sources. I understand that if at any time it is shown that I have significantly misrepresented material presented here, any degree or credits awarded to me on the basis of that material may be revoked. I declare that all statements and information contained herein are true, correct and accurate to the best of my knowledge and belief. This paper or part of it have not been published to date. It has thus not been made available to other interested parties or examination boards.

Tobias Koller	Place / Date

Foreword

Some background

1 Introduction

This section gives a brief introduction and overview of the Bitcoin and Lightning technology. It covers the history of digital cash, the advent of Bitcoin and explains the need for an off-chain scaling solution.

1.1 Bitcoin: Peer to Peer Electronic Cash

Bitcoin is a peer-to-peer electronic cash system first introduced in a white paper by the individual or group behind the pseudonym Satoshi Nakamoto (Nakamoto, 2008). This paper lines out the fundamental principles of the Bitcoin block chain that achieves digital transfer of value without a central third party. The next paragraph expands on the pre-bitcoin developments on electronic cash that eventually lead to the rise of Bitcoin. Additionally, we explain why there is a need for additional layer protocols.

glossary block chain

1.1.1 History of digital cash

Before the digital age cash was the dominant form of payment. A bank note or a coin embodies the respective face value to the bearer of it. Economical transactions can be made by simply exchanging this physical token by which the transaction was immediately settled. However, with the advent of e-commerce this simple and transparent mechanism was no longer possible. New institutions formed to fulfill the need of online transactions. Credit card companies and payment processors filled the gap of trust needed between the sender and the receiver of a transaction over the internet. This architecture came with significant drawbacks. Suddenly, the interacting parties are dependent on third parties which are collecting additional fees. Use of such systems require identification and since the intermediary can track all transactions, this reduces the user's privacy (Narayanan, Bonneau, Felten, Miller, & Goldfeder, 2016).

Bitcoin is by no means the first system introduced to allow for a digital cash system. Already in in 1983 Chaum worked on new cryptographic primitives that should make electronic banking services more private and offer improved auditability (Chaum, 1983). Although his technology still relied on a central "bank" server which issues electronic bills, blinded signatures allowed to anonymously transfer them. In 1989 the company **DigiCash** was found by Chaum to commercialise the idea and a few banks did later implement it. Technological complexity, patents on the invention and the incapability of handling user-to-user transactions prevented it from becoming a success Narayanan et al., 2016.

An important problem in the evolution of digital cash has been the so called **double spend problem**. Digital information has the property of being easily duplicated. This poses a problem to digital cash as this behaviour is generally unwanted. How can a receiver of digital cash be certain that the cash has not been spent to someone else before, thus eliminating its value? Satoshi Nakamoto introduced the concept of a global distributed ledger, a data structure that is append only and where any change must be disseminated to all participants. In order to keep the history of the ledger immutable Satoshi utilizes the idea of a time stamping server first proposed by Haber and Stornetta (1991) in 1991. It works by calculating the hash of a piece of data and publishing it to all the participants. This serves as a proof that the data existed at this given time since otherwise the hash could not have been calculated. The next piece of data to be published also contains the previous hash, effectively linking them and forming a chain. If someone would now want to change the underlying data this would change its hash and since it is included in the next element in the chain also this hash would need to be changed up until the most recent element.

explain hash and add to glossary

Maintaining a global state of transactions in a constantly changing network of participants which can not be trusted is challenging. A single user could spin up hundreds of nodes to overcome the conses of the network. How can a consensus be formed without a central authority? A solution to a similar problem was proposed by Back (2002) in 2002. To prevent e-mail spamming he introduced a mechanism that requires the sender to solve a puzzle that is computational heavy. This so called "proof of work" is requested by the receiver of the e-mail to trust that it is no spam. Since this computation can easily be done for one e-mail it becomes a big burden to do for thousand of e-mails therefore avoiding spammers. The puzzle simply involves finding a value whose hash starts with a certain amount of zero bits. Since the result of a hash function can not be predicted only brute force can be applied to find such a value. By selecting the number of leading zero bits one can change the difficulty of the puzzle. Each additional zero leads to the difficulty to be doubled. In order to add transactions to the Bitcoin ledger a participant constructs a block consisting of transactions, computes the proof of work and publishes it to the network. Only if all transactions are valid and the work done has been verified network participants append the transactions to their local copy of the ledger and further broadcast the block.

add pow to glossary

Combining proof of work with the chaining of hashes introduced in the last paragraph results in a strong security model. An attacker who wants to change the history of the ledger would need to recompute the proof of work of the changed and all subsequent blocks as they are linked by their hashes. Therefore, every new block makes it increasingly more difficult to change a transaction in the ledger. Number of blocks on top of a transaction in question is hence be referred to as **confirmations**.

