2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA) | 979-8-3503-4588-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SERA57763.2023.10197670

# Balance-aware Cost-efficient Routing in the Payment Channel Network

Suhan Jiang and Jie Wu
Department of Computer and Information Sciences
Temple University
{tug67249, jiewu}@temple.edu

Fei Zuo
Department of Computer Science
University of Central Oklahoma
fzuo@uco.edu

Alessandro Mei
Department of Computer Science
Sapienza University of Rome
mei@di.uniroma1.it

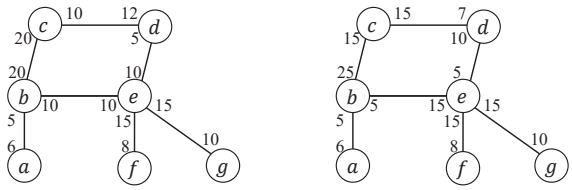
***Abstract*—Payment Channel Networks (PCNs) have been introduced as a viable solution to the scalability problem of the popular blockchain. In PCNs, a payment channel allows its end nodes to pay each other without publishing every transaction to the blockchain. A transaction can be routed in the network if there is a path of channels with sufficient funds, and the intermediate routing nodes can ask the transaction sender for a compensatory fee. However, a channel may eventually become depleted and cannot support further payments in a certain direction, as transaction flows from that direction is heavier than flows from the other direction. In this paper, we discuss a PCN node's possible roles and objectives, and analyze the strategies nodes should take under different roles by considering nodes' benefits and the network's performance. Then, we examine two basic network structures (ring and chord) and determine the constraints under which they constitute a Nash equilibrium. Based on the theoretical results, we propose a balance-aware fee-incentivized routing algorithm to guarantee cost-efficient routing, fair fee charging, and the network's long lasting good performance in general PCNs. Testbed-based evaluation is conducted to validate our theoretical results and to show the feasibility of our proposed approach.**

***Index Terms*—Balance-awareness, game theory, general payment channel network, ring and chord.**

## I. Introduction

For years, the blockchain [1] has faced a scalability issue, meaning there are challenges when the network tries to process more transactions simultaneously. Each transaction has to go through a long validation process before it becomes on-chain. Payment Channel Networks (PCNs) [2], a second-layer solution, have emerged to help improve processing times, build scalability, and lower the network's transaction fees. Nodes in a PCN can set up payment channels with pre-deposit funds, known as channel balances, and transfer values by re-adjusting balance allocation on the channels without publishing them on the blockchain. Non-connected nodes can transact with each other if they can find a path of channels with sufficient balances between them. The corresponding transactions will be routed along the path from the sender to the receiver. Usually, intermediate nodes along a payment path will charge fees as routing compensation and the routing fees are quite low compared with blockchain transaction fees. Each off-chain transaction is protected by a smart contract to guarantee the benefits of involved nodes. Obviously, PCNs can reduce payment overhead in terms of time and cost, and increase scalability of the whole system.

There is no central party in a PCN, meaning that transaction senders determine the payment path by themselves if existing multiple choices and all routing nodes individually set their fees without negotiation. In this situation, channels will be used in an unbalanced way, *i.e.*, transaction flows from one direction can be much heavier than flows from the other direction, since transaction senders always look for fast and cheap paths. One of a channel's end nodes that makes more transactions eventually runs out of balances and cannot send further payments unless committing a new transaction to the blockchain or performing a cyclic fund rebalancing [3], both of which are time-consuming and cost-inefficient. Meanwhile, those depleted channels reduce network-wide throughput.




(a) Original PCN. (b) Rebalanced PCN.

Fig. 1: A small PCN where the number associated with each channel close to a node is the deposit allocated by the node to the channel.

In this paper, we study how to avoid or at least how to alleviate the occurrence of depleted channels from the perspective of game theory, focusing on the interplay between three factors: transaction rates, routing fees, and channel balances. Routing fees charged by each node will affect the optimal path a transaction takes, thereby affecting the imbalances of the on-path channels. When the transaction rate across a channel is higher in one direction than the other, this channel will definitely be used in an unbalanced way. In this case, two possible solutions can be taken: one is to actively rebalance the channel so that more funds are allocated to that heavily-used direction, and the other is to passively adjust routing fees to make up for the channel imbalance.

Fig. 1(a) shows a small PCN in which $a$ sends \$1 to $f$ every hour, $g$ sends \$1 to $a$ every hour, and $f$ sends \$1 to $a$ every half hour, respectively. The shortest paths between $a$ and $f/g$ include the channel $b$-$e$. According to the transaction rates, the direction from $b$ to $e$ will be used more frequently than the opposite direction. For the purpose of a longer channel lifetime, it is better to have more balance on the end node $e$. Here, we suggest a balance distribution based on the ratio of the traffic rates in these two directions, *i.e.*, $1:3$ in the given example. Fig. 1(b) shows the result of a cyclic rebalancing among four nodes, *i.e.*, $b$, $e$, $d$, and $c$. That is, \$5 is moved along the channel $b \to e$, $e \to d$, $d \to c$, and $c \to e$. For

example, after sending \$5 to $e$, the channel value associated with $b$ is reduced by \$5 (from \$10 to \$5 in Fig. 1), and the channel value associated with $e$ is increased by \$5 (from \$10 to \$15 in Fig. 1). The total fund of each involved node on these corresponding channels keeps unchanged after rebalancing. For example, $b$'s total fund on channel $be$ and channel $bc$ is \$30 (\$10 + \$20 in Fig. 1(a) and \$5 + \$25 in Fig. 1(b)). Besides rebalancing, we can also adjust the channel imbalance through the fee policy. By setting the fees sent from $b$ to $e$ higher than that sent from $e$ to $b$, we accumulate more balance on $e$.

