Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Micropayment Channels Game

Master's Thesis

Rolf Scheuner

schrolf@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Zeta Avarikioti, Tejaswi Nadahalli
Prof. Dr. Roger Wattenhofer

May 19, 2019

Acknowledgements

I thank my supervisors Zeta Avarikioti and Tejaswi Nadahalli for their great support during my master thesis. Whenever I stuck with a problem they helped me out with a different point of view. In each meeting they shared their motivation and inspiration with me.

Abstract

The Bitcoin Lightning network is the most widely known micropayment network built on a blockchain. In this thesis we investigate the incentives of the nodes in general a micropayment network. We use two different game models: a simultaneous game model and a sequential game model. For the simultaneous model we evaluate different strategy combinations for different payment scenarios. We show that a star is a social optimum as well as a Nash equilibrium (NE) for a homogeneous payment scenario. For both models we make some statements about general NEs.

To set the results from the theoretic analyses of the two models into some context, we simulate a micropayment network using different payment scenarios and capital distributions as well as a simple, short-term-greedy policy for the nodes. This simulation confirms the (unproved) assumption from the theoretic part that NEs highly depend on the payment scenario. We present a simulation framework which is modular and can easily be fed with more complex scenarios and different policies for the nodes.

Finally we discuss our results and their applicability to the real world and we compare them to other works in the field of micropayment networks. We also present a wide range of approaches for future work.

Keywords: blockchain, cryptocurrency, layer 2, lightning protocol, game theory, Nash equilibrium

Contents

Acknowledgements **i**

Abstract **ii**

1 Introduction **1**

1.1 Motivation 1

1.2 Background 2

2 Preliminaries **5**

2.1 General Assumptions 8

3 Simultaneous Game Model **9**

3.1 Arbitrary Payment Scenario 11

3.2 Homogeneous Payment Scenario 13

3.2.1 Strategy Combination: Path 13

3.2.2 Strategy Combination: Star 13

3.2.3 Strategy Combination: Star with 2 Centers 14

3.2.4 Strategy Combination: Complete bipartite Graph 18

3.2.5 Strategy Combination: Clique 27

3.2.6 Conclusion 29

3.3 The Fee Game 29

3.3.1 Network Structure: Complete Bipartite Graph 29

3.3.2 Network Structure: Star 30

3.3.3 Conclusion 30

3.4 Producers and Consumers 31

3.4.1 Strategy Combination: Producer Clique 31

3.4.2 Conclusion 32

3.5 Local Subgroups 32

3.5.1 Strategy Combination: Tree of Height 2 33
3.6 Conclusion 34

4 Sequential Game Model 35
4.1 Without Network Fee 36
4.1.1 Strategy Combination: Connect to unreachable Payee . . 36
4.1.2 Conclusion 37
4.2 Global Fee 37
4.2.1 Strategy Combination: Connect to unreachable Payee . . 37
4.3 General Nash Equilibrium 38
4.4 Conclusion 40

5 Simulation 41
5.1 Network Model 41
5.2 Policy 41
5.3 Network Types 42
5.4 Payment Scenarios 42
5.5 Simulation Results 44

6 Related Work 55

7 Conclusion 57
7.1 Future Work 58

Bibliography 59

A Policy Variation Comparison A-1

B Resulting Network Graphs B-1

List of Figures

1.1 Schematic illustration of a payment routed through the Lightning network (Source: `https://lightning.network/lightning-network-summary.pdf`) . . . 4

3.1 Plot of the bounds for $N = 10^3$ with the upper bounds in red and the lower bounds in blue . . . 26

3.2 Zoomed in plot of the bounds for $N = 10^3$ with the upper bounds in red and the lower bounds in blue . . . 27

5.1 The start and resulting network graph for the simulation with the RND_1 network and the STL_1 scenario. . . . 45

5.2 Evolution number of channels in the network for the simulation with of STL scenarios. . . . 46

5.3 Number of channel creations in the network over time for the simulation of STL scenarios. . . . 46

5.4 Number of channel topups in the network over time for the simulation of STL scenarios. . . . 47

5.5 Evolution of correlation between node degree and initial capital over time for the simulation of STL scenarios. . . . 47

5.6 The start and resulting network graph for the simulation with the CP_1 network and BCP_1 scenario. . . . 48

5.7 Number of channels in the network over time for the simulation of BCP scenarios. . . . 49

5.8 Number of channel creations in the network over time for the simulation of BCP scenarios. . . . 49

5.9 Number of channel topups in the network over time for the simulation of BCP scenarios. . . . 50

5.10 Evolution of correlation between node degree and initial capital over time for the simulation of BCP scenarios. . . . 50

5.11 The start and resulting network graph for the simulation with the CP_1 network and the UCP_1 scenario. . . . 51

5.12 Number of channels in the network over time for the simulation with of UCP scenarios. 52
5.13 Number of channel creations in the network over time for the simulation of UCP scenarios. 52
5.14 Number of channel topups in the network over time for the simulation of UCP scenarios. 53
5.15 The correlation between node degree and initial capital over time for the simulation with of UCP scenarios. 53

B.1 The resulting network graph of the simulation with RND_1 and STL_1. . . . . B-2
B.2 The resulting network graph of the simulation with RND_1 and STL_2. . . . . B-3
B.3 The resulting network graph of the simulation with RND_1 and STL_3. . . . . B-4
B.4 The resulting network graph of the simulation with RND_1 and STL_4. . . . . B-5
B.5 The resulting network graph of the simulation with RND_1 and STL_5. . . . . B-6
B.6 The resulting network graph of the simulation with CP_1 and BCP_1. . . . . B-7
B.7 The resulting network graph of the simulation with CP_1 and BCP_2. . . . . B-8
B.8 The resulting network graph of the simulation with CP_1 and BCP_3. . . . . B-9
B.9 The resulting network graph of the simulation with CP_1 and BCP_4. . . . . B-10
B.10 The resulting network graph of the simulation with CP_1 and BCP_5. . . . . B-11
B.11 The resulting network graph of the simulation with CP_1 and UCP_1. . . . . B-12
B.12 The resulting network graph of the simulation with CP_1 and UCP_2. . . . . B-13
B.13 The resulting network graph of the simulation with CP_1 and UCP_3. . . . . B-14
B.14 The resulting network graph of the simulation with CP_1 and UCP_4. . . . . B-15

B.15 The resulting network graph of the simulation with CP_1 and UCP_5. . . . . B-16

List of Tables

3.1 Numerical results for the lower and bounds for a bipartite graph as a Nash equilibrium 25

5.1 Simulation results including the used network, payment scenario, the number of channels at the end of the simulation, the number of channel creations, the number of channel closings, the number of channel topups, the number of blockchain transactions 44

5.2 Simulation results averaged over the payment scenario type, including the network, payment scenario type, the number of channels at the end of the simulation, the number of channel creations, the number of channel closings, the number of channel topups, the number of blockchain transactions 45

A.1 Simulation Results including the used network, payment scenario, policy, the number of channels at the end of the simulation, the number of channel creations, the number of channel closings, the number of channel topups, the number of blockchain transactions A-1

Chapter 1

Introduction

1.1 Motivation

The recent raise of cryptocurrencies like Bitcoin [1] and Ethereum [2] also gave raise to their underlying technology: the blockchain. Many cryptocurrencies use blockchains as a distributed ledger. E.g. Bitcoin uses a blockchain which is replicated and stored by every Bitcoin node. To ensure all nodes agree on the same state of the blockchain, Nakamoto consensus mechanism is used [1].

When blockchains are used together with Nakamoto consensus (or similar), they have one big drawback: they scale very badly in terms of throughput, because every node who participates in this consensus mechanism needs to store the whole blockchain. E.g. Bitcoin can handle seven transactions per second [3] and Ethereum is able to handle up to 15 transactions per second [4], while payment systems like Visa handle thousands of payments per second.

The most prominent solution for this problem are payment channels. A payment channel between two nodes allows them to do instant payments between each other without committing a transaction to the blockchain for each payment. The only transactions committed to the blockchain are the one for creating a channel and the one for closing it. The blockchain is used as an arbiter in case of dispute. When many of these channels are created, they build a micropayment network on which payments can be routed through multiple hops. The nodes can ask fees for forwarding payments through their channels. Protocols like the Lighting protocol [5] for Bitcoin or the Raiden protocol [6] for Ethereum are able to provide this functionality in a trust-free way.

The nodes try to optimize multiple objectives: minimizing the number of blockchain transactions for channel creation and closing, minimizing the fees paid for sending payments, minimizing the number of on-chain payments, and maximizing the fees earned from forwarding payments. Since these are all monetary values, we will that assume the nodes try to minimize the sum of these costs. Nash equilibria and the Price of Anarchy in such systems are still not well studied yet. Therefore, we will define different game models and analyse them using game theoretic tools.

1.2 Background

This section gives some background on the fundamentals on which this thesis builds on. The reader is assumed to have a basic understanding how a blockchain works. We show why blockchains have a scaling problem and how payment channels can solve it. We explain micropayment networks and how they work. Further we give (non-mathematical) definitions of the game theoretic terms we use in this thesis.

Blockchain and Scalability. The blockchain technology serves as basis for various cryptocurrencies. However, as the cryptocurrency systems become more major, the blockchain turns out to be a severe bottleneck of cryptocurrencies which use it as distributed ledger. Current cryptocurrency systems are only capable of handling a handful of transactions per second (around seven for Bitcoin and 15 for Ethereum [3, 4]). Since the usual approach is to use one blockchain transaction to store and validate one payment, the number of payments per second is very limited. Nakamoto consensus mechanism (and most similar consensus mechanisms) requires every node to store and validate the whole blockchain. Increasing the block size or decreasing the block time will lead to a faster growing blockchain and therefore require the nodes to handle more data. This would lead to increasing centralization of the Bitcoin network and contradict the fundamental idea of Bitcoin. There are various proposed solutions to make faster blockchains [7, 8, 9, 10]. Most of them propose an approach with a different consensus mechanism which does not require all nodes to store the whole blockchain.

Payment Channels. Payment channels are a fundamentally different approach to handle the scalability problem. They aim to move the payment load off-chain, while blockchain transactions should only be used to fund payment channels and to close them (in collaboration or in case of dispute). The blockchain will act as fail-save or arbiter in case of fraud or dispute. There are different proposed approaches [5, 11], but they all share the same idea: the involved parties create a common account on the blockchain who's state can only be altered if both parities agree (Note: this could also be a common account of more than two nodes). The parties lock up a certain amount of funds in a channel, when creating it. The state of the channel is represented as a transaction giving each of the parties his part of the funds from the channel.

Payment channels enable the involved parties to exchange funds (make payments) without using the blockchain. This is done by creating, signing and exchanging transactions which represent the new state of the channel. However, these transactions are only committed to the blockchain if one of the parties disagree with the new state of the channel. This way, payment channels provide instant finality, i.e., the payments are executed almost instantly, in contrast with most blockchain protocols which have a relatively high confirmation time (e.g., for Bitcoin the confirmation time is six blocks, thus approximately one hour). Nodes can offer their channels to other nodes, the other nodes can then route their payments through these channels. The node who offers the channel asks a fee from the sender of the payment. With this mechanism the payment channels build a micropayment network, which can be used to send payments instantly, off-chain. Protocols describing such networks are also called Layer 2 protocols; they build on top of the consensus layer of the respective blockchain.

The Lightning protocol [5] describes a micropayment network (the Lightning network) on top of the Bitcoin blockchain. Currently, the Lightning network is the largest and most major micropayment network in the world of cryptocurrencies. At the time of writing, there are over 8000 participating nodes, over 35000 open channels and about 7 million dollar network capacity [12]. Nevertheless, the Lightning network is still in a very early state, far from being used in a commercial way. Channels are implemented as 2-of-2 signature address which means a channel can only be opened, closed or altered using a transaction signed by both participants. In the case one participant becomes unresponsive, the other participant can use a previously signed transaction to close the channel. He will have to wait a predefined dispute period. When closing a channel, one transaction containing the current state of the channel (and therefore the sum of all payments done on this channel) is committed to the blockchain. This greatly reduces the load on the blockchain [5].

Game Theory. In this thesis we use game theoretic tools to analyze and discuss the incentives of the nodes in a micropayment network. In a game we have a set of rational/selfish players. Every player chooses a *strategy*, which describes what the player does in every possible situation he could run into. It can be deterministic or randomized. The set containing all possible deterministic (*pure*) strategies of a player is called *strategy set*. A *mixed* strategy describes the strategy to randomly choose a pure strategy from the strategy set with certain probabilities. A *strategy combination* is set a containing a strategy for every player in the game. A strategy combination is called a *Nash equilibrium (NE)*, if no player could decrease his cost or increase his profit by unilaterally choosing a different strategy. A NE is called *pure* if it contains only pure strategies, otherwise it is called *mixed*. Furthermore, we study the *Price of Anarchy (PoA)* which is defined as the ratio between the social cost in the worst NE and the social cost in the social optimum. Note that the PoA depends on the definition of the social cost. We use the sum of the cost of all players as the social cost.

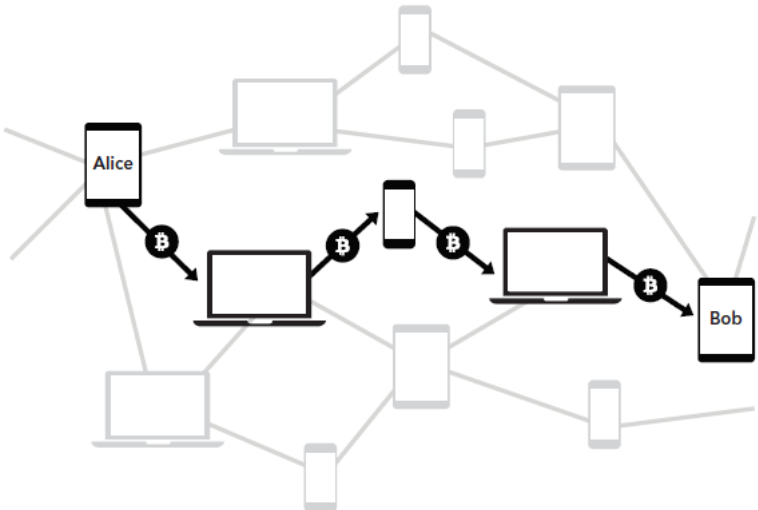

Figure 1.1: Schematic illustration of a payment routed through the Lightning network
(Source: https://lightning.network/lightning-network-summary.pdf)

Chapter 2

Preliminaries

In this chapter we introduce basic notations and make some definitions which will be used in the rest of thesis.

Definition 2.1 (Blockchain Fee)**.** The *blockchain fee* is the fee to commit a transaction to the blockchain. It is denoted as

$$F_B \in \mathbb{R}^+$$

It is paid for doing on-chain payments as well as for channel opening and closing. We assume the fee is always paid by the node who wants to open or close a channel.
Note: In the Lightning network, the fee is already included in the transaction which describes the state of the channel. However, our definition allows easier game theoretic analysis.

In the Bitcoin network, the blockchain fee corresponds to the transaction fee which is paid to the miners.

Definition 2.2 (Set of Nodes)**.** The *set of nodes* contains all nodes in the network and is denoted as

$$\boldsymbol{N} \text{ with cardinality (number of nodes) } N = |\boldsymbol{N}|$$

In general we assume $N > 3$.

Definition 2.3 (Set of Payments)**.** The *set of payments* is denoted as

$$\boldsymbol{P} \text{ with cardinality (number of payments) } P = |\boldsymbol{P}|$$

A concrete set of payments is also call payment scenario.
The time at which a payment is executed is denoted as

$$t(p) \in \mathbb{N} \text{ (discrete time)}$$

Definition 2.4 (Delegated Payment)**.** *Delegated payments* are used to model how a payment is routed through the network. A delegated payment is a payment from a node (hop) of a route to the next hop of that route through a channel between these two hops. All delegated payments of a route are coupled together, so either all of them are executed or none of them. The sender of the original payment (delegator) pays a fee to the hops (network fee) to use their channel.

Definition 2.5 (Sender and Receiver)**.** For a payment $p \in \boldsymbol{P}$ *sender* and *receiver* are written as

$$n_S(p) := \text{ sender/payer of } p$$

$$n_R(p) := \text{ receiver/payee of } p$$

and for a delegated payment d they are denoted as

$$h_S(d) := \text{ sender (hop) of } d$$

$$h_R(d) := \text{ receiver (hop) of } d$$

Definition 2.6 (Network State)**.** The *state of the network* at time $k \in \mathbb{N}_0$ (discrete time) is written as

$$x_k$$

It depends on the strategy combination of the nodes. However, we do not write this dependency explicitly to keep a good readability. Whenever we write x_k we mean the network state that resulted from the nodes applying their strategy until time k.
The initial network state is denoted as

$$x_0$$

If not stated differently we start with an empty network (no channels, just nodes).

Definition 2.7 (Route)**.** The *route* of a payment $p \in \boldsymbol{P}$ on the network in state x_k is denoted as

$$\boldsymbol{R}(x_k, p)$$

This returns a set of delegated payments. It always returns the cheapest route for $n_S(p)$. However, if there is no route cheaper than F_B, it returns an empty set and the payment has to be done on-chain.
Note: For the calculation of the cheapest route, the fee of the first delegated payment is not considered because the sender would pay that fee to himself. However, the first delegated payment is still in the returned set.

Definition 2.8 (Forwarding Fee)**.** The fee node $s \in \boldsymbol{N}$ asks for forwarding a payment to node $r \in \boldsymbol{N}$ is denoted as

$$f_d(x, s, r)$$

Definition 2.9 (Payment Fee)**.** The fee node $s \in \boldsymbol{N}$ has to pay for a payment to node $r \in \boldsymbol{N}$ if the network is in state x_k is defined as

$$f(x_k, s, r) := \begin{cases} \sum_{d \in \boldsymbol{R}(x_k, p) : h_S(d) \neq s} f_d(h_s(d), h_R(d)), & \boldsymbol{R}(x_k, p) \neq \emptyset \\ F_B, & \boldsymbol{R}(x_k, p) = \emptyset \end{cases}$$

Corollary 2.10 (Payment Fee Upper bound)**.**

$$f(x_k, n_S(p), n_R(p)) \leq F_B \forall x_k \forall p \in \boldsymbol{P}$$

Proof. By definition 2.7 and 2.9. □

Definition 2.11 (Revenue)**.** The *revenue* a node $n \in \boldsymbol{N}$ gets from payment $p \in \boldsymbol{P}$ when it is executed on the network in state x_k defined as

$$r(x_k, n, p) := \sum_{d \in \boldsymbol{R}(x_k, p) : h_S(d) = n \wedge n_S(p) \neq n} f_d(x_k, h_S(d), h_R(d))$$

Definition 2.12 (Action Set)**.** The *action set* of a node $n \in \boldsymbol{N}$ is the set of all actions available to a node. It is written as

$$\boldsymbol{A}_n$$

Definition 2.13 (Strategy Set)**.** The *strategy set* of a node $n \in \boldsymbol{N}$ is the set of all strategies available to node n. It is denoted as

$$\boldsymbol{S}_n$$

The strategy of node n is denoted as

$$\mu_n \in \boldsymbol{S}_n$$

Definition 2.14 (Strategy Combination)**.** A *strategy combination* is a set containing a strategy for every node. The set of all possible strategy combinations is defined as (using the Pi notation for Cartesian product)

$$\boldsymbol{S}^N := \prod_{n \in \boldsymbol{N}} \boldsymbol{S}_n$$

A strategy combination is denoted as

$$\boldsymbol{\mu} \in \boldsymbol{S}^N$$

Definition 2.15 (Number of Channels)**.** The *number of channels* created by a node $n \in \boldsymbol{N}$ is denoted as

$$n_{C,n}(\boldsymbol{\mu}, \mu_n)$$

The total number of channels created is

$$n_C(\boldsymbol{\mu}) = \sum_{n \in \boldsymbol{N}} n_{C,n}(\boldsymbol{\mu}, \mu_n)$$

Definition 2.16 (Number of on-chain Payments)**.** The *number of on-chain payments* by node $n \in \boldsymbol{N}$ is defined as

$$n_{OCP,n}(\boldsymbol{\mu}, \mu_n) := |\{p \in \boldsymbol{P} : n_S(p) = n \wedge \boldsymbol{R}(x_{t(p)}, p) = \emptyset\}|$$

The total number of on-chain payments is

$$n_{OCP}(\boldsymbol{\mu}) = \sum_{n \in \boldsymbol{N}} n_{OCP,n}(\boldsymbol{\mu}, \mu_n) = |\{p \in \boldsymbol{P} : \boldsymbol{R}(x_{t(p)}, p) = \emptyset\}|$$

2.1 General Assumptions

In this section we make some general assumptions. They are valid through the whole thesis if not stated differently.

