Understanding the Benefit of Being Patient in Payment Channel Networks

Qianlan Bai, Yuedong Xu, and Xin Wang

***Abstract*—Scaling blockchain efficiency is crucial to its widespread usage in which the payment channel is one of the most prominent approaches. With payment channels and the network they construct, two users can move some transactions off-chain to avoid expensive and time-consuming on-chain settlements. Existing works are devoted to designing high-throughput payment channel networks (PCNs) or efficient PCN routing policies to reduce the fee charged by intermediate nodes. In this paper, we investigate the PCN routing from a different perspective by answering whether the routing fee of transactions can be saved through being a bit more patient. The key idea is to reorder the processing sequence of atomic transactions, other than to handle each of them separately and immediately. We present two mechanisms, one is periodic transaction processing and the other is purely strategic waiting. In the former, the incoming transactions in a short time interval are processed collectively. We formulate an optimization model to minimize their total routing fee and derive the optimal permutation of processing transactions as well as the routing policy for each of them. A *Shapley value* based scheme is presented to redistribute the benefit of reordering among the transactions efficiently and fairly. In the latter, we model the waiting time of a strategic transaction on a single payment channel as the *first passage time* problem in queuing theory when the transaction value is higher than the channel balance upon its arrival. By capturing the balance dynamics, we are able to calculate the recursive expression of waiting time distribution that is useful to gauge a user's cost of patience. Experimental results manifest that our cost redistribution mechanism can effectively save routing fees for all the transactions, and the waiting time distribution coincides with the model well.**

***Index Terms*—Blockchain, payment channel, game theory, Shapley value.**

I. Introduction

SINCE the advent of Bitcoin in 2008 [1], we have witnessed the booming of various decentralized cryptocurrencies and the tremendous attentions they have gained. The historical transactions between cryptocurrency clients are recorded in a global and public data structure known as *blockchain*. It is envisioned that blockchain technology together with digital payment will embrace many new fields such as healthcare [2], manufacturing [3], edge computing [4], etc. Despite their ever-growing prosperity, cryptocurrencies suffer from unsatisfactory scalability. For instance, Bitcoin [5] processes 7 transactions per second and Ethereum processes 15 transactions per second [6]. In contrast, Visa network [7] can handle 1700 transactions per second, several orders of magnitude ahead of cryptocurrencies. The very low throughput hurdles its wide adoption. Scaling blockchain efficiency has become one of the most important issues that need to be solved in order to ravel the transitional means of payment.

At present, there are several effective ways of improving the scalability of blockchain including payment channel [8], [11], segregated witness [9] and sharding [10] in which the payment channel is the most prominent one. It is an off-chain solution that two cryptocurrency users are allowed to deposit tokens on the blockchain for the transactions between them. When establishing a payment channel, both parties agree on the amount of deposit, also called *capacity*, which measures the maximum value of transfer from one to the other at this moment. In the course of a transaction, the sum of capacity in both directions remains the same but the unilateral balance changes. The users (or nodes interchangeably) that do not have a direct channel can process transactions using multiple intermediate nodes and channels based on some protocols. Practical PCNs include Lightning network [8] in the Bitcoin system that utilizes the Hash-time Lock Contract (HTLC) scheme. A transaction can be routed from the sender to the receiver via intermediate nodes as long as the channel balances on this route are sufficient, or will be processed on the public chain otherwise. As a return, the intermediate nodes will charge each transaction a certain amount of fee for the interconnections provided, and this amount is usually much lower than the overall settlement on the public chain. Selecting the appropriate intermediate channels with the minimum cost has become an important issue in a blockchain PCN system.

The literature on route selection with the purpose of transaction cost reduction can be roughly categorized into two types. One is for an individual transaction. Zhang *et al.* [12] designed the Cheapay algorithm to find the cheapest available path with time and capacity constraints. A series of closely related works by Piatkivskyi *et al.* [16] and Rohrer *et al.* [17] presented new mechanisms which divided a transaction and transmitted on different paths to resist the capacity constraints.

Manuscript received May 28, 2021; revised January 16, 2022; accepted February 19, 2022. Date of publication February 25, 2022; date of current version May 23, 2022. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B010166003, in part by the Shanghai-Hong Kong Collaborative Project under Grant 18510760900, in part by the Natural Science Foundation of China under Grant 62072117, and in part by the Zhuhai Research Institute of Fudan University. Recommended for acceptance by Prof. Tie Qiu. *(Corresponding author: Yuedong Xu.)*

Qianlan Bai and Xin Wang are with the School of Computer Science and Technology, Fudan University, Shanghai 200237, China (e-mail: 17210 720025@fudan.edu.cn; xinw@fudan.edu.cn).

Yuedong Xu is with the School of Information Science and Engineering, Fudan University, Shanghai 200237, China (e-mail: ydxu@fudan.edu.cn).

Digital Object Identifier 10.1109/TNSE.2022.3154408

Ren *et al.* [14] proposed a new charging mechanism to maintain the balance of capacity, where the percentage of charges is defined as a reciprocal of the exponential function of the current capacity, so that more transactions can be processed successfully. The other is to design routing mechanisms for a sequence of transactions. Wang *et al.* in [13] proposed the Flash algorithm for PCN routing. They divided transactions into two categories according to the transaction amount, mainly focusing on the transaction fee cost of large transactions and the path detection cost of small-scale transactions. Varma and Maguluri utilized bilateral queues to process as many transactions as possible with the method of capacity rebalancing [18]. Sivaraman *et al.* proposed the Spider network with the idea of packetizing transactions and adopting a multipath transport protocol for high-throughput PCN routing [15]. However, the literature puts the emphasis on either the minimum cost or the high efficiency, while little attention is paid to their delicate trade-off.

In this paper, we study the payment channel routing from a novel perspective: *instead of pursuing extreme efficiency, can the users save their transaction routing fees by being a bit more patient?* Existing PCNs process the incoming transactions atomically and instantaneously [15]. Multiple transactions that arrive at different instants are routed on a first-come-first-serve (FCFS) basis. Two adverse effects might occur. One is that a transaction fails in the PCN and has to resort to the costly public chain if there is no path with sufficient balances; the other is that the overall routing fee of a set of transactions (from/to different nodes) is usually high. As the underlying reason, PCNs are *time evolving* with dynamic edge balances. For instance, Alice and Bob has built a payment channel. After a transaction from Alice to Bob, the balance of Alice to Bob decreases and that of Bob to Alice increases, and their total balance remains unchanged. Therefore, the FCFS processing is myopic in that a transaction possibly misses the cheap paths created by the subsequent transactions, or intercepts the cheap paths of subsequent transactions. Combined with the atomic transfer of transaction values in full, the FCFS means further increases the total routing fee. Inspired by this observation, we propose to reorder the processing of incoming transactions to reduce the routing fees, other than to process them immediately.

Our first mechanism stipulates that the incoming transactions are handled together at a fixed duration periodically or a fixed number of transactions. We formulate an optimization problem to minimize their total atomic routing fee whose output is the permutation of orders of transactions and the corresponding routing policy for each of them. Some transactions gain more, some gain less, while some others might lose. Under certain rules, we present a coalitional game framework to incentivize the form of a grand coalition using the famous Shapley value as the cost re-distribution mechanism. This mechanism is shown to be individual rational, efficient and fair such that all the transactions in the coalition benefit from the reduction of total routing fee, and the higher success rate of PCN processing is achieved.

Our second mechanism is rather intuitive. A "patient" transaction can wait for the increase of the edge balance if the initial balance upon its arrival is below its transaction value. This simple approach is also practical in the current payment channel network. However, computing how much time it should wait is a very challenging task even on a single PCN channel with bidirectional edges. We model the waiting time as the *first passage time* in queuing theory, that is, the first time that the initial balance increases above the transaction value, given the stochastic transaction arrival processes on both directions of a single payment channel. A novel stochastic model is built to capture the dynamics of edge balance. We compute the recursive expression of the distribution of waiting time that is useful for the transaction sender to gauge his cost of patience. Simulation results validate the accuracy of our model.

Our major contributions are briefly summarized as below.

- We propose a novel idea of reordering the transactions actively or opportunistically, other than the chronological processing in order to reduce the PCN routing cost.
- We present a periodic transaction processing scheme that yields an optimal transaction order, and formulate a coalitional game with Shapley value as the benefit redistribution mechanism.
- We present a novel transient queuing model to capture the balance dynamics of a payment channel, and calculate the recursive expression on the waiting time distribution of a strategic transaction.

The remainder of this paper is organized as follows. Section II describes the background and related work. Section III models the payment routing problem and introduces the concept of transaction reordering. The coalition mechanism design is presented in Section IV and the stochastic waiting time is mathematically analyzed in Section V. Section VI validates the Shapley-value based mechanism and the stochastic model. Section VII concludes our work.

II. Background and Related Work

In this section, we present the basics of the payment channel routing and describe the literature that is pertinent to our study.

A. A Premier on Payment Channel Network

Payment channel is a method to realize the off-chain overcoming the scalability issue [8]. It allows two users not to commit all processed transactions to the public blockchain. The users publish a transaction to the public blockchain when establishing a new payment channel where they deposit coins into a multi-signature address [19] controlled by both users. Once the channel is established, they can make transactions over the channel and maintain channel balances by themselves, without committing these transactions to the blockchain. When the two users want to close a channel, they need to publish a transaction to the public blockchain [32]. However, connecting each pair of users off-chain incurs a relatively high cost, especially when the direct transactions between them are infrequent. The users must pay the transaction fee on public chain when creating a new payment channel. To trade with others without direct channels in PCNs, the payments can be processed through multiple hops of channels

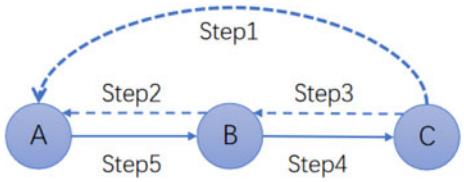


Fig. 1. The example of HTLC mechanism.

in the PCNs. This may lead to issues that the intermediate node denies performing payment transfer after receiving coins from the preceding one.