We aim to improve the PCN-wide throughput by lowering the occurrence of depleted channels. Since each PCN node is selfish, we first discuss their objectives and analyze their strategies when playing different roles in the network. We show that the individual objective is partially aligned with the network-wide objective. Thus, improving individual utility helps improve the throughput in the network. Then, we discuss some special topologies where we can find Nash equilibrium in which all nodes follow a relatively stable strategy. For the general topology, we propose a fee policy to guide each node to determine its channel fees in real time. Evaluation results show that our proposed policy improves nodes' utility as well as the network-wide throughput. The major contributions of this paper are as follows:

- We analyze the roles a PCN node can play and define its objectives under different roles.
- We determine three factors that affect a channel's lifetime and investigate how these factors affect nodes' strategies.
- In some specific topologies, we show that nodes will reach a Nash equilibrium where their path selections and routing fees are stable.
- For a general network topology, we propose a real-time fee policy to improve nodes' utilities, which also benefits the network-wide throughput.
- We perform the evaluation using real-world data on the testbed CLoTH, and the results show that our proposed fee policy benefits nodes' utilities, as well as the network-wide throughput.

## II. Background and Model

### A. Background

*Payment Channel in PCNs:* A channel allows two nodes to make multiple payments without the need to commit every payment to the blockchain. In Fig. 2, $u$ and $v$ jointly create a payment channel, in which they deposit funds. Suppose $u$ deposits \$5 and $w$ deposits \$2. After this transaction is committed to the blockchain, a channel with a *capacity* of \$7 is open between $u$ and $v$. Thereafter, $u$ and $v$ are able to perform payments back and forth freely by issuing transactions. At any moment, $u$ and $v$ can close the channel and refund the balance each one has in the channel by committing a closing transaction with their final balances to the blockchain. The balance of each node is updated after each successful transaction while keeping it private between two end nodes.

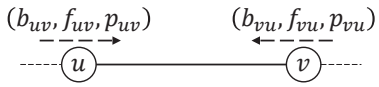



Fig. 2: Both $u$ and $v$ only act as intermediate routing nodes.

*Payment Path in PCNs:* In the lifecycle of a payment channel, a creating transaction and a closing transaction have to be committed to the blockchain, thereby causing transaction fees and waiting time. Payment channels are a suitable choice for any two nodes with long-term and high-frequency mutual transactions. Besides, two nodes that are not directly connected can make a transaction, as long as they can find a path consisting of multiple payment channels between them where the transaction amount is no larger than the minimum channel balance of the path. The transaction sender is required to reward intermediate routing nodes with a small routing fee.

### B. Model

*Channel State:* Let $C_{uv}$ be a payment channel between two nodes $u$ and $v$. The payment channel state is a pair $(b_{uv}, b_{vu})$, denoting the internal allocation of funds between $u$ and $v$. Here, we reuse the notation $C_{uv}$ as the capacity of the channel, where $C_{uv} = b_{uv} + b_{vu}$.

*Feasible Transactions:* Transactions will change the internal balance of a channel. Given the channel between $u$ and $v$, with the state of $(b_{uv}, b_{vu})$, a transaction of $a$ coins from $u$ to $v$ changes the state to $(b_{uv} - a, b_{vu} + a)$. The transaction is feasible if and only if $0 < a \leq b_{uv}$.

*Channel Liveness:* Although transfers in one direction are still possible when the balance is fully shifted to one side of the channel, we assume the channel is live unless it is ready for transfers in both directions. Any dead channel has to be rebalanced either through committing an onchain transaction or performing a cyclic fund rebalancing [3].

*Cost of Channel Rebalance:* We define $\eta_{uv}$ as the cost for the channel $C_{uv}$ to rebalance. It is always the smaller cost of either committing an onchain transaction or performing a cyclic fund rebalancing. Thus, the end nodes of each channel should dedicate their original balance distribution as well as their routing fees to prolong the channel lifetime and avoid rebalance costs.

## III. Payment Channel Lifetime

This section will focus on channel-level discussion and will investigate how to maximize a channel's lifetime. We start with a fee-free channel setting, meaning nodes will not charge routing fees. Given this simplified setting, we discuss how each end node of a channel initially distributes the balance in order to maximize the channel's lifetime, given the channel capacity is fixed. Based on what we obtain above, we then take routing fees into consideration.

Let's consider a channel between nodes $u$ and $v$. For simplicity, we assume that the channel capacity is $C$ and that $u$'s initial balance is $x$, indicating that $v$'s initial balance is $y = C - x$. We define $T_x$ as the expected lifetime of this channel given $u$'s initial balance is $x$. The channel lifetime

is characterized as the expected number of transactions this channel has executed before it needs a rebalancing. When the channel capacity is totally shifted to a specific end, $u$ and $v$ have to rebalance their channel. Obviously, $T_0 = 0$ since the channel capacity is totally shifted to $v$, and similarly, $T_C = 0$, since the channel capacity is totally shifted to $u$. Following [4], we assume that transaction arrival from one direction to the other follows a Poisson process. That is, on average, every second, there are $\lambda_{uv}$ transactions from $u$ to $v$ and $\lambda_{vu}$ transactions from $v$ to $u$. For each transaction, we simply assume its amount is $a = 1$.