Players. If not stated differently each node has unlimited capital available. Because nodes have unlimited capital, they will only create channels with unlimited funds. Channels with unlimited funds will never become unbalanced. Therefore, a node does not care how big a payment is, which he forwards. This leads to the situation that nodes will set constant fees on their channels (the fee does not depend on the amount of money of a forwarded payment). All the calculations will then become independent of the amount of money sent in a payment.

Information Model. Each node knows the complete payment scenario. They also know all channels and the fees on these channels.

CHAPTER 3

Simultaneous Game Model

In this game model we assume simultaneous gameplay. First, all nodes choose and apply their strategy. The network state becomes x_1. Then nodes must execute their payments on the network in state x_1.

Action Set. An action of a node n consists of choosing a subset of $\boldsymbol{N} \setminus n$, creating channels to these nodes and defining a constant, non-negative fee for each channel. The action set of a node n is written as

$$\boldsymbol{A}_n = (\mathbb{R}_0^+)^{N-1}, 0 \text{ means no channel is created to that node}$$

The action set might be restricted for some analyses.

Strategy Set. Since we have a simultaneous game with just one round, the strategy set of a node n is equal to his action set

$$\boldsymbol{S}_n = \boldsymbol{A}_n = (\mathbb{R}_0^+)^{N-1}$$

The set of all possible strategy combinations is

$$\boldsymbol{S}^N = \prod_{n \in \boldsymbol{N}} \boldsymbol{S}_n = (\mathbb{R}_0^+)^{N^2-N}$$

Note: When the action set is restricted in an analysis, the strategy sets of the nodes also change accordingly.

Cost Function. The cost function contains the cost for channel creation, on-chain payments, payments routed through the network and the earnings from forwarding payments. For a node n it is defined as

$$c_n(\boldsymbol{\mu}, \mu_n) = n_{C,n}(\boldsymbol{\mu}, \mu_n) \times F_B$$
$$+ \sum_{p \in \boldsymbol{P}: n_S(p)=n \wedge \boldsymbol{R}(x_1,p)=\emptyset} F_B$$

$$+ \sum_{p \in \boldsymbol{P} : n_S(p) = n} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) \neq n} f_d(x_1, h_S(d), h_R(d))$$

$$- \sum_{p \in \boldsymbol{P} : n_S(p) \neq n} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) = n} f_d(x_1, h_S(d), h_R(d))$$

Social Cost. The social cost (or negative welfare) is the sum of the costs of all nodes

$$-W = \sum_{n \in \boldsymbol{N}} c_n(\boldsymbol{\mu}, \mu_n)$$

$$= \sum_{n \in \boldsymbol{N}} (n_{C,n}(\boldsymbol{\mu}, \mu_n) \times F_B + \sum_{p \in \boldsymbol{P} : n_S(p) = n \wedge \boldsymbol{R}(x_1, p) = \emptyset} F_B$$

$$+ \sum_{p \in \boldsymbol{P} : n_S(p) = n} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) \neq n} f_d(x_1, h_S(d), h_R(d))$$

$$- \sum_{p \in \boldsymbol{P} : n_S(p) \neq n} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) = n} f_d(x_1, h_S(d), h_R(d)))$$

$$= \sum_{n \in \boldsymbol{N}} (n_{C,n}(\boldsymbol{\mu}, \mu_n) \times F_B) + \sum_{n \in \boldsymbol{N}} \; \sum_{p \in \boldsymbol{P} : n_S(p) = n \wedge \boldsymbol{R}(x_1, p) = \emptyset} F_B$$

$$+ \sum_{n \in \boldsymbol{N}} \; \sum_{p \in \boldsymbol{P} : n_S(p) = n} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) \neq n} f_d(x_1, h_S(d), h_R(d))$$

$$- \sum_{n \in \boldsymbol{N}} \; \sum_{p \in \boldsymbol{P} : n_S(p) \neq n} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) = n} f_d(x_1, h_S(d), h_R(d))$$

$$= F_B \times \sum_{n \in \boldsymbol{N}} n_{C,n}(\boldsymbol{\mu}, \mu_n) + F_B \times \sum_{p \in \boldsymbol{P} : \boldsymbol{R}(x_1, p) = \emptyset} 1$$

$$+ \sum_{p \in \boldsymbol{P}} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) \neq n_S(p)} f_d(x_1, h_S(d), h_R(d))$$

$$- \sum_{p \in \boldsymbol{P}} \; \sum_{d \in \boldsymbol{R}(x_1, p) : h_S(d) \neq n_S(p)} f_d(x_1, h_S(d), h_R(d))$$

$$= (n_C(\boldsymbol{\mu}) + n_{OCP}(\boldsymbol{\mu})) \times F_B$$

Social Optimum. The social optimum is the minimum of the social cost. As we see from the social cost above, it depends on the number of channels created and the number of payments executed on-chain. If we assume the payment traffic to span a global network, the social optimum is an arbitrary spanning tree. We assume all payments will be executed off-chain, because it costs the same amount to create a channel and to do an on-chain payment. The value of the social optimum is

$$\begin{aligned} &\min(-W) \\ &= \min_{\boldsymbol{\mu} \in \boldsymbol{S}^N} ((n_C(\boldsymbol{\mu}) + n_{OCP}(\boldsymbol{\mu})) \times F_B) \\ &= (N-1) \times F_B \end{aligned}$$

3.1 Arbitrary Payment Scenario

In this section, we attempt to find a Nash equilibrium for an arbitrary set of payments $\boldsymbol{P}$.

Action Set and Strategy Set. The nodes have to stick to a globally constant fee $f_0 \in \mathbb{R}_0^+$ on all of their channels. An action of a node n can therefore be represented as a subset of $\boldsymbol{N} \setminus \{n\}$. The action set of node n is the powerset of $\boldsymbol{N} \setminus \{n\}$, denoted as

$$\boldsymbol{A}_n = 2^{\boldsymbol{N} \setminus \{n\}}$$

As stated in the game model the strategy set is equal to the action set

$$\boldsymbol{S}_n = 2^{\boldsymbol{N} \setminus \{n\}}$$

The set of all possible strategy combinations is

$$\boldsymbol{S}^N = \prod_{n \in \boldsymbol{N}} 2^{\boldsymbol{N} \setminus \{n\}}$$

The payoff matrix for each node becomes an N-dimensional tensor with 2^{N-1} strategies per dimension.

Strategy Combination: Connect to all Payees. In this case, we only analyze one strategy combination. Specifically, we focus on the case where all nodes connect to all their payees.

The proposed strategy combination is not a NE, because their might be a cheap route for a payment, but the payer connects to his payee. The competitive ratio for a node to play this strategy is infinite. For example with following payment scenario: one node wants to pay every other node and all other nodes pay him. For this node the cost is $N \times F_B$ while his optimal cost would be zero.

With the payoff tensor we can (in theory) calculate the mixed NE where the nodes make each other indifferent between their strategies. A node chooses each of his 2^{N-1} strategies with a certain probability. The sum of all of these probabilities must be 1. Therefore he has $2^{N-1} - 1$ probabilities to calculate. However, since his probabilities depend on the probabilities of the other nodes, he has to calculate also the probabilities of all other nodes. In total there are $N \times (2^{N-1} - 1)$ probabilities to calculate and equally many equations to do so. The number of nodes (N) of the Lightning network is at the time of writing over 8000 [12]. This leads to an enormous number of equations. Nodes are not able to calculate such a number of equations in practice.

Even if we remove the option from the payment scenario that nodes can pay each other back, the provided payment scenario is not a NE. To demonstrate this, suppose a node pays a payee only few times (extreme case: one time). With this strategy, he creates a channel to this node even if there might have been a cheap route through the network to this node. For channel creation he will always pay F_B, the best he could possibly do is to pay f_0. Therefore, the strategies in this strategy combination have a competitive ratio of $\frac{F_B}{f_0}$.

Conclusion. We have illustrated one strategy combination which is not a NE. We tried many more, but for all of them we could easily find a payment scenario for which the suggested strategy combination was not a NE. It seems that for such unrestricted payment scenarios, there is no general, easy to describe NE. The presented option to calculate the mixed NE will neither find a general NE, but specific a one for a concrete payment scenario.

While trying to find a general description of a NE for an arbitrary payment scenario we observed the following:

Lemma 3.1 (No twice opened Channels at NE)**.** *In a pure Nash equilibrium there are no channels that are opened twice.*

Proof. (Towards contradiction.) Assume we have a pure NE in which a node opens a channel to another node which himself also opens a channel to the first node. Both nodes could reduce their cost by unilaterally deciding not to open this channel. This contradicts the assumption of a pure NE. □

Lemma 3.2 (No On-Chain Payments at NE)**.** *In a pure, strict Nash equilibrium, no payments are done on-chain.*

Proof. (Towards contradiction.) Assume we have a pure, strict NE in which a node sends a payment using a blockchain transaction. By changing his strategy to opening a channel to the payee of the mentioned payment, he would not spend more money. However, he might earn some fees from the additional channel. This contradicts the assumption of a strict NE. □

3.2 Homogeneous Payment Scenario

In this section, we analyse the simultaneous game for a homogeneous payment scenario. Every node makes $k_P \geq 1$ payments to every other node. The number of payments is $P = k_P \times N \times (N-1)$. The action set and the strategy set are the same as in section 3.1. For this payment scenario, we analyze multiple strategy combinations, i.e., different graph structures such as a path, a star, a complete bipartite graph and a clique, to discover under which parameters they are a NE.

3.2.1 Strategy Combination: Path

The first strategy combination we investigate is a path. Each node connects to the node with the next higher ID. The node with the highest ID does not create a channel, but he is connected to the network trough the node with the second highest ID. It is a social optimum since it is a spanning tree.

Social Cost. The social cost is $-W = (N-1) \times F_B$.

Alternative Strategies. For the proposed strategy combination to be a Nash equilibrium, none of the nodes must be able get a lower cost by unilaterally deviating from his strategy. This means all possible deviations of each node must lead to a higher cost for the deviating node.

For $f_0 = 0$ no one is interested to change anything. But for any $f_0 > 0$, e.g., the first node would have a lower cost if he connected to a node somewhere in the middle of the path and not to the second node.

Nash Equilibrium. We saw that a path can only be a NE for the special case where $f_0 = 0$. The reason is that nodes want to be closer to the center of the network to have shorter (and therefore cheaper) routes for their payments. This observation brings us to the idea of a star as a NE.

3.2.2 Strategy Combination: Star

The second strategy combination we investigate is a star: one node creates channels to everyone else, while the other nodes do not create any channels. As the path, the star is a spanning tree and therefore a social optimum. The strategy to create channels to $a \in [0, N-1]$ outer nodes is denoted as (a).

Cost Functions. The cost of the center node is $c_c(\boldsymbol{\mu}, (N-1)) = (N-1) \times F_B - (N-1) \times (N-2) \times k_P \times f_0$. The cost of the outer nodes is $c_o(\boldsymbol{\mu}, (0)) = (N-2) \times k_P \times f_0$. The social cost is $-W = (N-1) \times F_B$.

Alternative Strategies. The nodes have only one alternative strategy available (strategies covered by lemma 3.1 or lemma 3.2 not included): An outer node creates channels to $a \in [1, N-2]$ other outer nodes.

If an outer node created channels to $a \in [1, N-2]$ other outer nodes, his cost function would become

$$c_o(\boldsymbol{\mu}, (a)) = a \times F_B + (N-2-a) \times k_P \times f_0 - a \times (a-1) \times k_P \times \frac{1}{2} \times f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_o(\boldsymbol{\mu}, (0))$ for all a. Since the second derivative w.r.t a is strictly negative, we only have to check the edge cases $a = 1$ and $a = N-2$.
$a = 1$:

$$\begin{aligned} c_o(\boldsymbol{\mu}, (0)) &< c_o(\boldsymbol{\mu}, (1)) \\ \iff k_P \times f_0 \times (N-2) &< F_B + k_P \times f_0 \times (N-3) \\ \iff f_0 &< \frac{F_B}{k_P} \end{aligned}$$

$a = N-2$:

$$\begin{aligned} c_o(\boldsymbol{\mu}, (0)) &< c_o(\boldsymbol{\mu}, (N-2)) \\ \iff k_P \times f_0 \times (N-2) &< (N-2) \times F_B - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0 \\ \iff f_0 &< \frac{F_B}{k_P} - \frac{(N-3)}{2} \times f_0 \\ \iff \frac{N-1}{2} \times f_0 &< \frac{F_B}{k_P} \\ \iff f_0 &< \frac{F_B}{k_P} \times \frac{2}{N-1} \end{aligned}$$

Nash Equilibrium. Since we restricted $N > 3$ these two conditions reduce to:

$$f_0 < \frac{F_B}{k_P} \times \frac{2}{N-1}$$

The star is indeed a NE for the provided condition on f_0.

3.2.3 Strategy Combination: Star with 2 Centers

We have seen, that a star can be a Nash equilibrium, if there is a constant fee that is low enough. We also saw, that for a too high fee, other outer nodes can have a lower cost by becoming a second center of the star. This leads to the assumption that a star with two center nodes can also be a Nash equilibrium. In this section we prove our assumption. The proposed strategy combination in detail is as follows: There are two center nodes, each creating channels to all outer nodes, but not to each other. The outer nodes do not create any channels. We denote the strategy to create $a \in [0, 2]$ channels to center nodes and $b \in [0, N-2]$ channels to outer nodes as (a, b).

Cost Functions. The cost of the center nodes is

$$c_c(\boldsymbol{\mu}, (0, N-2)) = (N-2) \times F_B + k_P \times f_0 - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0$$

The cost of the outer nodes is

$$c_o(\boldsymbol{\mu}, (0, 0)) = (N-3) \times k_P \times f_0 - 2 \times k_P \times \frac{1}{(N-2)} f_0$$

The social cost is

$$-W = 2 \times (N-2) \times F_B$$

Alternative Strategies. The nodes have following alternative strategies available (strategies covered by lemma 3.1 or lemma 3.2 not included):

(Deviation A) A center node creates channels to $b \in [1, N-3]$ outer nodes.

(Deviation B) A center node creates channels to $b \in [0, N-2]$ outer nodes and creates a channel to the other center node.

(Deviation C) An outer node creates channels to $b \in [1, N-3]$ other outer nodes.

For the proposed strategy combination to be a Nash equilibrium, none of the nodes must be able get a lower cost by unilaterally choosing a different strategy, i.e., all alternative strategies of each node must lead to a higher cost for this node. Subsequently, we analyze below all the alternative strategies for this case.

(Deviation A) If a center node created channels to only $b \in [1, N-3]$ outer nodes, his cost function would become

$$c_c(\boldsymbol{\mu}, (0, b)) = b \times F_B + k_P \times f_0 + (N-2-b) \times k_P \times 2f_0 - b \times (b-1) \times k_P \times \frac{1}{2} f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_c(\boldsymbol{\mu}, (0, N-2))$ for all b. Since the second derivative w.r.t. b is strictly negative, we only have to check the edge cases $b = 1$ and $b = N-3$.

$b = 1$:

$$c_c(\boldsymbol{\mu}, (0, N-2)) < c_c(\boldsymbol{\mu}, (0, 1))$$

$$\iff (N-2) \times F_B + k_P \times f_0 - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0$$

$$
\begin{aligned}
&< F_B + k_P \times f_0 + (N-3) \times k_P \times 2f_0 \\
\iff &(N-3) \times \frac{F_B}{k_P} < (N-2) \times (N-3) \times \frac{1}{2} f_0 + (N-3) \times 2f_0 \\
\iff &\frac{F_B}{k_P} < (N-2) \times \frac{1}{2} f_0 + 2f_0 \\
\iff &f_0 > \frac{F_B}{k_P} \times \frac{2}{N+2}
\end{aligned}
$$

$b = N-3$:

$$
\begin{aligned}
&c_c(\boldsymbol{\mu}, (0, N-2)) < c_c(\boldsymbol{\mu}, (0, N-3)) \\
\iff &(N-2) \times F_B + k_P \times f_0 - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0 \\
&< (N-3) \times F_B + k_P \times f_0 + k_P \times 2f_0 - (N-3) \times (N-4) \times k_P \times \frac{1}{2} f_0 \\
\iff &\frac{F_B}{k_P} < (N-2) \times (N-3) \times \frac{1}{2} f_0 + f_0 - (N-3) \times (N-4) \times \frac{1}{2} f_0 \\
\iff &\frac{F_B}{k_P} < (2N-6) \times \frac{1}{2} f_0 + 2f_0 \\
\iff &\frac{F_B}{k_P} < (N-3) \times f_0 + 2f_0 \\
\iff &f_0 > \frac{F_B}{k_P} \times \frac{1}{N-1}
\end{aligned}
$$

(Deviation B) If an center node created a channel to the other center node and channels to $b \in [0, N-2]$ outer nodes, his cost function would become

$$
c_c(\boldsymbol{\mu}, (1, b) = (b+1) \times F_B + (N-2-b) \times k_P \times f_0 - b \times (b-1) \times k_P \times \frac{1}{2} f_0
$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_c(\boldsymbol{\mu}, (0, N-2))$ for all b. Since the second derivative w.r.t. b is strictly negative, we only have to check the edge cases $b = 0$ and $b = N-2$.

$b = 0$:

$$
\begin{aligned}
&c_c(\boldsymbol{\mu}, (0, N-2)) < c_c(\boldsymbol{\mu}, (1, 0)) \\
\iff &(N-2) \times F_B + k_P \times f_0 - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0 \\
&< F_B + (N-2) \times k_P \times f_0 \\
\iff &(N-3) \times \frac{F_B}{k_P} < (N-2) \times (N-3) \times \frac{1}{2} f_0 + (N-3) \times f_0 \\
\iff &\frac{F_B}{k_P} < (N-2) \times \frac{1}{2} f_0 + f_0
\end{aligned}
$$

$$\iff f_0 > \frac{F_B}{k_P} \times \frac{2}{N}$$

$b = N - 2$:

$$c_c(\boldsymbol{\mu}, (0, N-2)) < c_c(\boldsymbol{\mu}, (1, N-2))$$

$$\iff (N-2) \times F_B + k_P \times f_0 - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0$$

$$< (N-1) \times F_B - (N-2) \times (N-3) \times k_P \times \frac{1}{2} f_0$$

$$\iff k_P \times f_0 < F_B$$

$$\iff f_0 < \frac{F_B}{k_P}$$

(Deviation C) If an outer node created channels to $b \in [1, N-2]$ other outer nodes, his cost function would become

$$c_o(\boldsymbol{\mu}, (0, b)) = b \times F_B + (N-3-b) \times k_P \times f_0 - 2 \times k_P \times \frac{1}{(N-2)} f_0 - b \times (b-1) \times k_P \times \frac{1}{3} f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_o(\boldsymbol{\mu}, (0, 0))$ for all b. Since the second derivative w.r.t. b is strictly negative, we only have to check the edge cases $b = 1$ and $b = N - 3$.
$b = 1$:

$$c_o(\boldsymbol{\mu}, (0, 0)) < c_o(\boldsymbol{\mu}, (0, 1))$$

$$\iff (N-3) \times k_P \times f_0 - 2 \times k_P \times \frac{1}{(N-2)} f_0$$

$$< F_B + (N-4) \times k_P \times f_0 - 2 \times k_P \times \frac{1}{(N-2)} f_0$$

$$\iff (N-3) \times f_0 - (N-4) \times f_0 < \frac{F_B}{k_P}$$

$$\iff f_0 < \frac{F_B}{k_P}$$

$b = N - 3$:

$$c_o(\boldsymbol{\mu}, (0, 0)) < c_o(\boldsymbol{\mu}, (0, N-3))$$

$$\iff (N-3) \times k_P \times f_0 - 2 \times k_P \times \frac{1}{(N-2)} f_0$$

$$< (N-3) \times F_B - 2 \times k_P \times \frac{1}{(N-2)} f_0 - (N-3) \times (N-4) \times k_P \times \frac{1}{3} f_0$$

$$\iff (N-3) \times f_0 + (N-3) \times (N-4) \times \frac{1}{3} f_0 < (N-3) \times \frac{F_B}{k_P}$$

$$\iff f_0 + (N-4) \times \frac{1}{3} f_0 < \frac{F_B}{k_P}$$

$$\iff f_0 < \frac{F_B}{k_P} \times \frac{3}{N-1}$$

Nash Equilibrium. From the analysis above, we find following bounds on f_0 which must hold that a star with two centers is a NE:

- $\frac{F_B}{k_P} \times \frac{1}{N-1} < f_0$ *(lower bound)*
- $\frac{F_B}{k_P} \times \frac{2}{N+2} < f_0$ *(lower bound)*
- $\frac{F_B}{k_P} \times \frac{2}{N} < f_0$ *(lower bound)*
- $f_0 < \frac{F_B}{k_P}$ *(upper bound)*
- $f_0 < \frac{F_B}{k_P} \times \frac{3}{N-1}$ *(upper bound)*

Since we restricted $N > 3$ these conditions reduce to

$$\frac{F_B}{k_P} \times \frac{2}{N} < f_0 < \frac{F_B}{k_P} \times \frac{3}{N-1}$$

3.2.4 Strategy Combination: Complete bipartite Graph

We saw that stars with one or two center nodes can be a Nash equilibrium, if there is a constant fee that fulfills certain conditions. We also saw that for a too high f_0, outer nodes can get lower costs by creating channels to other outer nodes. This leads to the assumption that a star with an arbitrary number of center nodes could be a NE. The proposed strategy combination is as follows: $c \in [2, N/2]$ nodes build a center by creating channels to everyone else but each other. In this section we find conditions for the proposed strategy combination to be a NE. For easier reading we substitute the number of outer nodes $d := N - c$. This gives us a complete bipartite graph with c nodes in the smaller partition and d nodes in the larger partition. We denote the strategy to create channels to $a \in [0, c]$ center nodes and to $b \in [0, d]$ outer nodes as (a, b).