To address these issues, the Hashed Time-Lock Contract (HTLC) mechanism has been introduced [8], [20]. Consider a simple transaction with three nodes A (*Alice*), B (*Bob*) and C (*Carol*) in Fig. 1 where A transfers a fund (0.1 BTC) to C through the intermediary B. As the first step, C computes the hash value on an arbitrarily chosen random number R and sends it to A. In the second step, A and B sign a smart contract which stipulates that if B can provide the correct R to A before a certain time (i.e., the lock time chosen by A and B), A will settle the contract by paying B 0.11 BTC. If the lock time have elapsed, then the above clause is null and void. The third step is that B and C sign the same contract as in step 2, except for the amount and lock time, where the lock time is negotiated by B and C. The transfer value is 0.1 BTC, and the remaining 0.01 BTC is the transaction fee to be paid to node B. In the fourth step, C reveals the random number R to B before the specified time, then B transmits the corresponding amount to C after verifying R. Finally, the same operation occurs between A and B and the transaction has been processed successfully. It is noted that if C (resp. B) do not offer the correct R to B (resp. A), the transaction will fail.

B. Overview of Routing Algorithm

There has been a growing interest toward PCN routing. In what follows, we describe the recent progresses on the routing protocol design and the economic analysis of PCN routing.

Protocol Design. For each payment channel, the associated two nodes are aware of the channel information. In this regard, SilentWhispers [21] selected the nodes with high connectivity in the network as landmarks to generate the complete transaction path by combining the paths of both parties to landmark. Roos *et al.* [22] proposed SpeedyMurmurs to find the path between nodes in a distributed environment using a greedy routing algorithm based on graph embedding. Hoenisch and Weber brought in On-Demand Distance Vector (AODV) routing protocol from Mobile Ad Hoc Networks [35]. Mercan proposed a gateway selection algorithm to improve the transaction success rate which calculates the ratios of total inbound capacity to outbound capacity of each connected gateway and chooses the minimum among them [49]. Flare [23] presented a new hybrid routing algorithm so as to find the available path as quick as possible in PCNs. Each node stores the information of its neighborhood and selects a set of beacon nodes randomly. When a user initiates a transaction, he first checks the destination among his neighboring nodes, and resorts to the beacon nodes if the path is not found. To cope with the capacity constraint, a series of recent studies [15], [16], [26], [47] developed novel routing policies that turn an atomic payment into splittable smaller payments, and transfer them on multiple routes. The transfer, however, delays if any of the partial payments fail. To address this, Bagaria introduced Boomerang based on secret sharing and homomorphic one-way function, which allows reverting the transfer if the channel overdraws [51]. Rahimpour introduced Spear, a simple method with lower latency and less computation than Boomerang [52]. Splitting routing can significantly increase the rate of payment successes, but there is no trivial way to integrate fees in such protocol. LRFP extended the existing routing protocol to solve the integration of fees in the local routing protocol [50].

Economic Analysis. Cost efficient PCN routing has drawn considerable attentions since the senders seek to minimize their transaction fees charged by the payment channels. Engelmann *et al.* [33] proposed the first economic model of payment channel routing and computed the minimum cost route accordingly. Along this line, Zhang *et al.* [12] investigated the cheapest path in PCNs subject to time and capacity constraints. Avarikioti *et al.* [24] and Ersoy *et al.* [25] also described the models of finding the cheapest path in PCNs. In addition, some works aim to maintain the security and long-term availability of channels. Ren *et al.* [14] presented a new charging mechanism to achieve reasonably short path lengths and overall balanced channel capacities. Revive [28] allowed a user to utilize any other of his channels for re-balancing a particular channel. Engelshoven *et al.* proposed a new charging function to avoid channel depletion in that the fee for payment increases linearly with the degree to which deteriorates the balance of a channel [48]. Awathare *et al.* proposed REBAL, a distributed re-balancing mechanism which re-routes transactions from intermediate nodes around an unidirectional channel rather than propagating the failure back to the source [46]. Malavolta *et al.* [29] proposed Fulgor and Rayo schemes, where the former ensures the intermediate nodes to transfer funds honestly via a non-interactive zero-knowledge building block and the latter guarantees at least one of the concurrent transactions can be processed successfully. Yu *et al.* [27] designed a distributed algorithm to improve the payment success rate and reduce the system overhead. A deadlock in a payment network is a situation in which several simultaneous payments share edges in their paths in such a manner that none of the payments can go through. Werman *et al.* [30] solved the deadlock problem to improve the success rate of transaction by rearranging the transaction processing order.

III. Problem Description

In this section, we describe the mathematical model of payment channel networks (PCNs) and the motivation of being patient in PCNs by blockchain transactions.

A. Temporal Network Model

A payment channel network can be represented as a time-dependent directed graph $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}, \mathcal{C}, \mathcal{W}, \mathcal{B})$, where $\mathcal{N}$ is

the set of nodes and $\mathcal{E}$ is the set of edges. Each node $n_i \in \mathcal{N}$ represents a cryptocurrency account that has built one or more payment channel agreements with other nodes; each directed edge $e_{ij} \in \mathcal{E}$ represents a payment channel from node n_i to n_i. The edge e_{ij} is associated with a 3-tuple $(c_{ij}(t), w_{ij}, b_{ij})$, where the first is the maximum amount of cryptocurrency coins n_i can pay to n_j at time t and the second is the price per-unit of coin transfer charged by n_j on this edge if it is a relay node. Let $b_{ij} \in \mathbf{B}$ be the (flat-rate) base fee of using the channel e_{ij} as the relay, regardless the amount of transaction size. We denote by X_k the k^{th} transaction in the payment channel network that is expressed as a 4-tuple $X_k = (n_s(k), n_r(k), v_k, t_k)$. The sender and the receiver of X_k are denoted by $n_s(k)$ and $n_r(k)$, respectively. Here, v_k is the payment value measured by the number of coins and t_k is the time instant that the transaction takes place. One primary difference between the PCN and the traditional communication network is that a payment channel between n_i and n_j has a constant sum of balances on bidirectional edges, while the balance on each directed edge can change. When a transaction from n_i to n_j has been processed successfully, the balance of edge e_{ij} decreases while that of edge e_{ji} increases. For any pair of nodes n_s and n_r in the transaction X_k, a *feasible path* is an end-to-end path on $\mathcal{G}_t$ whose edge balances are sufficient to transfer the transaction value.

The business model of using payment channels is the following. If the nodes n_s and n_r form a payment channel directly, there is no need for n_s to pay for using this channel when n_r is the receiver. If the transaction X_k is forwarded along a path $\mathcal{P}_k(t)$ that starts at n_s and ends at n_r, the intermediate nodes and channels should be paid for the coin transfer. For instance, consider the path $\mathcal{P}_k(t) = \{n_1, n_2, n_3, n_4\}$. The node n_1 will pay a certain amount of fees to n_2 for using the channel e_{23}, and n_2 needs to pay to n_3 for using the channel e_{34}, but n_3 does not pay to the receiver n_4. Hence, the total routing fee that X_k needs to pay can be expressed as $f = (w_{34} \cdot v_k + b_{34}) + [w_{23} \cdot (v_k + v_k w_{34} + b_{34}) + b_{23}]$ [12], [35] [36]. Formally, the total routing fee on the $\mathcal{P}_k$ at graph $\mathcal{G}_t$ is given by:

$$f(X_k, \mathcal{G}_t) = \begin{cases} \xi & \mathcal{P}\mathrm{k(t)} = \varnothing; \\ \sum\limits_{\substack{e_{ij} \in \mathcal{P}_k \\ \forall n_i \neq n_s(k)}} (w_{ij} \cdot V_k(ij) + b_{ij}) & \text{otherwise.} \end{cases} \quad (1)$$

Here, $V_k(ij)$ represents the actual amount that X_k needs to transfer through e_{ij}, including v_k and the routing fees to be paid for subsequent edges. When the value of a transaction exceeds the edge balance, it can be sent to the public chain [15] that will incur a fixed amount of payment plus a much longer confirmation time. Without loss of generality, we deem the processing cost of a transaction as a constant ξ in the public chain that is higher than the routing fee in the PCN [41].

In the hop-by-hop value transfer, the transaction fee need to be transmitted with the original transaction, which means $f(X_k, \mathcal{G}_t) + v_k$ is the actual value that $n_s(k)$ needs to pay at time t. For the intermediate edges on $\mathcal{P}_k(t)$, the balance must be no less than the value plus the total charge on the remaining channels on a route. The feasible flow paths should satisfy:

$$\begin{cases} V_k(ij) \leq c_{ij}(t), \forall e_{ij} \in \mathcal{P}_k(t); \\ f(X_k, \mathcal{G}_t) \leq \xi. \end{cases} \quad (2)$$

We make a few commonly adopted assumptions to simplify our modeling efforts while complying with the real-world blockchain PCNs.

Assumption 1: (Information Availability) Each node can obtain the channel balance and the edge prices of other payment channels at the moment through probing the payment channel network, similar to [12], [13] and [23].

Assumption 2: (Immediate Processing) The transaction over a payment channel network takes effect immediately if it is scheduled for processing, similar to [13], [34].

Assumption 3: (Indivisible Transaction) A transaction cannot be split into multiple transactions of smaller values.

In PCNs, some nodes are reluctant to reveal their balances and prices, and if so, they cannot become the public intermediate nodes to forward the transactions. The splitting of transactions may involve the multi-path routing problem that is usually more complicated. We only consider indivisible transactions in this work, yet our problem and methodology are applicable to divisible transactions.

B. Transaction Reordering

The original purpose of constructing payment channel networks is to speed up the processing of Bitcoin transactions. The state-of-the-art efforts are devoted to designing high-throughput payment channel systems without sacrificing cryptocurrency security [8]. As a basic incentive to maintain the PCN, the payment channels may charge the sender of the transaction a small amount of *routing fee*. As a consequence, the user is inclined to select a feasible path that yields the minimum total routing fee [12].

Given a sequence of transactions sorted by their arrival times, the graph $\mathcal{G}_t$ changes after each successful transfer. In this sequential processing order, choosing the minimum cost path for each transaction might be myopic because the graph is dynamic. A transaction may have a chance to find another path with even lower cost if it can "wait" for some time. We hereby illustrate that being patient is "egoistic" or "altruistic".