1.1.2 Scaling solutions

One of Bitcoin's value propositions is being decentralised. At the same time every transaction ever made must be distributed and stored among all network peers. It becomes obvious that some trade-off has to be made to maintain those two properties: scalability. This is often referred to as the scalability trilemma which states that in distributed systems the three objectives **security**, **decentralisation** and **scalability** can not be achieved in full extent at the same time. While two can often be achieved, there are certain trade-offs to be made in the third domain. This section is explaining why this is true for Bitcoin and what main categories of solution exist.

trillema in glossary

While decentralisation can be described on many levels we focus here on the decentralisation of validating nodes. Those are network participants that verify the blocks published by miners. Decentralisation would be best achieved if every user of Bitcoin would run its own fully validating node to receive information about the ledger independently. On the other hand the network would be centralised when only few nodes would validate and users would need to trust those to tell the truth about the ledger state. To keep decentralisation high it is crucial to keep the hardware and network requirements as low as possible. The Bitcoin protocol restricts the amount of data to be processed to one block of 1 megabyte per 10 minutes. A full node in the network must be able to download at least 1 MB / 10 minutes in order to keep up with the tip of the chain. Lower bandwidth would cause it to get left behind without every being able to catch up. Additionally, the full node must keep the full ledger on storage. This yields to approximately 286 GB „blocks-size", n.d. at time of writing, increasing linearly in the future. This upper limit block size results in a throughput of approximately 7 transactions per second

add full node to glossary

(tps) Poon and Dryja, 2016. Clearly by several orders of magnitude smaller than what it would require to become a world wide payments network.

Bitcoin, by design, promotes security and decentralisation while sacrificing scalability. However, to be usable by everyone the scalability issue needs to be addressed. As Bitcoin is not owned by anyone there is no one party to decide on the future design decisions. This lead to a scaling debate with two ideological camps on how to progress. Scaling on-chain or scaling off-chain.

Scaling on-chain means lifting the 1 megabyte block limit, allowing for a higher transaction throughput. While this seems the most straight forward solution it can not be achieved without trade-off. As previously discussed decentralisation can only be maintained by keeping the hardware and network requirements low. Removing this restriction to allow worldwide usage would mean that nodes soon need to process hundreds of megabytes or even gigabytes per second, effectively reducing the number of nodes that can still keep up, leading to a more centralised network.

An off-chain scaling solution describes any system that acts outside of the Bitcoin protocol but is linked to it, in a way that leverages the number of economical transactions that can be performed per single on-chain transaction. These solutions build a second layer of abstraction. While still using functionality of the base layer they can reduce their dependency and make their own design decisions and trade-offs based on the scalability trilemma. The Lightning Network is only one possible off-chain solution and is described in the next section in more depth.

glossary second layer

glossary base layer

1.2 Lightning technology

The Lightning Network is a network protocol that utilizes Bitcoin as its underlying trust system. It can, therefore, be described as a “second layer” protocol building upon the Bitcoin “base layer”. Bitcoin does not scale on the base layer since it was designed with security and decentralisation in mind. Off-chain solutions like Lightning are developed to facilitate more transactions on a different layer without compromising the properties of the base layer (Poon & Dryja, 2016).

1.2.1 Payment channels

The transaction bottleneck in Bitcoin is imposed since every network participant needs to be updated about every transaction. This is required to ensure the integrity of the system and not because the transactions are of interest to the nodes. In fact they can learn very little about the parties involved in a transaction as pseudonymous public keys are used as identifiers. Poon and Dryja mention in the Lightning Network Paper that as long as only two participants care about a recurring transaction there is no need to inform the entire network about it (Poon & Dryja, 2016). They therefore propose that those two participants do not sent the transaction to the network but instead hold on to it and agree on their balances bilaterally. „Micropayment channels create a relationship between two parties to perpetually update balances, deferring what is broadcast to the block chain in a single transaction netting out the total balance between those two parties.“ (Poon and Dryja, 2016, p. 4)

Opening such a payment channel requires the two parties to create an on-chain Bitcoin transaction which spends an amount of bitcoin to a 2-of-2 multisignature contract. Only both parties collaboratively can spend it from there. At the same time they create a refund transaction that spends from the multisignature contract and distributes the current balance state. However, they could broadcast this transaction instead they can also decide to hold on to it and update it at a later stage. Holding the signed Bitcoin transaction ensures that they could at any time get their balance back on-chain. Every time they wish to transact with each other they just sign a new refund transaction which distributes the updated channel balance.