Cost Functions. The cost of the center nodes is

$$c_c(\boldsymbol{\mu}, (0, d)) = d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$

The cost of the outer nodes is

$$c_o(\boldsymbol{\mu}, (0, 0)) = (d-1) \times k_P \times f_0 - c \times (c-1) \times k_P \times \frac{1}{d} f_0$$

The social cost is

$$-W = c \times (N - c) \times F_B$$

Alternative Strategies. The nodes have following alternative strategies available (strategies covered by lemma 3.1 or lemma 3.2 not included):

(Deviation A) A center node creates channels to only $b \in [1, d-1]$ outer nodes.

(Deviation B) A center node creates channels to $a \in [1, c-1]$ center nodes and to $b = 0$ outer nodes.

(Deviation C) A center node creates channels to $a \in [1, c-1]$ center nodes and to $b \in [1, d]$ outer nodes.

(Deviation D) An outer node creates channels to $b \in [1, d-1]$ other outer nodes.

For the proposed strategy combination to be a Nash equilibrium, none of the nodes must be able get a lower cost by unilaterally choosing a different strategy, i.e., all alternative strategies of each node must lead to a higher cost for this node.

(Deviation A) If a center node created channels to only $b \in [1, d-1]$ outer nodes, his cost function would become

$$c_c(\boldsymbol{\mu}, (0, b)) = b \times F_B + (c-1) \times k_P \times f_0 + (d-b) \times k_P \times 2f_0 - b \times (b-1) \times k_P \times \frac{1}{c} f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_c(\boldsymbol{\mu}, (0, d))$ for all b. Since the second derivative w.r.t. b is strictly negative, we only have to check the edge cases $b = 1$ and $b = d - 1$.
$b = 1$:

$$\begin{aligned}
& c_c(\boldsymbol{\mu}, (0, d)) < c_c(\boldsymbol{\mu}, (0, 1)) \\
\iff & d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0 \\
& < F_B + (c-1) \times k_P \times f_0 + (d-1) \times k_P \times 2f_0 \\
\iff & (d-1) \times \frac{F_B}{k_P} < d \times (d-1) \times \frac{1}{c} f_0 + (d-1) \times 2f_0 \\
\iff & \frac{F_B}{k_P} < d \times \frac{1}{c} f_0 + 2f_0 \\
\iff & \frac{F_B}{k_P} < \frac{d+2c}{c} \times f_0 \\
\iff & \frac{F_B}{k_P} \times \frac{c}{d+2c} < f_0 \\
\iff & \frac{F_B}{k_P} \times \frac{c}{N+c} < f_0
\end{aligned}$$

$b = d - 1$:

$$c_c(\boldsymbol{\mu}, (0, d)) < c_c(\boldsymbol{\mu}, (0, d-1))$$

$$\iff d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$

$$< (d-1) \times F_B + (c-1) \times k_P \times f_0 + k_P \times 2f_0 - (d-1) \times (d-2) \times k_P \times \frac{1}{c} f_0$$

$$\iff \frac{F_B}{k_P} < d \times (d-1) \times \frac{1}{c} f_0 + 2f_0 - (d-1) \times (d-2) \times \frac{1}{c} f_0$$

$$\iff \frac{F_B}{k_P} < 2 \times f_0 + 2 \times (d-1) \times \frac{1}{c} \times f_0$$

$$\iff \frac{F_B}{k_P} < \frac{2c}{c} \times f_0 + \frac{2d-2}{c} \times f_0$$

$$\iff \frac{F_B}{k_P} < \frac{2d+2c-2}{c} \times f_0$$

$$\iff \frac{F_B}{k_P} \times \frac{c}{2d+2c-2} < f_0$$

$$\iff \frac{F_B}{k_P} \times \frac{c}{2N-2} < f_0$$

(Deviation B) If a center node created channels to $a \in [1, c-1]$ other center nodes and to $b \in [1, d]$ outer nodes, his cost function would become

$$c_c(\mu, (a, b)) = (a+b) \times F_B + (d-b) \times k_P \times f_0 + (c-1-a) \times k_P \times f_0$$
$$-b \times (b-1) \times k_P \times \frac{1}{c} f_0 - a \times (a-1) \times k_P \times \frac{1}{d+1} f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_c(\boldsymbol{\mu}, (0, d))$. Since the second derivatives w.r.t. a and b are strictly negative, we only have to check the edge cases $(a = 1, b = 1)$, $(a = 1, b = d)$, $(a = c-1, b = 1)$ and $(a = c-1, b = d)$.

$a = 1, b = 1$:

$$c_c(\boldsymbol{\mu}, (0, d)) < c_c(\boldsymbol{\mu}, (1, 1))$$

$$\iff d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$

$$< 2 \times F_B + (d-1) \times k_P \times f_0 + (c-2) \times k_P \times f_0$$

$$\iff (d-2) \times \frac{F_B}{k_P} < (d-c) \times f_0 + (c-2) \times f_0 + d \times (d+1) \times \frac{1}{c} f_0$$

$$\iff \frac{F_B}{k_P} < \frac{c \times (d-2) + d \times (d+1)}{c \times (d-2)} f_0$$

$$\iff \frac{F_B}{k_P} \times \frac{c \times (d-2)}{c \times (d-2) + d \times (d+1)} < f_0$$

$$\iff \frac{F_B}{k_P} \times \frac{c(N-c-2)}{c(N-c-2)+(N-c)(N-c+1)} < f_0$$
$$\iff \frac{F_B}{k_P} \times \frac{cN-c^2-2c}{N^2-cN+N-3c} < f_0$$

$a = 1, b = d$:

$$c_c(\boldsymbol{\mu}, (0,d)) < c_c(\boldsymbol{\mu}, (1,d))$$
$$\iff d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$
$$< (d+1) \times F_B + (c-2) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$
$$\iff f_0 < \frac{F_B}{k_P}$$

$a = c-1, b = 1$:

$$c_c(\boldsymbol{\mu}, (0,d)) < c_c(\boldsymbol{\mu}, (c-1,1))$$
$$d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$
$$< c \times F_B + (d-1) \times k_P \times f_0 - (c-1) \times (c-2) \times k_P \times \frac{1}{d+1} f_0$$
$$(d-c) \times \frac{F_B}{k_P} < (d-c) \times f_0 + d \times (d-1) \times \frac{1}{c} f_0 - (c-1) \times (c-2) \times \frac{1}{d+1} f_0$$
$$\frac{F_B}{k_P} < \frac{c \times (d+1) \times (d-c) + (d+1) \times d \times (d-1) - c \times (c-1) \times (c-2)}{c \times (d+1) \times (d-c)} \times f_0$$
$$\frac{F_B}{k_P} \times \frac{c \times (d+1) \times (d-c)}{c \times (d+1) \times (d-c) + (d+1) \times d \times (d-1) - c \times (c-1) \times (c-2)} < f_0$$
$$\frac{F_B}{k_P} \times \frac{c(N-c+1)(N-2c)}{c(N-c+1)(N-2c)+(N-c+1)(N-c)(N-c-1)-c(c-1)(c-2)} < f_0$$
$$\frac{F_B}{k_P} \times \frac{cN^2-3c^2N+cN+2c^3-2c^2}{N^3-2cN^2+cN-N+c^2-c} < f_0$$

$a = c-1, b = d$:

$$c_c(\boldsymbol{\mu}, (0,d)) < c_c(\boldsymbol{\mu}, (c-1,d))$$
$$\iff d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$
$$< (d+c-1) \times F_B - d \times (d-1) \times k_P \times \frac{1}{c} f_0 - (c-1) \times (c-2) \times k_P \times \frac{1}{d+1} f_0$$
$$\iff (c-1) \times f_0 + (c-1) \times (c-2) \times \frac{1}{d+1} f_0 < (c-1) \times \frac{F_B}{k_P}$$
$$\iff f_0 + \frac{c-2}{d+1} f_0 < \frac{F_B}{k_P}$$
$$\iff \frac{c+d-1}{d+1} f_0 < \frac{F_B}{k_P}$$
$$\iff f_0 < \frac{F_B}{k_P} \times \frac{N-c+1}{N-1}$$

(Deviation C) If a center node created channels to $a \in [1, c-1]$ other center nodes and to $b = 0$ outer nodes, his cost function would become

$$c_c(\boldsymbol{\mu}, (a, 0)) = a \times F_B + d \times k_P \times f_0 + (c - 1 - a) \times k_P \times 2f_0$$
$$-(a) \times (a-1) \times k_P \times \frac{1}{d+1} f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_c(\boldsymbol{\mu}, (0, d))$ for all a. Since the second derivative w.r.t. a is strictly negative, we only have to check the edge cases $a = 1$ and $a = c - 1$.
$a = 1$:

$$c_c(\boldsymbol{\mu}, (0, d)) < c_c(\boldsymbol{\mu}, (1, 0))$$
$$\iff d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$
$$< F_B + d \times k_P \times f_0 + (c-2) \times k_P \times 2f_0$$
$$\iff (d-1) \times \frac{F_B}{k_P} < (d - c + 1) \times f_0 + (c-2) \times 2f_0 + d \times (d-1) \times \frac{1}{c} f_0$$
$$\iff (d-1) \times \frac{F_B}{k_P} < (d + c - 3) \times f_0 + d \times (d-1) \times \frac{1}{c} f_0$$
$$\iff \frac{F_B}{k_P} < \frac{c \times (d + c - 3) + d \times (d-1)}{c \times (d-1)} \times f_0$$
$$\iff \frac{F_B}{k_P} \times \frac{c \times (d-1)}{c \times (d + c - 3) + d \times (d-1)} < f_0$$
$$\iff \frac{F_B}{k_P} \times \frac{c(N - c - 1)}{c(N-3) + (N-c)(N-c-1)} < f_0$$
$$\iff \frac{F_B}{k_P} \times \frac{cN - c^2 - c}{N^2 - cN - N + c^2 - 2c} < f_0$$

$a = c - 1$:

$$c_c(\boldsymbol{\mu}, (0, d)) < c_c(\boldsymbol{\mu}, (c-1, 0))$$
$$d \times F_B + (c-1) \times k_P \times f_0 - d \times (d-1) \times k_P \times \frac{1}{c} f_0$$
$$< (c-1) \times F_B + d \times k_P \times f_0 - (c-1) \times (c-2) \times k_P \times \frac{1}{d+1} f_0$$
$$(d - c + 1) \times \frac{F_B}{k_P} < (d - c + 1) \times f_0 + d \times (d-1) \times \frac{1}{c} f_0 - (c-1) \times (c-2) \times \frac{1}{d+1} f_0$$
$$\frac{F_B}{k_P} < \frac{c \times (d - c + 1) \times (d+1) + (d+1) \times d \times (d-1) - c \times (c-1) \times (c-2)}{c \times (d+1) \times (d - c + 1)} \times f_0$$
$$\frac{F_B}{k_P} \times \frac{c \times (d+1) \times (d - c + 1)}{c \times (d - c + 1) \times (d+1) + (d+1) \times d \times (d-1) - c \times (c-1) \times (c-2)} < f_0$$
$$\frac{F_B}{k_P} \times \frac{c(N - c + 1)(N - 2c + 1)}{c(N - 2c + 1)(N - c + 1) + (N - c + 1)(N - c)(N - c - 1) - c(c-1)(c-2)} < f_0$$
$$\frac{F_B}{k_P} \times \frac{cN^2 - 3c^2N + 2cN + 2c^3 - 3c^2 + c}{N^3 - 2cN^2 + 2cN - N} < f_0$$

(Deviation D) If an outer node created channels to $b \in [1, d-1]$ other outer nodes, his cost function would become

$$c_o(\boldsymbol{\mu}, (0, b)) = b \times F_B + (d-1-b) \times k_P \times f_0 - c \times (c-1) \times k_P \times \frac{1}{d} f_0 - b \times (n-1) \times k_P \times \frac{1}{c+1} f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_o(\boldsymbol{\mu}, (0, 0))$ for all b. Since the second derivative w.r.t. b is strictly negative, we only have to check the edge cases $b = 1$ and $b = d - 1$.
$b = 1$:

$$c_o(\boldsymbol{\mu}, (0, 0)) < c_o(\boldsymbol{\mu}, (0, 1))$$

$$\iff (d-1) \times k_P \times f_0 - c \times (c-1) \times k_P \times \frac{1}{d} f_0$$

$$< F_B + (d-2) \times k_P \times f_0 - c \times (c-1) \times k_P \times \frac{1}{d} f_0$$

$$\iff (d-1) \times k_P \times f_0 < F_B + (d-2) \times k_P \times f_0$$

$$\iff k_P \times f_0 < F_B$$

$$\iff f_0 < \frac{F_B}{k_P}$$

$b = d - 1$:

$$c_o(\boldsymbol{\mu}, (0, 0)) < c_o(\boldsymbol{\mu}, (0, d-1))$$

$$\iff (d-1) \times k_P \times f_0 - c \times (c-1) \times k_P \times \frac{1}{d} f_0$$

$$< (d-1) \times F_B - c \times (c-1) \times k_P \times \frac{1}{d} f_0 - (d-1) \times (d-2) \times k_P \times \frac{1}{c+1} f_0$$

$$\iff (d-1) \times k_P \times f_0 < (d-1) \times F_B - (d-1) \times (d-2) \times k_P \times \frac{1}{c+1} f_0$$

$$\iff f_0 < \frac{F_B}{k_P} - (d-2) \times \frac{1}{c+1} f_0$$

$$\iff (\frac{c+1}{c+1} + \frac{d-2}{c+1}) \times f_0 < \frac{F_B}{k_P}$$

$$\iff f_0 < \frac{F_B}{k_P} \times \frac{c+1}{N-1}$$

Nash Equilibrium. From the analysis of the alternative strategies, we derive upper and lower bounds for the value of f_0 for which this strategy combination is a Nash equilibrium:

- $f_0 > \frac{F_B}{k_P} \times \frac{c}{2N-2}$ *(lower bound 1)*
- $f_0 > \frac{F_B}{k_P} \times \frac{c}{N+c}$ *(lower bound 2)*
- $f_0 > \frac{F_B}{k_P} \times \frac{cN-c^2-2c}{N^2-cN+N-3c}$ *(lower bound 3)*
- $f_0 > \frac{cN^2-3c^2N+cN+2c^3-2c^2}{N^3-2cN^2+cN-N+c^2-c}$ *(lower bound 4)*
- $f_0 > \frac{F_B}{k_P} \times \frac{cN-c^2-c}{N^2-cN-N+c^2-2c}$ *(lower bound 5)*
- $f_0 > \frac{F_B}{k_P} \times \frac{cN^2-3c^2N+2cN+2c^3-3c^2+c}{N^3-2cN^2+2cN-N}$ *(lower bound 6)*
- $f_0 < \frac{F_B}{k_P}$ *(upper bound 1)*
- $f_0 < \frac{F_B}{k_P} \times \frac{N-c+1}{N-1}$ *(upper bound 2)*
- $f_0 < \frac{F_B}{k_P} \times \frac{c+1}{N-1}$ *(upper bound 3)*

Since we restricted $N > 3$ and $2 <= c <= {}^{N}\!/_{2}$ these conditions reduce to

$$f_0 > \frac{F_B}{k_P} \times \frac{cN - c^2 - 2c}{N^2 - cN + N - 3c}$$

$$f_0 > \frac{F_B}{k_P} \times \frac{cN - c^2 - c}{N^2 - cN - N + c^2 - 2c}$$

$$f_0 < \frac{F_B}{k_P} \times \frac{c+1}{N-1}$$

We have defined not only one Nash equilibrium in this analysis, but a whole class of NEs. Table 3.1 shows the numerical values for the bounds of a complete bipartite graph as NE. The figures 3.1 and 3.2 show a plot of the bounds for $N = 10^3$. The NEs lay in the thin area between the lowest red and the highest blue line.