Fig. 2 and Fig. 3 shows the different routing fees that need to be paid for the same three transactions in different orders. The three transactions are $X_0 = (D, C, 3, 0), X_1 = (E, C, 1, 1), X_2 = (E, D, 2, 2)$. Without loss of generality, we set the base transaction fee as 1 unit and the charge rate of each edge as 50% transaction amount. This setting has no influence for the basic property of problem which we want to illustrate. The red path (in dashed lines) represents the cheapest path through, the green path (in dotted lines) represents the edges that change in the opposite direction. The number on the line indicates the balance of the channel at the current time. Fig. 2

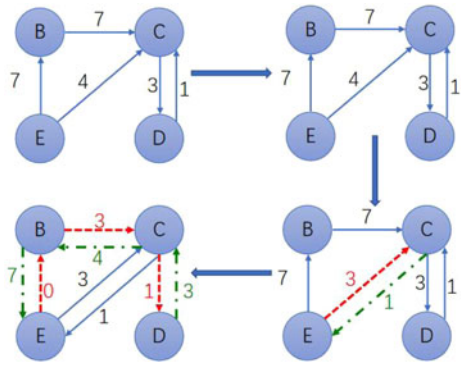


Fig. 2. The process of $X_0 = (D, C, 3, 0), X_1 = (E, C, 1, 1), X_2 = (E, D, 2, 2), b_{ij} = 1, w_{ij} = 0.5 \quad \forall e_{ij} \in \mathcal{E}$.

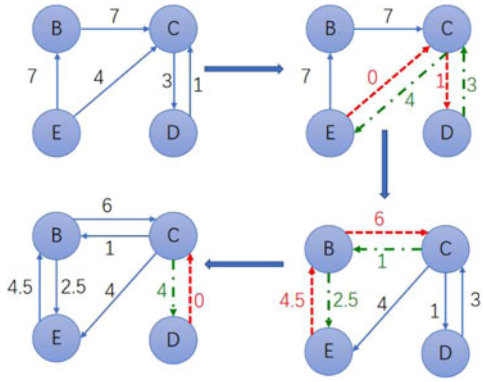


Fig. 3. The process of $X_2 = (E, D, 2, 2), X_1 = (E, C, 1, 1), X_0 = (D, C, 3, 0), b_{ij} = 1, w_{ij} = 0.5 \quad \forall e_{ij} \in \mathcal{E}$.

shows the change in PCN topology when each transaction chooses to be processed as soon as it comes into being. For each transaction, the routing fee is $f(X_0, G_0) = \xi, f(X_1, G_1) = 0, f(X_2, G_2) = 5$. If X_1 chooses to be processed at time $\eta_1 > 2$ and X_0 chooses to be processed at time $\eta_0 > \eta_1$, the sequential processing order is (X_2, X_1, X_0), the change in PCN topology is shown in Fig. 3. In this order,the routing fee are $f(X_0, G_{\eta_0}) = 0, f(X_1, G_{\eta_1}) = 1.5$ and $f(X_2, G_2) = 2$.

Egoistic Waiting. Egoistic waiting means that the waiting can reduce the user's own routing cost. If X_0 chooses not to wait, it must be processed in the public chain and the cost is ξ. By waiting for a short while, X_0 can be processed after X_1 and X_2. There is an available path due to the successful processing of X_1 and X_2, and henceforth the routing cost of X_0 decreases.

Altruistic Waiting. Altruistic waiting is defined as that the waiting of one user will cause its own routing cost to increase or remain unchanged, but other users' routing cost may decrease, leading to a decrease in the total cost. If X_1 is processed first, the balance of $E \to C$ is sufficient and the routing fee is 0. After waiting, X_1 can only choose the path $E \to B \to C$ due to the preemption of X_2. Then, the routing fee of X_1 increases to 1.5, and that of X_2 decreases to 2 while the total cost of processing X_1 and X_2 is less.

By reordering the transactions (other than first-come-first-serve (FCFS)), either some of the transactions can lower down their routing fees charged by the PCN, or these transactions as a whole can save a certain amount of routing fees. In order to take this advantage, there needs an incentive mechanism to encourage the users to be more patient. We hereby consider two scenarios, in which the former requires the intervention of the node(s) performing off-chain transactions, and the latter is completely compatible to the existing payment channel networks.

1) Periodic Transaction Processing (PTP): The PCN configures a buffer to store the information of incoming transactions. It processes all the transactions in the buffer cyclically when the number of transactions exceeds a certain threshold or a fixed duration has been reached. An optimal reordering policy will be implemented to minimize the total transaction cost, and the total routing fee will be redistributed among these transactions. As the underlying principle, no transaction will receive a higher routing fee in the reordered processing than in the FCFS processing.

2) Strategic Patience (SP): When a myopic user finds that the cheapest payment channel does not have the sufficient balance to transfer his transaction value, he may resort to an expensive channel or the public chain. While a strategic user can predict how much time he can wait until the balance of the cheaper channel is greater than his transaction value, given the arrival patterns of transactions at both directions of the channel.

IV. Coalition Mechanism Design

In this section, we formulate a cooperative game and formally define the benefit distribution mechanism that is compatible with the optimal transaction reordering principle.

A. Cooperative Game in PCN

Cooperative game (or coalition game narrowly) is a mathematical theory in revealing the behaviors of rational players in a cooperative setting. The players make agreements among themselves to form coalitions that affect their strategies and utilities, as opposed to the non-cooperative games. In what follows, we formulate the benefit redistribution mechanism as a cooperative game, namely CG.

- *Player:* A user who initiates a transaction is regarded as a player. If not mentioned explicitly, the transaction is equivalent to the user so that the set of players are expressed as $\mathbf{X} := \{X_0, X_1, \ldots, X_{N-1}\}$ with the arrival times $t_n \leq T, \forall\, 0 \leq n \leq N-1$.
- *Coalitional Function:* A function $\psi(S)$ where S is an arbitrary non-empty subset of $\mathbf{X}$.

We denote $\psi(S)$ as the *coalitional function* which measures the benefit produced by coalitions. The redistribution mechanism in the coalition is through Shapley value function.

Definition 1 (Grand coalition): If every player chooses to join in a coalition, the coalition is called *grand coalition*.

B. Worth of S

We name a few interesting features about the calculation of the coalitional function $\psi(S)$ in our game. First of all, $\psi(S)$ is an outcome of the minimum cost routing problem that demands an appropriate algorithm to generate this value.

Second, the sequence number of processing this coalition as a whole needs to be decided due to that different processing sequences lead to different transaction fees. Third, the possible "free-riding" users route through new available paths created by coalition for free. We need to devise a set of rules to handle the above problems so as to make the coalition feasible and implementable.

Rule 1: If a subset of transactions forms a coalition, their arrival times are modified to be aligned with the earliest transaction in this coalition. The arrival time of a transaction outside of the coalition remains unchanged. The processing priority of these transactions can be obtained by sorting the arrival times of the transactions.

Rule 2: The execution rule of transactions inside the coalition is decided for the purpose of reducing their total transaction fees charged by the payment channels, and the path of each transaction is provided accordingly.

Execution Order of the Coalition. The execution order of a coalition $S \in \mathbf{X}$ is determined by that of the earliest transaction in S. Denote by η_S the order of coalition S, and by η_k the order of X_k in the original set of transactions. There are

$$\tilde{t_k} = t_k, \quad \text{if} \quad X_k \notin S; \tag{3}$$

$$\tilde{t_S} = \arg\min_j \{t_j | X_j \in S\}, \quad \text{if} \quad X_k \in S. \tag{4}$$

We reorder these transactions on the basis of $\tilde{t_k}$ and $\tilde{t_S}$. Denote by η_k the processing sequence of transaction $X_k (X_k \notin S)$ after reordering. Denote by η_S the processing sequence of coalition S after reordering.

For instance, there exist four transactions X_0, X_1, X_2 and X_3 with the arrival time $t_0 < t_1 < t_2 < t_3$. When X_1 and X_3 form a coalition S, the processing order of S is determined by t_1 so that the processing sequence of X_0, X_S and X_2 satisfies $\eta_0 < \eta_S < \eta_2$. Hence, the coalition does not affect the processing of the transactions who arrive earlier than all the coalitional players.

Execution Order Within the Coalition. In S, the sequence of players should be rearranged so as to minimize the total routing fee. Consider the coalition $S = \{X_0, X_1, \ldots, X_{K-1}\}$, the PCN graph is redefined as $\mathcal{G}_k$ before processing the transaction X_k. The optimal routing for a particular X_k in the current topology of PCN is expressed below.

$$\min \quad \sum_{(i,j):e_{ij} \in \mathcal{E}} f_{ij}(X_k)\gamma_{ij}(X_k) \tag{5}$$

$$s.t. \quad \gamma_{ij}(X_k) := \begin{cases} 1, & \text{if} X_k \text{ is routed along } e_{ij}; \\ 0, & \text{else.} \end{cases} \tag{6}$$

$$\sum_{j \in \mathcal{N}} \gamma_{ij}(X_k) - \sum_{j \in \mathcal{N}} \gamma_{ji}(X_k) =: \begin{cases} 1, & n_i = n_s(k); \\ -1, & n_i = n_r(k); \\ 0, & \text{else.} \end{cases} \tag{7}$$

$$\gamma_{ij}(X_k) V_k(ij) \le c_{ij}(\eta_k), \quad \forall e_{ij} \in \mathcal{E}; \tag{8}$$

$$V_k(ij) = \sum_{h \in \mathcal{N}} \gamma_{jh}(X_k)(V_k(jh) + f_{jh}(X_k)); \tag{9}$$

$$f_{jh}(X_k) = \gamma_{jh}(X_k)(w_{jh} V_k(jh) + b_{jh}); \tag{10}$$

Algorithm 1: *PTP-SOLVER* function.

Input: Coalition, $\mathbf{X} = \{X_0, X_1, \ldots, X_n\}$; Initial topology of PCN, $G_0 = (\mathcal{N}, \mathcal{E}, \mathcal{C}, \mathcal{W}, \mathcal{B})$;
Output: fee: Set of transaction fee;
$Path$: Set of path;
G: Topology of PCN;

```
X_list = {S_1, S_2, ...};        ▷ S_i is the i^th permutation of X;
for all S_i do
    G ← G_0;
    fee ← [ξ, ξ, ..., ξ];        ▷ Maximum fee for each transaction is ξ;
    f, P, G' = SET − SOLVER(S_i, G_0);
    if sum(fee) > sum(f) then
        fee ← f;
        path ← P;
        G ← G';
    end if
    if sum (fee) = sum(f) then
    (fee, path, G) ← (f, P, G') with more successful transaction number;
    if successful number is same then
            (fee, path, G) ← (f, P, G') with more successful transaction value;
        end if
    end if
end for
return fee, path, G;
```

$$\sum_{h \in \mathcal{N}} \gamma_{hr}(X_k) V_k(hr) = v_k; \tag{11}$$

$$f_{ij}(X_k) = 0, \quad \text{if } n_i = n_s(k); \tag{12}$$

$$\sum_{(i,j):e_{ij} \in \mathcal{E}} f_{ij}(X_k)\gamma_{ij}(X_k) \le \xi. \tag{13}$$

The objective is the sum of the routing fees on all the edges. The binary variable $\gamma_{ij}(X_k) \in [0, 1]$ in (6) indicates the usage of edge e_{ij} for transaction X_k, and (7) represents the flow balance conditions. For all vertices except n_s and n_r, there is only one incoming edge and one outgoing edge (i.e., that it should be a part of the path from n_s to n_r).