### A. Fee-free Symmetric-transaction-rate Payment Channel

A channel with the symmetric transaction rate has $\lambda_{uv} = \lambda_{vu}$. Thus, for any transaction via this channel, the probability that this transaction is sent from $u$ to $v$ is 0.5 and vice versa. If $u$'s current balance is $x$, when a new transaction arrives, with a probability of 0.5, $u$'s balance turns into $x+1$, or with a probability of 0.5, $u$'s balance turns into $x-1$. Thus, we obtain the following relation:

$$T_x = 1 + 0.5T_{x+1} + 0.5T_{x-1}, \tag{1}$$

which yields a linear recurrence relation as below:

$$T_x = 2T_{x-1} - T_{x-2} - 2. \tag{2}$$

$$T_x = A_x - x^2 = \alpha_1 + \alpha_2 x - x^2. \tag{3}$$

To solve this linear non-homogeneous recurrence relation, let's assume $T_x = A_x - x^2$, then we have $A_x = 2A_{x-1} - A_{x-2}$. Thus, the characteristic equation based on $A_x$ is $r^2 - 2r + 1 = 0$, of which the solution is $r_1 = r_2 = 1$. Then, the format of $T_x$ can be expressed as $T_x = A_x - x^2 = \alpha_1 + \alpha_2 x - x^2$. Since $T_0 = 0$ and $T_C = 0$, we obtain $\alpha_1 = 0$ and $\alpha_2 = C$, which leads to $T_x = Cx - x^2$. When $x$ equals to $C/2$, $T_x$ reaches its maximal value. Based on the analysis above, we conclude that the optimal channel balance initialization for a fee-free channel with the symmetric transaction rate is $b_{uv} = b_{vu} = C_{uv}/2$.

### B. Fee-free Asymmetric-transaction-rate Payment Channel

Now, we consider a more complex setting where the channel has an asymmetric transaction rate. Suppose that, for any transaction via this channel, the probability that this transaction is sent from $u$ to $v$ is $p$ while the probability that this transaction is sent from $v$ to $u$ is $(1-p)$. For simplicity, we assume that $p \in (0, 0.5)$. Similarly, if $u$'s current balance is $x$, when a new transaction arrives, with a probability of $1-p$, $u$'s balance turns into $x+1$, or with a probability of $p$, $u$'s balance turns into $x-1$. Thus, we obtain the following relation:

$$T_x = 1 + (1-p)T_{x+1} + pT_{x-1}, \tag{4}$$

which yields a linear recurrence relation as below:

$$T_x = \frac{1}{1-p}T_{x-1} - \frac{p}{1-p}T_{x-2} - \frac{1}{1-p}, \tag{5}$$

the solution to which is the sum of the solution to the associated homogeneous recurrence system and a particular solution to the non-homogeneous case.

Given the associated homogeneous recurrence relation as

$$T_x^H = \frac{1}{1-p}T_{x-1}^H - \frac{p}{1-p}T_{x-2}^H, \tag{6}$$

its corresponding characteristic equation is $(1-p)r^2 - r + p = 0$, which yields $r_1 = 1, r_2 = p/(1-p)$. Then, the format of $T_x^H$ can be expressed as

$$T_x^H = \alpha_1 + \alpha_2(\frac{p}{1-p})^x. \tag{7}$$

Let $T_x^P$ be a particular solution to the non-homogeneous case. Since the non-homogeneous term is $1/p - 1$, a particular solution is of the form $T_x^P = \gamma x$. Since $T_x^P$ also follows the original recurrence relation, we have

$$\gamma x = \frac{1}{1-p}\gamma(x-1) - \frac{p}{1-p}\gamma(x-2) - \frac{1}{1-p}, \tag{8}$$

which yields $\gamma = 1/(2p-1)$. Thus, the format of $T_x$ can be expressed as

$$T_x = T_x^H + T_x^P = \alpha_1 + \alpha_2\left(\frac{p}{1-p}\right)^x + \frac{x}{2p-1}. \tag{9}$$

Since $T_0 = 0$ and $T_C = 0$, we obtain $\alpha_1 + \alpha_2 = 0$ and $\alpha_2 = \frac{C}{(1-2p)\left[\left(\frac{p}{1-p}\right)^C - 1\right]}$. Thus, we have

$$T_x = \left[\left(\frac{p}{1-p}\right)^x - 1\right]\alpha_2 + \frac{x}{2p-1}. \tag{10}$$

To find the maximal value of $T_x$, we check its concavity based on the sign of its second-order derivative. Listed below are $T_x$'s first-order and second-order derivatives, respectively.

$$\frac{\mathrm{d}T_x}{\mathrm{d}x} = \alpha_2\left(\frac{p}{1-p}\right)^x \ln\left(\frac{p}{1-p}\right) + \frac{1}{2p-1} \tag{11}$$

$$\frac{\mathrm{d}^2T_x}{\mathrm{d}x^2} = \alpha_2\left(\frac{p}{1-p}\right)^x \ln^2\left(\frac{p}{1-p}\right) \tag{12}$$

Since $\alpha_2 < 0$ for $\forall p \in (0, 0.5)$, then $\mathrm{d}^2T_x/\mathrm{d}x^2 < 0$ holds. Thus, $T_x$ is a concave function over $x$, and it reaches its maximum when $x$ satisfies $\mathrm{d}T_x/\mathrm{d}x = 0$, which yields $x = \left[\ln\left(z^C - 1\right) - \ln\left(C\ln z\right)\right] / \ln z$, given $z = p/(1-p)$.

### C. Payment Channel with Routing Fees

Now, let's take the routing fee into consideration. Let $f_{uv}$ ($f_{vu}$) be the routing fee charged by $v$ ($u$) when a transaction flows from $u$ ($v$) to $v$ ($u$). We assume that $u$ and all transactions executed by $u$ or $v$ are initiated by other nodes in the network, and they charge routing fees. Let $x$ be $u$'s initial balance, and after executing $m$ transactions, its balance turns into $x^{(m)}$. Among $m$ transactions, in expectation, $pm$ of them flow from $u$ to $v$, and the rest flow from $v$ to $u$. As we discussed before, channel rebalancing incurs an extra cost and this cost can be made up by charging routing fees. Let $F_{uv}$ be the expected routing fees earned by $u$ and $v$ via their channel $C_{uv}$, then $F_{uv}$ should be no less than *et al.*$_{uv}$.

