A Mathematical Theory of Payment Channel Networks

Rene Pickhardt[*]

July 9, 2024

Abstract

In this study, we explore the geometry of payment channel networks, such as the Lightning Network, which consist of n peers maintaining m payment channels. We demonstrate that the space L_G of liquidity states can be modeled as an m-dimensional hyperbox in $\mathbb{Z}^m$. By considering the transfer of wealth, we derive the $n-1$-dimensional bounded convex polytope W_G representing feasible distributions of C coins among the peers. This polytope is a subset of the hyperplane defined by $w_1 + \cdots + w_n = C$.

We prove that W_G is the quotient space of L_G when states resulting from circular self-payments are identified. Given that circular self-payments do not alter wealth distribution (neglecting routing fees), we study the quotient space $L_G/\sim_\pi$, where $\sim_\pi$ is the equivalence relation from these payments. We explicitly provide the isomorphism $W_G \cong L_G/\sim_\pi$.

Using geometric arguments like the concentration of measure, we predict phenomena such as channel depletion and infeasible payments. These issues, often attributed to suboptimal routing and liquidity management, are shown to be inevitable due to the network's geometric properties and the lack of credit between peers. Notably, the rate of feasible payments declines exponentially as the network size increases.

Finally, through Monte Carlo simulations and random walks, we present empirical evidence that k-party payment channel networks can achieve higher reliability and better service levels than two-party networks. The average access to liquidity for participants in k-party networks increases linearly with k, with a slope of $\frac{C}{n}$.

1 Introduction

The Bitcoin protocol faces significant limitations, notably its support for only about seven transactions per second and the requirement for users to wait several minutes on average for transaction confirmation. To address these issues, the Lightning Network was developed as a second-layer protocol. Barring exploitation of known DoS attacks [11, 20, 23], the Lightning Network allows users to conduct near real-time Bitcoin payments. Additionally, it technically supports a higher payment throughput compared to the Bitcoin protocol.

However, these enhancements come at the cost of reduced payment reliability on the Lightning Network, particularly as the desired payment amounts increase. To quantify this reduced reliability, we examine the geometry of two-party payment channel networks and their generalization to k-party payment channel networks. By analyzing the high-dimensional geometric properties of these networks, we explain the emergence of two phenomena frequently observed by Lightning Network participants, often associated with reduced payment reliability:

1. **Channel Depletion**: Most of the liquidity in payment channels is likely to be controlled by one of the two peers maintaining the channel.

2. **Infeasible Payments**: Even with numerous payment attempts and optimal routing techniques, a fraction of payments cannot be fulfilled.

Node operators often cite two primary reasons for channel depletion. First, the lack of a circular economy among network users results in the existence of source and sink nodes, leading to natural channel drains [13]. Second, it is believed that node operators have not yet optimized routing fee settings [25], which affect senders' payment route selections. Routing fees are thought to be useful for better flow control and channel balance [18]. Payment failure rates are often attributed to channel depletion [1], suboptimal payment routing strategies, liquidity management issues within channels [14], and users' incomplete information about the network's liquidity state [16].

We observe a contradiction between these reasons for channel depletion. While routing fees impact flow control, they cannot alter the distribution of payment requests among network participants, which leads to the emergence of sources and sinks. Regarding payment failure rates, the uncertainty about channel states has been addressed [16], and optimal methods for making routing decisions despite incomplete information are known [15]. Probability density functions for the depleted channel model also exist [3, 19].

[*] Due to personal circumstances related to my mental health I have not been able to receive a PhD - despite being a researcher since 2011. If you can give someone a PhD and think this work and my prior work about reliability in payment channel networks since 2018 reaches the merits of a PhD, then please feel free to get in contact with me. Thank you!

2 Review of some Graph Theory to describe Payment Channel Networks

We begin by reviewing the standard mathematical model used to describe the Lightning Network. For simplicity, we ignore routing fees and other metadata of channels throughout this document. The Lightning Network is typically modeled as an undirected weighted multigraph. However, for our considerations, multi-edges can be combined into one larger edge. Thus, we start with a weighted and undirected graph $G(V, E, cap)$. V corresponds to the set of $n = |V|$ vertices (also known as peers) in the network. E contains $m = |E|$ elements of $V \times V$. These are the m edges (also known as payment channels). The weighting of the edges corresponds to the **capacity** of the channel via the function $cap : E \longrightarrow \mathbb{N}$. We write $c_e = cap(e)$ and call $C = \sum_{e \in E} c_e$ the total capacity of the network. This is also called the total liquidity or the number of coins in the payment channel network.

The protocol specifies how many coins are owned by each of the peers within each channel.

Definition 2.1. *Let $e = (u, v) \in E$. We call $e_u, e_v \in \{0, \ldots, cap(e)\}$ the liquidity that u and v respectively own in the channel e.*

We can represent the assignment of liquidity in each channel (or the network state) to its peers using the following liquidity function:

Definition 2.2. *The liquidity function of the network $\lambda : E \times V \longrightarrow \{0, \ldots, C\}$ is defined through:*

$$\lambda(e, x) = \begin{cases} e_x & \text{if } x \in e \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Due to the design and properties of payment channels the channel states, and thus the $2 \cdot m$ variables e_x with $e \in E$ and $x \in e$, are not independent of each other. The constraints on the liquidity function are so crucial for the remainder of this document that we assign them a specific term:

Definition 2.3. *The constraints that the protocol imposes on the liquidity function*

$$\begin{aligned} \lambda(e, u) + \lambda(e, v) &= cap(e) \\ \Leftrightarrow e_u + e_v &= c_e \end{aligned} \tag{2}$$

is called ***conservation of liquidity****.*

The principle of conservation of liquidity implies that coins attached to a channel cannot leave it without an on-chain transaction. Despite this restriction, the ownership of coins can be routed through the network of channels as a network flow from one peer to another. However, not all payments are feasible due to the constraints imposed by the network's topology and liquidity function on the feasible flows.

Definition 2.4. *Given a fixed network $G(V, E, cap)$ and a fixed liquidity state of G defined by a liquidity function λ, we can create the associated liquidity network $\mathcal{L}(G, \lambda)$. $\mathcal{L}(G, \lambda)$ is a directed flow network (V', E', c) with a capacity function $c : E' \longrightarrow \mathbb{N}$. We set $V' = V$. For every $e = (u, v) \in E$, we add two directed edges (u, v) and (v, u) to E' such that $c(u, v) = \lambda(e, u)$ and $c(v, u) = \lambda(e, v)$.*

The liquidity network $\mathcal{L}(G, \lambda)$ thus encodes the possible flows of coins through the network G in a given state λ. A payment of amount a between users i and j is feasible if and only if the minimum ij-cut in $\mathcal{L}(G, \lambda)$ is greater than a. Determining the feasibility of a payment in a given state λ can be achieved in almost linear time [4, 5, 24]

3 Polytope of Liquidity States of a Payment Channel Network

The liquidity function is generally unknown to the peers within the network. Peers may have partial knowledge of the liquidity function in the region surrounding their own position within the network. For instance, a peer $v \in V$ is aware of the liquidity in their own channels. Also peers may learn about the current liquidity function while attempting to deliver payments or through actively probing the network [22]. Rather than focusing on or estimating a specific liquidity function, we address this uncertainty by examining the set of feasible liquidity functions and the geometry of that set.

Definition 3.1. *For a fixed payment channel network $G(V, E, cap)$, we define the set of all feasible liquidity functions, or the set of liquidity states of the network, as:*

$$L_G = \{\lambda : E \times V \longrightarrow \{0, \ldots, C\} \mid \lambda(e, u) + \lambda(e, v) = cap(e)\}.$$

This definition motivates the use of integer linear programming to examine L_G because the constraints are linear equations.

Lemma 3.1. *The set L_G of all feasible liquidity states can be embedded into $\mathbb{Z}^{2m}$.*

Proof. Consider a basis $\beta = \{b_{e,x} \in \mathbb{Z}^{2m} \mid e \in E, x \in e\}$ where $b_{e,x}$ are the $2m$ basis vectors that span $\mathbb{Z}^{2m}$. There is a natural embedding:

$$\begin{aligned} \iota : L_G &\hookrightarrow \mathbb{Z}^{2m}, \\ \lambda &\overset{\iota}{\longmapsto} \sum_{e \in E, x \in e} e_x \cdot b_{e,x} \\ &= \sum_{e=(u,v) \in E} \left(e_u \cdot b_{e,u} + e_v \cdot b_{e,v} \right). \end{aligned}$$

□

Let $e_i = (u_i, v_i)$ be the i-th edge, then we can fix an order of β via: $\beta = \langle b_{e_1,u_1}, b_{e_1,v_1}, \ldots, b_{e_m,u_m}, b_{e_m,v_m} \rangle$. In particular, we write:$c_i = \text{cap}(e_i)$.