N	c	lower bound $[\frac{F_B}{k_P}]$	upper bound $[\frac{F_B}{k_P}]$	active lb	active ub
10^3	2	$.2000000 \times 10^{-2}$	$.30030 \times 10^{-2}$	5	3
	3	$.2999991 \times 10^{-2}$	$.40040 \times 10^{-2}$	5	3
	5	$.4999925 \times 10^{-2}$	$.60060 \times 10^{-2}$	5	3
	10	$.9999192 \times 10^{-2}$	$.11011 \times 10^{-1}$	5	3
	100	$.9970024 \times 10^{-1}$	.10110	3	3
	499	.4975016	.50050	3	3
	500	.4984984	.50150	3	3
10^4	2	$.20000 \times 10^{-3}$	$.30003 \times 10^{-3}$	5	3
	3	$.29997 \times 10^{-3}$	$.40004 \times 10^{-3}$	5	3
	5	$.49995 \times 10^{-3}$	$.60006 \times 10^{-3}$	5	3
	10	$.99990 \times 10^{-3}$	$.11001 \times 10^{-2}$	5	3
	100	$.99981 \times 10^{-2}$	$.10101 \times 10^{-1}$	5	3
	1000	$.99971 \times 10^{-1}$	.10011	3	3
	4999	.49976	.50005	3	3
	5000	.49985	.50015	3	3
10^5	2	$.200000 \times 10^{-4}$	$.300003 \times 10^{-4}$	5	3
	3	$.299997 \times 10^{-4}$	$.400004 \times 10^{-4}$	5	3
	5	$.499995 \times 10^{-4}$	$.600006 \times 10^{-4}$	5	3
	10	$.999990 \times 10^{-4}$	$.110001 \times 10^{-3}$	5	3
	100	$.999990 \times 10^{-3}$	$.101001 \times 10^{-2}$	5	3
	1000	$.999971 \times 10^{-2}$	$.100101 \times 10^{-1}$	3	3
	10000	$.999971 \times 10^{-1}$	.100011	3	3
	49999	.499976	.500005	3	3
	50000	.499985	.500015	3	3

Table 3.1: Numerical results for the lower and bounds for a bipartite graph as a Nash equilibrium

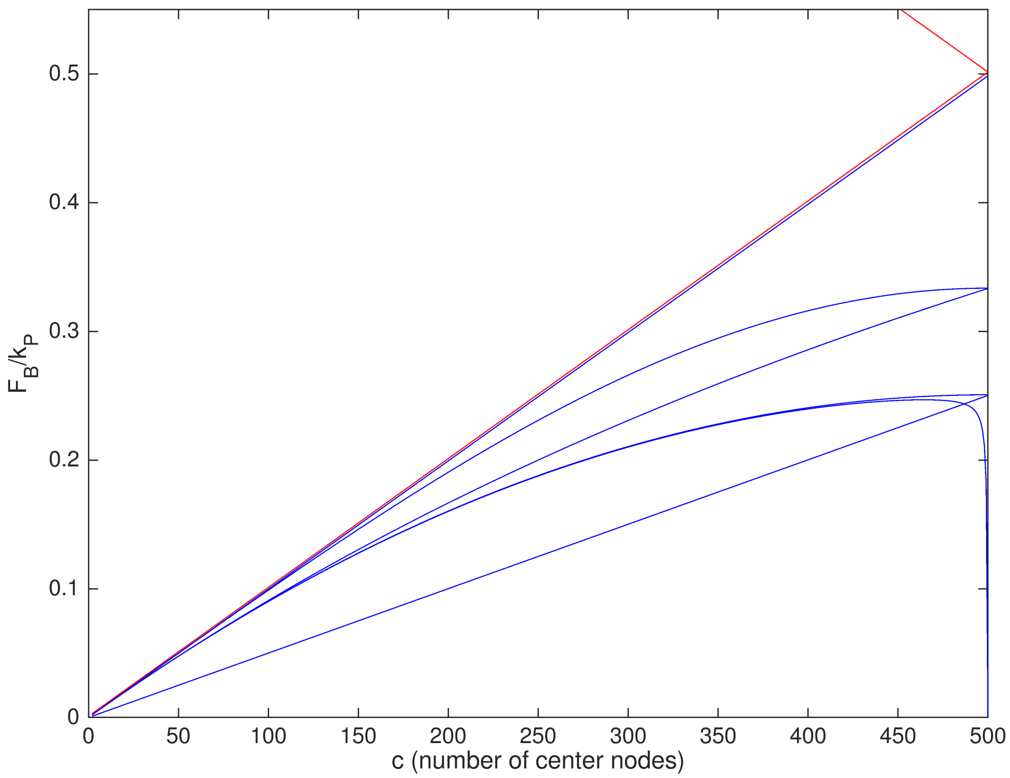

Figure 3.1: Plot of the bounds for $N = 10^3$ with the upper bounds in red and the lower bounds in blue

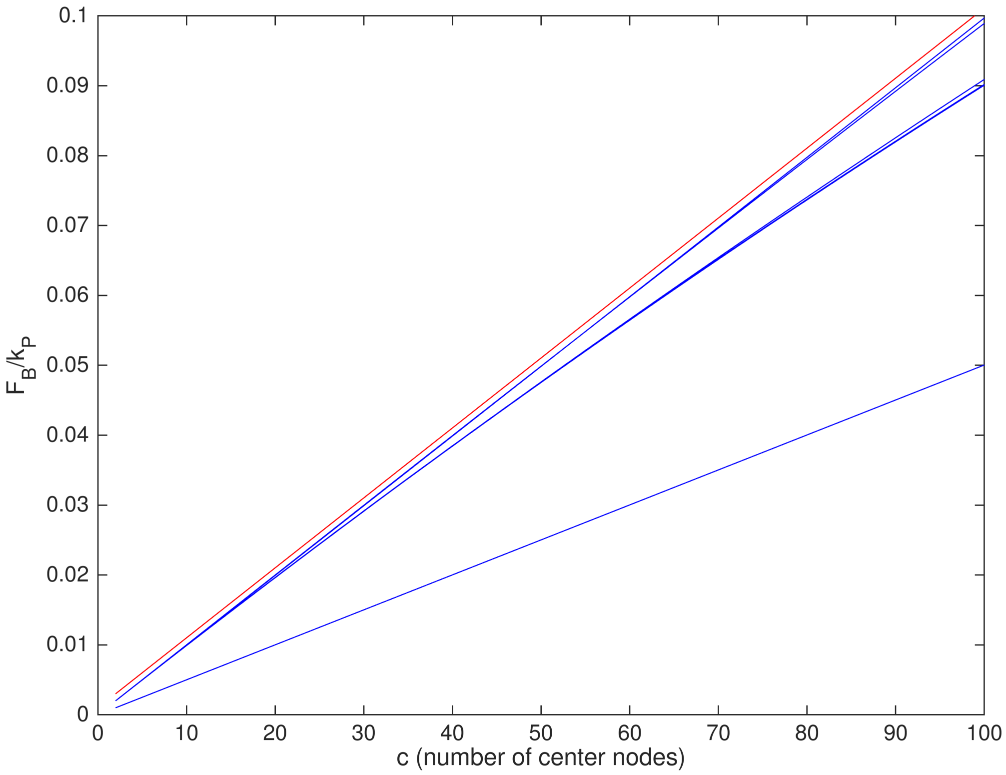

Figure 3.2: Zoomed in plot of the bounds for $N = 10^3$ with the upper bounds in red and the lower bounds in blue

3.2.5 Strategy Combination: Clique

The last network structure we want to investigate is the clique. The first node connects to all other nodes, the second one to all but the first one and so on. The i-th node opens $N - i$ channels. The strategy of creating channels to a nodes which create no channels to oneself is denoted as (a).

Cost Functions. The cost of the i-th node is $c_i(\boldsymbol{\mu}, (N - i)) = (N - i) \times F_B$. The social cost is $-W = \frac{N\times(N-1)}{2} \times F_B$.

Alternative Strategies. The nodes have following alternative strategies available (strategies covered by lemma 3.1 or lemma 3.2 not included):

(Deviation A) The first node creates channels to only $a \in [1, N - 2]$ other nodes

(Deviation B) Node i (not the first or last one) creates channels to only $a \in [0, N - i - 1]$

nodes from the set of nodes he would originally connect to (node $i+1$ to node N).

For the proposed strategy combination to be a Nash equilibrium, none of the nodes must be able get a lower cost by unilaterally choosing a different strategy, i.e., all alternative strategies of each node must lead to a higher cost for this node.

Deviation A If the first node created channels to $a \in [1, N-2]$ other nodes his cost function would become

$$c_1(\boldsymbol{\mu}, (a)) = a \times F_B + (N-1-a) \times k_P \times f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_1(\boldsymbol{\mu}, (N-1))$ for all a.

$$\begin{aligned} c_1(\boldsymbol{\mu}, (N-1)) &< c_1(\boldsymbol{\mu}, (a)) \\ \iff (N-1) \times F_B &< a \times F_B + (N-1-a) \times k_P \times f_0 \\ \iff (N-1-a) \times F_B &< (N-1-a) \times k_P \times f_0 \\ \iff f_0 &> \frac{F_B}{k_P} \end{aligned}$$

Deviation B If node i (not the first or last one) created channels to $a \in [0, N-i-1]$ nodes from the set of nodes he would originally connect to (node $i+1$ to node N), his cost functions would become

$$c_i(\boldsymbol{\mu}, (a)) = a \times F_B + (N-i-a) \times k_P \times f_0$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_i(\boldsymbol{\mu}, (N-i))$ for all a.

$$\begin{aligned} c_i(\boldsymbol{\mu}, (N-i)) &< c_i(\boldsymbol{\mu}, (a)) \\ \iff (N-i) \times F_B &< a \times F_B + (N-i-a) \times k_P \times f_0 \\ \iff (N-i-a) \times F_B &< (N-i-a) \times k_P \times f_0 \\ \iff f_0 &> \frac{F_B}{k_P} \end{aligned}$$

Nash Equilibrium. The clique is a NE if $f_0 > \frac{F_B}{k_P}$.

3.2.6 Conclusion

We assumed a homogeneous payment scenario and a globally defined network fee. With this assumption any bipartite graph with the smaller group of nodes creating the channels can be a Nash equilibrium for certain conditions on the network fee. In particular, for high network fees ($0.5F_B < f_0 < F_B$), we observe that the nodes start creating channels to the other nodes in their subgroup until there is a clique for very high fees.

3.3 The Fee Game

In this section we investigate how the nodes set the fees, if the network structure is fixed to one the Nash equilibria found in section 3.2. Especially we try to find a NE which also holds for the conditions from the previous section. Therefore we slightly change the model: x_0 is a NE from and section 3.2. The nodes can only set a constant fee on each of their channels. They can not create new channels. We still have simultaneous game play, i.e., the nodes must simultaneously choose their strategy before any payment happens. The payment scenario is still the same.

3.3.1 Network Structure: Complete Bipartite Graph

We start with a bipartite graph with $c \in [2, N/2]$ nodes in the smaller partition creating the channels. The goal of this analysis is to get a better intuition of how the fees will evolve and therefore how many hubs (nodes creating channels) will establish.

We make following statement about the NE of the described game.

Corollary 3.3 (Nash Equilibrium of the Fee Game, Complete Bipartite Graph). *A strategy setup is a (weak) Nash equilibrium $\iff$ for all indirect payments, there are at least 2 node-distinct routes which are free (zero fees).*

Proof. ($\rightarrow$) If there was a Nash equilibrium and there is payment for which only one node distinct free route exists, the forwarding node would have incentive to increase the fee for forwarding the payment up to just below the price of the second cheapest route. If there was no free route and a single cheapest route, the forwarder of the cheapest route would have incentive to increase the fee up to just below the price of the second cheapest route. If multiple cheapest (but not free) routes were available, each of the forwarders of these cheapest routes would have incentive to drop its price a little bit, in order to have all payments routed through him and not randomly chosen between the cheapest routes.

($\leftarrow$) Every node stands in competition to nodes which offer free routes. Therefore he will not get any payments to forward, if he asks a fee on his channels. Therefore no node could achieve a better cost by choosing a different strategy, i.e., it is a Nash equilibrium.

All payments are therefore routed for free through the network. □

Nash Equilibrium. As we see the fees must be zero for the proposed network structure to be a NE. But this contradicts the conditions for the complete bipartite graph with at least two nodes on each side. This leads to the assumption that the bipartite graph is reduced to a star ($c = 1$).

3.3.2 Network Structure: Star

In this subsection we analyze what happens with the fees if the network structure is given as a star where the center node creates the channels.

Nash Equilibrium. Since the center node is the only node to charge any fees he is the only active player in our game. He can set the fees as he likes. He also knows the result from 3.2: If the fees are too high, a second node will start competing him and change the network structure. Therefore the center node will set the fees just below that limit. We have following pure NE:

$$f_0 = \frac{F_B}{k_P} \times \frac{2}{N-1} - \epsilon$$

3.3.3 Conclusion

We found that the star is actually a NE under the conditions in section 3.2 and under the conditions in this section. We also saw that the complete bipartite graph, can not a NE if the nodes are free to choose the fees on their channels.

We recall the restrictions we have made so far:

- Nodes with unlimited capital
- Simultaneous game play with opening channels and setting fees
- Only constant fees
- Homogeneous payment traffic on the network

Note: If we apply these restrictions for independent subgroups of the nodes, then each subgroup has its own NE in the form of a star. The total network structure is a forest.

3.4 Producers and Consumers

In this analysis we assume that there are $c > 3$ producers $d > c$ consumers. Every consumer pays every producer k_P times. In total we have $P = k_P \times c \times d$ payments. The strategy set for the producers is the same as in section 3.1. The producers choose their strategy, then the consumers connect to the best connected (average distance to other producers) producer. If multiple producers are equally well connected, the consumers randomly choose a producer to connect to. Since the behavior of consumers can be determined from the producers actions, they are modeled as part of the game, not as players.

3.4.1 Strategy Combination: Producer Clique

The first network structure we analyze is following: The producers build a clique. The first producer connects to all other producers, the second one to all but the first one and so on. The i-*th* producer opens $c - i$ channels. The strategy (for a producer) to create channels to a producer which create no channels to oneself is denoted as (a).

Cost Functions. The cost of the i-*th* producer is

$$c_i(\boldsymbol{\mu}, (c-i)) = (c-i) \times F_B - \frac{d}{c} \times c \times k_P \times f_0 = (c-i) \times F_B - d \times k_P \times f_0$$

The cost of the consumers is

$$c_d(\boldsymbol{\mu}, \mu_d) = F_B + (c-1) \times k_P \times f_0$$

The social cost is

$$-W = \frac{c \times (c-1)}{2} \times F_B + d \times F_B$$

Alternative Strategies. The producers have only one alternative strategy available (strategies covered by lemma 3.1 or lemma 3.2 not included): A producer creates channels to only $a \in [0, c-i-1]$ other producers of the set of producers he would originally connect to (producer $i+1$ to producer N).

If a producer created channels to only $a \in [0, c-i-1]$ other producers, his cost function would become:

$$c_i(\boldsymbol{\mu}, (a)) = a \times F_B$$

For the proposed strategy combination to be a Nash equilibrium, this cost function must be higher than $c_i(\boldsymbol{\mu}, (c-i))$ for all $i \in [1, c], a \in [0, c-i-1]$. Since

$c_i(\boldsymbol{\mu}, (c-i))$ constantly decreases with i and $c_i(\boldsymbol{\mu}, (a))$ constantly increases with a we only have to check the case $(i = 1, a = 0)$.

$$
\begin{aligned}
& c_i(\boldsymbol{\mu}, (c-1)) < c_i(\boldsymbol{\mu}, (0)) \\
\iff & (c-1) \times F_B - d \times k_P \times f_0 < 0 \\
\iff & (c-1) \times F_B < d \times k_P \times f_0 \\
\iff & \frac{F_B}{k_P} \times \frac{c-1}{d} < f_0
\end{aligned}
$$

Nash Equilibrium. We see that for $f_0 > \frac{F_B}{k_P} \times \frac{c-1}{d}$ all producers can make profit from forwarding payments. However, if the fee is too low, they will just not connect to anyone. This is fine for them because they do not make any payments.

3.4.2 Conclusion

The first try for a more realistic payment scenario gave us a Nash equilibrium where the producers build a tightly connected core of the network. However, this payment scenario would be more realistic if there were local subgroups where most of the payments happen.

3.5 Local Subgroups

In this section we analyze how the nodes in the network behave if there are subgroups in which many payments happen while only few payments happen between different subgroups.

We have k_S node distinct subgroups with k_N nodes per subgroup in $\boldsymbol{N}$. Each node of a subgroup makes k_P payments to every other node in the subgroup. In addition each node makes one payment to every node which is not in the same subgroup. We have:

$$N = k_S \times k_N$$

$$
\begin{aligned}
P &= k_S \times k_N \times (k_N - 1) \times k_P + k_S \times k_N \times (N - k_N) \\
&= N \times ((k_N - 1) \times k_P + (k_S - 1) \times k_N)
\end{aligned}
$$

The action set and the strategy set are the same as in section 3.1.

3.5.1 Strategy Combination: Tree of Height 2

In this strategy combination, each subgroup has a hub which connects to all other nodes in the subgroup. One of these hubs also connects to all hubs of the other subgroups. We call this hub “global hub” an the others “local hubs”. The strategy of connecting to $a \in \{0,1\}$ global hubs, $b \in [0, k_S - 1]$ local hubs, $c \in [0, k_N - 1]$ outer nodes of the global hub, $d \in [0, (k_S - 1) \times (k_N - 1)]$ outer nodes of other subgroups and $e \in [0, k_N - 1]$ outer nodes of the own subgroup is denoted as (a, b, c, d, e).

Cost Function. The cost function of the outer nodes of local hubs is:

$$c_O L(\boldsymbol{\mu}, (0,0,0,0,0)) = k_P \times (k_N - 1) \times f_0 + (k_S - 2) \times (k_N - 1) \times 3f_0$$
$$+(k_N - 1) \times 2f_0 + (k_S - 2) \times 2f_0 + f_0$$

The cost function of the outer nodes of the global hub is:

$$c_{OG}(\boldsymbol{\mu}, (0,0,0,0,0)) = k_P \times (k_N - 1) \times f_0 + (k_S - 2) \times (k_N - 1) \times 2f_0$$
$$+(k_S - 1) \times f_0$$

The cost function of the local hubs is:

$$c_{LH}(\boldsymbol{\mu}, (0,0,0,0,k_N - 1)) = (k_N - 1) \times F_B + (k_S - 2) \times (k_N - 1) \times 2f_0$$
$$+(k_N - 1) \times f_0 + (k_S - 2) \times f_0$$

The cost function of the global hub is:

$$c_{GH}(\boldsymbol{\mu}, (0, k_S - 1, k_N - 1, 0, 0)) = (k_N + k_S - 2) \times F_B + (k_S - 1) \times (k_N - 1) \times f_0$$

The social cost is:

$$(N - 1) \times F_B$$

Alternative Strategies. The nodes have following alternative strategies available (strategies covered by lemma 3.1 or lemma 3.2 not included):

- An outer node of a local hub connects to the global hub.
- An outer node of a local hub becomes a second local hub in his subgroup.
- An outer node of a local hub becomes a second global hub.
- An outer node of the global hub connects to a local hub
- An outer node of the global hub becomes a second global hub
- A local hub becomes a second local hub in an other subgroup
- A local hub becomes a second global hub
- And many more, including various combination of the named ones

Nash Equilibrium. It looks like it is analytically possible to get results with this payment scenario. But what will this results look like? We have four different types of nodes giving us a much more complex cost function than in section 3.2. We will also have much more cases to check to ensure the proposed strategy combination is indeed a NE. The result probably looks like a few dozen conditions. This would be analytically fine, but in this thesis we try to get a good intuition of the mechanisms in such a network. Therefore, these few dozen conditions would not provide us much progress. Taking into account that we are still working on a very restricted scenario we decide that this effort exceeds what we might gain from the result.

3.6 Conclusion

In this chapter we formally analyzed various strategy combinations for different payment scenarios if they are a Nash equilibrium.

The most important result is the star and the complete bipartite graph as a NE for a homogeneous payment scenario with a globally defined fee which fulfills certain conditions (see table 3.1, figure 3.1 and figure 3.2). The star is also a NE when the nodes are allowed to create channels with their own fees. A second result is the NE for the producer consumer scenario, where the producers build a clique to forward the payments from the consumers. An interesting fact is that there is a trend for networks to become centralized. This trend was observed for different payment scenarios, e.g., the homogeneous payment scenario and the producer consumer payment scenario.

Further, we made two statements about NEs for general payment scenarios (lemma 3.1 and lemma 3.2) which say that no channels are opened twice and that no payments are done on-chain. This gives us some intuition on how the nodes behave in such a network. Since we have a simultaneous game without repetition there is also a mixed NE. Even tough game theory tells us how to calculate the mixed NE, doing so is not feasible in reality.

Chapter 4

Sequential Game Model

In this game model we assume sequential gameplay. In each round one node has to make a payment. He can choose either to open a channel to his payee or not. Each payment in $\boldsymbol{P}$ defines a round of the game. We denote the payment which defines the k-th round as p_k. $x_1 ... x_P$ are the states of the network after applying the actions of the nodes in the first to the P-th round. p_k is then executed on the network in state x_k.

Action Set An action of a node n consists of choosing to either open a channel to the payee or not and defining a constant fee for that channel. This might be restricted in some analyses. The action space is

$$\boldsymbol{A}_n = \mathbb{R}_0^+, \text{ 0 means no channel is created}$$

Strategy Set A pure strategy for a sequential game with perfect information assigns an action for every possible state of the game in every round a node has to take action. The state of the game can be represented as $x_k \in (\mathbb{R}_0^+)^{N \times (N-1)}$. The set containing all such strategies for a node n is denoted as

$$\begin{aligned} \boldsymbol{S}_n &= ((\mathbb{R}_0^+)^{((\mathbb{R}_0^+)^{N \times (N-1)})})^{|\{p_k \in \boldsymbol{P} : n_S(p_k) = n\}|} \\ &= (\mathcal{F}((\mathbb{R}_0^+)^{N \times (N-1)} \rightarrow (\mathbb{R}_0^+))^{|\{p_k \in \boldsymbol{P} : n_S(p_k) = n\}|} \end{aligned}$$

Where $\mathcal{F}(A \rightarrow B)$ denotes the function space of all functions from A to B.
Note: When the action set is restricted in an analysis, the strategy sets of the nodes will also change accordingly.