Example: Assume Alice and Bob open a payment channel and both send 0.5 BTC to a 2-of-2 multisignature contract. This channel has a total capacity of $0.5 + 0.5 = 1$ BTC. At the same time they also create a transaction that pays them back their initial balance of 0.5 BTC each. This transaction is not sent to the block chain but kept private by the two parties. If Alice wants to send Bob 0.2 BTC they update this transaction spending the initial 1 BTC to be distributed as follows: 0.3 BTC to Alice; 0.7 BTC to Bob. This payment was only agreed upon Alice and Bob, no one else needed to be notified. Unlimited future transactions like this can take place as long as the sender still has some balance. At any point in time Alice or Bob can decide to broadcast the latest transaction, distributing the agreed balance and effectively closing the payment channel. As depicted in figure 1.1 there can be many modifications to that balance.

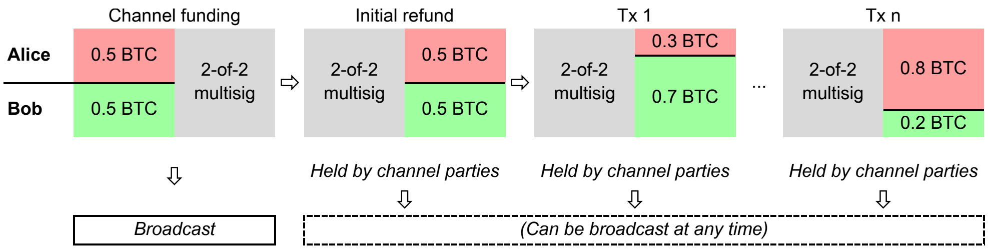

Figure 1.1: Life cycle of payment channel.

Whenever the parties decide to end there collaboration they can close the channel by broadcasting a closing transaction to the block chain which pays out each party its respective balance. While beneficial, collaborative channel closure is not required, each party can close the channel at any given time by broadcasting the latest transaction agreed upon.

Participants in such channels are referred to as Lightning nodes, a computer system that runs an implementation of the Lightning Network protocol. Those nodes are not to be confused with Bitcoin nodes. However, since most Lightning nodes need direct access to the Bitcoin block chain most Lightning nodes are Bitcoin nodes at the same time.

1.2.2 Using other channels for payments

Since a payment channel is a biparty relationship an update of channel balance can only ever represent a transaction between those parties. Instead of opening a channel to each participant a node wants to transact with there is the opportunity to route payments through multiple channels.

Often economical transactions to a specific recipient are made only once. It would therefore, defeat the purpose of Lightning to open a channel for this unique payment because both opening and closing of a channel takes each one transaction on the base block chain. Using Lightning would not use the load on-chain but rather increase it by a factor of two. This demonstrates that a payment channel should only be opened if the cost of doing so can be amortised over many transactions. This is either the case if the channel parties are expected to repeatedly transact with each other or if they can facilitate transactions between other nodes in the network through routing.

glossary econ. tx

Handling payments off-chain so far only worked between two parties that update Bitcoin transactions off-chain with the security of claiming their balance at any time on-chain. How can payments through multiple channels be accomplished without introducing trust into the system? If Alice wants to pay Carol through Bob, how can she be ensured that Bob will not keep the money and refuse to pay Carol? The solution are so called **Hash Time Locked Contracts** or short "HTLC" .

define htlc in glossary

First Alice asks the payee, Carol, to create and keep a secret **R** and only share the hash of it, $hash(R)$. Alice then uses this, so called *preimage* , to create a special HTLC transaction that promises Bob to receive the payment amount if he has knowledge of R. Bob also gets informed by Alice who is next in the path to find out this secret. Bob then himself creates an HTLC with the same preimage to Carol offering her the same amount upon disclosure of R. Carol is the payee in the transaction and has knowledge of R so she could technically claim the funds on-chain. There is however an easier solution than this. After she releases R, Carol and Bob can simply agree to update their channel to reflect this payment. The same is done between Alice and Bob leading to the state where all channel balances are updated and the HTLCs is rendered useless. This mechanism is called "time locked" because the offer of payment to the knowing R is only valid for a certain timespan. If R does not get released until the defined expiry the HTLC turns invalid. Therefore, it is important that each successive HTLC has a lower timespan.

glossary preimage

1.2.3 Routing

The previous chapter explained how a payment can utilize multiple channels to reach its destination without the introduction of trust between participants. This section describes how routing takes place and what trade-off needs to be made.