Lemma 3.2. *The set L_G of feasible liquidity functions on a network G is isomorphic to an m-dimensional hyperbox $H_G \subset \mathbb{Z}^m$. In particular:*

$$H_G = \{0, \dots, c_1\} \times \dots \times \{0, \dots, c_m\}.$$

Proof. We prove this by showing that the forgetful function $f : \iota(L_G) \longrightarrow H_G$ defined as

$$\iota(\lambda) = \sum_{e=(u,v)\in E} (e_u \cdot b_{e,u} + e_v \cdot b_{e,v}) \xmapsto{f} \sum_{e=(u,v)\in E} e_u \cdot b_{e,u}$$

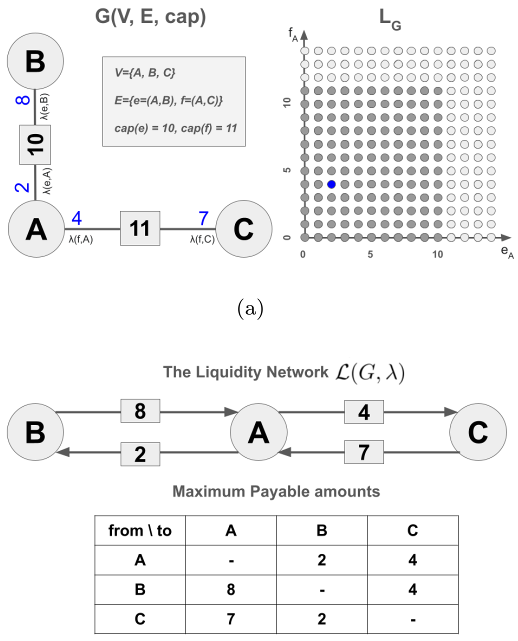
is bijective. This involves showing that f is both injective and surjective.

Injectivity: Let $\lambda \neq \mu \in L_G$. There must be at least one edge $e = (u, v) \in E$ such that $\lambda(e, u) \neq \mu(e, u)$. Consequently, by definition of f, we have $f(\iota(\lambda)) \neq f(\iota(\mu))$. This is sufficient to show that f is injective.

Surjectivity: To show that f is surjective, take an arbitrary point $x = (x_1, \dots, x_m) \in H$. We choose λ such that for all $e_i \in E$: $\lambda(e_i, u_i) = x_i$. Because of the conservation of liquidity, we have: $\lambda(e_i, v_i) = \text{cap}(e_i) - \lambda(e_i, u_i) = c_i - x_i$. Since $x_i \in \{0, \dots, c_i\}$, we can conclude that both $\lambda(e_i, u_i) = x_i$ and $\lambda(e_i, v_i) = c_i - x_i$ take values from $\{0, \dots, c_i\}$. Thus, we have constructed λ such that $f(\lambda) = x$, showing that f is also surjective.

This concludes our proof that f is bijective and that L_G is isomorphic to the m-dimensional hyperbox H_G. □

We can see a low dimensional visualization of the introduced concepts in figure 1.



(a)

Maximum Payable amounts

from \ to	A	B	C
A	-	2	4
B	8	-	4
C	7	2	-

(b)

Figure 1: An Example Network G with corresponding Polytope L_G of liquidity states and liquidity network $\mathcal{L}(G, \lambda)$ for a fixed $\lambda \in L_G$.

For two-party channels, as currently deployed on the Lightning Network, knowing the liquidity of one peer is sufficient to determine the entire state of the channel. Typically, e_v is only known to the peers u and v who maintain the channel e. The state e_v can change over time as the node conducts payments or fulfills routing requests from other nodes.

Corollary 3.2.1. *The number of feasible network states is given by the volume of L_G which can be computed as:*

$$vol(L_G) := |L_G| = \prod_{i=1}^{m} (c_i + 1).$$

Assuming the number of channels m divides the total number of coins C, and using the inequality of arithmetic and geometric means [21], it can be shown that the network in which all channels have the same capacity $\frac{C}{m}$ maximizes the volume of the corresponding polytope L_G of liquidity states. In particular, the maximum number of feasible network states is given by:

$$\text{vol}(L_G) = \left(\frac{C}{m} + 1\right)^m,$$

and is depicted in figure 2 for various number of coins and channels.

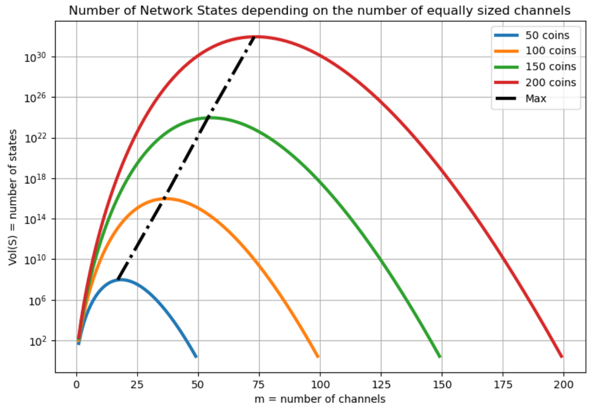


Figure 2: Volumes $\text{vol}(L_G) = \left(\frac{C}{m} + 1\right)^m$ of state polytopes with equally sized channels for various numbers C of coins and various numbers m of channels.

4 Polytope of feasible Wealth Distributions

We have focused on the liquidity states of payment channel networks so far. However, from a user's perspective, their overall wealth might be of more interest than how this wealth is allocated across different channels. Before exploring how payment channel networks impose additional constraints on the polytope of feasible wealth distributions, we will briefly examine the on-chain situation.

Definition 4.1. *Let $V = \{v_1, \dots, v_n\}$ be the set of n Bitcoin users who collectively own C coins. We define a function $\omega : V \longrightarrow \mathbb{N}_0$ as a wealth distribution if and only if $C = \sum_{i=1}^{n} \omega(v_i)$. We denote the set of Bitcoin wealth distributions as*

$$W(C,n) = \{\omega : V \longrightarrow \mathbb{N}_0 \mid C = \sum_{i=1}^{n} \omega(v_i)\}.$$

We use the same methodology as with liquidity functions to show that $W(C,n)$ is isomorphic to the polytope $\mathcal{W}(C,n)$, which is obtained by intersecting the n half-spaces $\mathbb{Z}_i = \{x \in \mathbb{Z}^n \mid x_i \geq 0\}$ with the $n-1$ dimensional hyperplane P_C defined by the linear equation:

$$P(C,n) = \{w \in \mathbb{Z}^n \mid w_1 + \dots + w_n = C\}.$$

Specifically,

$$\mathcal{W}(C,n) = P(C,n) \cap \left(\bigcap_{i=1}^{n} \mathbb{Z}_i \right).$$

To demonstrate this, we choose a basis $\beta = \{b_v \in \mathbb{Z}^n \mid v \in V\}$ with b_u being the n base vectors that span $\mathbb{Z}^n$. Similar to the case with liquidity functions, there is a natural embedding:

$$\iota : W(C,n) \hookrightarrow \mathbb{Z}^n$$
$$\omega \overset{\iota}{\longmapsto} \sum_{v \in V} \omega(v) \cdot b_v$$

Lemma 4.1. *The image of ι is $\mathcal{W}(C,n)$.*

Proof. Let $w \in \mathcal{W}(C,n)$ with

$$w = \sum_{i=1}^{n} w_i \cdot b_{u_i}$$

We select $\omega \in W(C,n)$ such that $\omega(v_i) = w_i$ This is the preimage of $\iota^{-1}(w)$. If $w' \in \mathbb{Z}^n \backslash \mathcal{W}(C,n)$ then no ω with $\iota(\omega) = w'$ exists. □

Because of the embedding, we may refer to wealth distributions as wealth vectors. From the lemma and the stars and bars theorem[1], it follows that the number $|\mathcal{W}(C,n)|$ of wealth distributions for n users with a fixed total of C coins is given by:

$$\text{vol}(\mathcal{W}(C,n)) := |\mathcal{W}(C,n)| = \binom{C+n-1}{n-1} \tag{3}$$

For example, assume the $n = 3$ users Alice, Bob, and Carol have $C = 21$ coins to distribute among themselves. They would have a total of 253 ways to do so. These distributions can be represented in a 2-dimensional polytope:

Knowing the volume of the polytope of feasible Bitcoin wealth distributions will serve as a reference value when studying the feasible wealth distributions W_G in a payment channel network G.