The PCN routing model has two major differences from traditional shortest-path models in networking. One is that each channel has a capacity constraint, and the balances of directed edges are alterable. The other is that the payment of a node should take into account the transaction fees charged by the downstream nodes on that path.

In (8), the maximum amount of transaction values that can be processed is no more than the balance of the edge. (9)∼(11) refer to the transaction fee calculations, i.e., $f_{ij}(X_k)$ represents the routing fee that X_k needs to pay to other nodes after edge e_{ij}. One can see that the transaction fee at a node depends on both the transaction value to be paid and the transaction fees charged on the rest of the path. Actually, the main constraints are about the selection of feasible paths, and our purpose is to

Algorithm 2: *SET-SOLVER* function.

Input: Transaction coalition $S = \{X_i, X_j, \ldots, X_k\}$; Initial topology of PCN $G_0 = (\mathcal{N}, \mathcal{E}, \mathcal{C}, \mathcal{W}, \mathcal{B})$;
Output: Set of transaction fee f; Set of path $\mathcal{P}$; Topology of PCN G;
1: $f \leftarrow \varnothing$;
2: $\mathcal{P} \leftarrow \varnothing$;
3: $G \leftarrow G_0$;
4: **for** all $X_j \in S_i$**do**
5: $f_j, \mathcal{P}_j, G = SINGLE - SOLVER(X_j, G)$;
6: $f \leftarrow f \cup f_j$;
7: $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_j$;
8: **end for**
9: **return** $f, \mathcal{P}, G$;

find the cheapest one among them. If such a path leads to a higher routing cost than ξ, the public chain will be selected for the value transfer (13).

Algorithm 1 is designed to calculate the minimum routing fee that the coalitional users need to pay. We need to permutate all the transactions in the coalition (line 1), calculate and compare the transaction fees required for each processing order (line 2-17). Lines 11-12 handle the corner case that different processing sequences yield the same amount of total transaction fees. We select the one with the largest number of transactions being processed successfully, and proceed to choose the one with the highest total transaction values (line 13-15). Algorithm 2 is used to calculate the minimum transaction fee of coalitional users in one processing order. For each transaction request, we can obtain the updated PCN topology, and the available minimum cost path in the current PCN topology. The new PCN topology is used as the input for the next transaction request (line 5).

Algorithm 3 calculates the cheapest path for a single transaction and updates the PCN topology (5) based on the Dijkstra algorithm. Unlike the traditional routing problem, the amount of payment at the first node of a path is the highest, and becomes less and less along this path because the intermediate nodes will keep their own transaction fees. Therefore, directly using the existing minimum-cost path algorithms does not work properly. We reverse the payment direction and search for the minimum cost available path from the receiver to the sender (line 10-19) alternatively. Meanwhile, we need to update the channel balances of the PCN topology after each successful processing of a transaction.

We can obtain the benefit of S

$$\psi(S) = \sum_{X_k \in S} f(X_k, G_{t_k}) - f(X_k, G_{\eta_S}) \tag{14}$$

and a set of important properties about $\psi(S)$ are shown below.

- *Cohesive:* A coalitional game $<\mathbf{X}, \psi>$ with transferable payoff is cohesive if

$$\psi(\mathbf{X}) \geq \sum_{k=1}^{n} \psi(S_k)$$

for every partition $\{S_1, \ldots, S_n\}$ of $\mathbf{X}$.

Algorithm 3: *SINGLE-SOLVER* algorithm.

Input: Transaction $X = (n_s, n_r, v)$; Current topology of PCN $G = (\mathcal{N}, \mathcal{E}, \mathcal{C}, \mathcal{W}, \mathcal{B})$;
Output: Transaction fee f; Path $\mathcal{P}$; New topology G';
1: $G' = (\mathcal{N}, \mathcal{E}, \mathcal{C}, \mathcal{W}, ' \mathcal{B}) \leftarrow G$
2: $\mathcal{N} \leftarrow G.nodes$; $\mathcal{E} \leftarrow G.edges$;
3: $R \leftarrow \{n_r\}$; $S \leftarrow \mathcal{N} \setminus n_r$;
4: $path \leftarrow \varnothing$;
5: $fee \leftarrow \varnothing$;
6: **for** $n_i \in \mathcal{N} \setminus n_r$**do**
7: $fee_i \leftarrow \xi, path_i \leftarrow [n_i]$, if $e_{ir} \notin \mathcal{E}$;
8: $w_{si} \leftarrow 0$, $b_{si} \leftarrow 0$, **if** $e_{si} \in \mathcal{E}$;
9: **end for**
10: **while**$S \neq \varnothing$ and fee was updated
11: $n_i \leftarrow \arg\min_{n_k \in R} fee$
12: **for** $n_j \in n_i.in\text{-}neighbors$ **do**
13: **if** $fee_i + w_{ji} \cdot (v + fee_i) + b_{ji} < fee_j$ **and** $(v + fee_i) \leq c_{ji}$ **then**
14: $fee_j \leftarrow fee_i + w_{ji} \cdot (v + fee_i) + b_{ji}$;
15: $path_j \leftarrow path_i \cup n_j$;
16: $R \leftarrow R \cup n_j$; $S \leftarrow S \setminus n_j$;
17: **end if**
18: **end for**
19: **end while**
$fee \leftarrow fee_s$; $path \leftarrow path_s$;
20: **for**$e_{ij} \in path$ **do**
21: $c_{ij} \leftarrow c_{ij} - (v + fee_j)$;
22: $c_{ji} \leftarrow c_{ji} + (v + fee_j)$;
23: **end for**
24: **return** $f \leftarrow fee_s, \mathcal{P} \leftarrow path_s, G' \leftarrow (\mathcal{N}, \mathcal{E}, \mathcal{C}, \mathcal{W}, ' \mathcal{B})$;

- *Weak Superadditivity:* In a cooperation game $<\mathbf{X}, \psi>$, for any S and S_1, $S \subset \mathbf{X}, S_1 \subset S$, there must be $\psi(S_1) \leq \psi(S)$. That is, the characteristic function ψ satisfies weak superadditivity.

The total benefit of the coalition may increase because of the existence of *altruistic waiting* users. It is necessary for *egoistic waiting* users to share part of their benefit with altruistic waiting user to compensate them for their losses. But there are some egoistic waiting users who want to monopolize the benefit and do not join the coalition. This kind of players is called *Free-riders*. In the social sciences, the free-rider problem is a type of market failure that occurs when those who benefit from resources, public goods (such as public roads or hospitals), or services of a communal nature do not pay for them or under-pay [45]. Free riders need to be prohibited because they may continue to access or use it although not paying for the good (either directly through fees or tolls or indirectly through taxes). We use an example to illustrate this problem and establish a constrain to avoid this kind of phenomenon.

As Fig. 4 shows, there are three transactions, $X_0 = (C, E, 1, 0), X_1 = (C, B, 3, 1), X_2 = (A, C, 6, 2)$. To simplify, we set $b_{ij} = 0$ and $w_{ij} = 0.5$ for all channels. If each transaction chooses not to cooperate, the benefits of three transactions are 0,0,0 separately. If X_2 and X_0 choose to cooperate, they should be processed firstly according the priority mechanism and the processing order is X_2, X_0, X_1. The benefits are $\xi -$

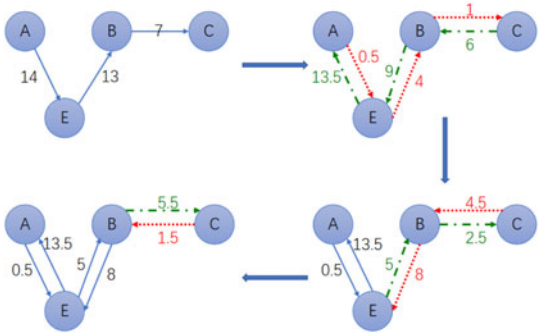


Fig. 4. The process of $X_2 = (A, C, 6, 2), X_0 = (C, E, 1, 0), X_1 = (C, B, 3, 1), b_{ij} = 0, w_{ij} = 0.5 \quad \forall e_{ij} \in E$.

$0.5, \xi, 0$ for X_0, X_1 and X_2. X_0 is egoistic waiting user and according the common sense, he should share some benefit with X_2. X_1 also benefit but he does not share his earning with X_2 because of not joining in coalition. We call X_1 *Free-rider*. X_2 is the altruistic waiting user and he is not paid for his service to free-rider. We need to set up an effective mechanism to resist this phenomenon because free-rider problem will lead to the instability of the coalition.

We address the free-rider problem by setting an additional restriction. All channels that update the balances as a result of coalition member transactions are stored. The path of the player who does not join in S is checked, and the cheaper path created by S is prohibited from being used by the free-riders. This solution can prevent the users in S from waiting but not getting the corresponding benefits. Intermediate nodes will be willing to abide by it consciously.

C. Benefit Redistribution Via Shapley Value

The reordering of the players may cause some of them paying less routing fee and while some others paying more. In order to stimulate the players to form a grand coalition, the redistribution of the benefit is inevitable. It needs to be *fair* to all the players, and yields a *unique* payoff vector known as the *value* of the coalitional game [44].

Definition 2: A *benefit redistribution mechanism* is an operator ϕ on a payment channel network $\langle \mathbf{X}, \psi \rangle$ that allocates a cost vector $\phi(\mathbf{X}, \psi) = (\phi_0, \phi_1, \ldots, \phi_{N-1})$ in $\mathbb{R}^N$ for all the players.

We hereby design a benefit redistribution mechanism with the following desirable properties among the players.

Property 1 (Rationality): If $\phi_i(S) \geqslant \psi(\{X_i\})$, it is the individual rationality ; If $\sum_{X_i \in S}(\phi_i(S)) = \psi(S)$, it is the group rationality .

Individual rationality [31] motivates the players to join the coalition and cooperate accordingly. Group rationality requires that the profit assigned equals the profit received from the coalition.