*1) Symmetric Transaction Rate:* As we show in the fee-free setting, a symmetric-transaction-rate channel can reach its longest lifetime when both of its end nodes have the same balance initialization, given a fixed channel capacity. Since the channel has a symmetric transaction rate, after executing $m$ transactions, the expected value of $x^{(m)}$ should be $E\left[x^{(m)}\right] = x + (f_{vu} - f_{uv})m/2$. If the initial value $x$ leads

to the maximal channel lifetime $T_x$, then to reach the maximal channel lifetime at the point of $x^{(m)}$, $E\left[x^{(m)}\right] = x$ should hold. That is, $f_{vu} = f_{uv}$, both of which we simplify as $f$.

By taking the routing fees into consideration, we can update the expected channel lifetime as follows:

$$\begin{aligned} T_x &= 1 + 0.5T_{x+1+f} + 0.5T_{x-1-f} \\ &= 1 + 0.5T_{x+(1+f)} + 0.5T_{x-(1+f)}, \end{aligned} \tag{13}$$

A recurrence equation defines a sequence based on a rule that gives the next term as a function of the previous term(s). Here, we define $\{A_n\}$ where $A_n = T_x$, $A_{n\pm1} = T_{x\pm(1+f)}$. Then, we obtain the recurrence relation of $b_n$ as $A_n = 2A_{n-1} - A_{n-2} - 2$, which is equivalent to Eq. (2). As we have shown in the fee-free setting, a symmetric-transaction-rate channel can reach its longest lifetime when both of its end nodes have the same balance initialization, given a fixed channel capacity. Here, we can obtain the same result. Thus, for a symmetric-transaction-rate channel, to make it a long lifetime, the initial balance of each end should be identical, and the routing fee charged by each node should be identical as well.

*2) Asymmetric Transaction Rate:* Given the asymmetric transaction rate, after executing $m$ transactions, the expected value of $x^{(m)}$ should be

$$E\left[x^{(m)}\right] = x + (1-2p)m + (1-p)mf_{vu} - pmf_{uv}. \tag{14}$$

If the initial value $x$ leads to the maximal channel lifetime $T_x$, then to reach the maximal channel lifetime at the point of $x^{(m)}$, $E\left[x^{(m)}\right] = x$ should hold. Thus, $f_{uv}$ and $f_{vu}$ satisfy a relation of $p(1+f_{uv}) = (1-p)(1+f_{vu})$.

We are ready to update the expected channel lifetime in this fee-charged setting. Given the current balance $x$, when a new transaction arrives, with a probability of $p$, the balance turns into $x-(1+f_{uv})$, and with a probability of $1-p$, the balance turns into $x+(1+f_{vu})$. Thus, we obtain the following relation:

$$T_x = 1 + pT_{x-(1+f_{uv})} + (1-p)T_{x+(1+f_{vu})}. \tag{15}$$

Based on $p(1+f_{uv}) = (1-p)(1+f_{vu})$, we obtain

$$T_x = 1 + pT_{x-\theta} + (1-p)T_{x-\theta+\theta/(1-p)}, \tag{16}$$

where $\theta = 1 + f_{uv}$. By solving the recurrence relation given in Eq. (16), we can further obtain the optimal balance initialization. Note that there is no explicit expression for $T_x$.

### D. From Channel to Network: the Gap

We have analyzed in detail the liveness of payment channels and several methods, *i.e.*, initial balance distribution and routing fee, to maximize a channel's lifetime. However, it is not completely applicable to build a long-lived network of channels as routing fees are paid alongside the delivery of funds by a chain passing through a number of channels. That is where the smart contract known as HTLC (hash-timelock-contracts) [5], come in. Here, we discuss the way they work and use an example to show how a payment is accomplished in the Lightning network.

In Fig. 1(a), we consider a transaction of 2 coins from $b$ to $g$ using the route $b-c-d-e-g$. Assume that the intermediate nodes $c$, $d$, and $e$ charge the same amount of routing fee, $0.1$. After executing this transaction, corresponding balances of $c$, $d$, and $e$ are updated as $b_{cb} = 20+1+0.3$, $b_{cd} = 10-1-0.2$, $b_{dc} = 12+1+0.2$, $b_{de} = 5-1-0.1$, and $b_{ed} = 10+1+0.1$. When transferring funds to the next node, all fees charged by the downstream routing nodes are sent together, which is not considered in our previous discussion. Such a fee delivery method will affect all routing channels except the last one.

Luckily, as mentioned in the Lightning Network's original whitepaper [6], the fees should asymptotically approach negligibility for many types of transactions, which has also been confirmed in reality [7, 8]. Since channel routing fees are negligible, they play a trivial role in a channel's balance shifting ratio. Meanwhile, according to [8], the average shortest path length of LN is around 2.8, and most multi-hop transactions can be completed within 4 hops, indicating that there are at most 3 routing nodes involved. It is obvious that even if we take the accumulated routing fees into consideration, a channel's lifetime won't be affected too much. Thus, when two channel end nodes determine their initial balance distribution (or rebalance their channel), it is reasonable to consider a relaxed setting where all channels are fee-free. When their channel starts to function, they still charge routing fees and follow the relationship we proposed previously. We will talk about how to determine their routing fees in the next section.

We have discussed the optimal balance initialization and routing fee charge of a single channel. However, when considering the whole network, things become complex since a payment path includes several channels and routing fees.

## IV. Nodes in the Payment Channel Network

This section will focus on nodes in the network. We will analyze different roles a PCN node can play and its objectives under different roles. Generally, there are three roles each node can play in a PCN, *i.e.*, a routing service provider, a transaction sender, and a transaction receiver.