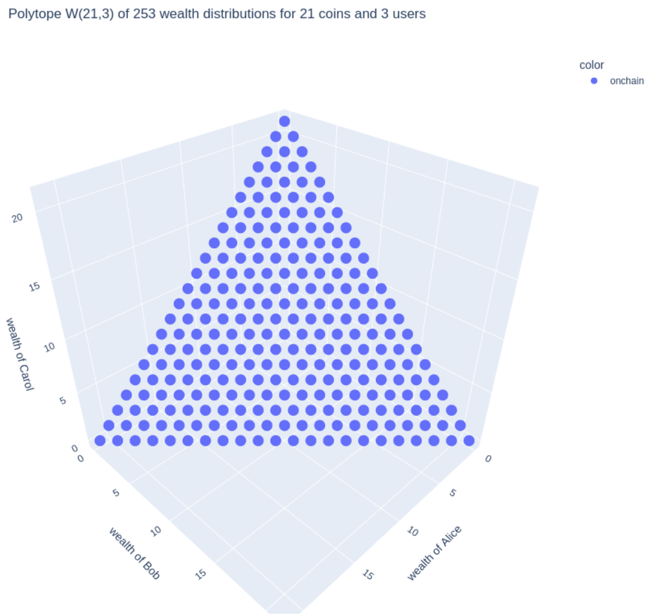


Figure 3: 3-dimensional embedding of the 2-dimensional polytope of wealth distributions of $C = 21$ coins among $n = 3$ users.

4.1 The Polytope W_G of feasible Wealth Distributions of a Payment Channel Network G

Instead of examining all wealth distributions of C coins among n users, we focus on those distributions that are feasible within a given payment channel network $G(V, E, cap)$.

Definition 4.2. *Given a payment channel network $G(V, E, cap)$ with $|V| = n$ users and C coins, a wealth distribution $\omega : V \longrightarrow \{0, \dots, C\}$ is called feasible in G if there exists a liquidity function λ such that for all $v \in V$:*

$$\sum_{e \in E : v \in e} \lambda(e,v) = \omega(v) \tag{4}$$

We denote the set of wealth distributions that are feasible in G as:

$$\{\omega \in W(C,n) | \exists \lambda \in L_G : \sum_{e \in E : v \in e} \lambda(e,v) = \omega(v) \forall v \in V\}$$

This set is referred to as W_G.

Next, we will study how the conservation of liquidity in payment channels reduces the volume of feasible wealth distributions W_G for a given payment channel network.

Lemma 4.2. *Let $G(V, E, cap)$ be a payment channel network. For $n > 2$, there exists $\omega \in W(C,n)$ such that $\omega \notin W_G$.*

[1] https://en.wikipedia.org/wiki/Stars_and_bars_(combinatorics)

Proof. We prove this by contradiction. Let $e = (u, v) \in E$ be an arbitrary channel with capacity $c_e > 0$. Since $n > 2$, there is a user w such that $u \neq w \neq v$. We define $\omega : V \longrightarrow \mathbb{N}$ as:

$$\omega(x) = \begin{cases} C & \text{if } x = w \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

By construction, we have $C = \sum_{x \in V} \omega(x)$. Thus, ω is feasible in $W(C, n)$, and specifically, $\omega(u) = \omega(v) = 0$.

To conclude the proof, assume $\omega \in W_G$. Due to the conservation of liquidity (equation 2) and the definition of ω as a feasible wealth function in G, we know that $\omega(u) + \omega(v) \geq c_e$. This contradicts the assumption that the capacity of the channel was non-zero. □

Corollary 4.2.1. *The number $|W_G|$ of feasible wealth distributions of C coins among $n > 2$ users in a payment channel network is strictly smaller than the number $|\mathcal{W}(C, n)|$ of wealth distributions that are feasible with on-chain transactions.*

We are interested in how the topology of G impacts the relative volume $r(G)$ that W_G has in comparison to $\mathcal{W}(C, n)$:

$$r(G) = \frac{|W_G|}{|\mathcal{W}(C, n)|} \tag{6}$$

For example, consider the network from the previous section in Figure 1. We can observe the feasible region W_G as a subpolytope of $\mathcal{W}(21, 3)$ in Figure 4.

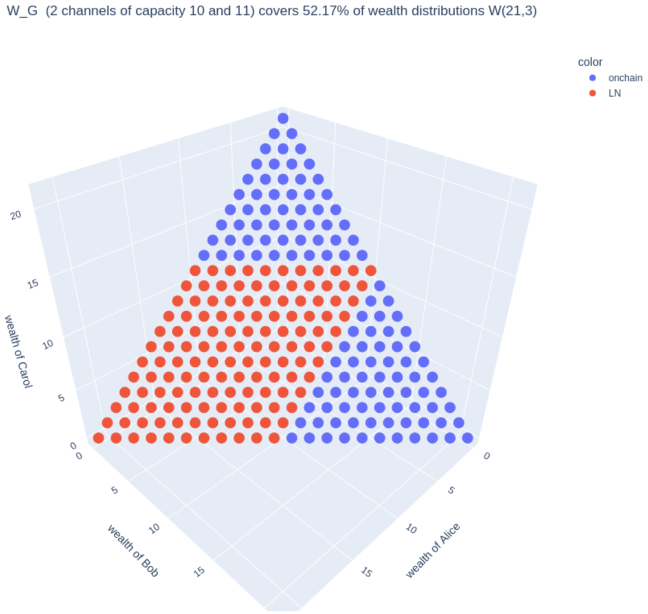
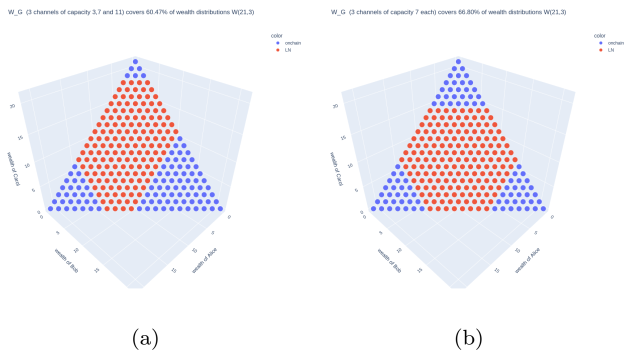


Figure 4: Only 52.17% of all wealth distributions in $\mathcal{W}(21, 3)$ are feasible if Alice, Bob, and Carol allocate the 21 coins into 2 channels of capacities 10 and 11.

We Note that the feasible reagion changes with the topology of the network. If Bob Spliced out 7 coins of his channel with Alice and created a new channel of 7 coins the feasible reagion would increase in size as can be seen in figure 5(a). However the increase is relatively small. It becomes larger if all channels had the same capcity as can be seen in figure 5 (b).



(a) (b)

Figure 5: 2 feasible regions of W_G for 2 different networks between Alice Bob and Carol who own in total 21 coins.

We are not aware of an analytically closed formula that describes the volume $|W_G|$ for a given network G. In particular we are not aware of a smart way to find the topology G that maximizes $r(G)$. However, we can estimate the volume through statistical sampling via Monte Carlo methods. Utilizing the Dirichlet Rescale Algorithm [9], we can uniformly and randomly select wealth distributions ω from $\mathcal{W}(C, n)$.[2] We can then check how often these distributions are feasible for a given network G. This gives us an estimate for $r(G)$:

$$r(G) \approx \frac{\text{number of } \omega \text{ such that } \omega \text{ is feasible in } G}{\text{number of sampled } \omega \in \mathcal{W}(C, n)} \tag{7}$$

In order to be able to evaluate this we need to be able to decide if a given wealth distribution in $\mathcal{W}(C, n)$ is feasible in W_G.