Property 2 (Balanced Contribution): A value ϕ satisfies the balanced contributions property if for every coalitional game with transferable benefit $< \mathbf{X}, \psi >$, we have

$$\phi_i(\mathbf{X}) - \phi_i(\mathbf{X} \backslash \{X_j\}) = \phi_j(\mathbf{X}) - \phi_j(\mathbf{X} \backslash \{X_i\}),$$

where $X_i \in \mathbf{X}$ and $X_j \in \mathbf{X}$.

The property of balanced contributions addresses the fairness between any pair of transactions in $\mathbf{X}$. If we start with a set of two transactions $(\mathbf{X}, \psi) = (\{X_1, X_2\}, \psi)$, the gain from cooperation is $\psi(\mathbf{X}) - \psi(X_1) - \psi(X_2)$. Thus, the egalitarian solution is

$$\phi_i(\mathbf{X}, \psi) = \psi(\{i\}) + \frac{1}{2}[\psi(\mathbf{X}) - \psi(\{X_1\}) - \psi(\{X_2\})].$$

Property 3 (Symmetry): If $\psi(S \cup X_i) = \psi(S \cup X_j)$ for all $S \in \mathbf{X} \backslash \{X_i, X_j\}$, then $\phi_i(S) = \phi_j(S)$.

The symmetry property requires that if two players contribute the same to every subset of other players, they should receive the same amount of cost.

Property 4 (Dummy): There is $\phi(X_k) = \psi(\{X_k\})$ for a dummy player k in coalition S.

In our game, if a player does not contribute to the reduction of routing fee, i.e. $\psi(S) + \psi(\{X_i\}) = \psi(S \cup X_i)$, the payoff of this player of joining the coalition is identical to that of not joining.

Property 5 (Additivity): For any two game (S, ψ_1) and (S, ψ_2) we have $\phi_i(\psi_1 + \psi_2) = \phi_i(\psi_1) + \phi_i(\psi_2)$ for all $i \in S$, where $\psi_1 + \psi_2$ is the game defined by $(\psi_1 + \psi_2)(S) = \psi_1(S) + \psi_2(S)$ for every coalition S.

The additivity property ensures that even if the charging rate of some edges changes, our redistribution mechanism is still available.

Shapley value is the unique value that satisfies all five properties. Then, it is defined as follows [43].

Definition 3 (Marginal contribution): The *marginal contribution* of player i to any coalition S with $i \in S$ in the game $\langle X, \psi \rangle$ to be

$$\triangle_i(S) = \psi(S) - \psi(S \backslash \{i\}).$$

Definition 4 (Shapley Value): The Shapley value ϕ is defined by the condition

$$\phi_i(S) = \sum_{S_1 \subseteq S} \frac{(|S| - |S_1|)!(|S_1| - 1)!}{|S|!} \triangle_i(S_1), \quad \forall X_i \in S. \tag{15}$$

where $|S|$ and $|S_1|$ represent the numbers of players in S and S_1, respectively. Actually, Shapley value represents the expected marginal contribution over all orders of this player to the set of players who precede him.

Lemma 1: If ψ satisfies the weak superadditivity property, the corresponding Shapley value satisfies the individual rationality.

Proof: If ψ satisfies the weak superadditivity, that means:

$$\psi(S \cup i) \geq \psi(S) + \psi(\{i\}), \forall S \in X, i \in X \backslash S.$$

In consenquence, $\triangle_i(S) \geq \psi(\{i\})$. According to (15), $\phi_i(S)$ satisfies individual rationality. ■

V. Stochastic Waiting Time

In this section, we formulate the stochastic model of balance dynamics on a simplified PCN and calculate the waiting time distribution of a strategic user.

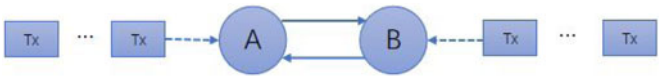


Fig. 5. A simple network for analyzing stochastic waiting time.

A. Stochastic Capacity Model

We will show that an individual "patient" transaction can achieve a lower cost by waiting for the feasibility of a payment channel when the balance of this channel is below the transaction value in the very beginning. Note that this type of patience does not demand the periodic transaction processing and the cost redistribution mechanisms. We consider a simplified PCN with only two nodes forming a payment channel in Fig. 5 that can be easily generalized to the network with parallel payment channels. Node A (resp. node B) is in charge of processing the left-side (resp. right-side) transactions on edge e_{AB} (resp. edge e_{BA}). The successful transactions from A to B increase the balance c_{BA} and decrease the balance c_{AB}, and vice versa. When a transaction finds the balance c_{AB} insufficient, it can keep patient until c_{AB} is larger than its transaction value. An interesting question is how much time a transaction needs to wait before successful processing, given the stochastic arrivals of other transactions on both sides of the channel.

Balance Dynamics. The time axis is set to $t = 0$ when a transaction arrives to node A and will be transferred to node B through edge e_{AB}. If the initial balance u is no less than the transaction value v, it is processed immediately. Otherwise, it will wait for until c_{AB} is greater than v. Without loss of generality, we denote this transaction as the tagged transaction $\hat{X}$. Denote by U_t the balance of edge e_{AB} at time t and there exists

$$U_t = u + \sum_{i=0}^{N_2(t)} v_{2i} - \sum_{i=0}^{N_1(t)} v_{1i}, \tag{16}$$

where v_{1i} indicates the value of the i^{th} transaction from A to B and v_{2i} indicates that of the i^{th} transaction from B to A. Here, $N_1(t)$ and $N_2(t)$ are the numbers of transactions on edge e_{AB} and e_{BA} by time t respectively. We make the following assumptions.

- The arrival process of transactions on edge e_{AB} is the Poisson process $\{N_1(t) : t \geq 0\}$ with parameter λ_1, and that on edge e_{BA} is the Poisson process $\{N_2(t) : t \geq 0\}$ with parameter λ_2.
- The transaction values on both directions, i.e. v_{1i} and v_{2i}, are independent and identically distributed (i.i.d) with the probability density function $g(\cdot)$ and expectation μ.

The assumption of Poisson arrival is commonly adopted in decentralized payment systems [37]–[40], and the bilateral transactions are deemed to have the same distribution of values but with different arrival rates [15]. Therefore, the evolution of $\{U_t; t \geq 0\}$ is a *compound* Poisson process.

The waiting time of the transaction $\hat{X}$ is actually the duration between 0 and the instant that the balance c_{AB} is greater than v for the first time. Thus, the waiting time can be modeled as the *first passage time* of U_t to v in queuing theory. Formally, we denote $\hat{t}$ as the waiting time that has

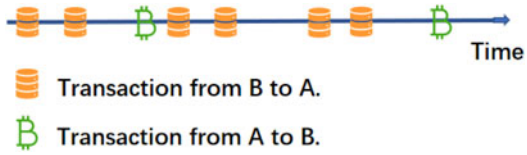


Fig. 6. An example of transactions arrival.

$$\widehat{t} = \arg\min_{t} \left\{ t \mid u - \sum_{i=0}^{N_1(t)} v_{1i} + \sum_{i=0}^{N_2(t)} v_{2i} \geq v \right\}. \tag{17}$$

Given the stochastic arrival of transactions on the edges e_{AB} and e_{BA}, $\hat{t}$ is a random variable by nature.

Before calculating the distribution of $\hat{t}$, we provide the precondition of waiting. The sum of c_{AB} and c_{BA} has been decided upon the construction of the payment channel so that the transaction $\widehat{X}$ cannot be processed in this channel if this sum is below v. We provide the following theorem with regard to the expectation and variance of the waiting time.

Proposition 1: The expected waiting time $\mathbb{E}[\widehat{t}]$ is calculated by

$$\mathbb{E}[\hat{t}] = \begin{cases} \frac{v-u}{(\lambda_2-\lambda_1)\mu} & \lambda 2 \ > \ \lambda 1. \\ \infty & \text{otherwise.} \end{cases} \tag{18}$$

The variance of waiting time is calculated by

$$\mathbb{V}ar[\widehat{t}] = \frac{(v-u)(\lambda_1+\lambda_2)\mathbb{E}(v^2)}{(\lambda_2-\lambda_1)^3\mu^3}. \tag{19}$$

The detailed proof can be referred to [53] with slight modifications. The expectation and variance of waiting time are both found to increase monotonically with λ_1 and decrease monotonically with λ_2 by derivative. Therefore, the smaller the λ_1 or the larger λ_2 is, the shorter and more stable the waiting time there has.

B. Computing Waiting Time Distribution

The expected waiting time overlooks the stochastic behavior of transaction arrivals and random transaction values, which is not sufficient to quantify the characteristic of the waiting time. We are more interested in how the chance of the successful processing increases over the waiting time. Hereby we analyze the probability distribution of the waiting time. Denote by $\Phi(v_k, u, t)$ the probability of a transaction being processed by time t

$$\Phi(v_k, u, t) = \mathbf{Pr}\{\hat{t} \leq t | U_0 = u, v = v_k\}, \tag{20}$$

where u is the initial balance observed by X.

Computing $\Phi(v_k, u, t)$ is a very challenging task. We use a timeline in Fig. 6 to illustrate the possible events that inspire our basic idea. One can observe that the first passage event must occur at the instant of processing a transaction on edge e_{BA}. Then, we can compute $\varphi(r, n)$, the probability that the first passage event takes place before or at the arrival of the n^{th} transaction on edge e_{BA}, where $r = v_k - u$. The calculation of $\Phi(v_k, u, t)$ is thus transformed into the union of $\varphi(r, n)$ for all n mutually exclusive events that happen before time t.

During the inter-arrival time of $(n-1)^{th}$ and n^{th} transaction from B to A, we need to scrutinize the number of transaction arrivals on edge e_{AB} and the distribution of their total value. Formally, we provide the following theorem on the distribution of waiting time.

Theorem 1: The distribution of the waiting time $\Phi(v_k, u, t)$ is expressed as an iterative equation:

$$\Phi(v_k,u,t)=\sum_{n=1}^{\infty}\sum_{j=n}^{\infty}e^{-\lambda_2 t}\frac{(\lambda_2 t)^j}{j!}(\varphi(r,n)-\varphi(r,n-1)); \quad (21)$$

$$\varphi(r,n)=\sum_{m=0}^{\infty}p_m\int_0^{\infty}q_n dG^{m^*}(z)+\varphi(r,1); \quad (22)$$

$$\varphi(r,1)=\sum_{m=0}^{\infty}p_m\int_0^{\infty}[1-G(r+z)]d(G^{m^*}(z)); \quad (23)$$

$$q_n=\int_0^{u+z}\varphi(u+z-x,n-1)dG(x); \quad (24)$$

$$p_m=\frac{\lambda_1^m\lambda_2}{(\lambda_1+\lambda_2)^{m+1}}; \quad (25)$$

$$r=v_k-u, \quad (26)$$

where $G(x)$ is the probability distribution function of x, $G^{m^*}(z)$ represents the convolution of $G(z)$ by m times.