### A. Routing Nodes

Routing nodes are incentivized to participate in others' transactions by charging a small fee for every transaction that was routed through their channels. Currently, there is no specific policy to regulate how much fee routing nodes should charge. In the previous section, we talked about how the end nodes of a channel negotiate their routing fees with the aim of maximizing their channel lifetime. However, we only discussed the relation between $f_{uv}$ and $f_{vu}$ rather than deterministic values. Obviously, the corresponding values of $f_{uv}$ and $f_{vu}$ are determined by the objectives of $u$ and $v$. In this paper, we assume that each node, when playing as a router, aims to make profits so that its channel rebalance cost can be compensated. That is, its expected fee profits during its lifetime are equal to its rebalance cost. Take node $u$ with an initial balance of $x$ as an example. It is obvious that $u$'s objective is $f_{vu} \cdot T_x = \eta_{uv}/2$, assuming that the rebalance cost is equally divided bewteen $u$ and $v$.

For a payment channel, the transaction rates of its two directions can be different so that one end node's balance is exhausted as $0$. In this case, no transactions from that

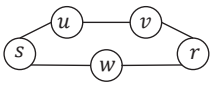



Fig. 3: Sender $s$ has two feasible path to reach Receiver $r$.

direction can be routed anymore, which is equivalent to an edge removal from a directed graph. What's worse, nodes of a channel applying a bad fee charging strategy can accelerate the channel exhaustion. To keep routing payments, some mechanisms are proposed to allow nodes to actively rebalance their corresponding channels by transferring funds from their high-balance channel to its low-balance channel through a payment channel circle. It is possible that no circle can be found to fulfill a rebalance, and such kind of active rebalancing costs money and time.

### B. Transaction Senders and Receivers

Assuming that there is a transaction of $a$ coins involving node $s$ as the sender and $r$ as the receiver. Given a channel with the sufficient balance, *i.e.*, $b_{sr} \geq a$, exists between them, $s$ can transfer the fund to $r$ directly. Otherwise, a payment path associated with routing fees is required. There may exist several paths connecting $s$ and $r$. When facing multiple feasible paths, it is the sender $s$'s responsibility to make a choice. In a traditional routing network, a sender usually takes two factors into account when choosing from paths. One is the path delay, measured by the number of hops of a path, and the other is the path fee, indicating the total fee charged along a path. These two factors are still applicable in a PCN.

We give a simple example in Fig. 3, where there exist two feasible paths between $s$ and $r$. For explanation, we assume that total routing fees charged by routing nodes $u$ and $v$ are fewer than that charged by $w$. Then, $s$ has two choices: the upside path of longer delay but less cost, and the downside path of shorter delay but more cost. Such cost-delay tradeoff will affect the decision made by $s$. We assume $s$ has a weight $w_s$ to show to what degree he cares about the path delay, and then his care on the fee cost is $1 - w_s$. Thus, we define the cost function for sender $s$ on a feasible path $P_{s,r,a}$ (simplified as $P$) as

$$C_{s,r,a,P} = (1 - w_s)\, f_{s,r,a,P} + w_s \left\| P \right\| \tag{17}$$

, where $f_{s,r,a,P} = \sum_{C_{uv} \in P} f_{uv}$ and $\|P\|$ is the path length.

Among all feasible paths, $s$ always selects the one that minimizes his cost by solving the following problem:

**Problem 1** ($\mathrm{OP_S}$)**.**

$$\textit{minimize} \quad C_{s,r,a,P_{s,r,a}} \tag{18}$$

Since a transaction receiver $r$ has nothing to do but receive the fund from the direction chosen by the sender, there is no specific objective for $r$.

### C. Discussion: Combination of All Roles

We need to combine all the roles for a node by integrating all the objectives. Generally, a node always wants to transfer money in a fast and cheap way, and meanwhile, it also wants to make as many routing fees as possible. Previously, we have discussed that the channel balance distribution can affect the channel lifetime and hence affect the accumulated routing fees in the long term. When a node acts as a sender, the path it picks provides a fast and cheap payment but may negatively affect the balance distribution of the first-hop channel, *i.e.*, the sender and the first routing node. There seemingly exists some objective conflict between the sender role and the routing node role. To refine this part, we can add a discounted factor related to the channel balance distribution. We use $\Delta_{s,r,a,P}$ to describe how the first-hop channel's balance distribution will be affected if choosing $P$. Given that $u$ is the first routing node in $P$, the expression of $\Delta_{s,r,a,P}$ is shown in the below:

$$\Delta_{s,r,a,P} = \left| \frac{b'_{su}}{b'_{us}} - \frac{\lambda_{us}}{\lambda_{su}} \right| - \left| \frac{b_{su}}{b_{us}} - \frac{\lambda_{su}}{\lambda_{us}} \right| \tag{19}$$

where $b'_{su} = b_{su} - a - f_{s,r,a,P}$ and $b'_{us} = b_{us} + a + f_{s,r,a,P}$. Then we update $C_{s,r,a,P}$ as follows:

$$C_{s,r,a,P} = (1 - w_s)\, f_{s,r,a,P} + w_s \left\| P \right\| - \Delta_{s,r,a,P} \tag{20}$$

We have discussed the individual objectives for nodes given some specific payment channels. However, all nodes coexist in a PCN and we want to know their behaviors mutually affect each other. In the following, we will first investigate some simple topologies and then use those achieved results to guide us in a large and complex topology.

## V. Network Analysis

We have analyzed a PCN at the level of individual nodes and individual channels. Obviously, lots of nodes coexist in a PCN, and mutually affect each other as well as corresponding channels. In this section, we take a PCN's topology into consideration and analyze the evolution of the whole network from the perspective of game theory. In reality, a PCN's topology and the transaction flows over it can be quite complex. Here we use some simple network topologies and traffic patterns as a starting point, in hopes of achieving results to guide us in a large and complex topology. Before deepening into specific network topology, let's introduce two traffic patterns that discussed the individual objectives for nodes given some specific payment channels. However, a PCN's topology and the transaction flows over it can be quite complex. All nodes coexist in a PCN, and we want to know their behaviors mutually affect each other. In the following, we will first investigate some simple topologies and then use those achieved results to guide us in a large and complex topology.