4.2 Deciding if a Wealth Distribution $\omega \in \mathcal{W}(C, n)$ is also feasible in W_G

Due to Lemma 4.2, we know there exists at least one vector $\omega \in \mathcal{W}(C, n)$ that is infeasible and not in W_G. For an arbitrary vector $\omega \in \mathcal{W}(C, n)$, we can determine if it is also an element of W_G and thus a feasible wealth distribution on the payment channel network. According to the definition of a feasible wealth distribution in W_G, we need to find $\lambda \in L_G$ such that for all $e \in E$ and $v \in V$, the following n linear equations hold:

$$\sum_{e \in E : v \in e} \lambda(e, v) = \omega(v) \tag{8}$$

[2] An open source implementation can be found at: `https://github.com/dgdguk/drs`

Due to the conservation of liquidity, λ requires an additional m equations for all $e \in E$:

$$\lambda(e, v) + \lambda(e, u) = \mathrm{cap}(e) \tag{9}$$

Thus, we need to solve a system of $n+m$ linear equations in $2 \cdot m$ variables:

$$\{e_{1,u_1}, e_{1,v_1}, \dots, e_{m,u_m}, e_{m,v_m}\}$$

These equations are not fully independent because summing the n equations from Equation 8 yields the same value as summing the m equations from Equation 9:

$$\sum_{v \in V} \sum_{e \in E : v \in e} \lambda(e, v) = C = \sum_{e=(u,v) \in E} (\lambda(e, v) + \lambda(e, u))$$

Therefore, we can eliminate one of the constraints, resulting in a total of $m + n - 1$ independent equations. Let $\sigma(G, \omega)$ be the solution space of feasible liquidity functions on G for the above system of linear equations over integers. If solved over $\mathbb{Z}^{2 \cdot m}$, the dimension of the solution space $\sigma(G, \omega)$ is at least $2 \cdot m - (m + n - 1) = (m - n + 1)$. However, we are seeking solutions over L_G, which is isomorphic to the bounded hypercube:

$$H_G = \{0, \dots, c_1\} \times \{0, \dots, c_m\}$$

Thus, for a wealth distribution to be feasible and a liquidity function λ to exist, we must require:

$$\sigma(G, \omega) \cap H_G \neq \emptyset \tag{10}$$

Finding a solution for the system of linear equations over integers in a bounded region can be done through integer linear programming. [3]

4.2.1 A Remark on Credit in Payment Channel Networks

The previous geometric example illustrates that the system of linear equations that come from conservation of liquidity and the network topology always has at least one solution over $\mathbb{Z}^{2 \cdot m}$. However if the intersection of the solution space $\sigma(G, \omega) \cap H_G$ is empty this means that all elements of $\sigma(G, \omega)$ have at least once component that is negative. This shows that some wealth distributions are infeasible on payment channel networks because the liquidity of the peers in the channel has to be always positive. If credit was allowed in channels a feasible network state could be constructed for any wealth distribution. A similar result has been shown before the Lightning Network was created [6] Of course credit requirs trust and is consequently not wished for by the users and developers of peer to peer electronic payment systems. However this geometric insight shows how difficult it is to create trustless electronic payment systems.

4.3 Application: Deciding the Feasibility of Payments

Making a payment is equivalent to changing the wealth vector. Let us assume $w = (w_1, \dots, w_n) \in W_G$ is the current wealth distribution of the network. If the i-th user wishes to pay user j the amount a, we can test the feasibility of the payment by checking if

$$w' = w + a \cdot b_j - a \cdot b_i \tag{11}$$

is still inside the polytope W_G of feasible wealth distributions. If $w' \notin W_G$, then given our initial wealth distribution w, the payment between i and j of amount a will fail.

Remarkably, deciding whether a payment is feasible in a given network without conducting additional on-chain operations does not depend on the exact state of the network. Instead, the feasibility of a payment depends only on the network topology and the current wealth distribution. Of course, similar to the liquidity state of the network, the current wealth distribution is unknown.

Thus we take a global perspective. Similarly to estimating $r(G)$, we can estimate the likelihood that a payment is feasible through Monte Carlo methods. For this, we uniformly randomly sample several feasible wealth distributions in W_G and compute how often the resultant distribution after executing the payment is feasible.[4] Node operators can use this to decide where to allocate liquidity an which channels to open or close.

Definition 4.3. *We call ρ the expected rate of infeasible payments.*

In particular, due to Lemma 4.2, it is noted that in payment channel networks, it is impossible for 100% of all conceivable payment requests to be feasible.

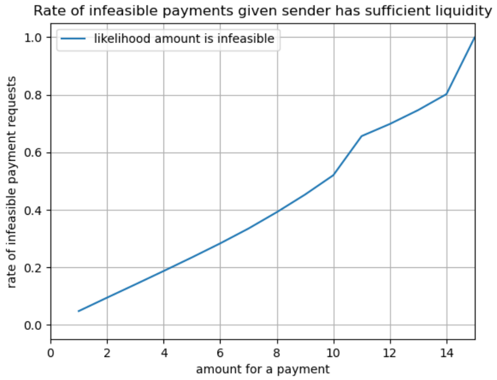


Figure 6: Expected Rate ρ of infeasible payments in the example network between Alice, Bob, and Carol with channels of sizes 3, 7, and 11

[3] As discussed with Stefan Richter. Instead of solving a system of linear equations over a bounded region, one could solve a max flow problem to test feasibility. For this, one would take an arbitrary wealth vector ω' and compute $\omega - \omega'$. The components of the difference are the supply and demand of the nodes in the network. If a multi-source, multi-sink flow exists that fulfills the supply and demand, then $\omega \in W_G$.

[4] This method has already been shared publicly before this paper was published.https://delvingbitcoin.org/t/estimating-likelihood-for-lightning-payments-to-be-in-feasible/973

4.4 Limited Ability of Payment Channel Networks to scale Blockchains

We emphasize that infeasible payment requests require at least one on-chain transaction to be executed successfully. This transaction can either change the wealth distribution and state of the payment channel network via an on-chain/off-chain swapping service or alter the network topology by opening and closing channels or through splicing. If ζ is the number of possible on-chain transactions per second, we can derive the maximum bandwidth of supported Lightning Network payments as:

$$\text{supported payments per second} = \mathcal{S} = \frac{\zeta}{\rho} \tag{12}$$

In the Lightning Network whitepaper [17], the authors state that the Visa Network supports 47,000 payments per second during peak times. Solving the equation for ρ and noting that the Bitcoin network allows $\zeta = 7$ transactions per second, we conclude that only $\rho = \frac{7}{47{,}000} \approx 0.0149\%$ of all conceivable payments can be infeasible if the Lightning Network is supposed have a comparable bandwidth to the visa network.

Thus, for payment channel networks to scale the payment throughput of blockchains, it is crucial that the rate ρ of infeasible payments is close to zero. This happens when the feasible region of wealth distributions W_G is large within $W(C, n)$ or if $r(G)$ is close to 1.

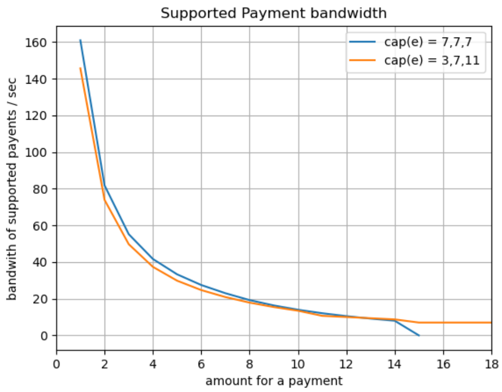


Figure 7: Rate of infeasible payments ρ depending on the desired payment amount a in the example network between Alice, Bob, and Carol with channels of sizes 3, 7, and 11

Using the data from Figure 6 we can derive $\mathcal{S}$ for various networks and payment amounts which is depicted in figure 7 The figure also demonstrates what users of payment channel networks have experienced before: Such networks are mainly capable to scale the bandwidth of small amount payments. There seems a small tradeoff between higher bandwidth but smaller supported maximum sendable amounts for networks with equally sized channels. We will see later in this document how multiparty channel networks values of $r(G)$ that are close to 1.

5 Relation between the Polytope L_G of Liquidity States and W_G of feasible Wealth Distributions

Given the topology of the lightning network as a weighted, undirected graph $G(V, E, cap)$. There is a geometric relation between the corresponding polytopes L_G of liquidity states and W_G of feasible wealth distributions. For every feasable state $\lambda \in L_G$ we have seen that the n participants $\{v_1, \ldots, v_n\}$ have a non negative wealth stored in their channels. Thus we can project any liquidity state to a feasible wealth distribution via the projection:

$$\begin{aligned} \pi : L_G &\longrightarrow W_G \\ \lambda &\xmapsto{\pi} \sum_{e=(u,v)\in E} (\lambda(e,u) \cdot b_u + \lambda(e,v) \cdot b_v) \end{aligned} \tag{13}$$

We proof that the image of π is indeed a subset of W_G. To do this we show that $\pi(\lambda)$ is feasible in W_G.