Proof: Our proof is carried out in two steps.

Step 1: Equivalence between $\Phi(v_k, u, t)$ and $\varphi(r, n)$. Define $U(m)$ the balance of edge e_{AB} when there are m transaction arrivals on edge e_{BA}. Define T_k the inter-arrival time of two consecutive arrivals, X_{k-1} and X_k, on edge e_{AB}. Then, the probability of the first passage event upon the arrival of Y_m is given by:

$$U(m)=\sum_{i=0}^{m}v_{2i}-\sum_{i=0}^{N_1(\sum_{k=1}^{m}T_k)}v_{1i}, \quad (27)$$

$$\varphi(r,n)=\mathbf{Pr}\{U(m)\geq r,\exists m\leq n\}. \quad (28)$$

The first passage event is the union of mutually exclusive events that the first passage happens at the n^{th} transaction arrival on edge e_{BA}. Accordingly, the waiting time distribution is expressed as

$$\Phi(v_k,u,t)=\sum_{n=1}^{\infty}\mathbf{Pr}\left\{\sum_{j=1}^{n}T_j<t|\lambda_2\right\}(\varphi(r,n)-\varphi(r,n-1)). \quad (29)$$

Since the arrival processes are Poisson, the number of transaction arrivals on each edge is given by:

$$\mathbf{Pr}\{N_1(t)=m\}=\frac{(\lambda_1 t)^m e^{-\lambda_1 t}}{m!}, \quad (30)$$

$$\mathbf{Pr}\{N_2(t)=m\}=\frac{(\lambda_2 t)^m e^{-\lambda_2 t}}{m!}. \quad (31)$$

The inter-arrival time of transactions obeys the memoryless exponential distribution so that we can write down the sum of n random variables in the form of Erlang distribution

$$\mathbf{Pr}\{T_n\leq t|\lambda\}=1-e^{-\lambda t}, \quad (32)$$

$$\mathbf{Pr}\left\{\sum_{i=1}^{n}T_i\leq t|\lambda_2\right\}=\sum_{j=n}^{\infty}e^{-\lambda_2 t}\frac{(\lambda_2 t)^j}{j!}. \quad (33)$$

Step 2: Calculation of $\varphi(r, n)$. According to the full probability formula, we can obtain

$$\Lambda=\{T_1=t,N_1(t)=y,\sum_{i=1}^{y}v_{1i}=z,v_{21}=x\}, \quad (34)$$

$$F_1(m,n,r)=\mathbf{Pr}\{U(m)\geq r,\exists m\leq n|\Lambda\}.$$

For simplify, $F_1(m, n, r)$ can be rewritten as F_1. It describes the probability that there is at least one moment when the balance of e_{BA} is greater than the value needed to transfer until the arrival of the n^{th} transaction on edge e_{BA} under condition Λ. Λ describes the condition during the time that the first event arrives from B to A.

According to the law of total probability, $\varphi(r, n)$ can be described as

$$\begin{aligned}\varphi(r,n)&=\mathbf{Pr}(N_1(t)=y)\int_0^{\infty}\mathbf{Pr}(T_1=t)\int_0^{\infty}\mathbf{Pr}\left(\sum_{i=1}^{y}v_{1i}=z\right)\\&\int_0^{\infty}F_1 d(G(x))dzdt\\&=\sum_{y=0}^{\infty}\frac{(\lambda_1 t)^y}{y!}e^{-\lambda_1 t}\int_0^{\infty}\lambda_2 e^{-\lambda_2 t}dt\int_0^{\infty}\mathbf{Pr}\left(\sum_{i=1}^{y}v_{1i}=z\right)\\&\int_0^{\infty}F_1 d(G(x))dz\\&=\sum_{y=0}^{\infty}\frac{\lambda_2\lambda_1^y}{(\lambda_1+\lambda_2)^{y+1}}\int_0^{\infty}\int_0^{\infty}F_1 d(G(x))d(G^{y^*}(z)), \quad (35)\end{aligned}$$

where $G^{y^*}(z)$ represents the convolution of $G(z)$ by y times.

Here, F_1 is sensitive to the number n. If $n = 1$, F_1 represents the probability that the passage event is bound to take place until the first transaction arrives from B to A under condition *con*, and if $n > 1$, F_1 represents the probability that the passage event takes place at least once until the arrival of n^{th} transactions of e_{BA}. We next derive the iterative formula as below:

- $n=1$: $\int_0^{\infty}F_1 d(G(x))=\int_{r+z}^{\infty}d(G(x))=1-G(r+z)$;
- $n>1$: $\int_0^{\infty}F_1 d(G(x))=1-G(r+z)+\int_0^{r+z}\varphi(r+z-x,n-1)d(G(x))$.

The detailed derivations are as follows under the condition $\{T_1=t,N_1(t)=y,\sum_{i=1}^{y}v_{1i}=z,v_{21}=x\}$.

- if $x-z\geq r$, $U(1)\geq r$. We can obtain

$$\begin{aligned}F_1(1,n,r)&=\mathbf{Pr}\{x-z\geq r|con\}\\&=\mathbf{Pr}\{x\geq r+z|con\}\\&=\int_{r+z}^{\infty}d(G(x))=1-G(r+z). \quad (36)\end{aligned}$$

- if $x-z<r$, it means $\{F_1(k,n,r),k>1\}$. For $F_1(k,n,r)$, we can express it as

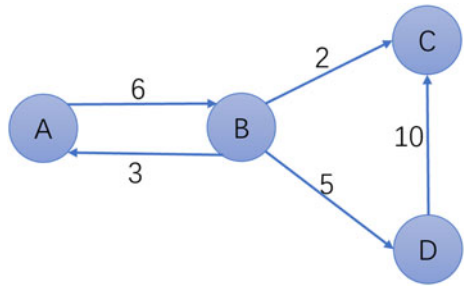


Fig. 7. Topology of PCNs in *PTP* case.

$$
\begin{aligned}
&F_1(k,n,r) = \mathbf{Pr}\{U(k) \geq r, \exists k \leq n | con, U(1) < r\} \\
&= \mathbf{Pr}\{\bigcup\nolimits_{k=1}^{n} U(k) \geq r | con, x - z < r\} \\
&= \mathbf{Pr}\{\bigcup\nolimits_{k=2}^{n} U(k) + x - z \geq r | con, x < z + r\} \\
&= \mathbf{Pr}\{\bigcup\nolimits_{k=1}^{n-1} U(k-1) \geq r + z - x | con, x < z + r\} \\
&= \varphi(r + z - x, n - 1). \qquad (37)
\end{aligned}
$$

According to (27), $F_1(k,n,r)$ can be written as

$$
\int_0^{\infty} F_1 d(G(x)) = 1 - G(r+z) + \int_0^{r+z} \varphi(r+z-x, n-1) d(G(x)). \qquad (38)
$$

Therefore, we can get

$$
\varphi(r,n) = \sum_{y=0}^{\infty} \frac{\lambda_2 \lambda_1^y}{(\lambda_1+\lambda_2)^{y+1}} \int_0^{\infty} [1 - G(r+z) + \int_0^{r+z} \varphi(r+z-x, n-1) d(G(x))] d(G^{y^*}(z)); \qquad (39)
$$

$$
\varphi(r,1) = \sum_{y=0}^{\infty} \frac{\lambda_2 \lambda_1^y}{(\lambda_1+\lambda_2)^{y+1}} \int_0^{\infty} [1 - G(r+z)] d(G^{y^*}(z)). \qquad (40)
$$

Submitting the above two equations to (29), the probability distribution of waiting time is obtained. This ends the proof.

VI. Experimental Study

In this section, we evaluate the performance of *PTP* mechanism under different PCNs and validate the theoretical result of the waiting time under *SP* mechanism.

A. Overall Result for PTP

First, we will use a specific PCN graph to study the coalitional mechanism. The initial topology of payment channel network is shown in Fig. 7. The integers represent the current balances. The charging rates of the channels are 0.1. The basic routing fees for all channels are set to 0 and ξ is set to 1. There are a set of transactions $\mathbf{X} = \{X_0, X_1, X_2, X_3\}$, where $X_0 = (C, D, 1, 0), X_1 = (B, C, 1, 1), X_2 = (A, C, 2, 2), X_3 = (B, A, 1, 3)$. If each transaction chooses not to cooperate with others (transfers at their arrival times), the routing cost of each transaction is $f(X_0, \mathcal{G}_0) = \xi, f(X_1, \mathcal{G}_1) = 0, f(X_2, \mathcal{G}_2) = 0.42, f(X_3, \mathcal{G}_3) = 0$ respectively. In this case, the benefit for each user is 0. If all the transactions choose to cooperate, the optimal order is X_2, X_1, X_0, X_3. The corresponding cost is $f(X_2, \mathcal{G}_0) = 0.2, f(X_1, \mathcal{G}_1) = 0.1, f(X_0, \mathcal{G}_2) = 0, f(X_3, \mathcal{G}_3) = 0$. The benefit for grand coalition is $\psi(X) = 1.12$. We can compute the benefits for the individual transaction.

TABLE I
Benefit Re-Distribution of FCFS and Coalition

$\psi(\{X_0\}) = 0$	$\psi(\{X_1\}) = 0$	$\psi(\{X_2\}) = 0$	$\psi(\{X_3\}) = 0$
$\phi_{X_0} = 0.52$	$\phi_{X_1} = 0.02$	$\phi_{X_2} = 0.58$	$\phi_{X_3} = 0$

The first row of Table I shows the benefit of each transaction if no one chooses to cooperate, and the second row shows that if a grand coalition is formed, each user's benefit will be distributed through Shapley value. After reordering, the routing fee decreases by 78% and the success rate of the number of transactions increases by 25%.

If two players choose to cooperate with each other, the benefits are shown in Table II. The benefits of the coalitions with three players are shown in Table III. The results show that the coalitional function and the re-distribution function completely satisfy the desired properties.