**Definition 1.** *A PCN is considered to have a uniform transaction flow pattern if any pair of its nodes have the same transaction rate.*

**Definition 2.** *A PCN is considered to have a proportional transaction flow pattern if any pair of its nodes have an identically proportional transaction rate. That is, for* $\forall\ i, j$, $\lambda_{ij} : \lambda_{ji} = r$.

### A. Ring

In a ring topology, nodes create a circular data path. Each node is connected to its two adjacent nodes, like points on a circle. In this topology, the fund travels from sender to receiver through intermediate nodes, clockwise or anti-clockwise until

it reaches the receiver. For $n$ nodes, the diameter is $n/2$, and the number of payment channels in the topology equals the number of nodes, $n$. In the simplest case, we assume that all nodes are homogeneous, indicating an identical budget, *i.e.*, the money they can allocate to their associated channels is identical, and the PCN has a uniform transaction flow pattern. Starting from scratch, we want to know how $n$ nodes distribute their budget to each channel, how they charge fees when acting as routing nodes, and how they decide when facing multiple paths in order to guarantee a long lasting good performance. We assume the decisions on the budget distribution and the fee that is going to charge are made before the network forms and cannot be changed. Thus, all nodes make their strategies in a long-term view rather than based on a certain network state. We are looking for an equilibrium where nodes have no incentive to change the strategies they make in the beginning.

**Theorem 1.** *Given a uniform transaction flow set among all nodes and the same starting budget $b$, in a pure equilibrium, all nodes evenly distribute their budgets on two channels, and charge equal routing fees of $\frac{4\eta}{b^2-4\eta}$ in both directions. When facing multiple choices, each transaction sender will choose the shortest path, and otherwise, it picks either path with an equal probability.*

*Proof.* For a pair of sender and receiver, without considering the balance insufficiency problem, there always exist two paths for the sender to choose from. Suppose that the sender is node $i$ and the receiver is node $j$. For the simplicity of writing, we assume that $i < j$ and $j \leq \lfloor n/2 \rfloor + i$. Thus, the short path $P_s$ is $i, i+1, \cdots, j$ while the long path $P_l$ is $i, i-1, \cdots, 1, n, n-1, \cdots, j$. Given $i$'s weight $w$ on the path delay, the costs of $P_s$ and $P_l$, *i.e.*, $C_s$ and $C_l$, are given below:

$$C_s = (1-w)\, f_s + w\,(j-i) \tag{21}$$

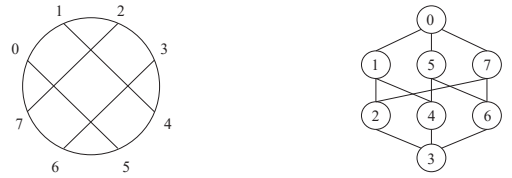
$$C_l = (1-w)\, f_l + w\,(n-j+i) \tag{22}$$

where $f_s$ and $f_l$ are the total routing fees of $P_s$ and $P_l$, respectively.

The problem is how sender $u$ chooses between $P_s$ and $P_l$. If $j = i+1$, meaning that $j$ is $i$'s next hop, then no matter what value $w$ is, $P_s$ is chosen. Meanwhile, there are three possible cases if $j > i+1$, as we show below:

1) If $C_s < C_l$, *i.e.*, $f_s - f_l < \frac{w}{1-w}\,[n-2\,(j-i)]$, then sender $i$ will choose $P_s$.
2) If $C_s = C_l$, *i.e.*, $f_s - f_l = \frac{w}{1-w}\,[n-2\,(j-i)]$, then sender $i$ can randomly pick either $P_s$ or $P_l$.
3) If $C_s > C_l$, *i.e.*, $f_s - f_l > \frac{w}{1-w}\,[n-2\,(j-i)]$, then sender $i$ will choose $P_l$.

Obviously, the second and third cases are not stable. Since $P_s$ takes advantages in terms of delay, it is always possible to decrease $f_s$ to a certain point to ensure $C_s < C_l$. Once $P_s$ is selected, no matter how small the routing fee is, each routing node will be better off than getting nothing. Thus, we can conclude that when reaching an equilibrium state, given two feasible paths, the total fee charges by the shorter path will be bounded by a certain value to ensure this path will be chosen by the sender. Thus, we can say, when facing two feasible paths, any sender will choose the shorter one. Based on this observation and our assumption that the transaction rate between any pair of nodes is identical, each node is isomorphic, as is each channel.



(a) Ring structure. (b) Flatten structure.

Fig. 4: A chordal ring topology of degree 3.

Therefore, a channel $C_{uv}$ is symmetric in terms of transaction rates, *i.e.*, $\lambda_{uv} = \lambda_{vu} = \frac{2}{n-1}$. According to our discussion in Section III and Section IV, we can conclude that each channel will have an identical balance on its two ends, and the routing fee of either channel direction is the same as well. Since each node has a budget of $b$, thus, either of its corresponding channels is allocated an initial balance of $b/2$. And by solving the following equation:

$$f \cdot \frac{b^2/2 - b^2/4}{2(1+f)} = \eta/2, \tag{23}$$

the corresponding routing fee is obtained as $f = \frac{4\eta}{b^2-4\eta}$. □

To make the setting more complex, we assume that the traffic rate between any two nodes is not identical. Instead, they follow some given patterns, and we still want to explore if there exists equilibrium in a long-term view.

**Corollary 1.** *Given that all nodes have the same starting budget while their in and out traffic rate is unbalanced as $r$, in an equilibrium, all nodes distribute their budgets and charge an amount of transaction fee with a specific rate (related to $r$) between two directions in an identical way.*

### *B. Chordal Ring*

*1) Chordal Ring of Degree k:* A chordal ring of degree k is a ring structured topology in which each node has an additional link, called a chord, to some other node across the network. The number of nodes in a Chordal Ring is assumed to be even. In Fig. 4(a), we show such a topology with $N = 8$. With those additional chords, the longest path of this topology is shortened from 4 hops to 3 hops, as is shown in Fig. 4(b). Nodes that are not directly connected can reach each other via different paths with the same length, meaning that routing delay is identical for a sender when facing multiple choices. Thus, all competitive routing nodes are involved in a price war, each trying to decrease a little on the current lowest fee, and hence leading to a zero fee. In this situation, the sender can randomly pick a path from all available choices, making all of them equally utilized in the long run. This is quite similar to a ring topology. Thus, we can get the following conclusion.