Lemma 5.1. *Let $w = \pi(\lambda)$. For a given base we can write $w = \sum_{v\in V} w_v \cdot b_v$. Then $\sum_{v\in V} w_v = C$*

Proof. We use the definition of our projection π in equation 13 as well as conservation of liquidity (equation 2) and the definition of C:

$$\begin{aligned} \sum_{v\in V} w_v &= \sum_{v\in V} \sum_{e\in E: v\in e} \lambda(e,v) \\ &= \sum_{e\in E} \sum_{v\in e} \lambda(e,v) \\ &= \sum_{e\in E} c_e \\ &= C \end{aligned} \tag{14}$$

□

Due to the methods introduced in Section 4.2, we can find a preimage of $w \in W_G$ under π. Thus, π is surjective. In topology, any surjective map induces an equivalence relation, from which a quotient space can be constructed.

Definition 5.1. *We call two states $\lambda, \mu \in L_G$ equivalent if and only if they are projected to the same wealth distribution, i.e., $\pi(\lambda) = \pi(\mu)$. In this case, we write $\lambda \sim_\pi \mu$.*

Lemma 5.2. *The relation $\sim_\pi$ is an equivalence relation. In particular, it is reflexive, symmetrical, and transitive.*

Proof. We need to show that the three properties of an equivalence relation are fulfilled:

1. **Reflexivity:** Since $\pi(\lambda) = \pi(\lambda)$, we have $\lambda \sim_\pi \lambda$ for any $\lambda \in L_G$.

2. **Symmetry:** Let $\lambda, \mu \in L_G$ with $\lambda \sim_\pi \mu$, which means $\pi(\lambda) = \pi(\mu)$. Since equality is symmetrical, it follows that $\pi(\mu) = \pi(\lambda)$, which means $\mu \sim_\pi \lambda$.

3. **Transitivity:** Let $\lambda, \mu, \nu \in L_G$ with $\lambda \sim_\pi \mu$ and $\mu \sim_\pi \nu$. We have $\pi(\lambda) = \pi(\mu) = \pi(\nu)$, from which it follows that $\lambda \sim_\pi \nu$.

□

Definition 5.2. *We call $[\lambda] = \{\mu \in L_G \mid \lambda \sim_\pi \mu\}$ the equivalence class of λ. The quotient space $L_G/\sim_\pi$ is the space of equivalence classes.*

In particular, the equivalence class $[\lambda] = \pi^{-1}(\{w\})$ is the preimage of $w \in W_G$ under the projection π. It follows that

$$L_G/\sim_\pi \cong W_G \tag{15}$$

for which we have explicitly provided the isomorphism. In one direction, it is the projection of the channel states to the wealth distribution as defined by π, and in the other direction, it is the construction of a feasible state for a feasible wealth distribution as described in Section 4.2.

5.1 Rebalancing Payment Channels and Circulations

We aim to study the number of elements in the equivalence class $|[\lambda]|$. Recall that for $\lambda, \mu \in [\lambda]$, we have $\pi(\lambda) = \pi(\mu)$. Thus, λ and μ project to the same wealth distribution. When routing fees are ignored, circular self-payments, also known as channel rebalancings, are the only payments that do not change the wealth distribution. However, they do alter the liquidity state of the network.

We will now prove that the size of the equivalence class $|[\lambda]|$ is equal to the number of strict circulations that exist on $\mathcal{L}(G, \lambda)$. Also, if $\pi(\lambda) = w$, then $|\pi^{-1}(\{w\})|$ is equal to the number of circulations.[5] Counting the number of circulations is possible with Ehrhart's theory [8] or by applying Barvinok's algorithm [2]. In particular, free open source software[6] exists [7] that can achieve this.

To follow these observations, we review a few elements from the theory of network flows.

Definition 5.3. *A flow network is a directed graph $G = (V, E)$ with a capacity function $c : E \to \mathbb{Z}^+$ and a flow function $f : E \to \mathbb{Z}$ such that $0 \leq f(e) \leq c(e)$ for all $e \in E$.*

Certain flows are of particular interest:

Definition 5.4. *A flow f is called a circulation if at every vertex $x \in V$ we have:*

$$\sum_{(u,x)\in E} f(u,x) = \sum_{(x,v)\in E} f(x,v) \tag{16}$$

Due to the particular construction of the liquidity graph $\mathcal{L}(G, \lambda)$, we require a more precise definition of circulations.

Definition 5.5. *In bidirectional flow networks, where for any $(u,v) \in E$ there is also $(v,u) \in E$, a circulation f is called strict if for all edges $(u,v) \in E$, we have either $f(u,v) = 0$ or $f(v,u) = 0$, or both $f(u,v) = 0 = f(v,u)$.*

We now demonstrate that there is a one-to-one correspondence between strict circulations on $\mathcal{L}(G, \lambda)$ and the elements of $[\lambda]$. First, let f be a circulation on $\mathcal{L}(G, \lambda)$. We show that λ', defined pointwise for every $e = (u,v)$ by

$$\lambda'(e,u) = \lambda(e,u) + f(v,u) - f(u,v) \tag{17}$$

and

$$\lambda'(e,v) = \lambda(e,v) + f(u,v) - f(v,u), \tag{18}$$

is indeed a member of $[\lambda]$. To establish this, we prove the following lemma.

Lemma 5.3. *Let $G(V, E, cap)$ be a payment channel network in an arbitrary feasible state $\lambda \in L_G$. Then $\lambda' \in L_G$ and, in particular, $\pi(\lambda) = \pi(\lambda')$.*

Proof. We use the fact that $\lambda \in L_G$ and conservation of liquidity to show that conservation of liquidity also holds for λ':

$$\begin{aligned}
\lambda'(e,u) + \lambda'(e,v) &= \lambda(e,u) + f(v,u) - f(u,v) \\
&\quad + \lambda(e,v) + f(u,v) - f(v,u) \\
&= \lambda(e,u) + \lambda(e,v) \\
&\quad \underbrace{+ f(u,v) - \underbrace{f(v,u) + f(v,u)}_{=0} - f(u,v)}_{=0} \\
&= \lambda(e,u) + \lambda(e,v) \\
&= \mathit{cap}(e)
\end{aligned}$$

Furhter more we have:

$$\lambda'(e,u) = \underbrace{\lambda(e,u) - \underbrace{f(u,v)}_{\leq \lambda(e,u)}}_{\geq 0} + \underbrace{f(v,u)}_{\geq 0} \geq 0 \tag{19}$$

and

$$\lambda'(e,u) = \underbrace{\lambda(e,u) + \underbrace{f(v,u)}_{\leq \mathit{cap}(e) - \lambda(e,u)}}_{\leq \mathit{cap}(e)} - \underbrace{f(u,v)}_{\geq 0} \leq \mathit{cap}(e) \tag{20}$$

This shows that $0 \leq \lambda'(e,u) \leq \mathit{cap}(e)$ indicating $\lambda' \in L_G$

To show that $\pi(\lambda) = \pi(\lambda')$, we recall the definition of λ':

$$\lambda' = \lambda + \underbrace{\sum_{(u,v)\in E} \big(f(u,v) \cdot b_{(u,v),u} - f(u,v) \cdot b_{(u,v),v}\big)}_{\mu}$$

[5] We are not the first to note that circular self-payments do not change the wealth distributions of the participants in a payment channel network [12].

[6] https://www.math.ucdavis.edu/~latte/

Since π is linear, we have $\pi(\lambda') = \pi(\lambda) + \pi(\mu)$. Thus, to show $\pi(\lambda) = \pi(\lambda')$, it is sufficient to prove that $\pi(\mu) = 0$.

$$\pi(\mu) = \sum_{(u,v)\in E} (f(u,v) \cdot b_u - f(u,v) \cdot b_v) \tag{21}$$

In particular, for an arbitrary $x \in V$ the x-th component of $\pi(\mu)$ can be written as:

$$(\pi(\mu))_x = \sum_{(x,v)\in E} f(x,v) - \sum_{(u,x)\in E} f(u,x)$$

Because of the conservation of flow and f being a circulation, this equation equals zero, so $(\pi(\mu))_x = 0$.

Therefore, $\pi(\mu) = 0$ and $\pi(\lambda) = \pi(\lambda')$. □

After establishing that a circulation f on $\mathcal{L}(G, \lambda)$ results in a new liquidity state λ', we aim to show that the provided mapping is injective.