B. Experimental Results of PTP.

We thereby show the performances of the coalitional mechanism under different network parameters. The transaction fee of trading on the public chain is much higher than that of trading on the PCNs. Therefore, we first focus on how much and how many of transactions can be successfully processed in PCNs through our mechanism. The metrics are measured by the successful transaction ratio (i.e. the number of successful transactions in the PCN divided by the total number) and the successful value ratio (i.e. the amount of transaction value processed via the PCN divided by the total transaction value). Next, the transactions that can not be processed in PCNs will be released to the public chain. We will explore the total transaction fees including the routing fees in PCNs and the transaction fees on public chain. We study the performance under different conditions, including different graph sizes and different number of transactions. Fifty random network topologies are generated. Fifty sets of transactions created under each network topology, and each set incorporates includes four transactions.

The maximum and minimum successful number, successful value and total transaction fee of each set of transactions are obtained. We calculate the average results for all sets of transactions under all topologies as the final results. The improvement ratio in successful transaction ratio can be calculated through the difference between the maximum and the minimum number of successful transactions (amount of successful transaction value) in PCNs, divided by the minimum number of successful transactions (minimum amount of successful transaction value) in PCNs. The reduction ratio of transaction fee can be measured by the difference between the maximum and minimum transaction fees under different processing order divided the maximum transaction fee.

TABLE II
THE BENEFIT OF INDIVIDUAL TRANSACTION IN A TWO-PLAYER COALITION

$S = \{X_0, X_1\}$	$\psi(S) = 0$	$\psi(\{X_2\}) = 0$	$\psi(X_3) = 0$	$\phi_{X_0} = 0$	$\phi_{X_1} = 0$
$S = \{X_0, X_2\}$	$\psi(S) = 1.12$	$\psi(\{X_1\}) = 0$	$\psi(\{X_3\}) = 0$	$\phi_{X_0} = 0.56$	$\phi_{X_2} = 0.56$
$S = \{X_0, X_3\}$	$\psi(S) = 0$	$\psi(\{X_1\}) = 0$	$\psi(\{X_2\}) = 0$	$\phi_{X_0} = 0$	$\phi_{X_3} = 0$
$S = \{X_1, X_2\}$	$\psi(S) = 0.12$	$\psi(\{X_0\}) = 0$	$\psi(\{X_3\}) = 0$	$\phi_{X_1} = 0.06$	$\phi_{X_2} = 0.06$
$S = \{X_1, X_3\}$	$\psi(S) = 0$	$\psi(\{X_0\}) = 0$	$\psi(\{X_2\}) = 0$	$\phi_{X_1} = 0$	$\phi_{X_3} = 0$
$S = \{X_2, X_3\}$	$\psi(S) = 0$	$\psi(\{X_0\}) = 0$	$\psi(\{X_1\}) = 0$	$\phi_{X_2} = 0$	$\phi_{X_3} = 0$

TABLE III
THE BENEFIT OF INDIVIDUAL TRANSACTION IN A THREE-PLAYER COALITION

$S = \{X_0, X_1, X_2\}$	$\psi(S) = 1.12$	$\psi(\{X_3\}) = 0$	$\phi_{X_0} = 0.52$	$\phi_{X_1} = 0.02$	$\phi_{X_2} = 0.58$
$S = \{X_0, X_1, X_3\}$	$\psi(S) = 0$	$\psi(\{X_2\}) = 0$	$\phi_{X_0} = 0$	$\phi_{X_1} = 0$	$\phi_{X_3} = 0$
$S = \{X_0, X_2, X_3\}$	$\psi(S) = 1.12$	$\psi(\{X_1\}) = 0$	$\phi_{X_0} = 0.56$	$\phi_{X_2} = 0.56$	$\phi_{X_3} = 0$
$S = \{X_1, X_2, X_3\}$	$\psi(S) = 0.12$	$\psi(\{X_0\}) = 0$	$\phi_{X_1} = 0$	$\phi_{X_2} = 0.06$	$\phi_{X_3} = 0.06$

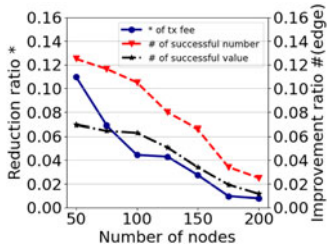


Fig. 8. The performance of *PTP* under same edges w.r.t. number of nodes.

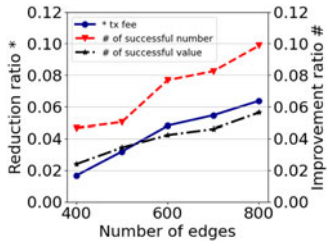


Fig. 9. The performance of *PTP* w.r.t. number of edges.

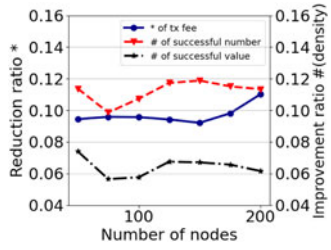


Fig. 10. The performance of *PTP* under same network density w.r.t. number of nodes.

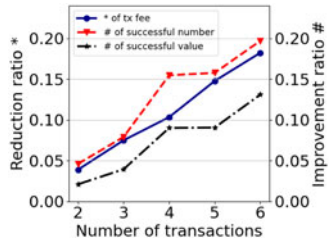


Fig. 11. The performance of *PTP* w.r.t. number of transactions.

In each topology, the number of nodes is set to 100, the number of edges is set to 700, the capacity of channels is randomly selected in the range of 10 to 15 coins. We randomly select the transaction value in the range of 1 to 12 coins. The charging rate of each channel is chosen randomly from 0.1% to 0.5%. The base routing fee is 0 for all channels. The transaction fee on public chain is 5 coins.

The performance of *PTP* is related to the network density. We define the network density [42] of PCNs with j nodes and k edges as:

$$\rho = \frac{2 \cdot k}{j(j-1)}$$

where the value of ρ ranges from 0 to 1. The higher the value is, the denser the network is.

First, the improvement ratios of successful transactions in PCNs and the reduction ratios of transaction fees under different network densities are explored. Fig. 8 shows that when the number of nodes increases, the improvement ratios of successful transactions in PCNs show a downward trend in the numbers and the values. The improvement ratios are still more than 1% even the number of nodes increases to 200. At this point, the network density is extremely low, decreasing to 0.02. The reduction ratios of transaction fees also exhibit a downward trend, but they are lower than the improvement ratios of the successful number under the same conditions. The reason is that although transactions processed on PCNs do not need to pay transaction fees processed on the public chain, they still need to pay routing fees in PCNs. Then, Fig. 9 plots the performance of *PTP* under different number of edges when the number of nodes are same. The improvement ratios of successful transactions and reduction ratios of transaction fees both increase with the increase of the number of edges.

We want to explore further whether the size or the density of PCN affects the performance of *PTP* mechanism. Fig. 8 and Fig. 9 show the improvement ratio of the success rate under different network densities. We then scale the network sizes and fix the network densities. As observed in Fig. 10, the improvement ratios of successful transactions in PCNs and reduction ratios of transaction fees are relatively stable with the increase in the number of nodes when $\rho = 0.3$. The result

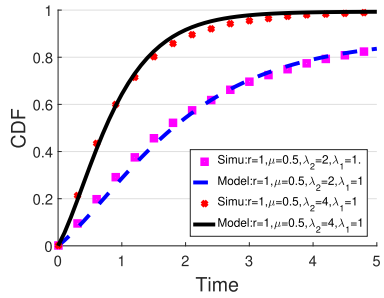


Fig. 12. The CDF of waiting time: model vs simulation.

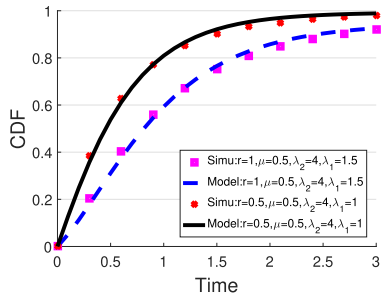


Fig. 13. The CDF of waiting time: model vs simulation.

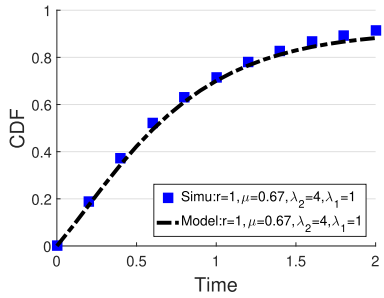


Fig. 14. The CDF of waiting time: model vs simulation.

shows that the performance of *PTP* mechanism is relatively stable with the same density despite the node changes. These results validate that the performance of the *PTP* is related to the network density. The high network density indicates the better performance of the *PTP*.

The number of transactions affects the performance of *PTP* mechanism. As depicted in Fig. 11, the improvement ratio of the successful transactions and the reduction ratio of transaction fees in PCNs increase with the number of transactions when the number of nodes is 100 and the number of channels is 5000. However, a lot of transactions means great processing time. Therefore, *PTP* will process all the transactions in the buffer cyclically when the number of transactions exceeds a certain threshold or a fixed duration has been reached.

C. Experiment Results of Stochastic Waiting Time

We assume that the transaction sizes obey the exponential distribution with the expectation μ, the transaction from A to B is a Poisson process with rate λ_1, and from B to A is a Poisson process with λ_2. We recall r is the difference between the required transaction value and the current balance of e_{AB}.

Figs. 12–14 show the simulation results and the calculation results of SP. In the three figures, the markers represent the simulation results, the lines represent the model results. The horizontal axis is the waiting time, and the vertical axis is the cumulative probability function of the waiting time. We simulate and calculate to get the waiting time distribution under different setting of parameters which include r, μ, λ_1 and λ_2. The simulation results match well with the theoretical results under different conditions.