**Theorem 2.** *Given a uniform transaction pattern and the same starting budget, in a pure equilibrium, all nodes evenly distribute their budgets on $k$ channels, and charge zero fee in both directions. When facing multiple choices, each trans-*

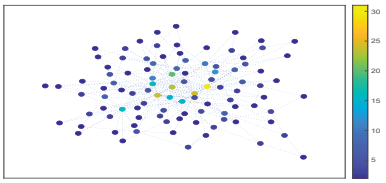


Fig. 5: Topology of the custom network.

*action sender will choose the shortest path, and otherwise, it picks any path with an equal probability.*

### C. Traffic-aware Balance Redistribution in General PCNs

In reality, it is impossible for nodes to know the traffic distribution until it stays in the PCN for a while. Thus, when creating channels, end nodes cannot immediately decide their initial channel balances and fee policies in the optimal manner. Here, we assume that each sender-receiver pair in the PCN has its transaction pattern, *i.e.*, the sender periodically issues identical-amount transactions to the receiver. Obviously, when all senders fix their own routing policies, the network traffic can form some relatively stable patterns, meaning that end nodes of a channel can learn the traffic rate of their channel gradually, based on which they can adjust their balance distribution and fee policies, following the rules we mentioned in Section III and Section IV. Eventually, the network can reach a relatively stable state, meaning that each channel fully utilizes its liquidation before it is depleted, and each balance redistribution requires as little extra cost, *i.e.*, the part that routing fees cannot cover, as possible. However, in reality, transactions cannot be always of an identical amount, meaning that learning the transaction rate is not enough for channels. In this case, each node should learn the transaction rate as well as the amount for a better channel balance distribution.

## VI. Evaluation

This Evaluation will focus on PCN-wide analysis to validate the proposed fee policy. We conduct experiments on CLoTH [9], a testbed for HTLC payment channel networks. We compare our balance distribution method with an existing mechanism, Revive, which aims to improve the performance of the lightning network (LN) [10] by rebalancing channel funds as equally as possible. The outcome proves the feasibility of our method in real time, and also shows better performance when compared with Revive.

*1) Setup:* In our validation process, we will use the clustering coefficient as a measure of the degree to which nodes in a graph tend to cluster together. Various evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit clusters characterized by a relatively high density of ties. In the simulation, we generate LN topologies using the GraphStream library [11] in Java [12] and implement routing algorithms using the Graph package in Matlab R2021a [13].

*2) Generation of Network and Transaction:* We generate a network, as is shown in Fig. 5, based on the BA model with 25 nodes and 51 edges. Each channel's capacity is set randomly from an interval ranging from $[50000, 75000)$ with a probability of $50\%$, $[75000, 100000)$ with a probability of $35\%$, and $[100000, 125000)$ with a probability of $15\%$. For each transaction, the sender-receiver pair is randomly selected. For each channel, the balances are randomly distributed between two nodes at the beginning of an experiment. For the transaction size, we use heterogeneous settings: $40\%$ of them are micro, with the transfer amount ranging from $(0, 1000]$; $30\%$ of them are small from $(1000, 4000]$; $20\%$ of them are medium from $(4000, 5000]$; and $10\%$ of them are large from $(5000, 8000]$. All selections are random. Each node's minimal HTLC is set as 1000 millisatoshi, and its timelock is set as 144ms. To get rid of the external effects, we set the routing fee charged by each routing node as an identical value of 10 millisatoshi, which is far less than the channel balances and the transaction amounts. We also generate a flow of 8000 heterogeneous transactions, evenly separated in 4 rounds. We specify a start time for each transaction and ensure the interval between any two subsequent transactions is within the range of $[50, 250)$ms. We compare the transaction flow success ratio under 3 different network mechanisms, *i.e.*, (1) our proposed traffic-aware rebalancing, (2) Revive rebalancing every 1000 transactions, and (3) Revive rebalancing every 2000 transactions.

| Round | Traffic-aware rebalancing | Revive rebalancing | |
|---|---|---|---|
| | | every 1000 txs | every 2000 txs |
| 1 | 0.3594 | 0.3742 | 0.3594 |
| 2 | 0.4333 | 0.3789 | 0.3742 |
| 3 | 0.4545 | 0.4008 | 0.4008 |
| 4 | 0.5083 | 0.4432 | 0.4225 |
| **Improvement** | **41.43%** | **18.44%** | **17.56%** |

TABLE I: Success ratio updates where each round contains 2000 txs.

*3) Performance and Discussion:* The corresponding results are shown in Table I. Obviously, our traffic-aware rebalancing mechanism still shows good performance when running in the real-time testbed. Although increasing the rebalancing frequency can lead to a higher success ratio for Revive scheme, its performance is still lower than that of our mechanism. Meanwhile, in the experiment, we ignore the rebalancing time which is quite time-consuming in reality. This means those involved channels cannot be used for routing during the rebalancing period, which may lead some transactions to fail due to the path being unavailable.

In fact, our fee mechanism may leak some important information about a channel's balance distribution, which violates the privacy requirement that the balance information of the channel should be only known to its owners. Remember that our goal is to balance a channel by adjusting transaction fees in different directions based on its current balance distribution. Rather than completely following the optimal fee policy, we suggest adding some random noises on the fee, *i.e.*, make the real charged amount different from the optimal amount by doing some random addition or reduction.