Lemma 5.4. *Let $f \neq g$ be strict circulations on $\mathcal{L}(G, \lambda)$. Then the associated liquidity states $\lambda_f \neq \lambda_g$.*

Proof. We prove this by contradiction. Assume $\lambda_f = \lambda_g$. This means that for all $e = (u, v) \in E$, we have:

$$\begin{aligned} \lambda(e,u) + f(v,u) - f(u,v) &= \lambda(e,u) + g(v,u) - g(u,v) \\ \Leftrightarrow f(v,u) - f(u,v) &= g(v,u) - g(u,v) \end{aligned}$$

Because f and g are strict circulations, at least one term on each side of the equation equals zero. We consider all cases, noting that $f, g \geq 0$:

1. **Case:**

$$f(v,u) = g(v,u) = 0 \Rightarrow -f(u,v) = -g(u,v)$$

2. **Case:**

$$f(u,v) = g(u,v) = 0 \Rightarrow f(v,u) = g(v,u)$$

3. **Case:**

$$f(v,u) = g(u,v) = 0 \Rightarrow -f(u,v) = g(v,u) = 0$$

4. **Case:**

$$f(u,v) = g(v,u) = 0 \Rightarrow f(v,u) = -g(u,v) = 0$$

This shows that $f = g$, which contradicts our assumption that $f \neq g$. Therefore, $\lambda_f \neq \lambda_g$. □

Finally, we prove that our correspondence between strict circulations and equivalent liquidity states is surjective.

Lemma 5.5. *For any $\lambda' \in [\lambda]$, there exists a strict circulation f on $\mathcal{L}(G, \lambda)$ such that $\lambda'(u,v) = \lambda(u,v) + f(v,u) - f(u,v)$.*

Proof. For any $e = (x, y)$, we define:

$$f(x,y) = \begin{cases} \lambda(e,x) - \lambda'(e,x) & \text{if } \lambda(e,x) \geq \lambda'(e,x) \\ 0 & \text{otherwise} \end{cases}$$

From the definition, it follows that:

$$0 \leq f(x,y) \leq c(x,y)$$

The x-th component of the wealth vector $\pi(\lambda)$ is computed as:

$$(\pi(\lambda))_x = \sum_{e\in E: x\in e} \lambda(e,x)$$

Since $\lambda' \in [\lambda]$, we have $\pi(\lambda') = \pi(\lambda)$. It follows for any $x \in V$:

$$0 = \sum_{e\in E: x\in e} (\lambda(e,x) - \lambda'(e,x))$$

If $\lambda(e,x) - \lambda'(e,x) \geq 0$, then we have:

$$f(x,y) = \lambda(e,x) - \lambda'(e,x)$$

Otherwise, we use the conservation of liquidity to show:

$$\begin{aligned} \lambda(e,x) - \lambda'(e,x) &= c(e) - \lambda(e,y) - (c(e) - \lambda'(e,y)) \\ &= c(e) - \lambda(e,y) - c(e) + \lambda'(e,y) \\ &= -\lambda(e,y) + \lambda'(e,y) \\ &= -\underbrace{(\lambda(e,y) - \lambda'(e,y))}_{f(y,x)} \\ &= -f(y,x) \end{aligned}$$

Therefore, we can replace $\lambda(e,x) - \lambda'(e,x)$ with $f(x,y) - f(y,x)$, yielding:

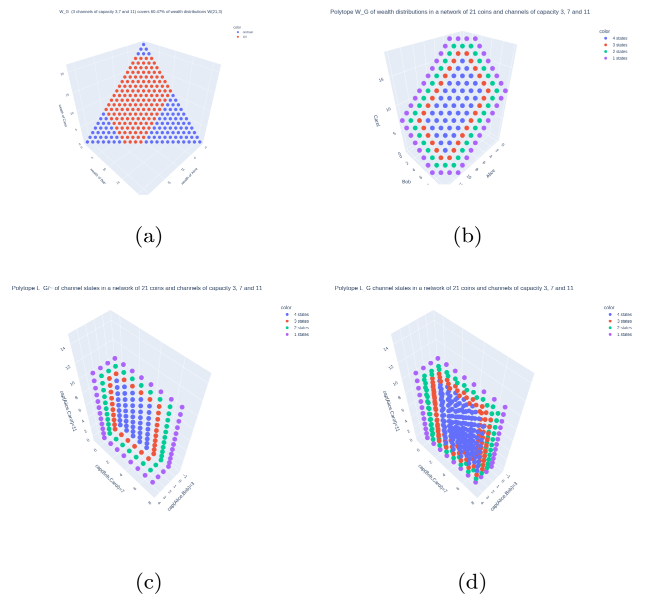
$$\begin{aligned} 0 &= \sum_{e\in E: x\in e} (\lambda(e,x) - \lambda'(e,x)) \\ &= \sum_{e\in E: x\in e} (f(x,y) - f(y,x)) \\ &= \sum_{e\in E: x\in e} f(x,y) - \sum_{e\in E: x\in e} f(y,x) \\ \Leftrightarrow \sum_{(y,x)} f(y,x) &= \sum_{(x,y)} f(x,y) \end{aligned} \tag{22}$$

This proves that f is a strict circulation on $\mathcal{L}(G, \lambda)$. □

From these lemmas, the main theorem follows:

Theorem 5.6. *The fiber $\pi^{-1}(\{w\})$ of any feasible wealth distribution $w \in W_G$ is an equivalence class $[\lambda]$ of liquidity states. The number of strict circulations on $\mathcal{L}(G, \lambda)$ is the same as the number $|[\lambda]|$ of distinct liquidity states of the network with the same fixed wealth distribution $w \in W_G$.*

Circular rebalancing of liquidity does not change the feasibility of a payment since circulations leave the wealth distribution invariant, and the feasibility of payments is



(a) (b)

(c) (d)

Figure 8: (a) shows the feasible region W_G as in figure 5 (b) depicts the size of the equivalence classe for each feasible wealth distribution. (c) Space $L_G/\sim_\pi$ of equivalance classes as a subset of the surface of L_G (d) full polytope L_G of all liquidity states

determined by testing if the change in wealth distribution still results in a feasible wealth distribution. However, depending on which liquidity state $\lambda \in \pi^{-1}(\{w\})$ the network is in, the speed at which payment planning strategies of nodes can find the necessary liquidity for feasible payments can be impacted.

Figure 8 demonstrates in 3-dimensional geometry how the L_G in combination with π can be seen as a bundle over W_G and how $W_G \cong L_G/\sim_\pi$. In particular in subfigure (c) one can realize that the feasible region W_G corresponds to just 3 sides of the hypercube:

$$H_G = \{0,1,2,3\} \times \{0,2,3,4,5,6,7\} \times \{0,\ldots,11\}$$

Each side corresponds to the subspace in which one dimension (state) equals 0. Meaning the liquidity state which is rebalanced so long until one channel is depleted. This obviously works only if cycles are present and not in trees.

5.2 Special Case of Spanning Trees

Let us assume G has the shape of a tree. In this case,

$$m = |E| = |V| - 1 = n - 1$$

We recall that the dimension of $\sigma(G,\omega)$ was: $m - n + 1$. Replacing m with $n-1$, we see that the dimension of $\sigma(G,\omega)$ is:

$$m - n + 1 = (n-1) - n + 1 = 0$$

A zero-dimensional space is just a single point. If ω is feasible, this point lies in W_G, and the corresponding liquidity function is unique. Thus, in tree-shaped networks, there is a one-to-one correspondence between the set W_G and the set of feasible liquidity functions L_G. In particular, we can write:

$$W_G \cong L_G$$

The careful reader may have noticed that the polytope of feasible liquidity states L_G in Figure 1 had the same number of points as the feasible region W_G in Figure 4. As we have seen, this is not surprising but to be expected.

This means that for a uniform distribution of feasible wealth distributions in a spanning tree, the liquidity in channels would also be uniformly distributed, preventing depletion. This is remarkable because it has been shown [10] that under the assumption of balanced flows, there is a stable state in which the liquidity within most channels is depleted, but there is a spanning tree in which the liquidity is uniformly distributed.

6 Channel Depletion as a Consequence of the Goemetry of Cycles

WARNING! FROM HERE ON THE PAPER NEEDS TO BE REWRITTEN AND FINALIZED We have seen that under the assumption that all feasible wealth distributions in W_G are equally likeli that in spanning trees the liquidity states is also uniformly distributed. In particular the channel states are equally likely and idependent of each other.

In figure 8 (b) we have seen that the the feasible wealth distributions $w \in W_G$ with $|\lambda| = |\pi^{-1}(\{w\})|$ are located on the boundary of W_G. Assuming every node in the network is part of a cycle then the wealth distributions with $|\lambda| = 1$ are because of theorem 5.6 those that do only have the trivial circulation ($f = 0$) on the associated liquidity network $\mathcal{L}(G,\lambda)$. The fact that no other circulations exists means that at least one channel has to be depleted. It is a well known fact most of the volume of high dimensional convex bodyes is located close to the boundary. To see this for an n-dimensional hyper we realize that its n dimensional volume is described as:

$$vol = l^n$$

If we want to see how much volume is on the boundary we have to substract the volume of a cube that has a length of $l-2$:

$$vol(boundary) = l^n - (l-2)^n$$

Use concentration of measure to show how depletion has to occur.