Cost Functions. The cost function contains the cost for channel creation, on-chain payments, payments routed through the network and the earnings from forwarding payments. For a node n it is defined as

$$c(\boldsymbol{\mu}, \mu_n) := n_C(\boldsymbol{\mu}, \mu_n) \times F_B$$

$$+ \sum_{p_k \in \boldsymbol{P} : n_S(p_k) = n \wedge \boldsymbol{R}(x_k, p_k) = \emptyset} F_B$$
$$+ \sum_{p_k \in \boldsymbol{P} : n_S(p_k) = n} \sum_{d \in \boldsymbol{R}(x_k, p_k) : h_S(d) \neq n} f_d(x_k, h_S(d), h_R(d))$$
$$- \sum_{p_k \in \boldsymbol{P} : n_S(p_k) \neq n} \sum_{d \in \boldsymbol{R}(x_k, p_k) : h_S(d) = n} f_d(x_k, h_S(d), h_R(d))$$

Social Cost. The social cost is calculated in the same way as in the simultaneous game model in chapter 3:

$$-W = (n_C(\boldsymbol{\mu}) + n_{OCP}(\boldsymbol{\mu})) \times F_B$$

Social Optimum. The social optimum is also the same as in chapter 3:

$$\min(-W) = (N-1) \times F_B$$

4.1 Without Network Fee

In this section we assume every node makes k_P payments. This gives $P = k_P \times N$ payments in total.

Action Set and Strategy Set There is a globally constant the fee $f_0 = 0$ on all channels. An action of a node n can therefore be represented as a subset of $\boldsymbol{N} \setminus \{n\}$. The action set of node n is the powerset of $\boldsymbol{N} \setminus \{n\}$, denoted as

$$\boldsymbol{A}_n = 2^{\boldsymbol{N} \setminus \{n\}}$$

As there are $2^{\frac{N^2-N}{2}}$ possible states of the game the strategy set is

$$\boldsymbol{S}_n = (\boldsymbol{A}_n^{(2^{\frac{N^2-N}{2}})})^{k_P}$$

4.1.1 Strategy Combination: Connect to unreachable Payee

The first strategy setup we investigate is following: A node creates a channel to his payee if he is not yet connected to him through the network.

Cost Functions. The cost function of a node is

$$c(n) = \sum_{p_k \in \boldsymbol{P} : \boldsymbol{R}(x_{k-1}, p_k) = \emptyset \wedge n_S(p_k) = n} F_B$$

The model assumes the payments to span a global network, therefore the graph will be connected. Since nodes only connect to nodes they are not connected to already, no loops will occur in the network graph. The network structure will be a spanning tree and therefore the social cost is

$$-W = (N - 1) \times F_B$$

Nash equilibrium. The nodes minimize their expenditures for channel openings and there is no way they could earn anything. Therefore there is no way they can do better and presented strategy combination is a NE.

4.1.2 Conclusion

When we have no network fee, the strategy combination where everyone connects to his payee, if the payee is not reachable through the network, is a Nash equilibrium.

4.2 Global Fee

We assume each node makes k_P payments. This gives $P = k_P \times N$ payments in total. There is a globally defined fee for all channels $f_0 \in \mathbb{R}$. The action set and the strategy set are the same as in section 4.1.

4.2.1 Strategy Combination: Connect to unreachable Payee

The first strategy setup we investigate is the same as in section 4.1: A node creates a channel to his payee if he is not yet connected to him through the network.

Cost Functions. The cost function of each node is the number of channels he opens times F_B plus the fees for the payments he sends trough the network minus the fees he earns from forwarding payments.

The model assumes the payments to span a global network, therefore the graph will be connected. Since nodes only connect to nodes they are not connected to already, no loops will occur in the network graph. The network structure will be a spanning tree and therefore the social cost is $-W = (N - 1) \times F_B$.

Nash Equilibrium. The nodes minimize their expenditures for channel openings because they only open a channel if it is really necessarily. However, we observe there are cases were this strategy is suboptimal. Below we present two such cases;

- A node pays fees for repeated payments to the same payee (to which he is connected through hops) which sum up to over F_B. In that case it would be better for him to open a channel to his payee before he does the first payment.
- A node opens a channel to a loosely connected node in a subnetwork to which he is not connected yet but to which he want s to make a payment. If he wants to make many payments to that subnetwork it might be better for him to connect to a well connected node in that subnetwork to safe fees on the future payments.

This list is not complete but it shows that this strategy combination is not a NE and it gives an intuition of the complexity of the problem. A node has to take into account the whole future (set of payments) to optimize his strategy.

4.3 General Nash Equilibrium

We could not find a general NE for the case of a globally constant fee $f_0 \neq 0$. However, the discussion about proposed strategy combination gives the intuition that the NE is highly dependent on the payment scenario. In this section, we write this intuition more formally and prove it.

Existence of a Pure Nash Equilibrium. Since we have a sequential game and perfect information, there must be a pure NE (every node can draw the game tree and choose the best strategy for him, as he knows what everyone else will do).

Definition 4.1 (Game Tree)**.** A *game tree* is a representation of a game as a tree with the payouts at the leaves and the decisions on the edges.

Since in our case each node has just two actions available at each of his turns, the game tree is a binary tree. In this section we calculate two upper bounds on the size of the game tree.

Lemma 4.2. *A channel will never increase fees on any future payment.*

Proof. (Towards contradiction.) Assume there is a channel, who's existence leads to a higher fee for a payment. Since the channel has an influence on the payment, the payment must be routed through that channel. The route ($r1$) of the payment in the network without the channel, is cheaper than the route ($r2$) in the network with the channel. But the route $r2$ still exists, even if the additional channel is created. This contradicts the definition of the route (definition 2.7). □

Lemma 4.3. *A node will never lose income if he creates an additional channel.*

Proof. If an additional channel changes the route of a payment, the additional channel is part of this route. Therefore the owner of this channel does not loose income in this case. The other case is that the additional channel does not change the route of a payment, then income does not change neither. □

Lemma 4.4. *In an Nash equilibrium, if a sender is not connected to his payer, he always opens a channel.*

Proof. The cost of creating a channel for the payment is equal to the cost of sending the payment by using the blockchain (F_B). By lemma 4.3 the node will have the same or higher income in the future after creating a channel. Additionally he might profits from lower fees (lemma 4.2). This means a node is not able to reduce his cost function by not creating a channel to a payee, to which he is not connected yet. □

Lemma 4.5. *In a Nash equilibrium, a channel is always created on the first possibility.*

Proof. By lemma 4.2, 4.3 and 4.4 a node will never have a lower cost or higher profit by deferring the creation of a channel. □

Corollary 4.6 (Upper Bound on the Size of the Game Tree)**.** *For any N, P the size of the game tree (number of nodes in the tree) of the sequential game will never exceed:*

- $2^P - 1$
- $P \times 2^{\frac{N^2-N}{2}}$

Proof. The first one is just the exact tree size for a binary tree of height P.
The second one:
There are only $\frac{N^2-N}{2}$ channels to open, because no channels are closed. This gives a maximum width of the tree of $2^{\frac{N^2-N}{2}}$. The height of the tree is P. Therefore the maximum size of the tree is width times height equals $P \times 2^{\frac{N^2-N}{2}}$. □

Note: even if the second upper bound is much better then the first one for large P, in reality, with $N > 8000$ for the Lightning network at the time of writing, it is not feasible for the nodes to calculate the whole game tree.

Corollary 4.7. *For any N, P there exists a set of payments $\boldsymbol{P}$ of size P such that removing the last payment or not will change the optimal strategy of the third sending node.*

Proof. (By construction.) Assume:

$$\boldsymbol{N} = \{A, B, C, ...\}$$

$$\boldsymbol{P} = \{A \to B, B \to C, k' \times (C \to A)\}$$

Obviously for any F_B and f_0 we can find a k' such that C's optimal strategy at the third round will change depending on whether the last payment is removed or not.

$$k' \times f_0 > F_B \wedge (k' - 1) \times f_0 < F_B$$

$$\iff k' = \lceil \frac{F_B}{f_0} \rceil$$

□

4.4 Conclusion

We observe that there is a pure Nash equilibrium for every payment scenario. However, we showed that this NE is very specific for different payment scenarios (corollary 4.7). Further we proved two upper bounds on the size of the game tree (corollary 4.6). Nevertheless, in general it is not feasible for the nodes to calculate the whole game tree.

Chapter 5

Simulation

To gain more realistic insight into how rational nodes behave in a micropayment network, we implemented a simulation [13] which can generate groups of nodes and payment scenarios with some random parameters. We implemented four variations of a short term greedy policy, two network types, and three payment scenario types.

5.1 Network Model

The network model is the main state object used in our simulation (beside the payment scenario as second state object). It holds all the information about the network, which are needed for the simulation. The nodes are represented as a list of integers from 0 to $N-1$. A list of length N stores the types of the nodes, e.g., 'consumer' or 'producer'. Another list of length N holds the policies of the nodes. The capital of the nodes and the allocated funds in the channels are represented as a matrix with the nodes as indices, e.g., the entry (i,j) denotes how much funds node i has on its channel to node j. The entry (i,i) is the free capital of node i. This restricts the nodes to have only one open channel to the same node at the same time. This is reasonable because opening a second channel to the same node is equally expensive as topping up the existing one. Another matrix denotes the fees on the network in the same way as the funds.

The nodes can create, close, and top up channels. The topping up is the possibility to increase the funds on one side of the channel by using a blockchain transaction as described in [14].

5.2 Policy

A policy is implemented as a method which takes the network state and a payment as input. It returns an action object, which is then applied to the network. We have implemented one short term greedy policy, which has four different variations. All of them try to minimize the cost for the current payment. However, if they have multiple equally expensive possibilities to handle a payment, they have different preferences.

The policy works as follows: If there is a route (cheaper than F_B) use it. If not, find the cheapest possibility to create such a route using a channel topup. Also find the cheapest possibility to create such a route using a channel opening. To find these possibilities, also channel closings are considered in case a node does not have enough capital to fund or topup a channel. The two possibilities are compared and the cheaper one is returned. The channels are always created with a fee of $10^{-5} \times payment_value$.

There is the situation where multiple equally expensive possibilities exist. Firstly, if the option to use a topup and the option to create a new channel are equally expensive. And secondly, if a channel has to be closed and multiple channels are suitable: should the one with the most capital be closed, or the one with the least capital? These two situations give us four variations of our short term greedy policy. These four variations were compared in different situations (see appendix A). There was no big difference between them, so we continued with the one with preference for topping up channels and releasing maximum capital.

5.3 Network Types

In this section the two implemented types of network generation methods are described.

Random Network (RND). A random network is generated given the number of nodes and the mean capital of the nodes. The capital of a node is then randomly chosen around the mean capital of the node using an exponential distribution.

Consumer Producer Network (CP). This network generation method takes the number of consumers and producers as well as their corresponding mean capital as input. The actual capital of the nodes is then randomly chosen around the provided mean using a exponential distribution. Additionally the nodes are labeled as consumer or producer.

5.4 Payment Scenarios

A payment scenario is an object which is generated before the simulation. When asked, the payment scenario object itself will then generate the payments for a payment period, based on the current state of the network. This allows us to create a dynamic payment scenario which adapts to changes in the capital distribution.

Balanced and Unbalanced Payment Scenarios. We call a payment scenario balanced, if for each payment period, each node has an expected capital change of zero (except of the fees). We call it unbalanced otherwise.
Note: Only the expected capital change must be zero. This means for a concrete payment period of this payment scenario the capital of the nodes can change. This might cause a node to go bankrupt.

Regular and Spontaneous Payments. At the time of generation, a payment scenario can define a set of payments which will reoccur in every payment period. These payments are called regular payments. On the other hand, payments which are actually generated at the time of the payment period generation, are called spontaneous payments.

Small to Large Scenario (STL). This payment scenario can be generated on any network, it does not require any labeled nodes. For a new payment, it randomly chooses a node as sender for the payment. The receiver is chosen with weighted randomness: probability of a node being chosen as a receiver corresponds to his capital divided by the network capital. The amount of money for the payment is uniformly distributed between half a percent and one percent of a node's capital. This leads to the situation that big (rich) nodes receive more payments, on the other side, big nodes send bigger payments. The goal of this payment scenario is to imitate the real world in a natural way, i.e., without using fixed structures. Nevertheless, big nodes act somehow as companies and smaller nodes as people: people frequently pay companies and get only few payments (salary) from the companies.

Balanced Consumer Producer Scenario (BCP). This payment scenario requires the nodes to be labelled as consumers and producers. It generates salaries and regular expenses which reoccur in every payment period. It also generates some spontaneous expenses and some inter-producer payments for each payment period. It is is a completely balanced payment scenario, i.e., in every payment period, each node spends exactly the same amount of money as he receives (except the fees). This enables us to run the payment scenario for a very long time without having nodes which go bankrupt.

Unbalanced Consumer Producer Scenario (UCP). As the previous payment scenario, this one also requires the nodes to be labeled as consumers and producers. It generates a salary for each consumer which will be paid in every payment period. The consumers spend a fixed percentage (in our case 80%) of their salary in every payment period. Since the consumers do not spend all of their salary at some point a producer will run out of money. This payment scenario is not balanced since the consumers do not spend all their salary.

5.5 Simulation Results

We generated one instance per network type. In the RND_1 network the 30 nodes have a mean capital of $10^5 \times F_B$. We generated five STL scenarios for this network. In the CP_1 network, the 25 consumers have a mean capital of $10^5 \times F_B$ and the 5 producers $10^7 \times F_B$. Five BCP and five UCP scenarios were generated for that network. The simulation was run for 96 payment periods for all of these combinations. The nodes make 10 payments per payment period in average. Table 5.1 shows the most important results from the simulations.

network	scenario	n_channels	c_created	c_closed	c_topup	bc_tx
RND_1	STL_1	94	94	0	14	108
RND_1	STL_2	99	99	0	10	109
RND_1	STL_3	82	82	0	10	92
RND_1	STL_4	79	79	0	5	84
RND_1	STL_5	86	86	0	16	102
CP_1	BCP_1	48	48	0	8	56
CP_1	BCP_2	47	47	0	7	54
CP_1	BCP_3	51	51	0	8	59
CP_1	BCP_4	55	55	0	5	60
CP_1	BCP_5	53	53	0	6	59
CP_1	UCP_1	78	90	12	388	490
CP_1	UCP_2	90	109	19	612	740
CP_1	UCP_3	90	105	15	539	659
CP_1	UCP_4	87	96	9	351	456
CP_1	UCP_5	86	101	15	547	663

Table 5.1: Simulation results including the used network, payment scenario, the number of channels at the end of the simulation, the number of channel creations, the number of channel closings, the number of channel topups, the number of blockchain transactions

Table 5.2 shows the averages over the five different payment scenarios per type.

network	scenario type	n_channels	c_created	c_closed	c_topup	bc_tx
RND_1	STL	88	88	0	11	99
CP_1	BCP	50.8	50.8	0	6.8	57.6
CP_1	UCP	86.2	100.2	14	487.4	601.6

Table 5.2: Simulation results averaged over the payment scenario type, including the network, payment scenario type, the number of channels at the end of the simulation, the number of channel creations, the number of channel closings, the number of channel topups, the number of blockchain transactions

In the following paragraphs we show and discuss, the number of channels in the network over time, the number of channel creations over time, the number of channel topups over time, and the correlation between the initial capital of a node and his degree in the graph over time. We also show the initial and final network state. Here, we show only one final network state per experiment (5 runs), because they all look similar. All final network states can be found in B.

RND Network and STL Payment Scenario. For the STL scenario there is a near constant increase of the number of channels over time and the correlation stabilizes between 0.6 and 0.9 which indicates that there is a clear tendency for larger nodes to have more channels. The initial and the final network graphs are shown in figure 5.1. Figures 5.2, 5.3, 5.4, 5.5 show how the number of channels, the number of channel creations, the number of channel topups, and the correlation between the initial capital of a node and the number of channels of a node evolve over time.

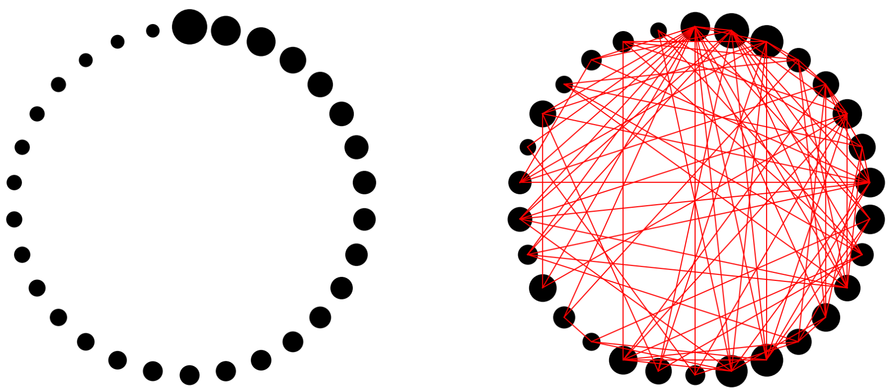

Figure 5.1: The start and resulting network graph for the simulation with the RND_1 network and the STL_1 scenario.

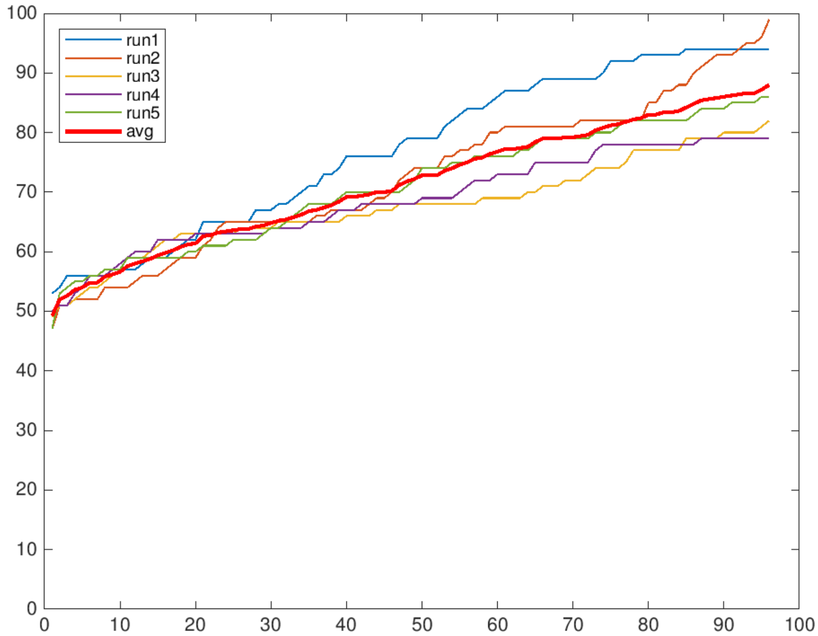

Figure 5.2: Evolution number of channels in the network for the simulation with of STL scenarios.

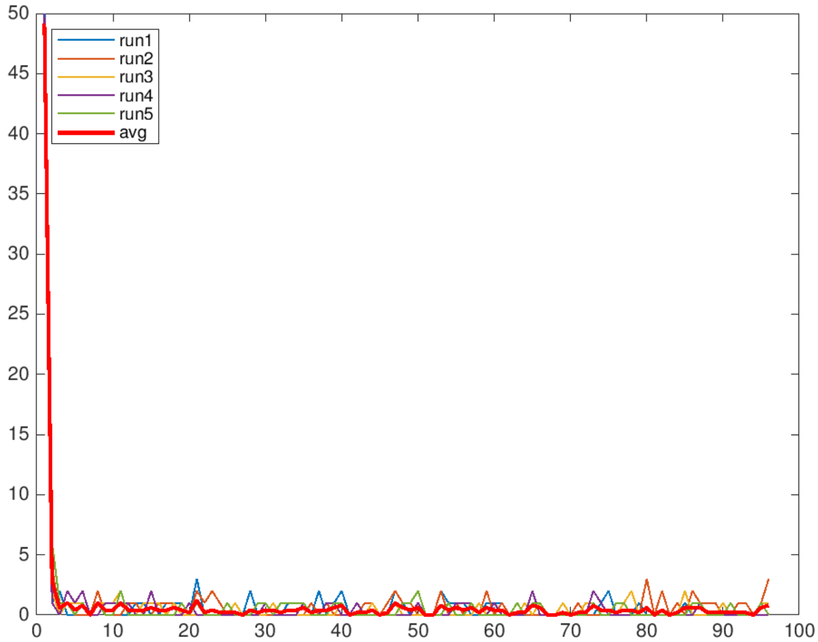

Figure 5.3: Number of channel creations in the network over time for the simulation of STL scenarios.