## VII. Related Work

### A. Path Determination in PCNs

A BGP-like protocol is proposed in the original routing algorithm in the Lightning Network white paper [6], which

requires nodes to store a global topology. This solution works for smaller networks without considering the increasing size of the payment network. Flare [14] supports scalability by reducing the routing tables' size maintained by nodes, while introducing beacon nodes to supplement a node's local view, which violates the spirit of decentralization. Landmark-based routing schemes are considered by both SilentWhispers [15] and SpeedyMurmurs [16]. All the above routing algorithms belong to static routing, meaning that the payment channel dynamics are not captured. Thus, Revive [3], Spider [17], and Flash [18] propose dynamic routing algorithms, both of which leads to a higher throughput and success volume of an LN.

### B. Routing Fee Policy in PCNs

In LN's white paper [6], an intermediate node can specify a base fee that is fixed for each payment and a fee rate which is a percentage fee charged on the value of the payment. The authors of [19] consider that the same transaction would cause a larger imbalance for channels with smaller capacities. They propose an optimal fee structure, whereby channels with large capacities charge smaller fees to minimize the balance difference between end nodes. Another paper [20] discusses global fee optimization in PCN design and examines the optimal graph structure and fee assignment to maximize profits from the perspective of routing nodes. Our paper combines three roles a PCN node can play and also takes channel imbalance into consideration when designing our fee policy. Besides, we also consider the transaction flow pattern as an important factor for fee policy.

### C. Game Theoretical Analysis in PCNs

There exist some works using game theory to analyze node strategies and network formations in PCN. [21–23] focus on determining which connection points are preferable for nodes joining the network with respect to their connectivity or revenue and show a theoretical result that profit-optimal join strategies tend to promote network centralization. Our paper focuses on how nodes reach Nash equilibrium on their path selections and fee determinations, given some specific PCN topologies and transaction flow patterns.

## VIII. Conclusion

In this paper, we consider the roles a PCN node can play and its objectives and strategies under different roles. Then we align the individual utility with the network-wide throughput, where we give some suggestions for the channel balance allocation and fee policy determination to benefit both individual nodes as well as the throughput of the whole payment network. For some specific PCN topologies like ring and chordal ring, we show the existence of Nash equilibrium where nodes have stable path selection and fee strategies. For general topologies, we propose a fee policy so that nodes adjust their strategies for channel balancing and utility maximization. Numerical evaluation is conducted to show the feasibility of our proposed policy.

## References


[1] M. Pilkington, "Blockchain technology: principles and applications," in *Research handbook on digital transformations*, 2016.

[2] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proceedings of the 2017 ACM SIGSAC*, 2017.

[3] R. Khalil and A. Gervais, "Revive: Rebalancing off-blockchain payment networks," in *Proc. of ACM CCS*, 2017.

[4] N. Papadis and L. Tassiulas, "State-dependent processing in payment channel networks for throughput optimization," *arXiv preprint arXiv:2103.17207*, 2021.

[5] bitcoinwiki, "Hashed timelock contracts." [Online]. Available: https://en.bitcoinwiki.org/wiki/Hashed_Timelock_Contracts

[6] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," *White Paper*, 2016.

[7] N. Khan *et al.*, "Lightning network: A comparative review of transaction fees and data analysis," in *International Congress on Blockchain and Applications*, 2019.

[8] F. Béres, I. A. Seres, and A. A. Benczúr, "A cryptoeconomic traffic analysis of bitcoin's lightning network," *arXiv preprint arXiv:1911.09432*, 2019.

[9] M. Conoscenti, A. Vetrò, J. C. De Martin, and F. Spini, "The cloth simulator for htlc payment networks with introductory lightning network performance results," *Information*, 2018.

[10] S. Martinazzi and A. Flori, "The evolving topology of the lightning network: Centralization, efficiency, robustness, synchronization, and anonymity," *Plos one*, 2020.

[11] Y. Pigné, A. Dutot, F. Guinand, and D. Olivier, "Graphstream: A tool for bridging the gap between complex systems and dynamic graphs," *arXiv preprint arXiv:0803.2093*, 2008.

[12] "Eclipse 2019-09." [Online]. Available: https://www.eclipse.org/downloads/packages/release/2019-09

[13] N. MathWorks Inc, "Ma. 2018. matlab r2018a."

[14] P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun, "Flare: An approach to routing in lightning network," *White Paper*, 2016.

[15] G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei, "Silentwhispers: Enforcing security and privacy in decentralized credit networks." in *Proc. of NDSS*, 2017.

[16] S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling payments fast and private: Efficient decentralized routing for path-based transactions," *arXiv preprint arXiv:1709.05748*, 2017.

[17] V. Sivaraman, S. B. Venkatakrishnan, M. Alizadeh, G. Fanti, and P. Viswanath, "Routing cryptocurrency with the spider network," *arXiv preprint arXiv:1809.05088*, 2018.

[18] P. Wang, H. Xu, X. Jin, and T. Wang, "Flash: efficient dynamic routing for offchain networks," *arXiv preprint arXiv:1902.05260*, 2019.

[19] A. H. J. Ren, L. Feng, S. A. Cheong, and R. S. M. Goh, "Optimal fee structure for efficient lightning networks," in *Proc. of ICPADS*, 2018.

[20] G. Avarikioti, G. Janssen, Y. Wang, and R. Wattenhofer, "Payment network design with fees," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, 2018.

[21] O. Ersoy, S. Roos, and Z. Erkin, "How to profit from payments channels," in *International Conference on Financial Cryptography and Data Security*, 2020.

[22] Y. Sali and A. Zohar, "Optimizing off-chain payment networks in cryptocurrencies," *arXiv preprint arXiv:2007.09410*, 2020.

[23] Z. Avarikioti, L. Heimbach, Y. Wang, and R. Wattenhofer, "Ride the lightning: The game theory of payment channels," in *Proc. of Financial Cryptography and Data Security*, 2020.