7 Multiparty channels

Everyhing we saw generalizes easily to k-party channels. A k-party channel is a channel is an element in V^k thus the channels build a subset $E_k \subset V^k$. The capacity function $cap_k : E_k \longrightarrow \mathbb{N}$ is generalized. We for $2 \leq k \leq n$ we call $G_k(V, E_k, cap_k)$ a k-party payment channel network. If $k == 2$ we omit k and just have a regular graph. Obviously the principle of conservation of liquidity must still hold:

$$cap(e) = \sum_{x \in e} \lambda(e, x) \forall e \in E_k \tag{23}$$

Obviously the number of possible k party channels in a network of n participants is defined by:

$$\binom{n}{k}$$

Thus $|E_k| \leq \binom{n}{k}$.

We can easily see that the hypberbox B which describes the variables is $k \cdot m$ dimensional. If the m linear independent constraints are divided out we get the state polytope $S_k \subset \mathbb{Z}^{(k-1)\cdot m}$. We again have the cannonical projection $\pi : S \longrightarrow W_{G_k}$ to the polytope of feasible wealth distributions with k-party channels.

Definition 7.1. *Let $G_k(V, E_k, cap)$ be a k-party channel network. We call $G_{k+1}(V, E_{k+1}, cap_{k+1})$ an extension if*

$$|E_{k+1}| \leq |E_k|$$

and

$$\forall e = \{x_1, \ldots, x_k\} \in E_k \quad \exists! e' \in E_{k+1} : e \subset e'$$

and

$$cap_{k+1}(e') = \sum e \in \epsilon cap_k(e)$$

with $\epsilon = \{e_1, \ldots, e_N\} \subset E_k$ be the subset of edges in E_k such that $|e' \backslash e_i| = 1$. ϵ' is the set of edges that are being extended to e'.

Graph extensions are useful because we can proof the extensability theorem which indicates that multi party channels allow for more feasible wealth distributions.

Theorem 7.1. ***Extensability Theorem:*** *Let G_{k+1} be an extension of G_{k+1} then*

$$W_{G_k} \subsetneq W_{G_{k+1}}$$

Proof. [7] Let $w \in W_{G_k}$ be a a feasible wealth distribution. We need to show that $w \in W_{G_{k+1}}$ Because w is feasible there exists a feasible state $x \in \pi_k^{-1}(w)$ that satisfisfies the wealth distribution. For any edge $e' \in G_{k+1}$ Beacuse G_{k+1} is an exstension ϵ has at least one element. Now we set:

$$e'_x = \begin{cases} \sum_{e \in \epsilon} e_x & \text{if } \exists e \in \epsilon : x \in e \\ 0 & \text{(otherwise)} \end{cases} \tag{24}$$

e' fulfills conservation of liquidity. Thus we have constructed a feasible state x in the $(k+1)$-party channel network with $\pi(x) = w$. This concludes our proof and demonstrates that in the multi party channel extension there are at least as many wealth distributions feasible as before. □

While the communications protocol for multiparty channels are complicated and developer currently avoid implementing such channels several cryptographic secure constructs for multiparty channels exist.

7.1 Example how multiparty channel network achieves more feasible wealth distributions

Assume we have 4 participants bringing 3 coins each. for $k = 2$ we could create the following network:

We have 6 payment channels of capacity 2 each and each user is part of 3 of the 4 channels. In particular a user could allocate 1 coin to each channel. This topology corresponds to a fully connected graph. Each user could own at most 6 coins in this topology.

On the other hand a setup where these 4 participants select $k = 3$ to create 3-party channels we see that in this particular case at most 4 such channels are possible. Each user would be part in 3 of the 4 multiparty channels. Again in this particular topology each user could have broght 1 coin to each channel where the capacity of the channels is now 3. Each user could theoretically own up to 9 coins in this instantiation of the payment channel network.

Of course fully connected networks - in particluar with a small number of users - are rather artificial. but the fact that in this particular case we need only 4 instead of 6 on chain transactions and users can not only have more coins but overall more wealth distributions would be possible.

8 Emperical Results from Simulations and Experiments

There are currently a few outdated ipython notebooks on the github repository of this paper. In those we used a trick to more quickly decide if a wealth distribution is feasible. Unfortunately this resulted in getting false positives. We will update this section as soon as the notebooks are fixed

9 Conclusion

Generally speaking we seek a preferable[8] tradeoff between the expenses to maintain channels and the limits and burden they produce to the feasible wealth distributions for the users of payment channel networks.

[7] I assume one can crate a projection matrix that mapps the constraints from the $k + 1$ party channel network down to k-party channel networks

[8] for some suitable definition of preferable

We have seen that the two party channel design seems too constraint to allow for capital to be deployed efficiently as too many wealth distributions become infeasible. In particular it seems unlikely that the rate of infeasible payments can be reduced to a level that the payment channel network can relay on liquidity management through on chain transactions to fulfill all conceivable payment requests.

Because of the concentration of measure we can see that channel depletion is to be expected - which leads to a higher rate of payment attempts - even when delivering feasible payments. The communities wish to avoid credit in payment channel networks is another reason for an increased rate of infeasible payments.

While the narrative around the likelihood has shifted from being a peer to peer payment system for end users towards being a peer to peer settlement layer between service providers our results indicate that even a fairly small and professionally run payment channel network probably lacks a sufficient service level in terms of reliability and rate of payments being feasible and might depend on too many on chain transactions to to satisfies the user's requests.

Our results indicte that the idea that a two party Lightning Network can be used as a payment / settlement network between federations seems less promessing than understanding federations as multi party channels. Instead of connecting federations via two party channels it might seems more promesing to create an open standard for a cross federation / multiparty routing protocol. Of course this could also come in the form of an extension to the lighting network that allows the creation of multi party payment channels and routing though them.

In particular we have seen that the average access to capital of users grows linear in the number of peers per channel ($k \cdot \frac{C}{n}$). Also the Polytope W_k of feasible wealth distributions of a k-party channel network is a subset of the polytope W_{k+1} of wealth distributions for any extension of the k-party channel network to an $(k+1)$-party channel network. This is a strong indicator that multi party channels are useful.[9]

10 Acknowledgements

This work was funded by Open Sats Inc. and various Patreons. I would like to thank the reviewers of earlier versions. In particular Stefan Richter, Christian Decker, Christian Kuemmerle, Anastasios Sidiropoulos (who pointed out an error in my initial description of the wealth distributions), Elias Roher and bitromortac. Thanks to Rene Treffer for sharing an example with me that indicated how dropping the assumption that liquidity in channels is distributed independently can be used to show how bimodal liquidity distributions emerge. Furthermore since I am not a native english speaker I passed the manuscript through chatGPT for copy editing and revised its output. My origninal manuscript is as comments in the latex files.

References

[1] Sebastian Alscher. Price of anarchy in the lightning network. 2023.

[2] Alexander I Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19(4):769–779, 1994.

[3] bitromortac. Blazing the trails: Improving lnd pathfinding reliability, 2024.

[4] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623. IEEE, 2022.

[5] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Almost-linear-time algorithms for maximum flow and minimum-cost flow. *Communications of the ACM*, 66(12):85–92, 2023.

[6] Pranav Dandekar, Ashish Goel, Ramesh Govindan, and Ian Post. Liquidity in credit networks: A little trust goes a long way. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 147–156, 2011.

[7] Jesús A De Loera, Raymond Hemmecke, Jeremiah Tauzer, and Ruriko Yoshida. Effective lattice point counting in rational convex polytopes. *Journal of symbolic computation*, 38(4):1273–1302, 2004.

[8] Eugene Ehrhart. On homothetic rational polyedras à n dimensions. *CR Acad. Sci. Paris*, 254:616, 1962.

[9] David Griffin, Iain Bate, and Robert I Davis. Generating utilization vectors for the systematic evaluation of schedulability tests. In *2020 IEEE Real-Time Systems Symposium (RTSS)*, pages 76–88. IEEE, 2020.

[10] Gregorio Guidi. Paper - modeling a steady-state lightning network economy, 2019.

[11] Jona Harris and Aviv Zohar. Flood & loot: A systemic attack on the lightning network. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 202–213, 2020.