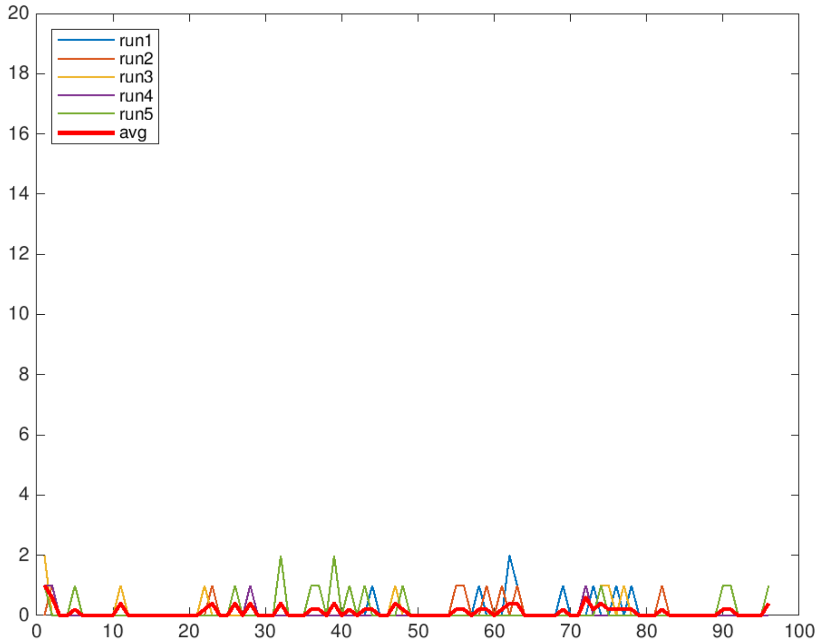

Figure 5.4: Number of channel topups in the network over time for the simulation of STL scenarios.

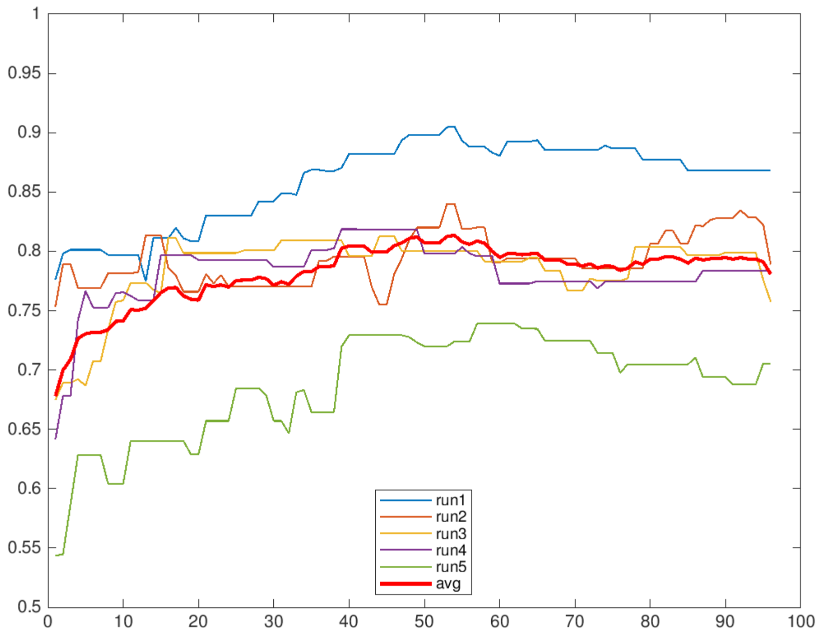

Figure 5.5: Evolution of correlation between node degree and initial capital over time for the simulation of STL scenarios.

CP Network and BCP Payment Scenario. This simulation shows an almost constant number of channels after the first few payment periods. The main reason for this is the tightly connected core of producers. Another reason is the high number of reoccurring payments: only about half of the payments are generated spontaneously, the rest are payments reoccur in every payment period. Further, there is a very high correlation between the capital of a node and the number of channels he has (between 0.9 and 1). This means small nodes are only connected to a few large nodes, while large nodes have channels to much more nodes. The initial and the final network graphs are shown in figure 5.6. Figures 5.7, 5.8, 5.9, 5.10 show how the number of channels, the number of channel creations, the number of channel topups, and the correlation between the initial capital of a node and the number of channels of a node evolve over time.

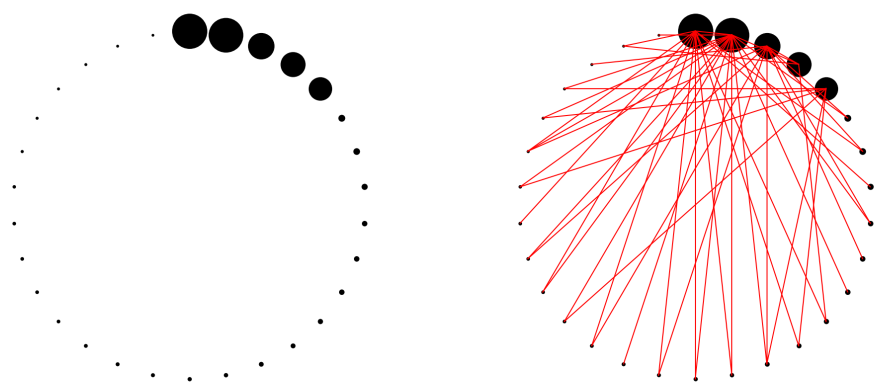

Figure 5.6: The start and resulting network graph for the simulation with the CP_1 network and BCP_1 scenario.

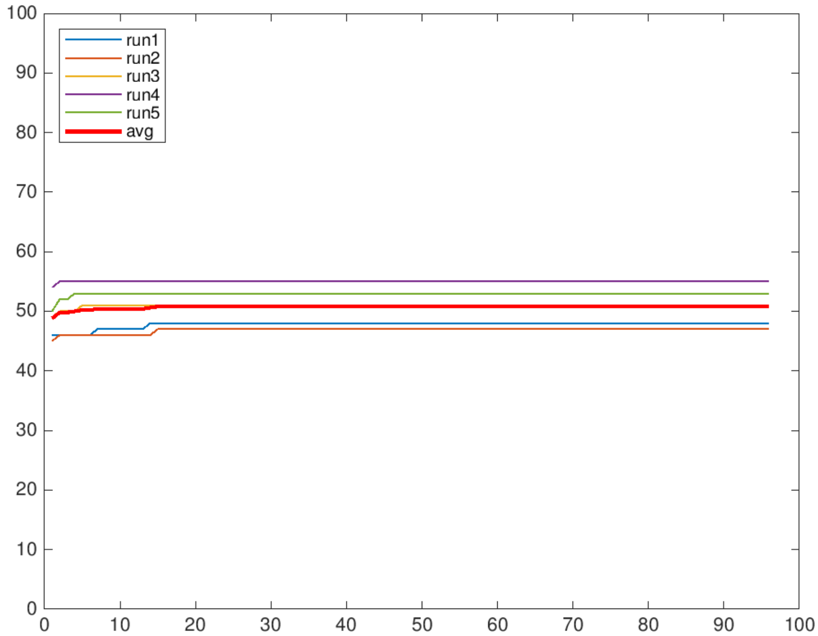

Figure 5.7: Number of channels in the network over time for the simulation of BCP scenarios.

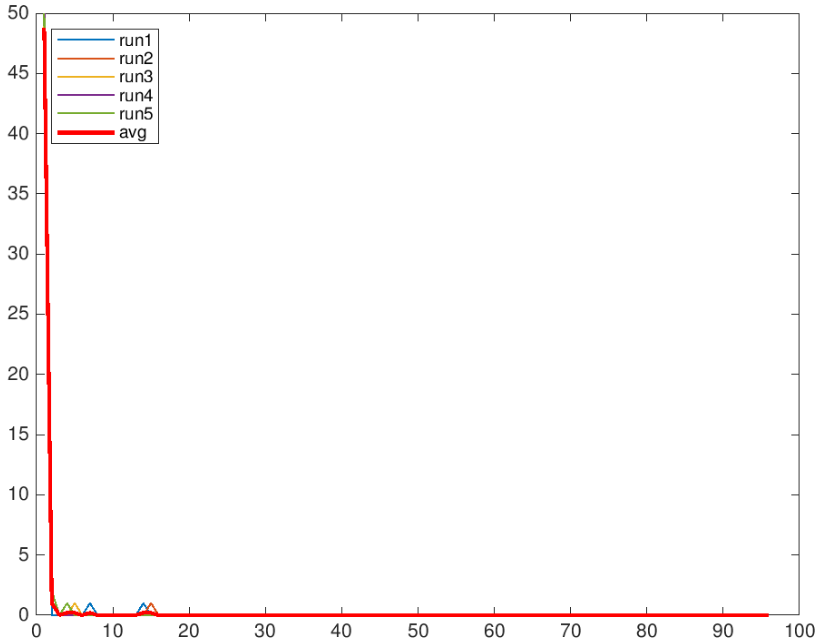

Figure 5.8: Number of channel creations in the network over time for the simulation of BCP scenarios.

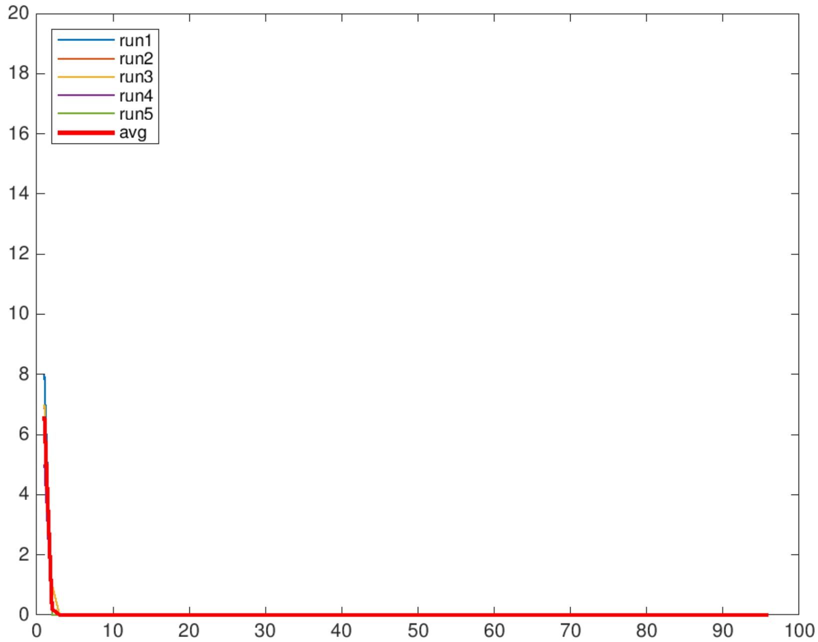

Figure 5.9: Number of channel topups in the network over time for the simulation of BCP scenarios.

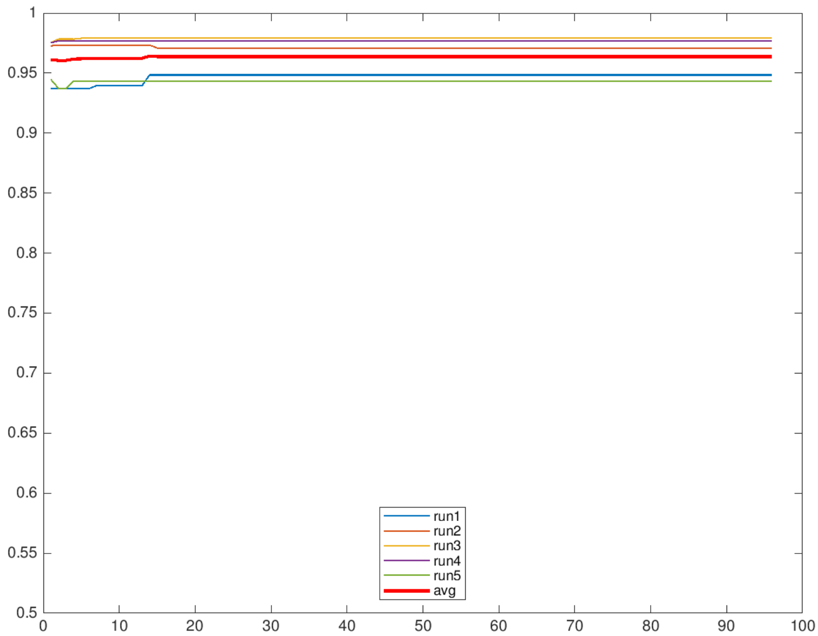

Figure 5.10: Evolution of correlation between node degree and initial capital over time for the simulation of BCP scenarios.

CP Network and UCP Payment Scenario. For the UPC scenario the number of channel increases, but starts to saturate after some time. The correlation between the capital of a node and the number of channels he has starts at the same level as for the BCP scenario. However, it drops and stabilizes between 0.8 and 0.9. Figures 5.12, 5.13, 5.14, 5.15 show how the number of channels, the number of channel creations, the number of channel topups, and the correlation between the initial capital of a node and the number of channels of a node evolve over time.

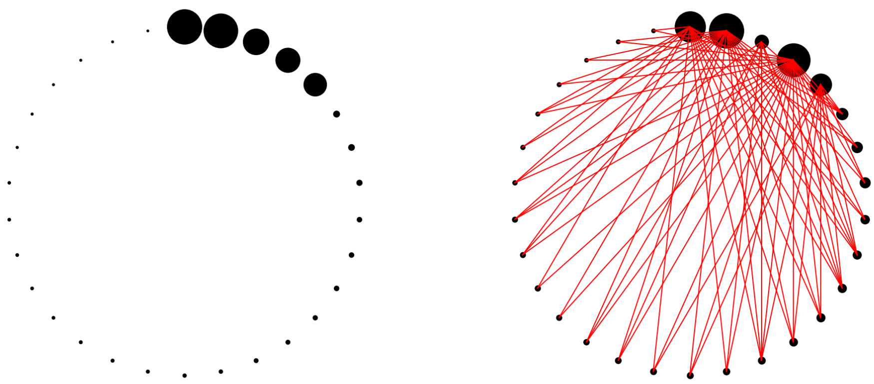

Figure 5.11: The start and resulting network graph for the simulation with the CP_1 network and the UCP_1 scenario.

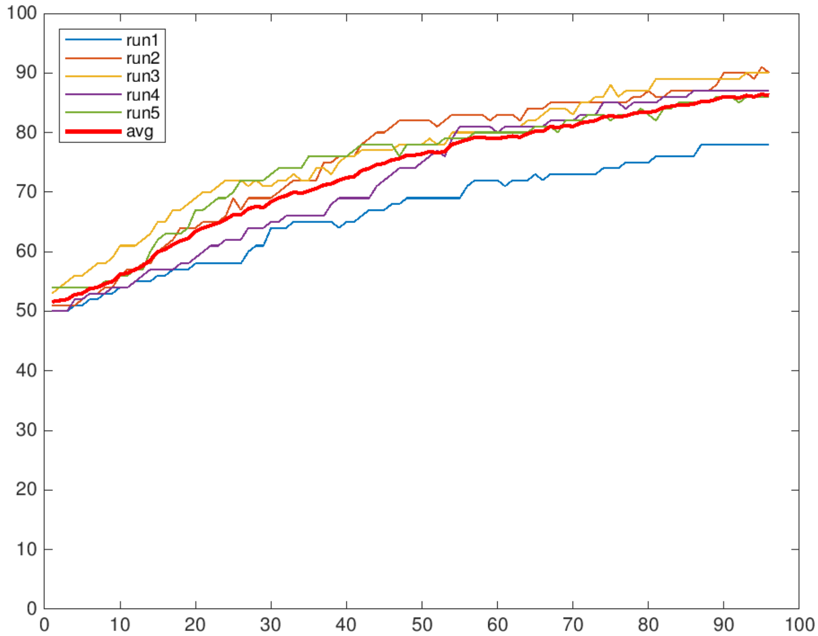

Figure 5.12: Number of channels in the network over time for the simulation with of UCP scenarios.

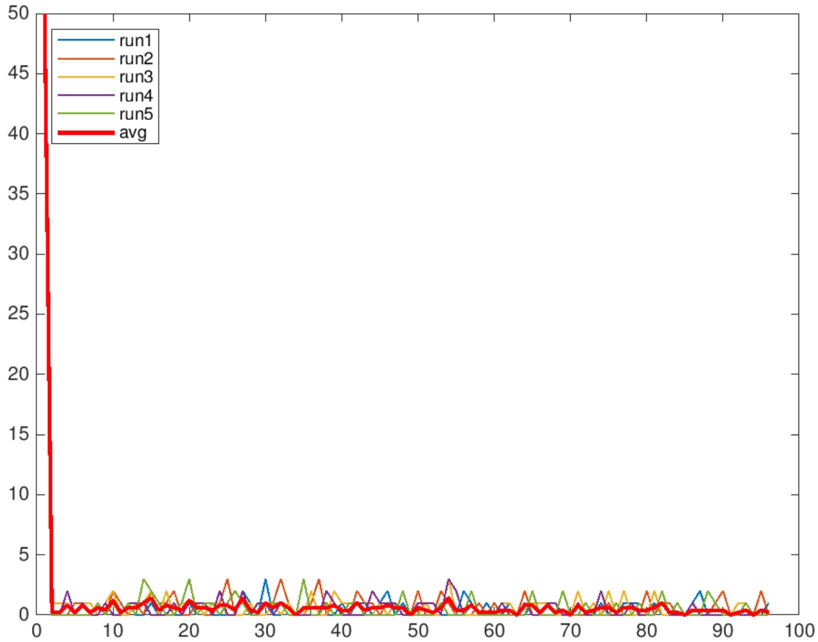

Figure 5.13: Number of channel creations in the network over time for the simulation of UCP scenarios.

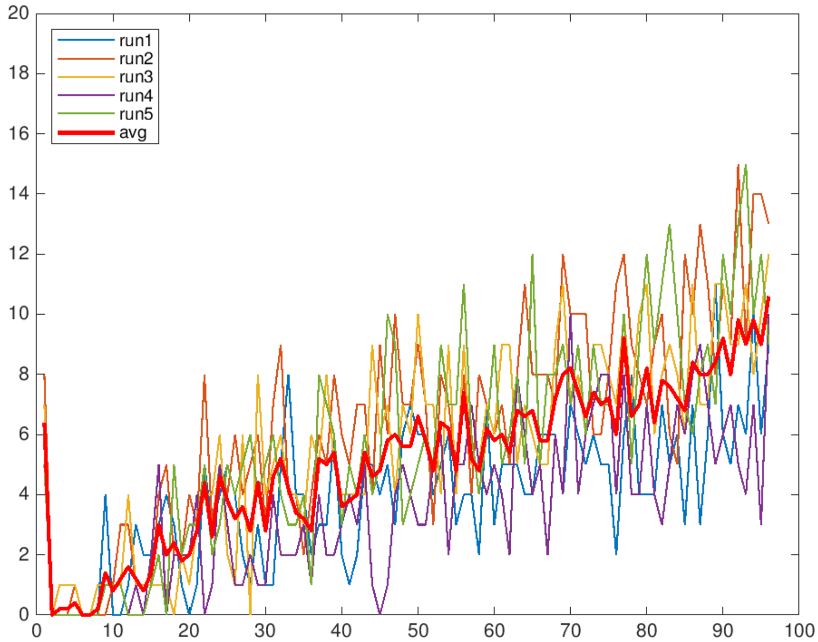

Figure 5.14: Number of channel topups in the network over time for the simulation of UCP scenarios.