To ensure privacy of payments the routing protocol uses onion routing . The sender of the payment encrypts the information in multiple layers so that each hop only learns about the next hop in the path. Each intermediary node only knows where the payment comes from and where it goes next. Who is the payer and the payee remains secret. Only the last node in the path finds out that it is the actual receiver of a payment.

define onion routing in glossary

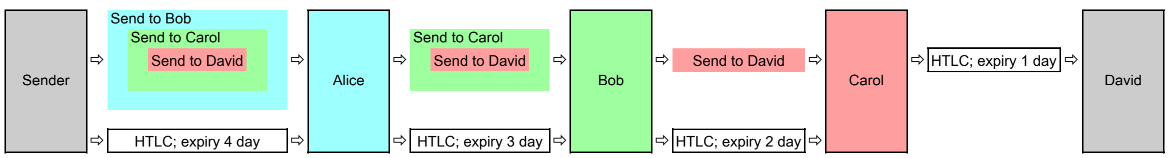

Figure 1.2: Onion routing preserves privacy over the chose path.

Figure 1.2 color codes the information which is only readable by the respective node with the same color. In each step the outermost layer is decrypted and the next hop gets revealed. The remaining encrypted data is passed on down the line. This ensures that each node only has a local view of the path knowing only about its predecessor and successor. Thus follows, the source of the payment needs to determine the complete path before sending a payment.

Sending a payment requires the sender to find a viable path to the recipient. This is called source routing and necessitates some information about the channel graph. When a new channel opens the parties notify the network about the channel, its capacity and the fee policies. Each other network participants stores this information and builds up its local view of the network. When a node constructs a payment it can query a path with enough capacity along all its channels, though this is not enough to be certain that the payment can be routed. The payment in

glossary source routing

figure 1.3 of 0.5 BTC can not be routed from Bob to Carol as Bob only owns 0.2 BTC of the total channel balance. This information however is only known to Bob and Carol, not to Alice. Alice only knows the public information which tells her that the channel has 1 BTC in capacity.

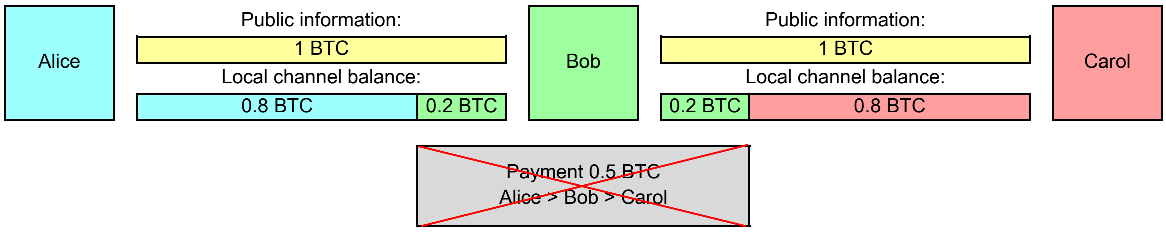

Figure 1.3: Impossible to route payment when balances are not well distributed.

A payment channel is always opened between two nodes. One of the participants acts as the initiator and provides funds for the channel in the form of bitcoin. This leads to the total capacity being allocated to his or her balance within the channel. As soon as he starts to make payments towards the other node, his or her balance decreases, and the partner's balance increases (total capacity remains constant). Transactions can only be executed if the amount is smaller or equal than the channel's capacity and if the sending node has enough local balance. Channels and their capacities are announced to the network but the distribution of balances remains private to the channel partners.

1.2.4 Source Routing

explain source routing privacy implications. onion routed. no knowledge of full route.