[12] Dmytro Piatkivskyi. Rebalancing argument, 2018.

[9] In particular for $k = n$ we have only one large multiparty in which all wealth distributions that are feasible on chain will also be feasible in the payment channel. This is similar to systems like Ark.

[13] Rene Pickhardt. The power of valves for better flow control, improved reliability and lower expected payment failure rates on the lightning network, 2022.

[14] Rene Pickhardt and Mariusz Nowostawski. Imbalance measure and proactive channel rebalancing algorithm for the lightning network. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5. IEEE, 2020.

[15] Rene Pickhardt and Stefan Richter. Optimally reliable & cheap payment flows on the lightning network. *arXiv preprint arXiv:2107.05322*, 2021.

[16] Rene Pickhardt, Sergei Tikhomirov, Alex Biryukov, and Mariusz Nowostawski. Security and privacy of lightning network payments with uncertain channel balances. *arXiv preprint arXiv:2103.08576*, 2021.

[17] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[18] Alvin Heng Jun Ren, Ling Feng, Siew Ann Cheong, and Rick Siow Mong Goh. Optimal fee structure for efficient lightning networks. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 980–985. IEEE, 2018.

[19] Emanuele Rossi, Vikash Singh, et al. Channel balance interpolation in the lightning network via machine learning. *arXiv preprint arXiv:2405.12087*, 2024.

[20] Clara Shikhelman and Sergei Tikhomirov. Unjamming lightning: A systematic approach. *Cryptology ePrint Archive*, 2022.

[21] J Michael Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities.* Cambridge University Press, 2004.

[22] Sergei Tikhomirov, Rene Pickhardt, Alex Biryukov, and Mariusz Nowostawski. Probing channel balances in the lightning network. *arXiv preprint arXiv:2004.00333*, 2020.

[23] Saar Tochner, Aviv Zohar, and Stefan Schmid. Route hijacking and dos in off-chain networks. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 228–240, 2020.

[24] Jan Van Den Brand, Li Chen, Richard Peng, Rasmus Kyng, Yang P Liu, Maximilian Probst Gutenberg, Sushant Sachdeva, and Aaron Sidford. A deterministic almost-linear time algorithm for minimum-cost flow. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 503–514. IEEE, 2023.

[25] Yunqi Zhang and Shaileshh Bojja Venkatakrishnan. Rethinking incentive in payment channel networks. In *2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 61–66. IEEE, 2023.

A Low Dimensional Example

We have already seen that for different topologies the feasible region and thus $r(G)$. For our example we set:

$$\begin{gathered} V = \{x, y, z\} \\ E = \{e = (x, y), f = (y, z), g = (x, z)\} \\ c_e = 3, c_f = 7, c_g = 11 \end{gathered}$$

We want to see if the wealth vector $w = (5, 6, 10)$ is feasible in this network. thus we construct our system of linear equations as.

We have m constraints related to conservation of liquidity.

$$\begin{aligned} e_x + e_y &= c_e \\ f_y + f_z &= c_f \\ g_x + g_z &= c_g \end{aligned}$$

and n constraints related to the wealth vector.

$$\begin{aligned} e_x + g_x &= w_x \\ e_y + f_y &= w_y \\ f_z + g_z &= w_z \end{aligned}$$

Together this results in the following system of linear equations.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ f_y \\ f_z \\ g_x \\ g_z \end{bmatrix} = \begin{bmatrix} c_e \\ c_f \\ c_g \\ w_x \\ w_y \\ w_z \end{bmatrix}$$

Again we ca replace the variables e_y, f_z and g_z with e_x, f_y and g_x respectively.

$$\begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} e_x \\ f_y \\ g_x \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} - \begin{bmatrix} 0 \\ c_e \\ c_f + c_g \end{bmatrix}$$

The solution space is 1 dimensional and can be parmeterized via:

$$\sigma = \{ \begin{pmatrix} 0 \\ w_y - c_e \\ w_x \end{pmatrix} + t \cdot \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \in \mathbb{Z}^3 | t \in \mathbb{Z}\}$$

Plugging in the wealth vector $w = (5, 6, 10)^t$ and the capacities we get:

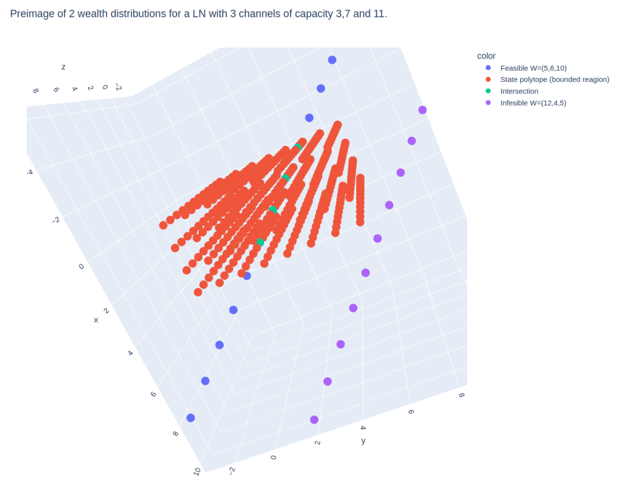


Figure 9: We see the preimage of $\pi|_{\mathbb{Z}^m}$ for 2 wealth distributions. One does have an empty intersection with the bounding box of the state polytope and is thus infeasible on the example network.

t	$0 \leq e_x \leq 3$	$0 \leq f_y \leq 7$	$0 \leq g_x \leq 11$	**feasable**
-1	-1	2	6	N
0	0	3	5	Y
1	1	4	4	Y
2	2	5	3	Y
3	3	6	2	Y
4	4	7	1	N

$$\sigma = \{ \begin{pmatrix} 0 \\ 3 \\ 5 \end{pmatrix} + t \cdot \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \in \mathbb{Z}^3 | t \in \mathbb{Z}\}$$

Finally we racall that our state space $S = \{0, \ldots, 3\} \times \{0, \ldots, 7\} \times \{0, \ldots, 11\}$. We define $F = \sigma \cap S$ to be the set of feasible solutions. We see that the parameter t cannot be negative. Thus we get the following table of feasible solutions.

If y wants to make a payment of 2 coins to z we take the wealth vector $w = (5, 6, 10)^t$ and compute:

$$w' = \begin{pmatrix} 5 \\ 6 \\ 10 \end{pmatrix} - 2 \cdot b_y + 2 \cdot b_z = \begin{pmatrix} 5 \\ 4 \\ 12 \end{pmatrix}$$

We get:

$$\sigma' = \{ \begin{pmatrix} 0 \\ 1 \\ 5 \end{pmatrix} + t \cdot \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \}$$

For $t = 0$ we have $(0, 1, 5)^t \in S$. Thus the wealth distribution $w' \in W$ and the payment was feasible.

Assuming we have the wealth distribution w' and user z wants to pay 10 coins to x then we had the wealth distribution $\omega = (15, 4, 2)$. As we have alredy seen in figure **??** that the solution space of ω does have an empty intersection with S. Thus the payment of 10 coins is infeasible.

B Open ends

B.1 Conjecture

Let $\sigma_w \subset \mathbb{Z}^n$ be the solution space for a wealth distribution w. Then:

$$\frac{|\left(\bigcup_{w \in W} \sigma_w\right) \cap S|}{|S|} \tag{25}$$

is the liklihood that a payment is infeasible.

B.2 TODO: Compare method to reality and optimally realiable payment flows

We use a data set of some network probes However we can estimate the likelihood for a payment to be feasible through statistical sampeling

B.3 Learn from successfull payments

if a payment was successfull we can find a subset of feasible states from where we could have started

B.4 TODO: Compare with min cut distribution

of course one could also compute the min cut distribution for various uniformly distributed wealth vectors (this is certainly more sound than initilizing channels randomly)

B.5 understand the subset of equivalence classes

For a general Network we have:

$$W_G \cong \bigcup_{i=1}^{m} \left(H_G \cap \{(\lambda_1, \ldots, \lambda_m) \in L_G | \lambda_i = 0\}\right) \tag{26}$$

B.6 Questions

1. Assuming all coins must be put into payment channels with a static topology. How should the capacities of the payment channels for a given wealth distribution being chosen? Is the goal that the space of possible wealth distributions is maximized favourable?

2. Assume only a limited amount of payment channels can be created (due to the fact that payment channels are achored as multisig transactions in the blockchain which has limited space). What is a desireable topology of channels at blockheight h for a fixed rate of channels per block?

3. Given that a subset of the existing coins may be put in payment channels. How many coins should be used to create channels?