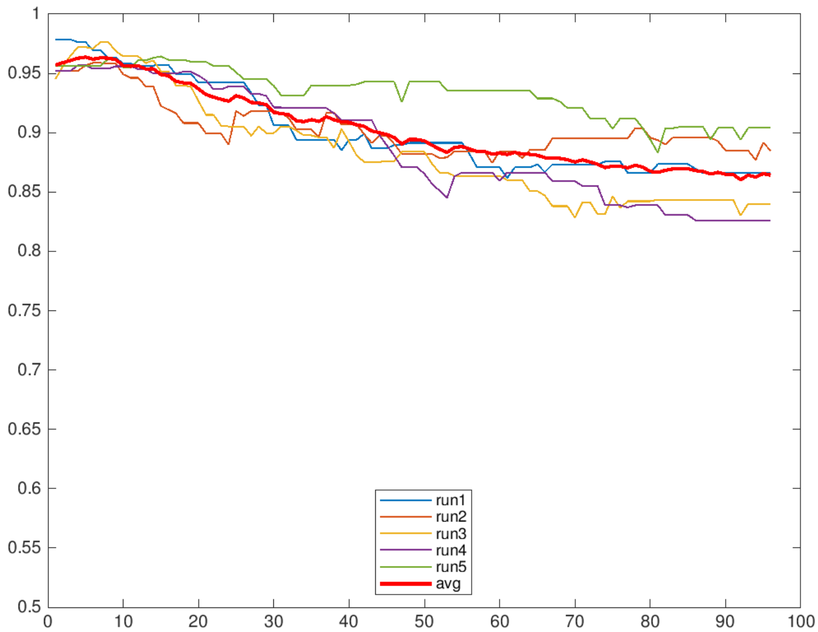

Figure 5.15: The correlation between node degree and initial capital over time for the simulation with of UCP scenarios.

We see that that the result of the simulation strongly depends on the payment scenario. This underlines the results of the theoretic part (chapters 3 and 4) which indicate a strong dependence of the Nash equilibrium on the payment scenario.

Chapter 6

Related Work

Payment channels were originally introduced by Spilman [15]. The original idea was to use unidirectional channels with a predefined sender and receiver. Later, various constructions for bidirectional payment channels were proposed [5, 11, 15, 16]. They all use a common account for the parties and off-chain exchange of signed transactions proving the state of the channel. The creation of multiple such channels on a common blockchain network leads to the formation of channel networks, such as the Lightning network [5] on Bitcoin [1]. In this work, we studied different strategies for the nodes in such a payment channel network, independent of which payment channel construction method is used. Thus, this work applies to all payment channel solutions.

Avarikioti et al. [17, 18] formulated a similar problem to the one studied in this thesis. They aimed to find an optimal strategy for a central instance, a so-called payment service provider. We found that their near-optimal solution (the star as network structure) is also a Nash equilibrium in an uncoordinated situation. Note that the two approaches are completely different and yet they produced similar results. This is a strong indication that the Lightning network (and similar others) will eventually form a more centralized network structure.

Network creation games, originally introduced in by Fabrikant et al. [19], are used to model distributed networks with rational players. Each player wants to maximize/minimize a profit/cost function which represents the cost of creating and using the network. Fabrikant et al. [19] modeled the Internet using Network Creation Games. They used a cost function containing the network creation cost and the sum of the distances to the other nodes. For their model, they proved upper and lower bounds for the Price of Anarchy(PoA). They also conjectured that Nash equilibria in this game are trees, however this was disproved by Albers et al. [20]. Alon et al. [21] aimed for stronger bounds on PoA of the Network Creation Game (sum and local-diameter version). However, all of these works use very basic cost functions. In contrast, in this work the cost function is very complicated since the usage cost contains the earned and paid fees which depend on the state of the network which itself contains the individual fee policies of the nodes and changes over time. Fabrikant et al. [19] and Alon et al. [21] both have temporally separated creation and usage of the network. Especially the sequential game model differs from these works as it does not temporally separate the creation and the usage of the network.

Chapter 7

Conclusion

We defined a simultaneous and a sequential game model to study the behavior of nodes in a micropayment network. We found Nash equilibria (NE) for several cases of these models. For the simultaneous model with a homogeneous payment scenario (section 3.2) we found that a star, a complete bipartite graph, and a clique as NE for different values of a globally defined fee. The star can also be a NE, if there is no globally defined fee, but the nodes choose the fees for their channels individually (section 3.3). Further, we formulated a producer consumer based payment scenario for the simultaneous game model and found a NE were the producers build a clique.

For the sequential game model (chapter 4), for a specific payment scenario, there is always a NE. In theory, we can calculate this NE, but in real life this method is not feasible (corollary 4.6). However, we could not formulate a generally applicable optimal strategy. We proved that NE in this model are indeed very specific for different payment scenarios (corollary 4.7).

The limit we reached with the mathematical analysis was set by the strong dependence of NEs on a concrete payment scenario. However, the NEs we found indicate increasing centralization of micropayment networks in the future. This is also indicated by Avarikioti et al. [17, 18] which found the star as a near optimal solution for a payment service provider. Further, we made some general statements about NEs which might help future works to define upper and lower bounds on the Price of Anarchy.

The simulation (chapter 5) of short term greedy nodes, gave us a different point of view. This provided us a more general understanding of the behavior of rational nodes in a micropayment network. The simulation showed (once more) a strong dependence of the resulting network graph on the payment scenario (section 5.5).

In this work, we freely defined some payment scenarios. One could also have tried to use real world data. However, even when using real world data one must always keep in mind, that also data of existing payment systems can not be applied 1-to-1 to a micropayment network, because new fee systems will lead to new payment habits.

7.1 Future Work

Model Analysis. The game models provide a solid basis for a deeper, more mathematical analysis of the micropayment channels game. However, one has always to keep in mind, that concrete NE's heavily depend on the chosen payment scenario. The game theoretic approaches however, reach their limit when dealing with very large strategy sets, e.g., function spaces for the sequential model, where even defining these strategy sets can be a challenge.

Another approach is to research the dependence between the NEs and the payment scenario. This would help to see specific solutions, e.g., for a homogeneous payment scenario, in a bigger context. For highly parameterized payment scenarios, more graph theoretical tools will be needed than used in this thesis.

A third approach in the direction of model analysis is to make a step back and to define a cost function which is independent of the payment scenario, but which is motivated by different payment scenarios (maybe real world data).

We always assumed a fixed payment scenario to see how the network structures and fees will look like. However, as stated in chapter 7 one could also do the exact opposite: Assume a fixed network structure and fees and analyse how the payment habits will change. Furthermore, one could also define a new model. We suggest following (motivated by the simulation): the nodes know the payments of a payment period in advance, they choose their action for a payment period simultaneously and then the payments are executed.

Simulation. The architecture of the simulation code is very modular. It can easily be extended with new network types, new payment scenarios or new policies. The simulation can also be run with data from csv files. Therefore, it can also be fed with real world data.

Another interesting approach would be to use neural networks to optimize the policy for payment scenarios. These payment scenarios could be randomly generated or based real world data.

Bibliography

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," accessed: 2019-05-10. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] V. Buterin, "Ethereum white paper," accessed: 2019-05-13. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[3] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–125.

[4] M. Scherer, "Performance and scalability of blockchain networks and smart contracts," Ph.D. dissertation, Umeå University, 2017. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-136470

[5] T. D. Joseph Poon, "The bitcoin lightning network: Scalable off-chain instant payments," accessed: 2019-05-10. [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[6] "Raiden network documentation," accessed: 2019-05-13. [Online]. Available: https://raiden-network.readthedocs.io/en/stable/

[7] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ser. ICDCN '16. New York, NY, USA: ACM, 2016, pp. 13:1–13:10. [Online]. Available: http://doi.acm.org/10.1145/2833312.2833321

[8] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol," Cryptology ePrint Archive, Report 2016/1159, 2016. [Online]. Available: https://eprint.iacr.org/2016/1159

[9] S. B. D. H. G. D. Mustafa Al-Bassam, Alberto Sonnino, "Chainspace: A sharded smart contracts platform," accessed: 2019-05-14. [Online]. Available: https://arxiv.org/pdf/1708.03778.pdf

[10] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 583–598, 2018.

[11] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Stabilization, Safety, and Security of Distributed Systems*, A. Pelc and A. A. Schwarzmann, Eds. Cham: Springer International Publishing, 2015, pp. 3–18.

[12] "1ml," accessed: 2019-05-12. [Online]. Available: https://1ml.com/

[13] R. Scheuner, "Payment channel network simulations," accessed: 2019-05-19. [Online]. Available: https://gitlab.ethz.ch/disco-students/fs19/payment-channel-network-simulations

[14] L. Lab, "Dynamic channel deposit and withdrawl for lightning network," accessed: 2019-05-16. [Online]. Available: https://github.com/litexnetwork/papers/blob/master/FundCasc.pdf

[15] J. Spilman, "Anti dos for tx replacement," accessed: 2019-05-14. [Online]. Available: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-April/002433.html

[16] O. O. Christian Decker, Rusty Russell, "eltoo: A simple layer2 protocol for bitcoin," accessed: 2019-05-14. [Online]. Available: https://blockstream.com/eltoo.pdf

[17] G. Avarikioti, G. Janssen, Y. Wang, and R. Wattenhofer, "Payment network design with fees," in *DPM/CBT@ESORICS*, 2018.

[18] G. Avarikioti, Y. Wang, and R. Wattenhofer, "Algorithmic blockchain channel design," *CoRR*, vol. abs/1810.07603, 2018.

[19] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker, "On a network creation game," in *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, ser. PODC '03. New York, NY, USA: ACM, 2003, pp. 347–351. [Online]. Available: http://doi.acm.org/10.1145/872035.872088

[20] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty, "On nash equilibria for a network creation game," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, ser. SODA '06. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006, pp. 89–98. [Online]. Available: http://dl.acm.org/citation.cfm?id=1109557.1109568

[21] N. Alon, E. D. Demaine, M. Hajiaghayi, and T. Leighton, "Basic network creation games," *SIAM Journal on Discrete Mathematics*, vol. 27, pp. 106–113, 01 2010.

Appendix A

Policy Variation Comparison

The comparison of the four variations of the policy gave the result shown in table A.1.

network	scenario	policy	channel	create	close	topup	bctx
RND_N1	STL_S1	topup/min	84	84	0	5	89
RND_N1	STL_S1	topup/max	84	84	0	5	89
RND_N1	STL_S1	open/min	84	84	0	5	89
RND_N1	STL_S1	open/max	84	84	0	5	89
CP_N1	STL_S2	topup/min	168	274	106	728	1108
CP_N1	STL_S2	topup/max	183	198	15	687	900
CP_N1	STL_S2	open/min	177	285	108	606	999
CP_N1	STL_S2	open/max	191	206	15	599	820
CP_N1	BCP_S1	topup/min	50	50	0	9	59
CP_N1	BCP_S1	topup/max	50	50	0	9	59
CP_N1	BCP_S1	open/min	50	50	0	9	59
CP_N1	BCP_S1	open/max	50	50	0	9	59
CP_N1	UCP_S1	topup/min	88	110	22	481	613
CP_N1	UCP_S1	topup/max	91	112	21	473	606
CP_N1	UCP_S1	open/min	88	110	22	481	613
CP_N1	UCP_S1	open/max	91	112	21	473	606

Table A.1: Simulation Results including the used network, payment scenario, policy, the number of channels at the end of the simulation, the number of channel creations, the number of channel closings, the number of channel topups, the number of blockchain transactions

Appendix B

Resulting Network Graphs

In this chapter we show all resulting network graphs form the simulations from chapter 5.

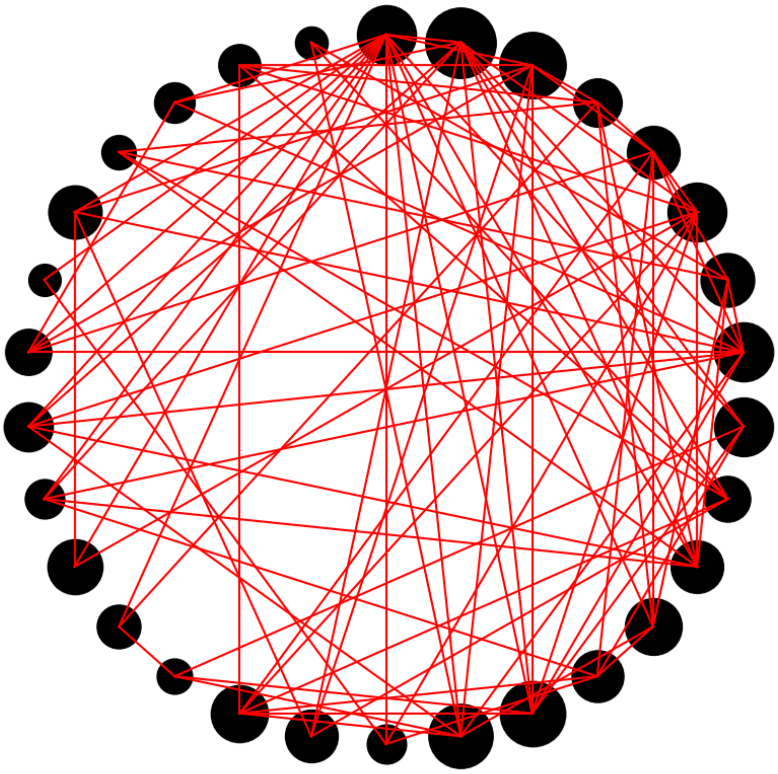

Figure B.1: The resulting network graph of the simulation with RND_1 and STL_1.

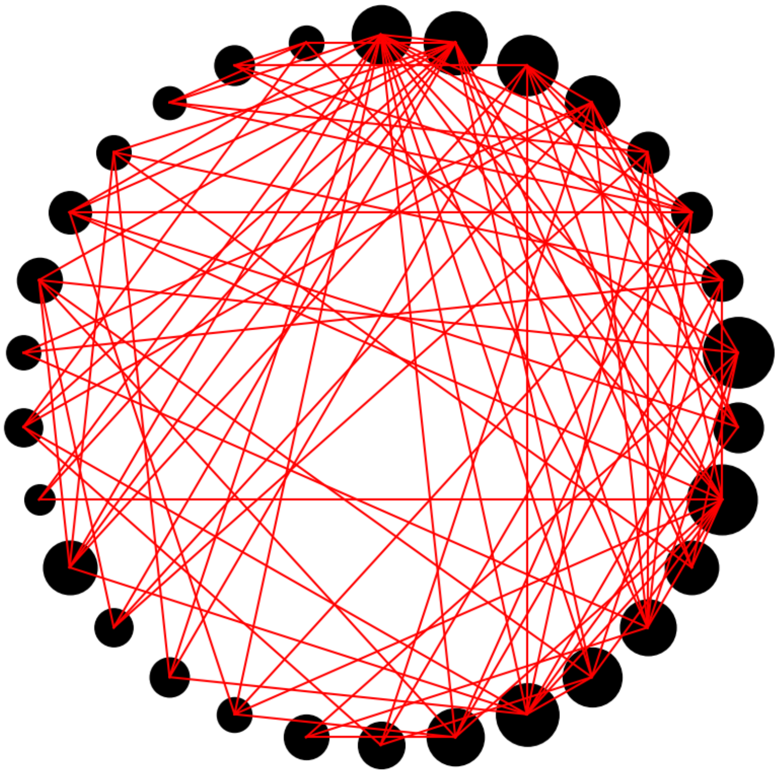

Figure B.2: The resulting network graph of the simulation with RND_1 and STL_2.

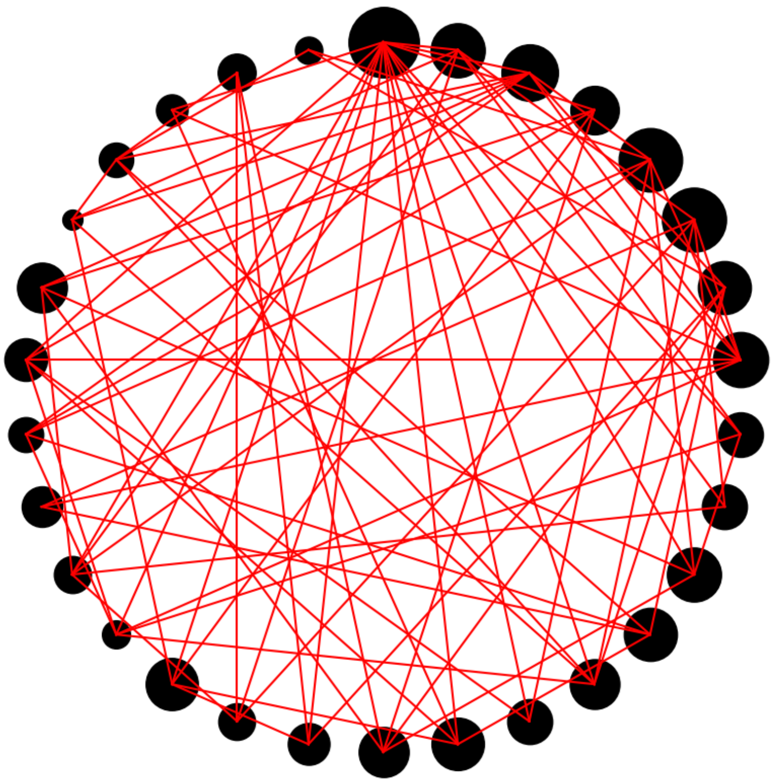

Figure B.3: The resulting network graph of the simulation with RND_1 and STL_3.

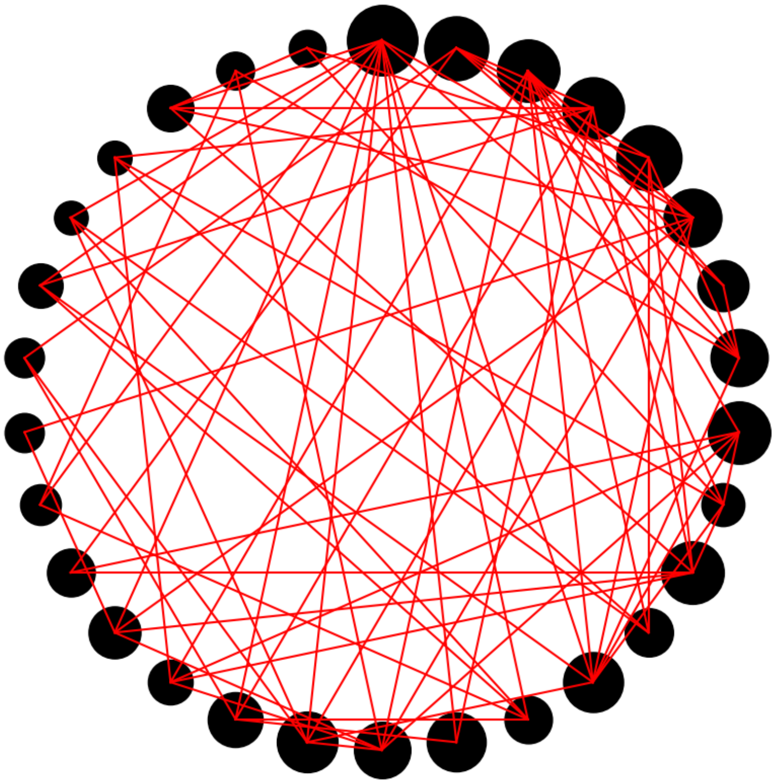

Figure B.4: The resulting network graph of the simulation with RND_1 and STL_4.

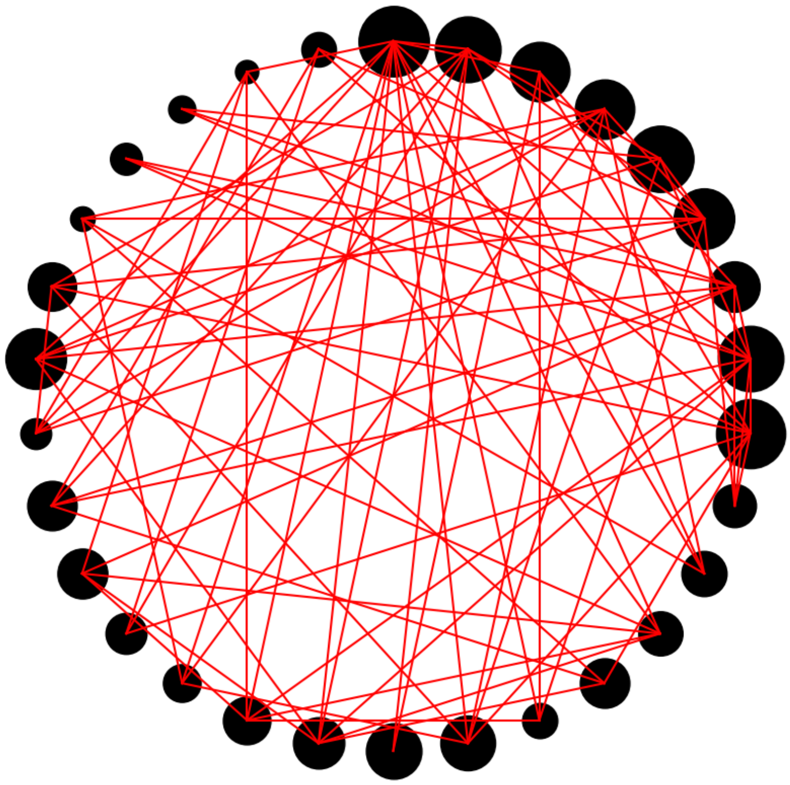

Figure B.5: The resulting network graph of the simulation with RND_1 and STL_5.