2 Problems in graph theory

Go from general graph theory problems into more specific areas, 1 paragraph per topic

Explain why finding a path in the lightning network can be so difficult

2.1 Overview of broad problems

one paragraph per

- coloring
- enumeration
- subgraphs
- route problems (explained more depth)
- network flow (explained more depth)

2.2 Route problems

kd

2.3 Network flow

kdf

2.4 Application in the Lightning Network

problem to be solved in ln is network flow: find route that accommodates enough liquidity without knowing the balances. In experiment algorithm does rebalancing by trying to reallocate funds differently (greedy)

Nodes trying to find a path in the Lightning Network work with limited information. While they know what channels are available and what their capacities are, they do not know about the balances and therefore whether the nodes can forward their payment or not. Hence, it is likely that a payment attempt fails because a node had insufficient balance. The paying node needs to find another route and retry the payment until it succeeds. If the payment fails repeatedly it can cause delays that are bad for the user experience.

2.5 Problem statement

Only give a brief informal overview of the problem but forward reference to the next chapter

These payment channel networks are decentralized by nature and no protocol change can be forced upon the node operators. Therefore, the question arises on how effective this protocol change will be assuming only partial participation of nodes. What are the effects of different levels of participation on the imbalance measure [1] of the network during repeated rebalancing cycles? What is the effect of different levels of participation on the network's ability to route payments between random nodes?

[1] Defined as the inequality of the distribution of a nodes channel balance coefficients

2.6 Previous work

René Pickhardt's and Mariusz Nowostawski's publication "Imbalance measure and proactive channel rebalancing algorithm for the Lightning Network" Pickhardt and Nowostawski, 2019 serves as a base to formulate the question for this thesis. In their work, they present a solution for the pathfinding problem in a privacy-aware payment channel network. The proposed solution includes a rebalancing protocol which the nodes of the network should follow to achieve a higher balancedness (for itself but also the entire network). It consists of instructions to proactively rebalance their channels within their friend of a friend's network, redistributing the relative funds owned in a channel but leaving total funds owned unchanged.

Rebalancing is an activity where one node engages in a circular payment that pays itself. This is only possible when the node has at least two channels with different peers. The payment gets routed **out** through one channel and is **received back** over another. On the way, it can use one or more hops to find back to the sender node. This procedure enables a node to change the balances of the individual channels while the total node balance stays the same. In practice, there would be a fee collected by the intermediate nodes whose channels are used. In the proposed rebalancing protocol nodes would forego the fee and only participate in the rebalancing attempt if their balancedness improves as well.

3 Experiment

explain all the performance measures

explain methodology

explain the experimental setup

3.1 Preprocessing

explain which nodes and channels where selected

3.2 Lightning properties

3.3 Implementing routing protocol

Describe how the proposed protocol change is impolemented in the **Network** class.

show illustrations with a dummy network

ask Rene to his dummy network

3.4 Methodology

explain how to get the network information for a running node. quickly show how I extracted the data from the full node. Explain the structure of the python class **Network**

. And some more **bold text**.

To model the network, public information from the Lightning Network is used. From a Lightning node, all the channel and node information can be extracted.

For all further manipulations and calculations, the programming language Python will be used. This includes writing code that facilitates:

- The selection of nodes, participating in the protocol change.
- Implement the proposed algorithm Pickhardt and Nowostawski, 2019, p. 3.
- Performing rebalancing in the network.
- Storing different network states for different scenarios.
- Calculate different performance measures.
- Aggregate data.
- Plot graphs to visualize the results.

4 Results

5 Conclusion & Outlook

References

Back, A. (2002, August 1). *Hashcash - a denial of service counter-measure.* Retrieved from http://cypherspace.org/hashcash/hashcash.pdf

blocks-size. (n.d.). Retrieved July 6, 2020, from https://www.blockchain.com/charts/blocks-size

Chaum, D. (1983). Blind signatures for untraceable payments. In D. Chaum, R. L. Rivest, & A. T. Sherman (Eds.), *Advances in cryptology* (pp. 199–203). doi:10.1007/978-1-4757-0602-4_18

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, *3*(2), 99–111. doi:10.1007/BF00196791

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system, 9. Retrieved June 7, 2020, from https://bitcoin.org/bitcoin.pdf

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies.*

Pickhardt, R., & Nowostawski, M. (2019). *Imbalance measure and proactive channel rebalancing algorithm for the Lightning Network.*

Poon, J., & Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments, 59.

Glossary

Glossary

double spend problem Problem in digital cash systems that a digital token can be duplicated at will and used as payment to multiple receivers at the same time making it difficult to detect the fraud.. 1

A Some appendix

maybe. . .