VII. Conclusion

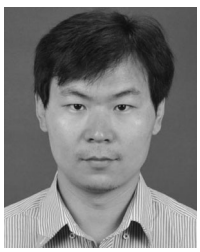
In this paper, we study the cost efficient PCN routing problem in which the first-come-first-serve way of transaction processing may block inexpensive paths, thus forcing the late transactions to traverse more expensive paths or resort to on-chain settlements. Two novel approaches, *periodic transaction processing (PTP)* and *strategic patience (SP)*, are proposed that require the PCN users to wait for the reordered processing of transactions. In the former, the order of dealing with transactions is determined to minimize their total cost. A Shapley value mechanism is proposed to redistribute the cost to all the users so as to incentive their willingness to participate. In the later, a transaction can wait for the accumulation of the balance of the payment channel edge when the initial balance is insufficient upon its arrival. Under the general transaction value distribution and the assumption of Poisson arrivals, we present a stochastic model to capture the distribution of waiting time. Experimental results manifest that being a bit more patient can effectively reduce the cost of all the participated transactions in *PTP* and the stochastic model of the waiting time is accurate in *SP*.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *White Paper*, 2008.
[2] Z. Shae and J. J. P. Tsai, "On the design of a blockchain platform for clinical trial and precision medicine," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, Atlanta, GA, USA, 2017, pp. 1972–1980.
[3] K. Karlsson *et al.*, "Vegvisir: A partition-tolerant blockchain for the Internet-of-Things," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, Vienna, 2018, pp. 1150–1158.
[4] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang, "Resource allocation and consensus on edge blockchain in pervasive edge computing environments," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, Dallas, TX, USA, 2019, pp. 1476–1486.
[5] [Online]. Available: https://bitcoin.org/en/bitcoin-core/
[6] [Online]. Available: http://www.ethereum.org/
[7] "Visa acceptance for retailers," [Online]. Available: https://usa.visa.com/about-visa/visanet.html
[8] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," *White Paper*, 2016.
[9] A. Singh, R. M. Parizi, M. Han, A. Dehghantanha, H. Karimipour, and K.-K. R. Choo, "Public blockchains scalability: An examination of sharding and segregated witness," *Blockchain Cybersecur., Trust Privacy*, vol. 79, pp. 203–232, 2020.
[10] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, 2016, pp. 17–30.
[11] Raiden Network, "What is the raiden network," 2018. [Online]. Available: https://messari.io/asset/raiden-network/profile
[12] Y. Zhang, D. Yang, and G. Xue, "CheaPay: An optimal algorithm for fee minimization in blockchain-based payment channel networks," in *Proc. IEEE Int. Conf. Commun.*, Shanghai, China, 2019, pp. 1–6.

[13] P. Wang, H. Xu, X. Jin, and T. Wang, "Flash: Efficient dynamic routing for offchain networks," in *Proc. Int. Conf. Emerg. Netw. Exp. Technol.*, Orlando, Florida, 2019, pp. 370–381.
[14] A. H. Jun Ren, L. Feng, S. A. Cheong, and R. S. Mong Goh, "Optimal fee structure for efficient lightning networks," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst.*, Singapore, 2018, pp. 980–985.
[15] S. Vibhaalakshmi *et al.*, "High throughput cryptocurrency routing in payment channel networks," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2020, pp. 777–796.
[16] D. Piatkivskyi and M. Nowostawski, "Split payments in payment networks," *Data Privacy Manage. Cryptocurrencies Blockchain Technol.*, vol. 11025, pp. 67–75, 2018.
[17] E. Rohrer, J.-F. Laß, and F. Tschorsch, "Towards a concurrent and distributed route selection for payment channel networks," *Data Privacy Manage. Cryptocurrencies Blockchain Technol.*, vol. 10436, pp. 411–419, 2017.
[18] S. M. Varma and S. T. Maguluri, "Throughput optimal routing in blockchain based payment systems," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 4, pp. 1859–1868, Dec. 2021.
[19] [Online]. Available: https://en.bitcoin.it/wiki/Multi-signature
[20] P. McCorry, M. Möser, S. F. Shahandasti, and F. Hao, "Towards Bitcoin payment networks," *Inf. Secur. Privacy*, vol. 9722, pp. 57–76, 2016.
[21] G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei, "SilentWhispers: Enforcing security and privacy in decentralized credit networks," presented at the Network and Distributed System Security Symposium, San Diego, CA, USA, 2017.
[22] S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling payments fast and private: Efficient decentralized routing for path-based transactions," Dec. 2017, *arXiv:1709.05748*.
[23] P. Prihodko *et al.*, "Flare: An approach to routing in lightning network," *White Paper*, 2016.
[24] Z. Avarikioti, L. Heimbach, Y. Wang, and R. Wattenhofer, "Ride the Lightning: The game theory of payment channels," *Financial Cryptogr. Data Secur.*, vol. 12059, pp. 264–283, 2020.
[25] O. Ersoy, S. Roos, and Z. Erkin, "How to profit from payments channels," *Financial Cryptogr. Data Secur.*, vol. 12059, pp. 284–303, 2020.
[26] O. Osuntokun, "AMP: Atomic multi-path payments over lightning," 2018.
[27] R. Yu, G. Xue, V. T. Kilari, D. Yang, and J. Tang, "CoinExpress: A fast payment routing mechanism in blockchain-based payment channel networks," in *Proc. Int. Conf. Comput. Commun. Netw.*, Hangzhou, China, 2018, pp. 1–9.
[28] R. Khalil and A. Gervais, "Revive: Rebalancing off-blockchain payment networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, 2017, pp. 439–453.
[29] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, 2017, pp. 455–471.
[30] S. Werman and A. Zohar, "Avoiding deadlocks in payment channel networks," *Data Privacy Manage. Cryptocurrencies Blockchain Technol.*, vol. 11025, pp. 175–187, 2018.
[31] M. Sasan *et al.*, "The Shapley value for a fair division of group discounts for coordinating cooling loads," *PLoS One*, vol. 15, no. 1, 2020, Art. no. e0 227049.
[32] E. Rohrer, J. Malliaris, and F. Tschorsch, "Discharged payment channels: Quantifying the lightning network's resilience to topology-based attacks," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, Stockholm, Sweden, 2019, pp. 347–356.
[33] F. Engelmann, H. Kopp, F. Kargl, F. Glaser, and C. Weinhardt, "Towards an economic analysis of routing in payment channel networks," in *Proc. 1st Workshop Scalable Resilient Infrastructures Distrib. Ledgers*, Las Vegas, NV, USA, 2017, pp. 1–6.
[34] N. Khan and R. State, "Lightning network: A comparative review of transaction fees and data analysis," *Blockchain Appl.*, vol. 1010, pp. 11–18, 2020.
[35] P. Hoenisch and I. Weber, "AODV–based routing for payment channel networks," *Blockchain – ICBC*, vol. 10974, pp. 107–124, 2018.
[36] F. Beres, I. A. Seres, and A. A. Benczur, "A cryptoeconomic traffic analysis of Bitcoin's lightning network," Jul. 2020, *arXiv:1911.09432*.
[37] C. N. Cordi, "Simulating high-throughput cryptocurrency payment channel networks," Ph.D. thesis, 2017.
[38] N. Papadis and L. Tassiulas, "State-dependent processing in payment channel networks for throughput optimization," Mar. 2021, *arXiv:2103.17207*.
[39] X. Ding, L. Ren, Z. Sang, Z. Zhang, Y. Du, and P. Yan, "Routing optimization for high speed photon state-channel architecture," *Blockchain Technol. Appl.*, vol. 1176, pp. 231–241, 2020.
[40] Y. Sali and A. Zohar, "Optimizing off-chain payment networks in cryptocurrencies," Jul. 2020, *arXiv:2007.09410*.
[41] [Online]. Available: https://ycharts.com/indicators/bitcoin_average_transaction_fee
[42] F. Katherine, "Comparing social networks: Size, density, and local structure," *Adv. Methodol. Statist.*, vol. 3, no. 2, pp. 185–216, 2006.
[43] W. Eyal., "The Shapley value," *Handbook Game Theory Econ. Appl.*, vol. 3, no. 2, pp. 2025–2054, 2002.
[44] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA, USA: MIT Press, 1994.
[45] [Online]. Available: https://en.wikipedia.org/wiki/Free-rider-problem
[46] N. Awathare, Suraj, V. J. Akash Ribeiro, and U. Bellur, "REBAL: Channel balancing for payment channel networks," in *Proc. Int. Symp. Model., Anal., Simul. Comput. Telecommun. Syst.*, Houston, TX, USA, 2021, pp. 1–8.
[47] H. Xue, Q. Huang, and Y. Bao, "EPA-Route: Routing payment channel network with high success rate and low payment fees," in *Proc. Int. Conf. Distrib. Comput. Syst.*, DC, USA, 2021, pp. 227–237.
[48] Y. Van Engelshoven and S. Roos, "The merchant: Avoiding payment channel depletion through incentives," in *Proc. IEEE Int. Conf. Decentralized Appl. Infrastructures*, U.K., 2021, pp. 59–68.
[49] S. Mercan, E. Erdin, and K. Akkaya, "Improving transaction success rate in cryptocurrency payment channel networks," *Comput. Commun.*, vol. 166, pp. 196–207, Jan. 2021.
[50] O. Neut, "LRFP: Extending local routing protocols in layer 2 networks with a secure fee model," Bachelor thesis, 2021.
[51] V. Bagaria, J. Neu, and D. Tse, "Boomerang: Redundancy improves latency and throughput in payment-channel networks," in *Financial Cryptography and Data Security*, vol. 12059. Berlin, Germany: Springer, 2020, pp. 304–324.
[52] S. Rahimpour and M. Khabbazian, "Spear: Fast multi-path payment with redundancy," in *Proc. ACM Conf. Adv. Financial Technol.*, Arlington, VA, USA, 2021, pp. 183–191.
[53] M. Gao, "Time analysis of the surplus reaching a given level firstly in the double compound possion risk model," *J. Quantitative Econ.*, vol. 27, no. 1, pp. 81–84, 2010.

Qianlan Bai is currently working toward the Ph.D. degree in computer science with the Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China. Her research interests include economic analysis and security analysis about blockchain.



Yuedong Xu received the B.S. degree from Anhui University, Hefei, China, the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong. He is currently a tenured Associate Professor with the School of Information Science and Technology, Fudan University, Shanghai, China. From 2009 to 2012, he was a Postdoc with INRIA Sophia Antipolis, Biot, France, and Universite d'Avignon, Avignon, France. His research interests include performance evaluation, optimization, security, data analytics and economic analysis of communication networks, and mobile computing.

Xin Wang was born in 1973. He received the B.S. degree in information theory and the M.S. degree in communication and electronic systems from Xidian University, Xi'an, China, in 1994 and 1997, respectively, and the Ph.D. degree in computer science from Shizuoka University, Shizuoka, Japan, in 2002. He is currently a Professor with Fudan University, Shanghai, China. His research interests include quality of network service, next-generation network architecture, mobile internet, and network coding. He is a Distinguished Member of CCF.