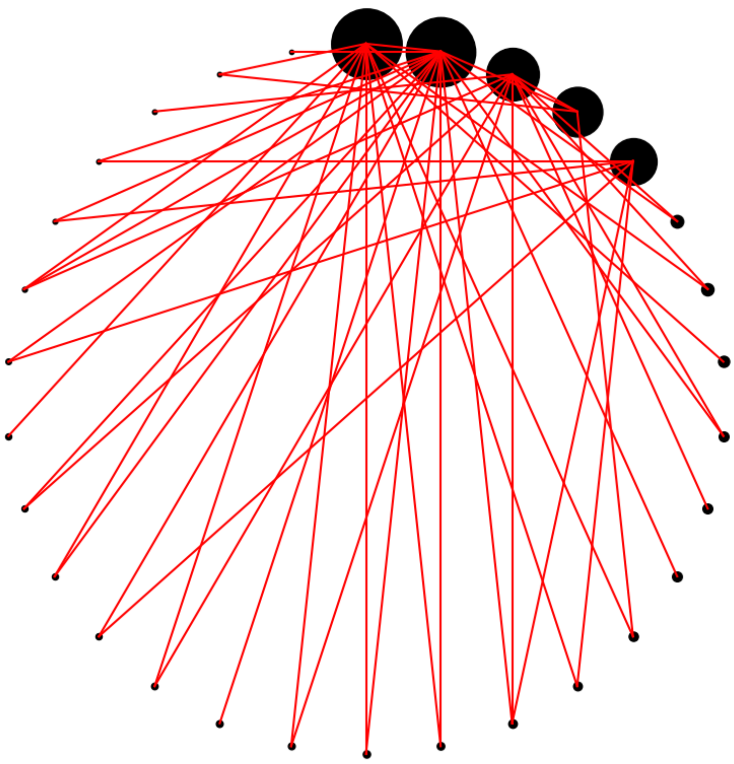

Figure B.6: The resulting network graph of the simulation with CP_1 and BCP_1.

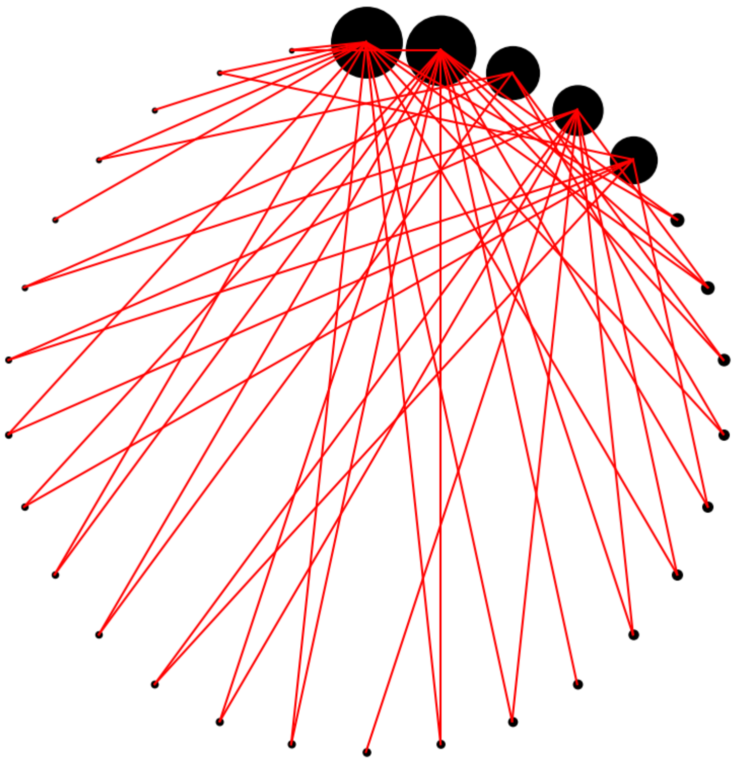

Figure B.7: The resulting network graph of the simulation with CP_1 and BCP_2.

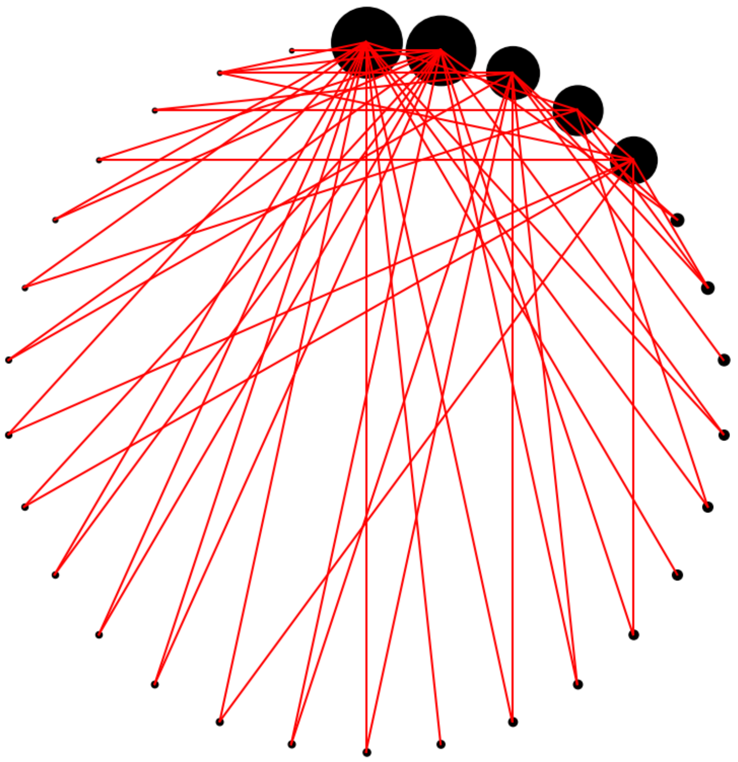

Figure B.8: The resulting network graph of the simulation with CP_1 and BCP_3.

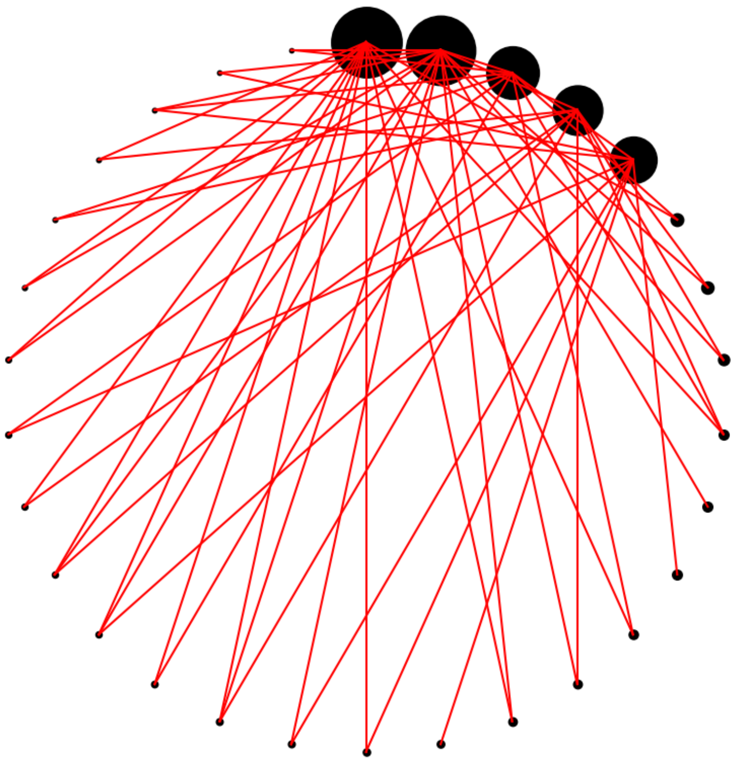

Figure B.9: The resulting network graph of the simulation with CP_1 and BCP_4.

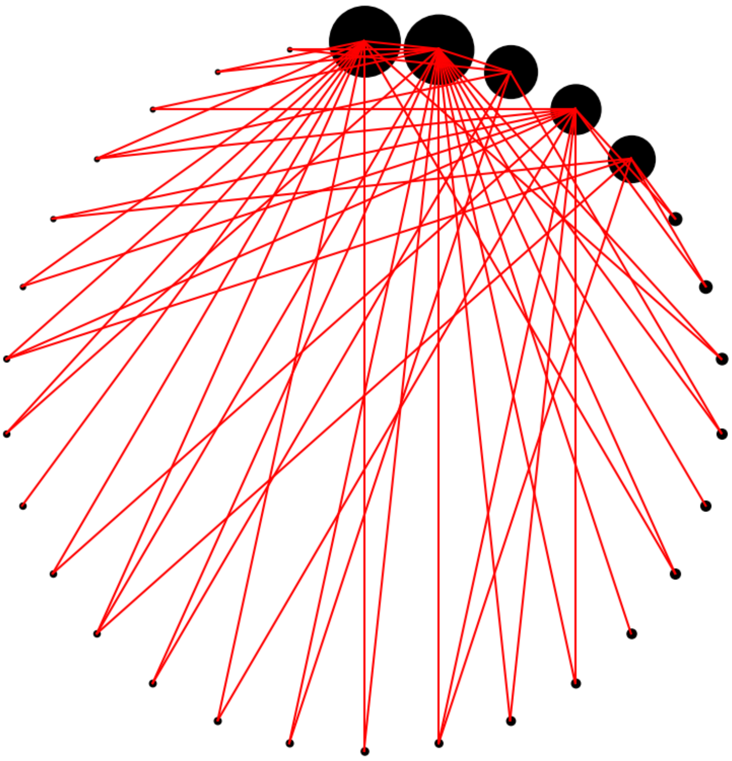

Figure B.10: The resulting network graph of the simulation with CP_1 and BCP_5.

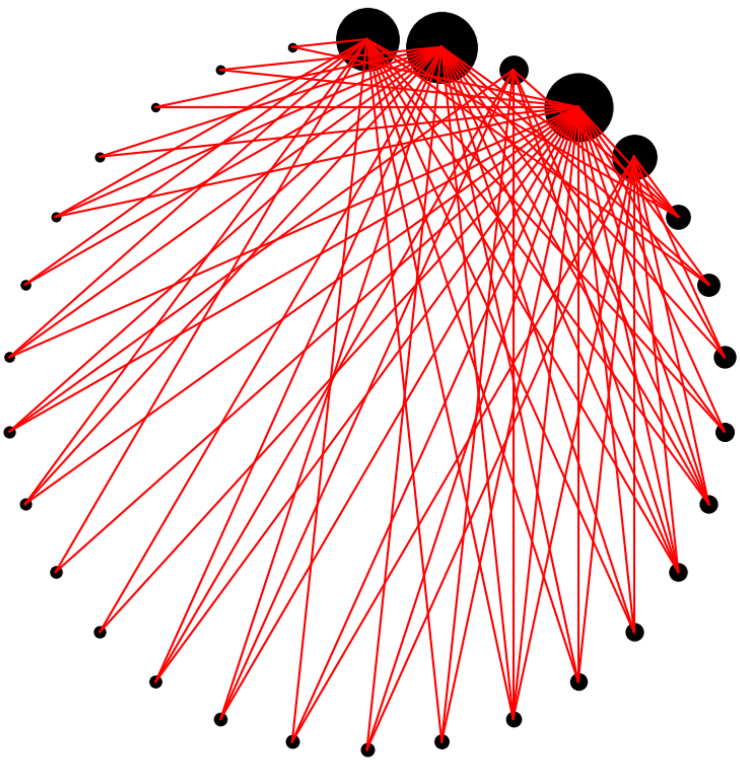

Figure B.11: The resulting network graph of the simulation with CP_1 and UCP_1.

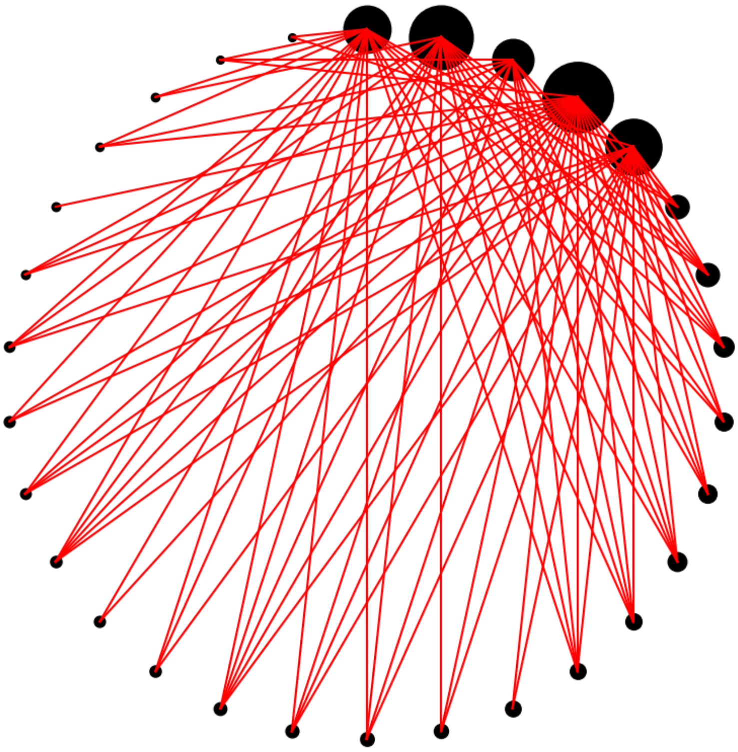

Figure B.12: The resulting network graph of the simulation with CP_1 and UCP_2.

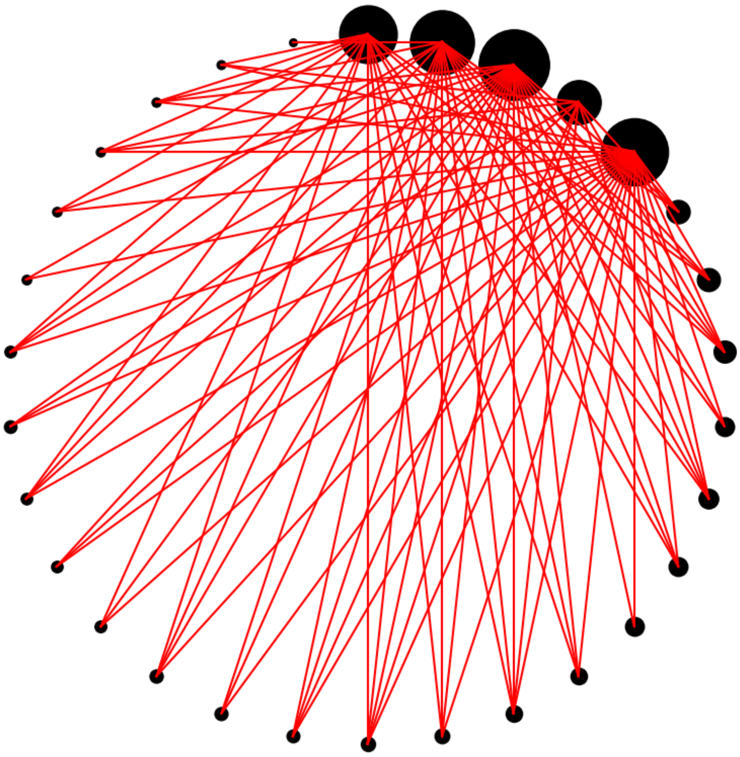

Figure B.13: The resulting network graph of the simulation with CP_1 and UCP_3.

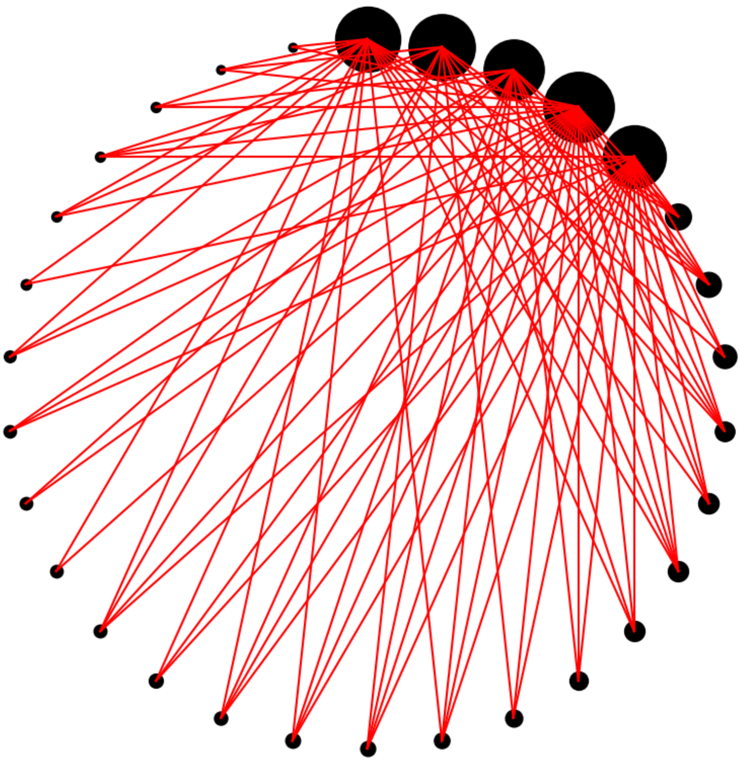

Figure B.14: The resulting network graph of the simulation with CP_1 and UCP_4.

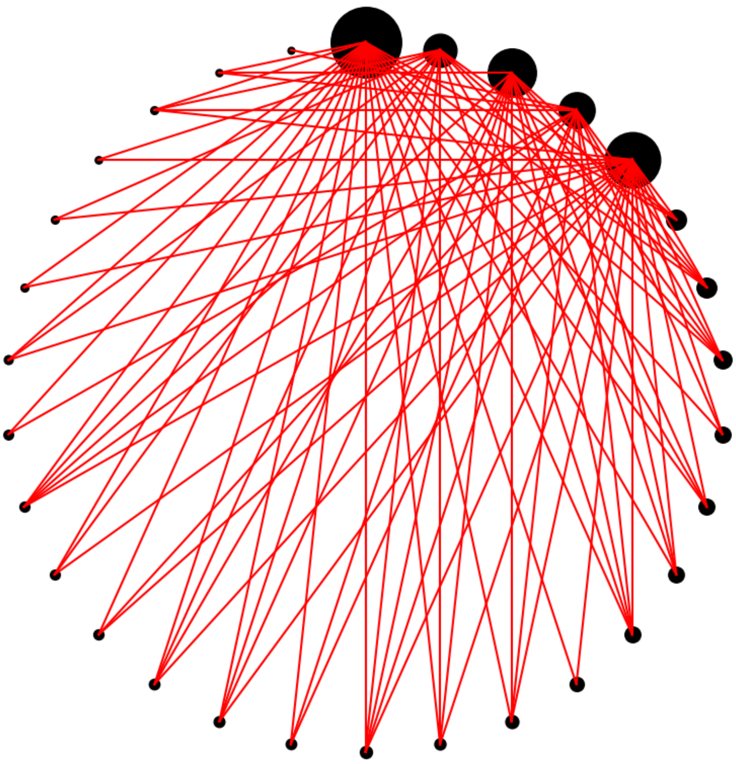

Figure B.15: The resulting network graph of the simulation with CP_1 and UCP_5.